

# **FootPrints 12.1.09**

## Table Of Contents

FootPrints Subject Areas and Diagrams	1
000 - Very High Level Domain Model	13
Entity Level	13
Table Level	14
001 - High Level Domain Model	15
Entity Level	15
Table Level	16
010 - Translation	17
Logical	17
Physical	18
040 - Container	19
Logical	19
041 - Definition Detail	20
Logical	20
Physical	21
046 - Workspace Item Assignment	22
Logical	22
Physical	23
047 - Workspace Time Tracking	24
Logical	24
Physical	25
048 - Item Field Journaling	26
Logical	26
Physical	27
050 - Item Relationship	28
Logical	28
Physical	29
051 - Item Field Collection	30
Logical	30
Physical	31
060 - Role	32
Logical	32
Physical	33
065 - Configuration Setting	34
Logical	34
Physical	35
066 - Container Snapshot	36
Logical	36
Physical	37
070 - User	38
Logical	38
Physical	39
071 - Team	40
Logical	40
Physical	41
080 - User Profile	42
Logical	42
Physical	43
096 - Email Notification	44
Logical	44
Physical	45
101 - Data Access System	46
Logical	46
Physical	47
102 - Data Access Scripts	48
Logical	48
Physical	49
110 - Id Generation	50
Logical	50
Physical	51
120 - Job Scheduler	52
Logical	52
Physical	53
140 - Workspace Calendar and Operation Schedule	54
Logical	54
Physical	55
141 - Calendar Appointments and Items	56
Logical	56
Physical	57
142 - Calendar Sync	58
Logical	58
Physical	59
160 - Templating	60

Logical	60
Physical	61
170 - Knowledge Base	62
Logical	62
Physical	63
190 - Password Reset	64
Logical	64
Physical	65
210 - Audit Trail	66
Logical	66
Physical	67
230 - Rules	68
Logical	68
Physical	69
240 - File Storage	70
Logical	70
Physical	71
250 - Apache Jack Rabbit JCR	72
Logical	72
Physical	73
262 - Import Items and Users	74
Logical	74
Physical	75
280 - Session and Locking	76
Logical	76
Physical	77
290 - Security	78
Logical	78
Physical	79
300 - Ticket Data Mart	80
Logical	80
Physical	81
305 - Data Mart Load	82
Logical	82
Physical	83
310 - Auxiliary Tables	84
Logical	84
Physical	85
320 - Report	86
Logical	86
Physical	87
332 - Change Management Voting	88
Logical	88
Physical	89
340 - Search	90
Logical	90
Physical	91
350 - Chat	92
Logical	92
Physical	93
360 - Survey	94
Logical	94
Physical	95
370 - Deployment	96
Logical	96
Physical	97
380 - Portal	98
Logical	98
Physical	99
390 - Business Process Template	100
Logical	100
Physical	101
400 - Asset Management	102
Logical	102
Physical	103
410 - Publishing	104
Logical	104
Physical	105
430 - UI Component State	106
Logical	106
Physical	107
900 To Be Deleted	108
Physical	108
901 - Tables Added	109

Logical	109
Physical	110
902 - Columns Added	111
Logical	111
Physical	112
903 - Tables Deleted	113
Logical	113
Physical	114
904 - Columns Deleted	115
Logical	115
Physical	116
Legend	117
Legend	117
Renoir Domain Model Detail	118
Subject Area(s) of Model "Renoir Domain Model Detail"	118
ER Diagram(s) of Subject Area "000 - Very High Level Domain Model"	119
Entity/Table(s) of Subject Area "000 - Very High Level Domain Model"	119
Attribute/Column(s) of Entity/Table "Address Book Container"	124
Attribute/Column(s) of Entity/Table "Application Element"	124
Attribute/Column(s) of Entity/Table "Application Element String"	124
Attribute/Column(s) of Entity/Table "Application Role"	125
Attribute/Column(s) of Entity/Table "Application User"	125
Attribute/Column(s) of Entity/Table "Application User Profile Link"	125
Attribute/Column(s) of Entity/Table "Audit Trail"	125
Attribute/Column(s) of Entity/Table "Audit Transaction Audit Trail"	125
Attribute/Column(s) of Entity/Table "CMDB Container"	125
Attribute/Column(s) of Entity/Table "Configuration Item"	126
Attribute/Column(s) of Entity/Table "Configuration Setting"	126
Attribute/Column(s) of Entity/Table "Contact Item"	126
Attribute/Column(s) of Entity/Table "Container"	126
Attribute/Column(s) of Entity/Table "Container Definition"	126
Attribute/Column(s) of Entity/Table "Definition"	126
Attribute/Column(s) of Entity/Table "Definition Relationship"	127
Attribute/Column(s) of Entity/Table "Definition Version"	127
Attribute/Column(s) of Entity/Table "Field Definition"	127
Attribute/Column(s) of Entity/Table "Item"	127
Attribute/Column(s) of Entity/Table "Item Definition"	127
Attribute/Column(s) of Entity/Table "Item Field Definition"	127
Attribute/Column(s) of Entity/Table "Item Relationship"	128
Attribute/Column(s) of Entity/Table "Item Relationship Definition"	128
Attribute/Column(s) of Entity/Table "Locale"	128
Attribute/Column(s) of Entity/Table "Schedule"	128
Attribute/Column(s) of Entity/Table "String Resource Value"	128
Attribute/Column(s) of Entity/Table "Template"	129
Attribute/Column(s) of Entity/Table "Tenant Partition"	129
Attribute/Column(s) of Entity/Table "Ticket Item"	129
Attribute/Column(s) of Entity/Table "User Profile"	129
Attribute/Column(s) of Entity/Table "Workspace Container"	129
ER Diagram(s) of Subject Area "001 - High Level Domain Model"	130
Entity/Table(s) of Subject Area "001 - High Level Domain Model"	130
Attribute/Column(s) of Entity/Table "Activity Definition"	138
Attribute/Column(s) of Entity/Table "Address Book Container"	138
Attribute/Column(s) of Entity/Table "Agent"	138
Attribute/Column(s) of Entity/Table "Application Element"	138
Attribute/Column(s) of Entity/Table "Application Element String"	138
Attribute/Column(s) of Entity/Table "Application Role"	138
Attribute/Column(s) of Entity/Table "Application User"	139
Attribute/Column(s) of Entity/Table "Asset Item"	139
Attribute/Column(s) of Entity/Table "Audit Trail"	139
Attribute/Column(s) of Entity/Table "Audit Transaction Audit Trail"	139
Attribute/Column(s) of Entity/Table "Business Rule Definition"	139
Attribute/Column(s) of Entity/Table "CMDB Container"	139
Attribute/Column(s) of Entity/Table "Configuration Item"	139
Attribute/Column(s) of Entity/Table "Configuration Setting"	139
Attribute/Column(s) of Entity/Table "Configuration Setting Value"	140
Attribute/Column(s) of Entity/Table "Contact Item"	140
Attribute/Column(s) of Entity/Table "Container"	140
Attribute/Column(s) of Entity/Table "Container Definition"	141
Attribute/Column(s) of Entity/Table "Container Group Definition"	141
Attribute/Column(s) of Entity/Table "Container Group Relationship"	141
Attribute/Column(s) of Entity/Table "Container Home Page"	141
Attribute/Column(s) of Entity/Table "Customer"	141
Attribute/Column(s) of Entity/Table "Database Object Definition"	141
Attribute/Column(s) of Entity/Table "Default Rule"	141



Attribute/Column(s) of Entity/Table "Definition"	141
Attribute/Column(s) of Entity/Table "Definition Subtype"	141
Attribute/Column(s) of Entity/Table "Definition Type"	142
Attribute/Column(s) of Entity/Table "Definition Version"	142
Attribute/Column(s) of Entity/Table "Event Workflow Process Definition"	142
Attribute/Column(s) of Entity/Table "External Data Source Definition Database Object Definition"	142
Attribute/Column(s) of Entity/Table "Field Datatype"	142
Attribute/Column(s) of Entity/Table "Field Definition"	142
Attribute/Column(s) of Entity/Table "Field Dependency Rule"	142
Attribute/Column(s) of Entity/Table "Item"	142
Attribute/Column(s) of Entity/Table "Item Definition"	143
Attribute/Column(s) of Entity/Table "Item Field"	143
Attribute/Column(s) of Entity/Table "Item Field Definition"	143
Attribute/Column(s) of Entity/Table "Item Relationship"	143
Attribute/Column(s) of Entity/Table "Item Vote"	143
Attribute/Column(s) of Entity/Table "Knowledge Base Container"	143
Attribute/Column(s) of Entity/Table "License Type"	143
Attribute/Column(s) of Entity/Table "Locale"	143
Attribute/Column(s) of Entity/Table "Localized Column"	143
Attribute/Column(s) of Entity/Table "Localized Column Value"	144
Attribute/Column(s) of Entity/Table "Master Contact Item"	144
Attribute/Column(s) of Entity/Table "Request For Change Item"	144
Attribute/Column(s) of Entity/Table "Schedule"	144
Attribute/Column(s) of Entity/Table "Security Right"	144
Attribute/Column(s) of Entity/Table "Service Item"	144
Attribute/Column(s) of Entity/Table "Service Level Target Item"	144
Attribute/Column(s) of Entity/Table "Service Portfolio Container"	144
Attribute/Column(s) of Entity/Table "Solution Item"	144
Attribute/Column(s) of Entity/Table "String Resource Value"	145
Attribute/Column(s) of Entity/Table "Survey Item"	145
Attribute/Column(s) of Entity/Table "System Environment"	145
Attribute/Column(s) of Entity/Table "System Log"	145
Attribute/Column(s) of Entity/Table "Template"	145
Attribute/Column(s) of Entity/Table "Tenant Partition"	145
Attribute/Column(s) of Entity/Table "Tenant Partition Definition"	145
Attribute/Column(s) of Entity/Table "Ticket Item"	145
Attribute/Column(s) of Entity/Table "UI Element"	145
Attribute/Column(s) of Entity/Table "UI Element Definition"	145
Attribute/Column(s) of Entity/Table "UI Field"	145
Attribute/Column(s) of Entity/Table "UI Field Definition"	145
Attribute/Column(s) of Entity/Table "UI Form"	146
Attribute/Column(s) of Entity/Table "UI Form Definition"	146
Attribute/Column(s) of Entity/Table "UI Tab"	146
Attribute/Column(s) of Entity/Table "UI Tab Definition"	146
Attribute/Column(s) of Entity/Table "User Preference Definition"	146
Attribute/Column(s) of Entity/Table "User Profile"	146
Attribute/Column(s) of Entity/Table "Valid Values Rule"	146
Attribute/Column(s) of Entity/Table "Validation Rule"	146
Attribute/Column(s) of Entity/Table "Workflow Process Definition"	146
Attribute/Column(s) of Entity/Table "Workflow Process Definition Workflow Action"	147
Attribute/Column(s) of Entity/Table "Workspace Container"	147
Attribute/Column(s) of Entity/Table "WorkTarget Item"	147
ER Diagram(s) of Subject Area "010 - Translation"	147
Entity/Table(s) of Subject Area "010 - Translation"	147
Attribute/Column(s) of Entity/Table "Application Element"	148
Attribute/Column(s) of Entity/Table "Application Element Namespace"	148
Attribute/Column(s) of Entity/Table "Application Element String"	148
Attribute/Column(s) of Entity/Table "Application Element String Usage"	148
Attribute/Column(s) of Entity/Table "Application Element Type"	149
Attribute/Column(s) of Entity/Table "Internal Identifier Type"	149
Attribute/Column(s) of Entity/Table "Locale"	149
Attribute/Column(s) of Entity/Table "String Resource Value"	149
Attribute/Column(s) of Entity/Table "String Usage Type"	149
ER Diagram(s) of Subject Area "040 - Container"	149
Entity/Table(s) of Subject Area "040 - Container"	149
Attribute/Column(s) of Entity/Table "Address Book Container"	150
Attribute/Column(s) of Entity/Table "CMDB Container"	151
Attribute/Column(s) of Entity/Table "Container"	151
Attribute/Column(s) of Entity/Table "Container Definition"	151
Attribute/Column(s) of Entity/Table "Definition"	151
Attribute/Column(s) of Entity/Table "Field Definition"	151
Attribute/Column(s) of Entity/Table "Workspace Container"	151
ER Diagram(s) of Subject Area "041 - Definition Detail"	152
Entity/Table(s) of Subject Area "041 - Definition Detail"	152

Attribute/Column(s) of Entity/Table "Association Type"	153
Attribute/Column(s) of Entity/Table "Container Definition"	153
Attribute/Column(s) of Entity/Table "Definition"	153
Attribute/Column(s) of Entity/Table "Definition Relationship"	154
Attribute/Column(s) of Entity/Table "Definition Relationship Type"	154
Attribute/Column(s) of Entity/Table "Definition Subtype"	154
Attribute/Column(s) of Entity/Table "Definition Type"	154
Attribute/Column(s) of Entity/Table "Definition Version"	154
Attribute/Column(s) of Entity/Table "Definition Version Association"	154
Attribute/Column(s) of Entity/Table "Definition Version Status"	155
Attribute/Column(s) of Entity/Table "Field Datatype"	155
Attribute/Column(s) of Entity/Table "Field Definition"	155
Attribute/Column(s) of Entity/Table "Item Definition"	155
Attribute/Column(s) of Entity/Table "Item Field Definition"	155
Attribute/Column(s) of Entity/Table "Tenant Partition Definition"	155
ER Diagram(s) of Subject Area "046 - Workspace Item Assignment"	155
Entity/Table(s) of Subject Area "046 - Workspace Item Assignment"	155
Attribute/Column(s) of Entity/Table "Application User"	157
Attribute/Column(s) of Entity/Table "Assignment"	157
Attribute/Column(s) of Entity/Table "Assignment Type"	158
Attribute/Column(s) of Entity/Table "Configuration Item"	158
Attribute/Column(s) of Entity/Table "Contact Item"	158
Attribute/Column(s) of Entity/Table "Definition"	158
Attribute/Column(s) of Entity/Table "Item"	158
Attribute/Column(s) of Entity/Table "Item Field"	158
Attribute/Column(s) of Entity/Table "Team"	158
Attribute/Column(s) of Entity/Table "Ticket Item"	158
ER Diagram(s) of Subject Area "047 - Workspace Time Tracking"	159
Entity/Table(s) of Subject Area "047 - Workspace Time Tracking"	159
Attribute/Column(s) of Entity/Table "Application User"	160
Attribute/Column(s) of Entity/Table "Definition"	160
Attribute/Column(s) of Entity/Table "Item"	160
Attribute/Column(s) of Entity/Table "Time Tracking"	160
Attribute/Column(s) of Entity/Table "Time Tracking Type"	161
ER Diagram(s) of Subject Area "048 - Item Field Journaling"	161
Entity/Table(s) of Subject Area "048 - Item Field Journaling"	161
Attribute/Column(s) of Entity/Table "Application User"	162
Attribute/Column(s) of Entity/Table "Item"	162
Attribute/Column(s) of Entity/Table "Item Definition"	162
Attribute/Column(s) of Entity/Table "Item Field"	162
Attribute/Column(s) of Entity/Table "Item Field Definition"	162
Attribute/Column(s) of Entity/Table "Item Field Journal"	163
Attribute/Column(s) of Entity/Table "Ticket Item"	163
ER Diagram(s) of Subject Area "050 - Item Relationship"	163
Entity/Table(s) of Subject Area "050 - Item Relationship"	163
Attribute/Column(s) of Entity/Table "Item"	164
Attribute/Column(s) of Entity/Table "Item Definition"	164
Attribute/Column(s) of Entity/Table "Item Relationship"	164
Attribute/Column(s) of Entity/Table "Item Relationship Definition"	164
Attribute/Column(s) of Entity/Table "Link Type Definition Version"	165
ER Diagram(s) of Subject Area "051 - Item Field Collection"	165
Entity/Table(s) of Subject Area "051 - Item Field Collection"	165
Attribute/Column(s) of Entity/Table "Item"	166
Attribute/Column(s) of Entity/Table "Item Definition"	166
Attribute/Column(s) of Entity/Table "Item Field"	166
Attribute/Column(s) of Entity/Table "Item Field Collection"	166
Attribute/Column(s) of Entity/Table "Item Field Definition"	166
ER Diagram(s) of Subject Area "060 - Role"	166
Entity/Table(s) of Subject Area "060 - Role"	166
Attribute/Column(s) of Entity/Table "Administrator CMDB Role"	168
Attribute/Column(s) of Entity/Table "Administrator Type"	168
Attribute/Column(s) of Entity/Table "Agent Role"	168
Attribute/Column(s) of Entity/Table "Application Role"	169
Attribute/Column(s) of Entity/Table "CMDB Role"	169
Attribute/Column(s) of Entity/Table "Configuration Setting Value"	169
Attribute/Column(s) of Entity/Table "Create Edit CMDB Role"	169
Attribute/Column(s) of Entity/Table "Custom CMDB Role"	170
Attribute/Column(s) of Entity/Table "Customer Read Knowledge Base and Requests Role"	170
Attribute/Column(s) of Entity/Table "Customer Read Knowledge Base Role"	170
Attribute/Column(s) of Entity/Table "Customer Read/Submit Role"	170
Attribute/Column(s) of Entity/Table "Customer Read/Submit/Edit Role"	170
Attribute/Column(s) of Entity/Table "Customer Role"	170
Attribute/Column(s) of Entity/Table "Definition"	170
Attribute/Column(s) of Entity/Table "Definition Subtype"	170

Attribute/Column(s) of Entity/Table "Read Only CMDB Role"	171
Attribute/Column(s) of Entity/Table "Role Group"	171
Attribute/Column(s) of Entity/Table "Role Group Role"	171
Attribute/Column(s) of Entity/Table "System Administrator Role"	171
Attribute/Column(s) of Entity/Table "Template Application Role"	171
Attribute/Column(s) of Entity/Table "User Defined Agent Role"	171
Attribute/Column(s) of Entity/Table "User Defined Customer Role"	171
Attribute/Column(s) of Entity/Table "User Profile"	171
Attribute/Column(s) of Entity/Table "User Type"	172
Attribute/Column(s) of Entity/Table "Workspace Administrator Role"	172
ER Diagram(s) of Subject Area "065 - Configuration Setting"	172
Entity/Table(s) of Subject Area "065 - Configuration Setting"	172
Attribute/Column(s) of Entity/Table "Application Role"	174
Attribute/Column(s) of Entity/Table "Configuration Allowed Value"	174
Attribute/Column(s) of Entity/Table "Configuration Area Type"	175
Attribute/Column(s) of Entity/Table "Configuration Category"	175
Attribute/Column(s) of Entity/Table "Configuration Category Type"	175
Attribute/Column(s) of Entity/Table "Configuration List Type"	175
Attribute/Column(s) of Entity/Table "Configuration Object"	176
Attribute/Column(s) of Entity/Table "Configuration Setting"	176
Attribute/Column(s) of Entity/Table "Configuration Setting Owner Type"	176
Attribute/Column(s) of Entity/Table "Configuration Setting Value"	176
Attribute/Column(s) of Entity/Table "Configuration Value List"	177
Attribute/Column(s) of Entity/Table "Default Configuration Setting Value"	177
Attribute/Column(s) of Entity/Table "Definition"	177
Attribute/Column(s) of Entity/Table "Definition Version"	178
Attribute/Column(s) of Entity/Table "tmp_cfg_setting_val"	178
Attribute/Column(s) of Entity/Table "User Profile"	178
Attribute/Column(s) of Entity/Table "Value Type Code"	179
ER Diagram(s) of Subject Area "066 - Container Snapshot"	179
Entity/Table(s) of Subject Area "066 - Container Snapshot"	179
Attribute/Column(s) of Entity/Table "Configuration Setting"	181
Attribute/Column(s) of Entity/Table "Configuration Setting Value"	182
Attribute/Column(s) of Entity/Table "Container"	182
Attribute/Column(s) of Entity/Table "Container Definition"	183
Attribute/Column(s) of Entity/Table "Container Snapshot"	183
Attribute/Column(s) of Entity/Table "Definition"	183
Attribute/Column(s) of Entity/Table "Definition Lock"	183
Attribute/Column(s) of Entity/Table "Definition Relationship"	183
Attribute/Column(s) of Entity/Table "Definition Relationship Type"	183
Attribute/Column(s) of Entity/Table "Definition Version"	184
Attribute/Column(s) of Entity/Table "Definition Version Association"	184
Attribute/Column(s) of Entity/Table "Definition Version Status"	184
Attribute/Column(s) of Entity/Table "Dynamic Table Script History"	184
Attribute/Column(s) of Entity/Table "Field Definition"	184
Attribute/Column(s) of Entity/Table "Snapshot Configuration Setting Value"	184
Attribute/Column(s) of Entity/Table "Snapshot Definition"	185
Attribute/Column(s) of Entity/Table "Snapshot Definition Relationship"	185
Attribute/Column(s) of Entity/Table "Snapshot Definition Version"	185
Attribute/Column(s) of Entity/Table "Snapshot Definition Version Assoc"	186
Attribute/Column(s) of Entity/Table "Snapshot Status"	186
Attribute/Column(s) of Entity/Table "Snapshot Type"	186
Attribute/Column(s) of Entity/Table "Workspace Container"	186
ER Diagram(s) of Subject Area "070 - User"	186
Entity/Table(s) of Subject Area "070 - User"	186
Attribute/Column(s) of Entity/Table "Agent"	187
Attribute/Column(s) of Entity/Table "Anonymous Customer"	187
Attribute/Column(s) of Entity/Table "Application User"	187
Attribute/Column(s) of Entity/Table "Application User Profile Link"	187
Attribute/Column(s) of Entity/Table "Customer"	188
Attribute/Column(s) of Entity/Table "Customer Login Type"	188
Attribute/Column(s) of Entity/Table "License Type"	188
Attribute/Column(s) of Entity/Table "Shared Customer"	188
Attribute/Column(s) of Entity/Table "Unique Customer"	188
Attribute/Column(s) of Entity/Table "User Profile"	188
ER Diagram(s) of Subject Area "071 - Team"	189
Entity/Table(s) of Subject Area "071 - Team"	189
Attribute/Column(s) of Entity/Table "Agent Team"	190
Attribute/Column(s) of Entity/Table "Application User"	190
Attribute/Column(s) of Entity/Table "Assignment"	190
Attribute/Column(s) of Entity/Table "Assignment Type"	191
Attribute/Column(s) of Entity/Table "Definition"	191
Attribute/Column(s) of Entity/Table "Item"	191
Attribute/Column(s) of Entity/Table "Team"	191

Attribute/Column(s) of Entity/Table "Team Email Notification Type"	192
Attribute/Column(s) of Entity/Table "Team Item Association"	192
ER Diagram(s) of Subject Area "080 - User Profile"	192
Entity/Table(s) of Subject Area "080 - User Profile"	192
Attribute/Column(s) of Entity/Table "Agent Authentication Type"	194
Attribute/Column(s) of Entity/Table "Agent Profile"	194
Attribute/Column(s) of Entity/Table "Anonymous Customer Profile"	194
Attribute/Column(s) of Entity/Table "Application Role"	194
Attribute/Column(s) of Entity/Table "Application User"	195
Attribute/Column(s) of Entity/Table "Application User Profile Link"	195
Attribute/Column(s) of Entity/Table "Auto Add Customer Profile"	195
Attribute/Column(s) of Entity/Table "Customer Authentication Type"	195
Attribute/Column(s) of Entity/Table "Customer Profile"	195
Attribute/Column(s) of Entity/Table "Definition"	195
Attribute/Column(s) of Entity/Table "Definition Subtype"	195
Attribute/Column(s) of Entity/Table "Definition Type"	196
Attribute/Column(s) of Entity/Table "Role Group"	196
Attribute/Column(s) of Entity/Table "Shared Customer Profile"	196
Attribute/Column(s) of Entity/Table "Standard Agent Profile"	196
Attribute/Column(s) of Entity/Table "System Administration Profile"	196
Attribute/Column(s) of Entity/Table "Unique Customer Profile"	196
Attribute/Column(s) of Entity/Table "User Profile"	196
Attribute/Column(s) of Entity/Table "User Type"	197
Attribute/Column(s) of Entity/Table "Workspace Administration Profile"	197
ER Diagram(s) of Subject Area "096 - Email Notification"	197
Entity/Table(s) of Subject Area "096 - Email Notification"	197
Attribute/Column(s) of Entity/Table "Email Notification"	197
Attribute/Column(s) of Entity/Table "Email Notification Attachment"	198
Attribute/Column(s) of Entity/Table "Email Notification Type"	198
Attribute/Column(s) of Entity/Table "Notification Recipient Status"	198
ER Diagram(s) of Subject Area "101 - Data Access System"	198
Entity/Table(s) of Subject Area "101 - Data Access System"	198
Attribute/Column(s) of Entity/Table "Database Data Type"	199
Attribute/Column(s) of Entity/Table "Dynamic Column Constraint Specification"	199
Attribute/Column(s) of Entity/Table "Dynamic Column Specification"	199
Attribute/Column(s) of Entity/Table "Dynamic Table Index"	199
Attribute/Column(s) of Entity/Table "Dynamic Table Index Column"	200
Attribute/Column(s) of Entity/Table "Dynamic Table Specification"	200
Attribute/Column(s) of Entity/Table "Index Type"	200
Attribute/Column(s) of Entity/Table "Logical Data Type"	200
Attribute/Column(s) of Entity/Table "Target DBMS"	200
ER Diagram(s) of Subject Area "102 - Data Access Scripts"	200
Entity/Table(s) of Subject Area "102 - Data Access Scripts"	200
Attribute/Column(s) of Entity/Table "Container Definition"	201
Attribute/Column(s) of Entity/Table "Definition Subtype"	201
Attribute/Column(s) of Entity/Table "Dynamic Table Script"	202
Attribute/Column(s) of Entity/Table "Dynamic Table Script History"	202
Attribute/Column(s) of Entity/Table "Dynamic Table Script Type"	202
Attribute/Column(s) of Entity/Table "Item Definition"	202
Attribute/Column(s) of Entity/Table "Target DBMS"	202
ER Diagram(s) of Subject Area "110 - Id Generation"	203
Entity/Table(s) of Subject Area "110 - Id Generation"	203
Attribute/Column(s) of Entity/Table "Id Generator"	203
ER Diagram(s) of Subject Area "120 - Job Scheduler"	203
Entity/Table(s) of Subject Area "120 - Job Scheduler"	203
Attribute/Column(s) of Entity/Table "Job Configuration Detail"	203
Attribute/Column(s) of Entity/Table "Job Scheduler Configuration"	203
ER Diagram(s) of Subject Area "140 - Workspace Calendar and Operation Schedule"	204
Entity/Table(s) of Subject Area "140 - Workspace Calendar and Operation Schedule"	204
Attribute/Column(s) of Entity/Table "Agent"	205
Attribute/Column(s) of Entity/Table "Calendar Item"	205
Attribute/Column(s) of Entity/Table "Calendar Schedule"	206
Attribute/Column(s) of Entity/Table "Calendar Type"	206
Attribute/Column(s) of Entity/Table "Schedule"	206
Attribute/Column(s) of Entity/Table "Schedule Actual Period"	206
Attribute/Column(s) of Entity/Table "Time Zone Local Hour"	207
Attribute/Column(s) of Entity/Table "Time Zone Region"	207
Attribute/Column(s) of Entity/Table "Workspace Container"	207
ER Diagram(s) of Subject Area "141 - Calendar Appointments and Items"	208
Entity/Table(s) of Subject Area "141 - Calendar Appointments and Items"	208
Attribute/Column(s) of Entity/Table "Application User"	210
Attribute/Column(s) of Entity/Table "Calendar Collection Item"	210
Attribute/Column(s) of Entity/Table "Calendar Container Item"	210
Attribute/Column(s) of Entity/Table "Calendar Content Data"	211

Attribute/Column(s) of Entity/Table "Calendar Event Stamp"	211
Attribute/Column(s) of Entity/Table "Calendar Field"	211
Attribute/Column(s) of Entity/Table "Calendar Field Dictionary Value"	211
Attribute/Column(s) of Entity/Table "Calendar Field Multistring Value"	211
Attribute/Column(s) of Entity/Table "Calendar Item"	211
Attribute/Column(s) of Entity/Table "Calendar Schedule"	212
Attribute/Column(s) of Entity/Table "Calendar Stamp"	212
Attribute/Column(s) of Entity/Table "Calendar Subscription"	212
Attribute/Column(s) of Entity/Table "Calendar Ticket"	212
Attribute/Column(s) of Entity/Table "Calendar Ticket Privilege"	213
Attribute/Column(s) of Entity/Table "Calendar Workspace"	213
Attribute/Column(s) of Entity/Table "Definition"	213
Attribute/Column(s) of Entity/Table "Item"	213
ER Diagram(s) of Subject Area "142 - Calendar Sync"	213
Entity/Table(s) of Subject Area "142 - Calendar Sync"	213
Attribute/Column(s) of Entity/Table "Application User"	214
Attribute/Column(s) of Entity/Table "Calendar Item"	214
Attribute/Column(s) of Entity/Table "Definition"	215
Attribute/Column(s) of Entity/Table "PIM Appointment Information"	215
Attribute/Column(s) of Entity/Table "PIM Record Information"	215
Attribute/Column(s) of Entity/Table "Syncing Job Detail"	216
ER Diagram(s) of Subject Area "160 - Templating"	216
Entity/Table(s) of Subject Area "160 - Templating"	216
Attribute/Column(s) of Entity/Table "Application Element"	217
Attribute/Column(s) of Entity/Table "Definition"	217
Attribute/Column(s) of Entity/Table "Item Field Definition"	217
Attribute/Column(s) of Entity/Table "Template Attribute Definition"	217
Attribute/Column(s) of Entity/Table "Template Category"	217
Attribute/Column(s) of Entity/Table "Template Content Type"	217
Attribute/Column(s) of Entity/Table "Template Definition"	218
Attribute/Column(s) of Entity/Table "Template Definition Attribute"	218
Attribute/Column(s) of Entity/Table "Template Group"	218
ER Diagram(s) of Subject Area "170 - Knowledge Base"	218
Entity/Table(s) of Subject Area "170 - Knowledge Base"	218
Attribute/Column(s) of Entity/Table "Application User"	219
Attribute/Column(s) of Entity/Table "Definition"	219
Attribute/Column(s) of Entity/Table "Item"	220
Attribute/Column(s) of Entity/Table "Item Definition"	220
Attribute/Column(s) of Entity/Table "Item Useful Detail"	220
Attribute/Column(s) of Entity/Table "Item Useful Summary"	220
Attribute/Column(s) of Entity/Table "Solution Item"	220
Attribute/Column(s) of Entity/Table "Useful Code"	220
ER Diagram(s) of Subject Area "190 - Password Reset"	221
Entity/Table(s) of Subject Area "190 - Password Reset"	221
Attribute/Column(s) of Entity/Table "Application User"	221
Attribute/Column(s) of Entity/Table "Password Reset Question"	221
Attribute/Column(s) of Entity/Table "Password Reset Question Answer"	221
Attribute/Column(s) of Entity/Table "Password Reset Registered Account"	221
ER Diagram(s) of Subject Area "210 - Audit Trail"	222
Entity/Table(s) of Subject Area "210 - Audit Trail"	222
Attribute/Column(s) of Entity/Table "Application User"	223
Attribute/Column(s) of Entity/Table "Audit Event Type"	223
Attribute/Column(s) of Entity/Table "Audit Field Configuration"	223
Attribute/Column(s) of Entity/Table "Audit Log"	223
Attribute/Column(s) of Entity/Table "Audit Message Template"	224
Attribute/Column(s) of Entity/Table "Audit Property Change"	224
Attribute/Column(s) of Entity/Table "Audit Property Dimension"	225
Attribute/Column(s) of Entity/Table "Audit Rule Name"	225
Attribute/Column(s) of Entity/Table "Audit Transaction"	225
Attribute/Column(s) of Entity/Table "Item Field Definition"	226
Attribute/Column(s) of Entity/Table "System Component"	226
Attribute/Column(s) of Entity/Table "Value Type Code"	226
ER Diagram(s) of Subject Area "230 - Rules"	226
Entity/Table(s) of Subject Area "230 - Rules"	226
Attribute/Column(s) of Entity/Table "Rule Usage Statistic"	226
ER Diagram(s) of Subject Area "240 - File Storage"	226
Entity/Table(s) of Subject Area "240 - File Storage"	226
Attribute/Column(s) of Entity/Table "Definition"	227
Attribute/Column(s) of Entity/Table "File Information"	227
Attribute/Column(s) of Entity/Table "Item Attachment"	227
ER Diagram(s) of Subject Area "250 - Apache Jack Rabbit JCR"	227
Entity/Table(s) of Subject Area "250 - Apache Jack Rabbit JCR"	227
Attribute/Column(s) of Entity/Table "Default Binval"	228
Attribute/Column(s) of Entity/Table "Default Bundle"	228

Attribute/Column(s) of Entity/Table "Default Names"	228
Attribute/Column(s) of Entity/Table "Default Refs"	228
Attribute/Column(s) of Entity/Table "Security Binval"	228
Attribute/Column(s) of Entity/Table "Security Bundle"	228
Attribute/Column(s) of Entity/Table "Security Names"	228
Attribute/Column(s) of Entity/Table "Security Refs"	228
ER Diagram(s) of Subject Area "262 - Import Items and Users"	228
Entity/Table(s) of Subject Area "262 - Import Items and Users"	228
Attribute/Column(s) of Entity/Table "Application Role"	230
Attribute/Column(s) of Entity/Table "Application User"	230
Attribute/Column(s) of Entity/Table "Definition"	230
Attribute/Column(s) of Entity/Table "Import"	230
Attribute/Column(s) of Entity/Table "Import Assignment"	231
Attribute/Column(s) of Entity/Table "Import Definition Rule"	231
Attribute/Column(s) of Entity/Table "Import Error Result"	231
Attribute/Column(s) of Entity/Table "Import Execution Record"	231
Attribute/Column(s) of Entity/Table "Import Scheduled Task"	232
Attribute/Column(s) of Entity/Table "Import User Workspace Role"	232
Attribute/Column(s) of Entity/Table "Scheduled Task"	232
Attribute/Column(s) of Entity/Table "Scheduled Task Notification"	232
Attribute/Column(s) of Entity/Table "Team"	232
ER Diagram(s) of Subject Area "280 - Session and Locking"	233
Entity/Table(s) of Subject Area "280 - Session and Locking"	233
Attribute/Column(s) of Entity/Table "Application User"	234
Attribute/Column(s) of Entity/Table "Container Definition"	234
Attribute/Column(s) of Entity/Table "Definition Lock"	234
Attribute/Column(s) of Entity/Table "Item"	234
Attribute/Column(s) of Entity/Table "Item Definition"	234
Attribute/Column(s) of Entity/Table "Item Lock"	234
Attribute/Column(s) of Entity/Table "System Session"	235
ER Diagram(s) of Subject Area "290 - Security"	235
Entity/Table(s) of Subject Area "290 - Security"	235
Attribute/Column(s) of Entity/Table "Security Policy"	235
Attribute/Column(s) of Entity/Table "Target Match"	235
Attribute/Column(s) of Entity/Table "Target Match Type"	236
ER Diagram(s) of Subject Area "300 - Ticket Data Mart"	236
Entity/Table(s) of Subject Area "300 - Ticket Data Mart"	236
Attribute/Column(s) of Entity/Table ".:Item_Shortname Balance"	237
Attribute/Column(s) of Entity/Table ".:item_shortname Event Sum"	238
Attribute/Column(s) of Entity/Table ".:item_shortname Status"	238
Attribute/Column(s) of Entity/Table ".:item_shortname Transition"	239
Attribute/Column(s) of Entity/Table ".:item_shortname Transition Sum"	239
Attribute/Column(s) of Entity/Table "Life Cycle Event"	239
Attribute/Column(s) of Entity/Table "Previous Timing"	239
Attribute/Column(s) of Entity/Table "Time Dimension"	241
Attribute/Column(s) of Entity/Table "Time Zone Local Hour"	241
Attribute/Column(s) of Entity/Table "Time Zone Region"	242
ER Diagram(s) of Subject Area "305 - Data Mart Load"	242
Entity/Table(s) of Subject Area "305 - Data Mart Load"	242
Attribute/Column(s) of Entity/Table "Audit Transaction"	243
Attribute/Column(s) of Entity/Table "Data Load Type"	244
Attribute/Column(s) of Entity/Table "Data Mart Load History"	244
Attribute/Column(s) of Entity/Table "Data Mart Load Lock"	244
Attribute/Column(s) of Entity/Table "Data Mart Request History"	244
Attribute/Column(s) of Entity/Table "Data Mart Request Queue"	245
Attribute/Column(s) of Entity/Table "Data Mart Request State"	245
Attribute/Column(s) of Entity/Table "Data Mart Request Type"	245
Attribute/Column(s) of Entity/Table "Definition"	245
ER Diagram(s) of Subject Area "310 - Auxiliary Tables"	245
Entity/Table(s) of Subject Area "310 - Auxiliary Tables"	245
Attribute/Column(s) of Entity/Table "Configuration File Repository"	247
Attribute/Column(s) of Entity/Table "External System Certificate"	247
ER Diagram(s) of Subject Area "320 - Report"	247
Entity/Table(s) of Subject Area "320 - Report"	247
Attribute/Column(s) of Entity/Table "Application User"	248
Attribute/Column(s) of Entity/Table "Definition"	248
Attribute/Column(s) of Entity/Table "Definition Version"	249
Attribute/Column(s) of Entity/Table "Recur Period Type"	249
Attribute/Column(s) of Entity/Table "Recur Schedule Type"	249
Attribute/Column(s) of Entity/Table "Report"	249
Attribute/Column(s) of Entity/Table "Report Configuration"	250
Attribute/Column(s) of Entity/Table "Report Format"	250
Attribute/Column(s) of Entity/Table "Report Scheduled Task"	250
Attribute/Column(s) of Entity/Table "Report Visibility Type"	250

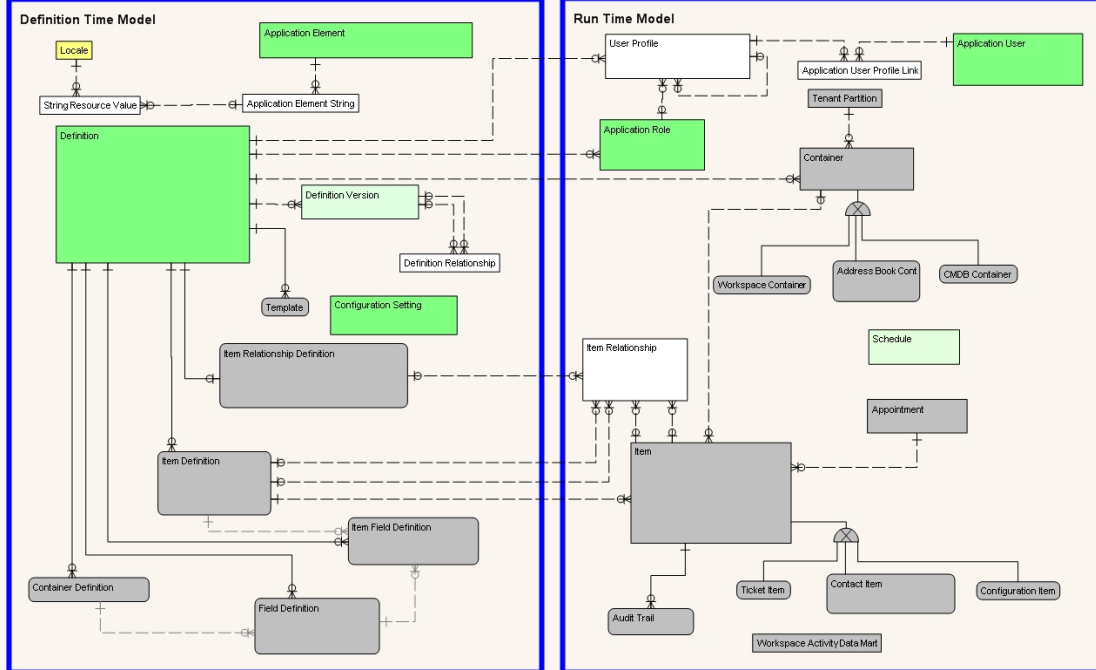
Attribute/Column(s) of Entity/Table "Schedule Task Run Statistic"	250
Attribute/Column(s) of Entity/Table "Scheduled Task"	251
Attribute/Column(s) of Entity/Table "Scheduled Task Notification"	251
Attribute/Column(s) of Entity/Table "Scheduled Task Run"	251
Attribute/Column(s) of Entity/Table "Scheduled Task Statistic"	251
Attribute/Column(s) of Entity/Table "Scheduled Task Status"	252
Attribute/Column(s) of Entity/Table "Scheduled Task Type"	252
Attribute/Column(s) of Entity/Table "Value Type Code"	252
ER Diagram(s) of Subject Area "332 - Change Management Voting"	252
Entity/Table(s) of Subject Area "332 - Change Management Voting"	252
Attribute/Column(s) of Entity/Table "Application User"	253
Attribute/Column(s) of Entity/Table "Definition"	253
Attribute/Column(s) of Entity/Table "Item"	253
Attribute/Column(s) of Entity/Table "Vote"	254
Attribute/Column(s) of Entity/Table "Vote History"	254
Attribute/Column(s) of Entity/Table "Voting Result"	254
ER Diagram(s) of Subject Area "340 - Search"	254
Entity/Table(s) of Subject Area "340 - Search"	254
Attribute/Column(s) of Entity/Table "Application User"	255
Attribute/Column(s) of Entity/Table "Definition"	255
Attribute/Column(s) of Entity/Table "Definition Version"	256
Attribute/Column(s) of Entity/Table "Logical Data Type"	256
Attribute/Column(s) of Entity/Table "Search Column"	256
Attribute/Column(s) of Entity/Table "Search Condition"	256
Attribute/Column(s) of Entity/Table "Search Constraint Type"	256
Attribute/Column(s) of Entity/Table "Search Item Query"	256
Attribute/Column(s) of Entity/Table "Search Query"	257
Attribute/Column(s) of Entity/Table "Search View"	257
ER Diagram(s) of Subject Area "350 - Chat"	257
Entity/Table(s) of Subject Area "350 - Chat"	257
Attribute/Column(s) of Entity/Table "Alert Role Type"	259
Attribute/Column(s) of Entity/Table "Application User"	259
Attribute/Column(s) of Entity/Table "Broadcast Alert"	259
Attribute/Column(s) of Entity/Table "Broadcast Message"	259
Attribute/Column(s) of Entity/Table "Broadcast Message Type"	260
Attribute/Column(s) of Entity/Table "Chat Channel"	260
Attribute/Column(s) of Entity/Table "Chat Channel Subscriber"	260
Attribute/Column(s) of Entity/Table "Chat Message"	260
Attribute/Column(s) of Entity/Table "Chat Status"	260
Attribute/Column(s) of Entity/Table "Chat User Status"	260
Attribute/Column(s) of Entity/Table "Definition"	260
Attribute/Column(s) of Entity/Table "Definition Version"	261
Attribute/Column(s) of Entity/Table "Item"	261
Attribute/Column(s) of Entity/Table "Item Definition"	261
Attribute/Column(s) of Entity/Table "Ticket Alert"	261
Attribute/Column(s) of Entity/Table "Workspace Container"	261
ER Diagram(s) of Subject Area "360 - Survey"	262
Entity/Table(s) of Subject Area "360 - Survey"	262
Attribute/Column(s) of Entity/Table "Definition Version"	262
Attribute/Column(s) of Entity/Table "Survey Sent"	262
ER Diagram(s) of Subject Area "370 - Deployment"	262
Entity/Table(s) of Subject Area "370 - Deployment"	262
Attribute/Column(s) of Entity/Table "Deployment Update"	262
ER Diagram(s) of Subject Area "380 - Portal"	263
Entity/Table(s) of Subject Area "380 - Portal"	263
Attribute/Column(s) of Entity/Table "Application User"	263
Attribute/Column(s) of Entity/Table "Console"	263
Attribute/Column(s) of Entity/Table "Portal"	263
Attribute/Column(s) of Entity/Table "Portal Console"	264
Attribute/Column(s) of Entity/Table "Widget"	264
ER Diagram(s) of Subject Area "390 - Business Process Template"	264
Entity/Table(s) of Subject Area "390 - Business Process Template"	264
Attribute/Column(s) of Entity/Table "Business Process"	265
Attribute/Column(s) of Entity/Table "Business Process Definition"	265
Attribute/Column(s) of Entity/Table "Definition"	265
Attribute/Column(s) of Entity/Table "Definition Subtype"	265
Attribute/Column(s) of Entity/Table "Template Package"	266
Attribute/Column(s) of Entity/Table "Template Package Type"	266
ER Diagram(s) of Subject Area "400 - Asset Management"	266
Entity/Table(s) of Subject Area "400 - Asset Management"	266
Attribute/Column(s) of Entity/Table "Definition"	267
Attribute/Column(s) of Entity/Table "Item"	267
Attribute/Column(s) of Entity/Table "Pending Advanced Action"	267
ER Diagram(s) of Subject Area "410 - Publishing"	267

Entity/Table(s) of Subject Area "410 - Publishing"	267
Attribute/Column(s) of Entity/Table "Application User"	268
Attribute/Column(s) of Entity/Table "Definition"	268
Attribute/Column(s) of Entity/Table "Item"	269
Attribute/Column(s) of Entity/Table "Publish Error"	269
Attribute/Column(s) of Entity/Table "Publish Log"	269
ER Diagram(s) of Subject Area "430 - UI Component State"	269
Entity/Table(s) of Subject Area "430 - UI Component State"	269
Attribute/Column(s) of Entity/Table "Application User"	270
Attribute/Column(s) of Entity/Table "UI Component State"	270
ER Diagram(s) of Subject Area "900 To Be Deleted"	270
Entity/Table(s) of Subject Area "900 To Be Deleted"	270
Attribute/Column(s) of Entity/Table "Calendar Ticket Privilege"	271
Attribute/Column(s) of Entity/Table "Localized Column"	271
Attribute/Column(s) of Entity/Table "Localized Column Value"	271
Attribute/Column(s) of Entity/Table "Role Group"	271
Attribute/Column(s) of Entity/Table "Role Group Role"	271
Attribute/Column(s) of Entity/Table "vt\$item_fid_defn_NetworkAdapter"	272
Attribute/Column(s) of Entity/Table "vt\$item_fid_NetworkAdapter\$changes"	272
ER Diagram(s) of Subject Area "901 - Tables Added"	272
Entity/Table(s) of Subject Area "901 - Tables Added"	272
Attribute/Column(s) of Entity/Table "Container Snapshot"	272
Attribute/Column(s) of Entity/Table "Snapshot Configuration Setting Value"	273
Attribute/Column(s) of Entity/Table "Snapshot Definition"	273
Attribute/Column(s) of Entity/Table "Snapshot Definition Relationship"	273
Attribute/Column(s) of Entity/Table "Snapshot Definition Version"	274
Attribute/Column(s) of Entity/Table "Snapshot Definition Version Assoc"	274
Attribute/Column(s) of Entity/Table "Snapshot Status"	274
Attribute/Column(s) of Entity/Table "Snapshot Type"	274
ER Diagram(s) of Subject Area "902 - Columns Added"	274
Entity/Table(s) of Subject Area "902 - Columns Added"	274
Attribute/Column(s) of Entity/Table "Audit Transaction"	275
ER Diagram(s) of Subject Area "903 - Tables Deleted"	275
ER Diagram(s) of Subject Area "904 - Columns Deleted"	275
ER Diagram(s) of Subject Area "Legend"	275
Entity/Table(s) of Subject Area "Legend"	275
Attribute/Column(s) of Entity/Table "Added Column"	276
Attribute/Column(s) of Entity/Table "Added Table"	276
Attribute/Column(s) of Entity/Table "Configuration Setting"	276
Attribute/Column(s) of Entity/Table "Configuration Setting Owner Type"	277
Attribute/Column(s) of Entity/Table "Delete Entry"	277
Attribute/Column(s) of Entity/Table "Deleted Column"	277
Attribute/Column(s) of Entity/Table "Deleted Table"	277
Attribute/Column(s) of Entity/Table "Item Definition"	277
Attribute/Column(s) of Entity/Table "To"	277
Attribute/Column(s) of Entity/Table "To"	277
Attribute/Column(s) of Entity/Table "To"	277
Attribute/Column(s) of Entity/Table "To"	277
Attribute/Column(s) of Entity/Table "To One Entity"	278
Attribute/Column(s) of Entity/Table "To One or More Entities"	278
Attribute/Column(s) of Entity/Table "To Zero or More Entities"	278
Attribute/Column(s) of Entity/Table "To Zero or One Entity"	278
Index	

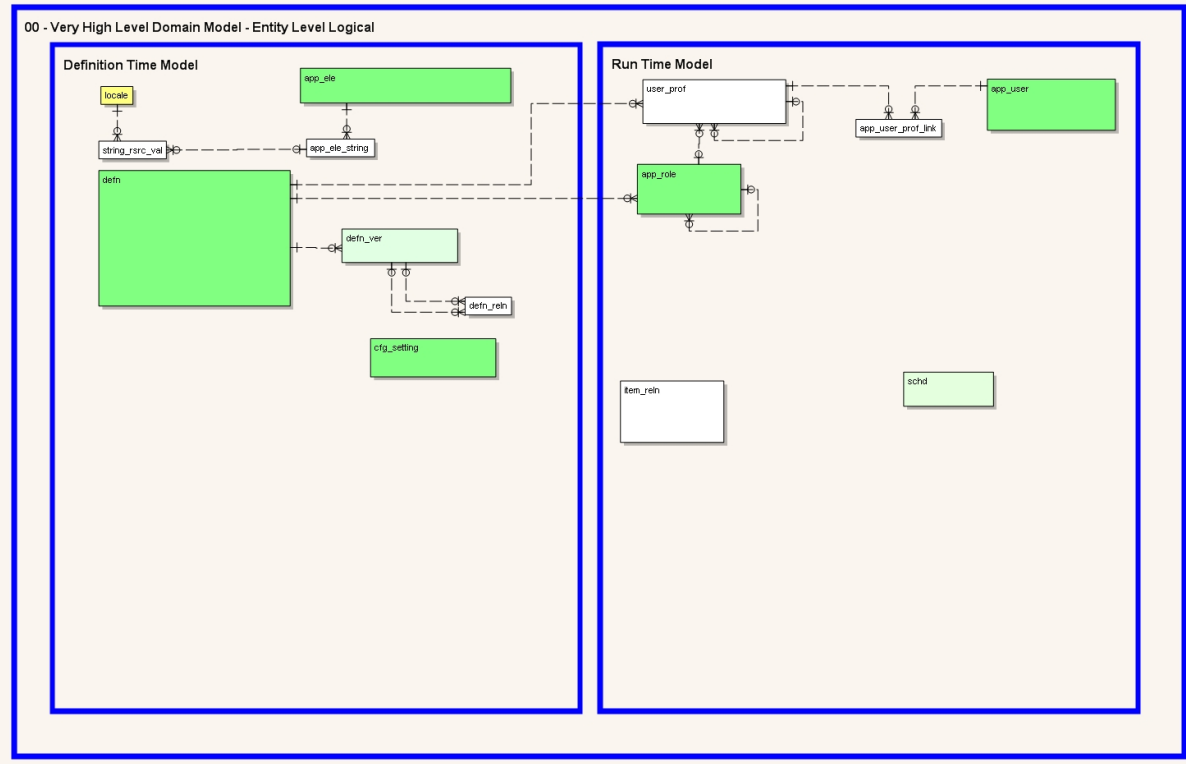


## Entity Level

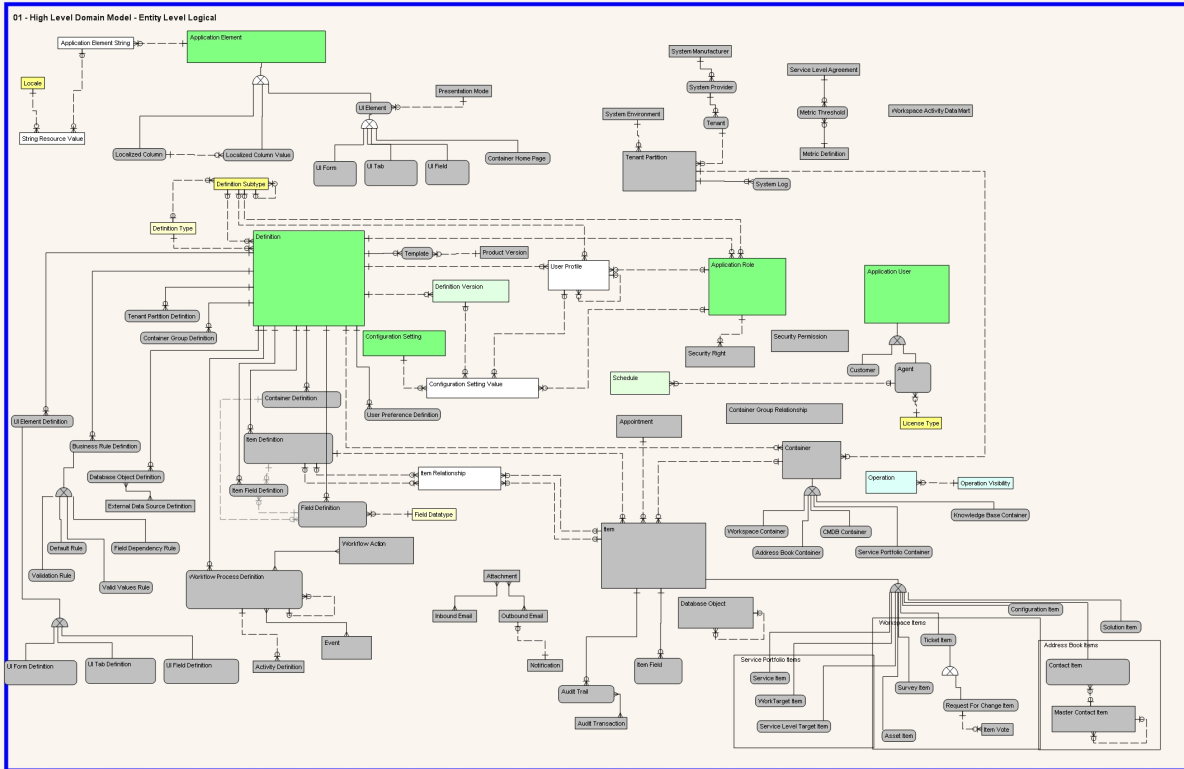
### 00 - Very High Level Domain Model - Entity Level Logical



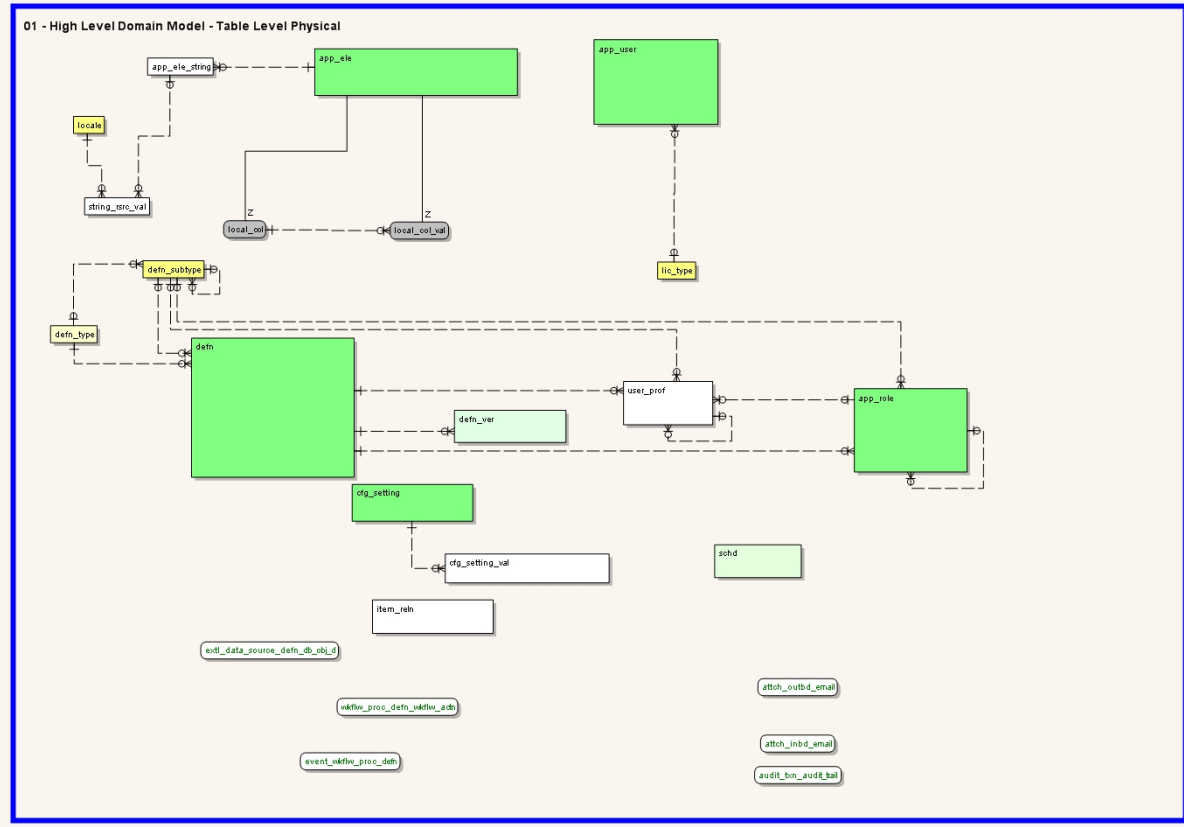
## Table Level



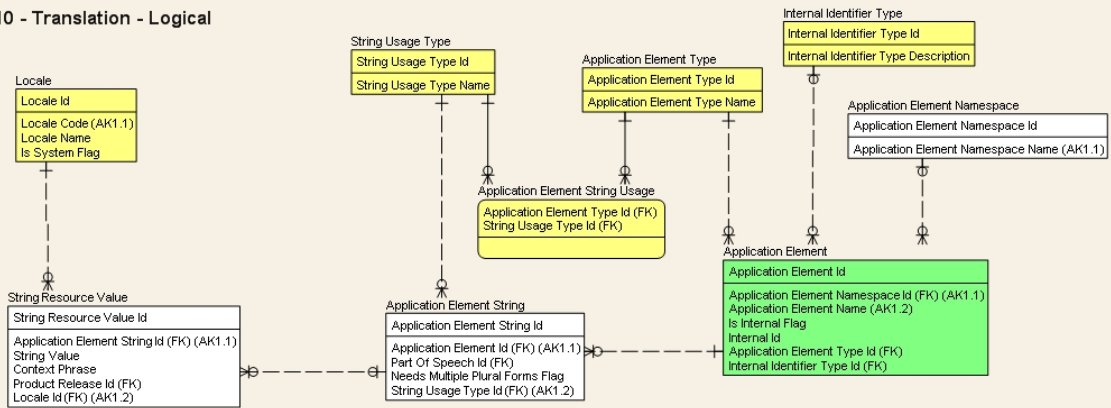
Entity Level
--------------



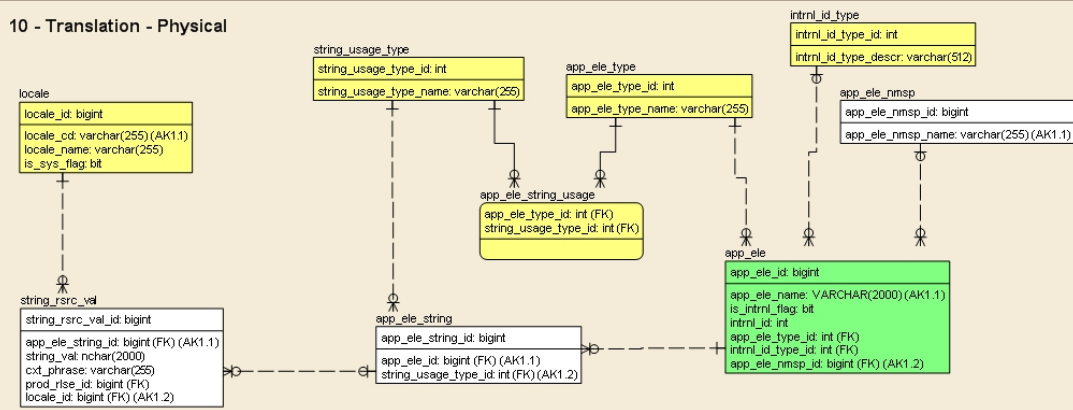
## Table Level

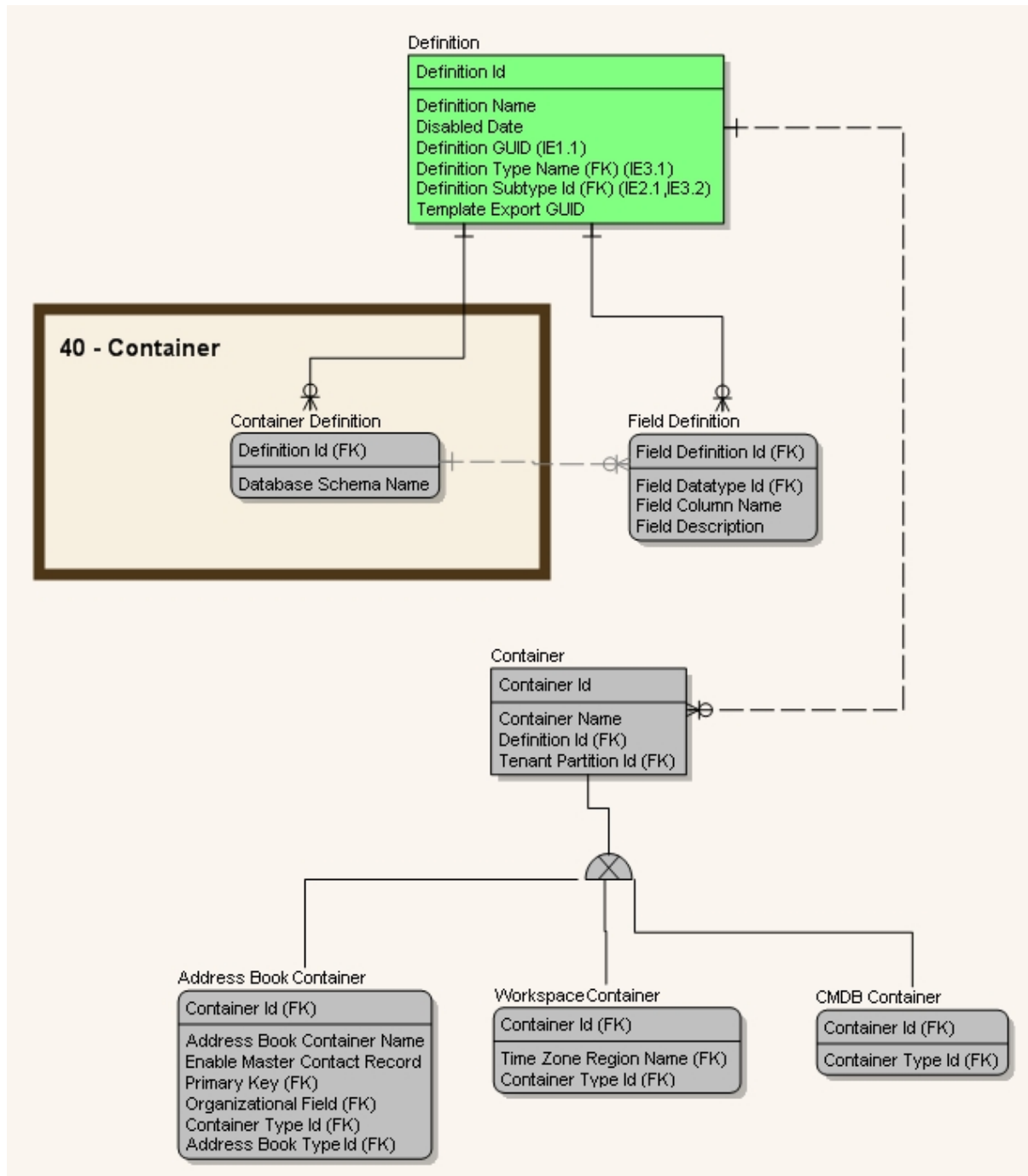


## 10 - Translation - Logical

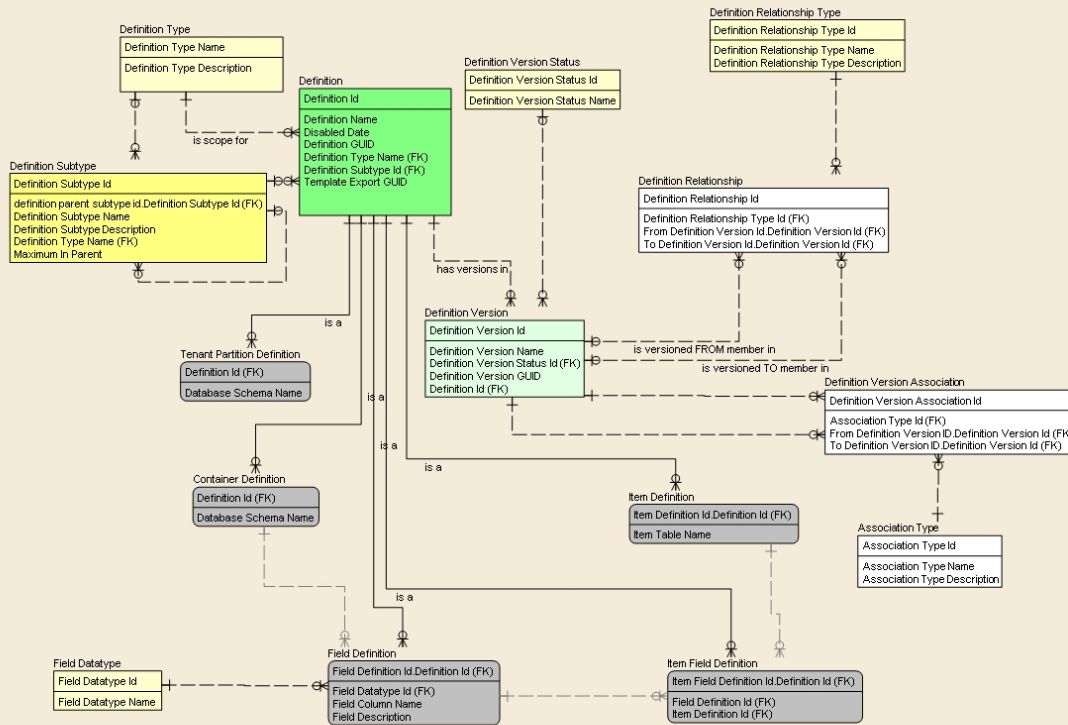


## 10 - Translation - Physical



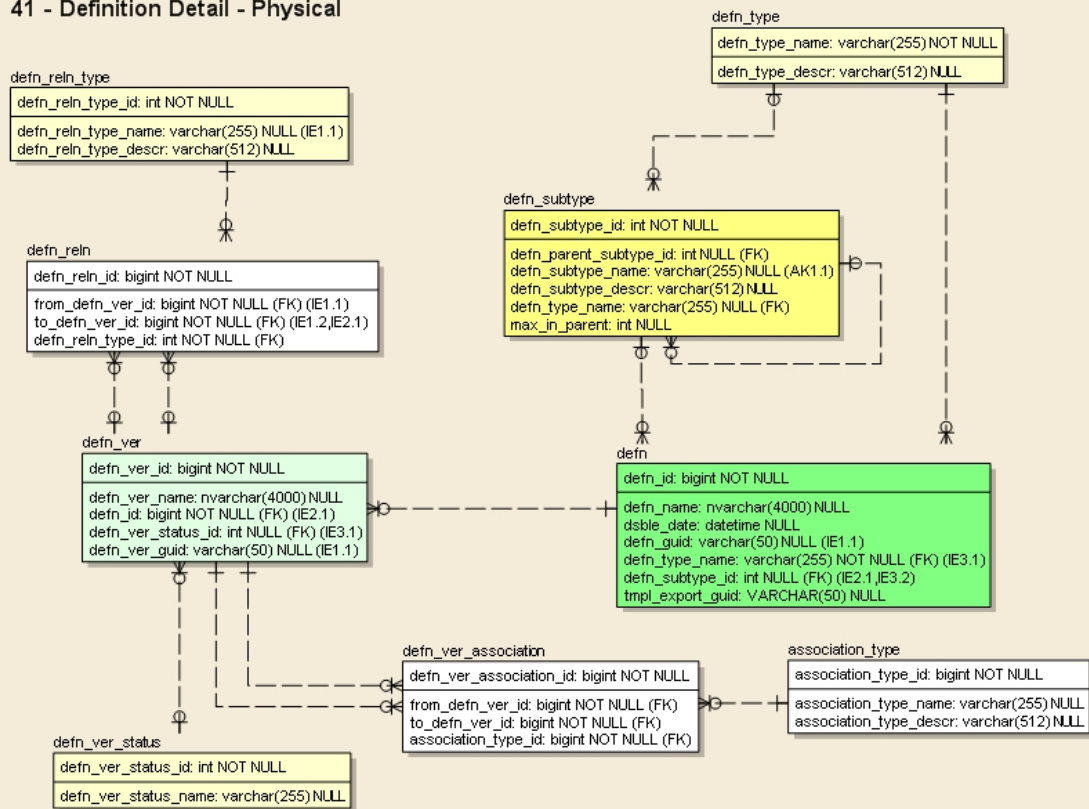


#### 41 - Definition Detail - Logical

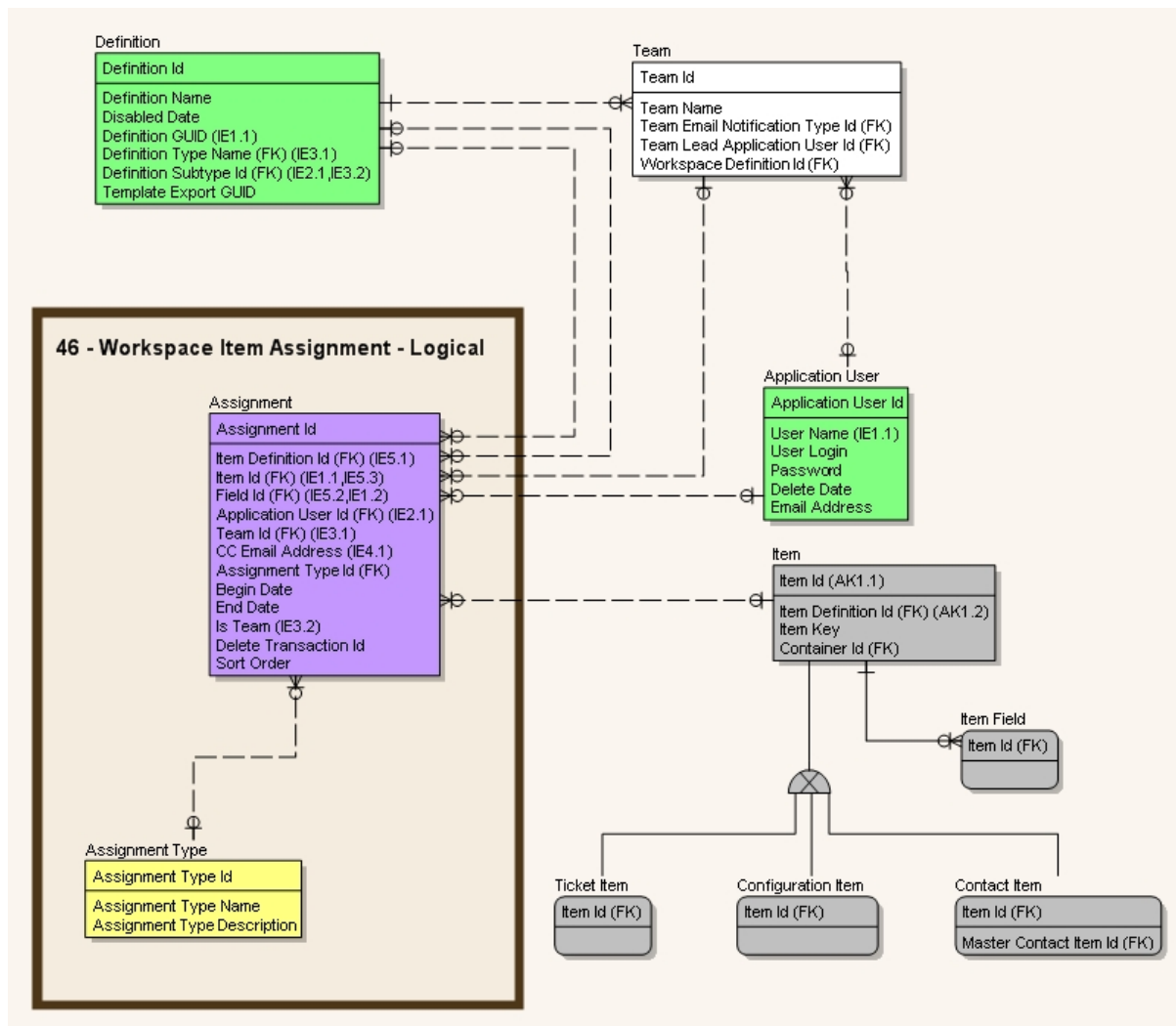




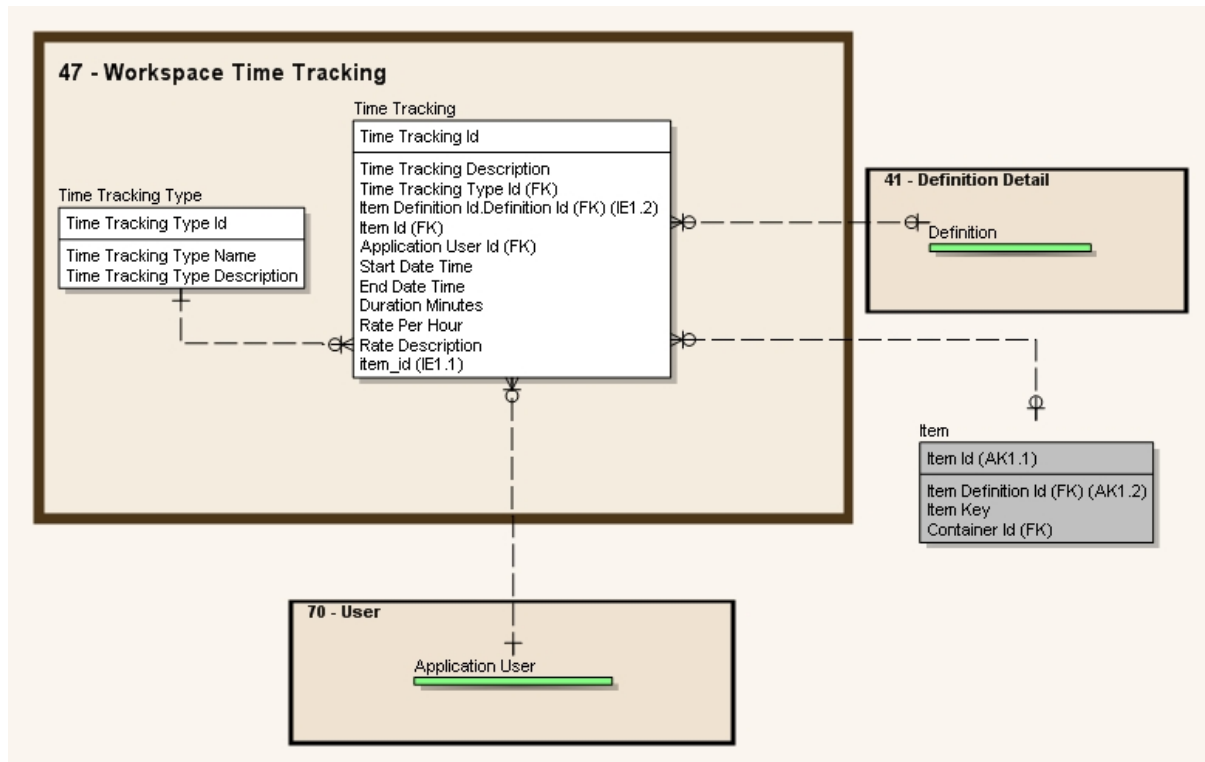
## 41 - Definition Detail - Physical

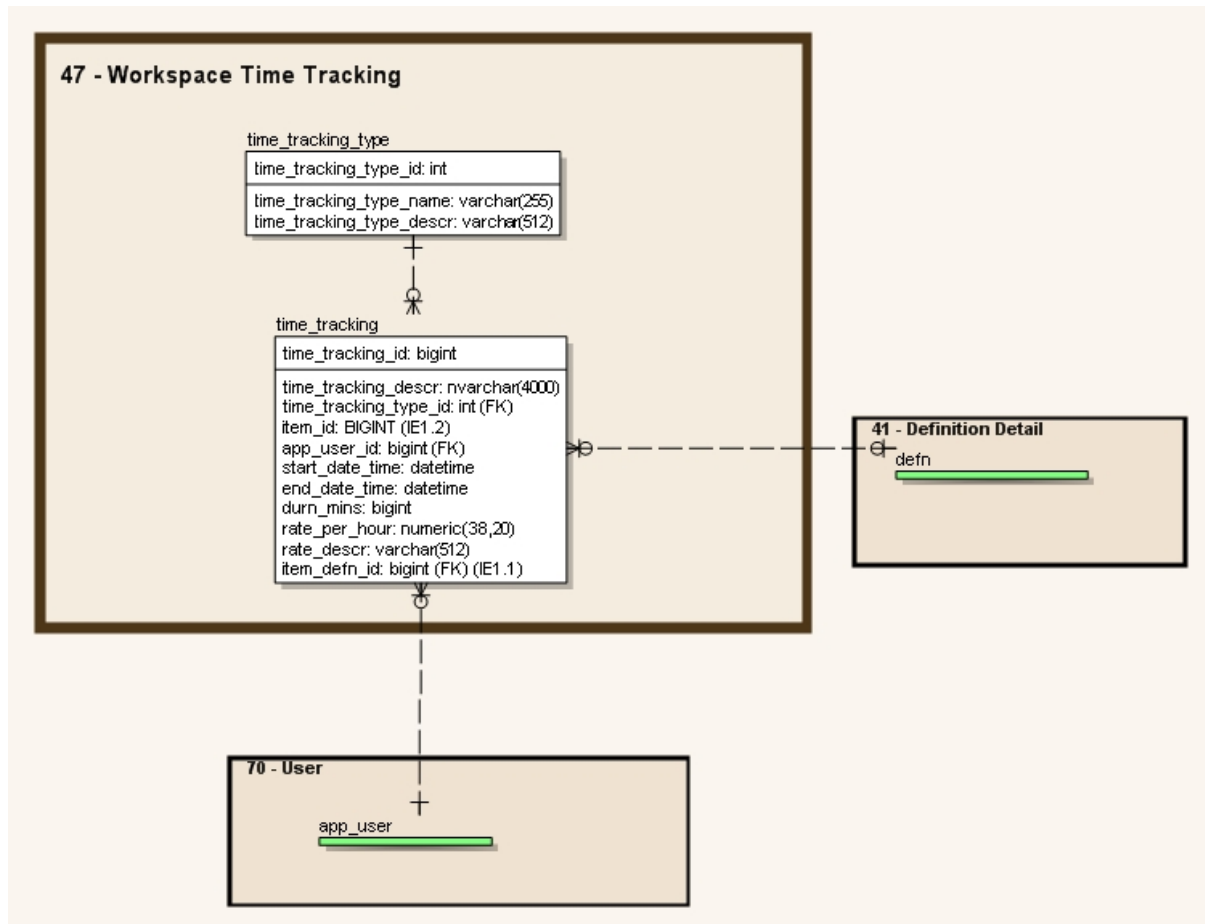


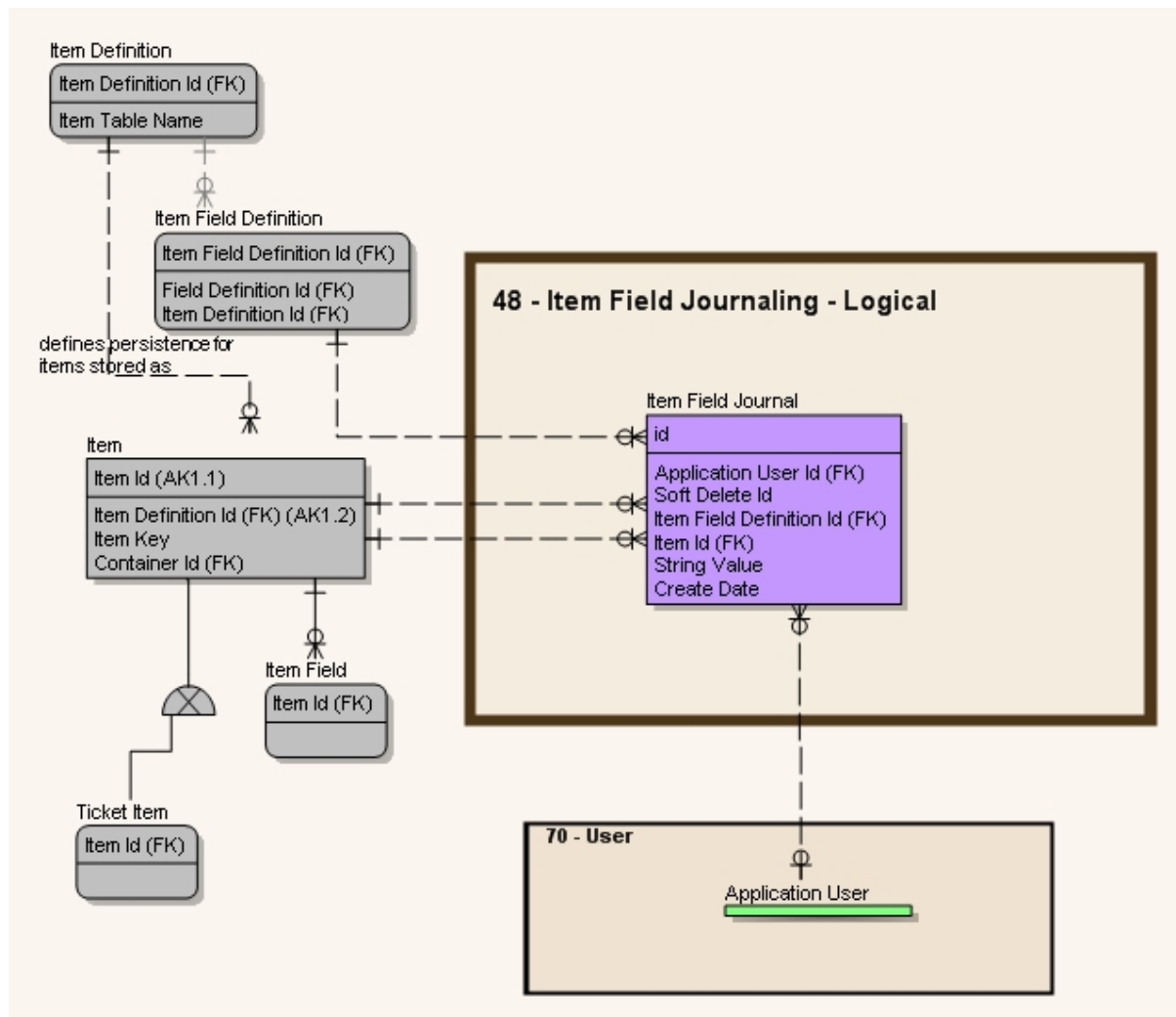
## Logical











## 48 - Item Field Journaling - Physical

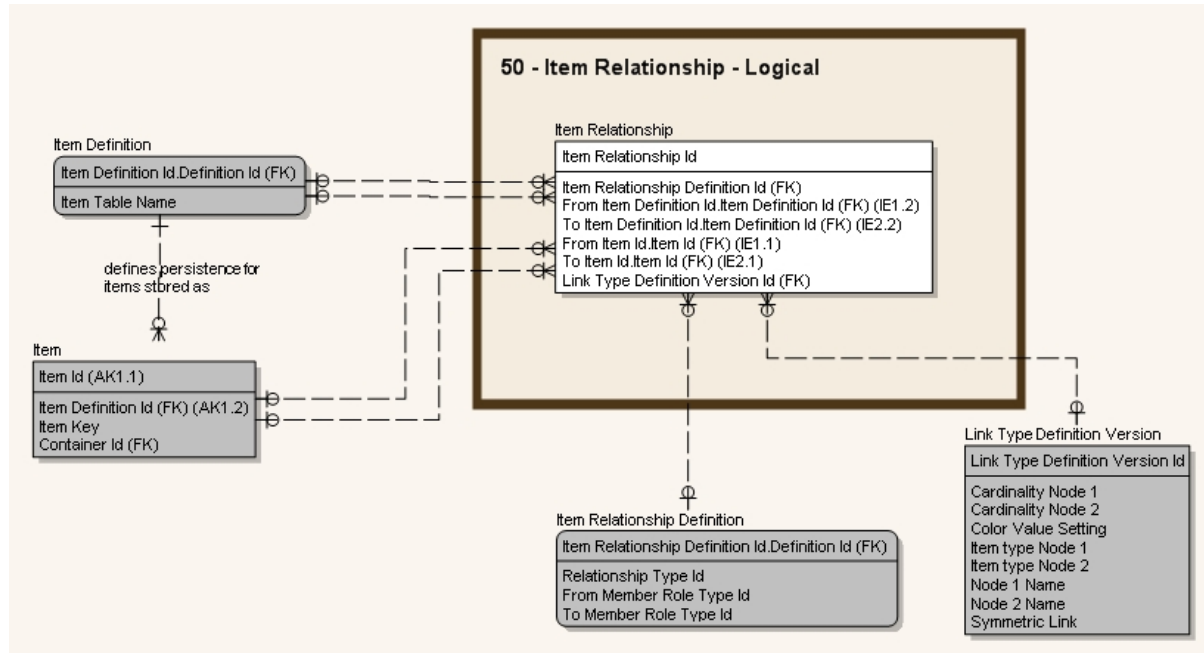
:container\_schema.item\_field\_journal

id: bigint  
app\_user\_id: bigint (FK)  
soft\_delete\_id: int  
item\_field\_defn\_id: bigint (FK)  
item\_id: bigint  
string\_val: varchar(255)  
create\_date: datetime



## 70 - User

:fp\_schema.app\_user





## 50 - Item Relationship - Logical

item\_reln

item\_reln\_id: bigint NOT NULL

item\_reln\_defn\_id: bigint NULL (FK)

from\_item\_defn\_id: bigint NOT NULL (FK) (IE1.1)

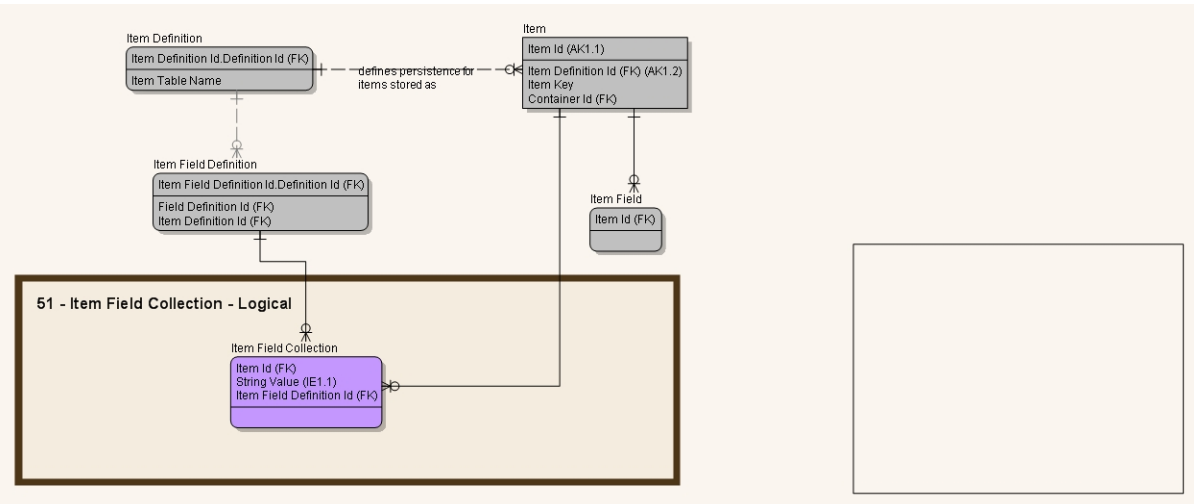
to\_item\_defn\_id: bigint NOT NULL (FK) (IE2.1)

from\_item\_id: bigint NOT NULL (FK) (IE1.2)

to\_item\_id: bigint NOT NULL (FK) (IE2.2)

link\_type\_defn\_ver\_id: bigint NULL (FK)

## Logical

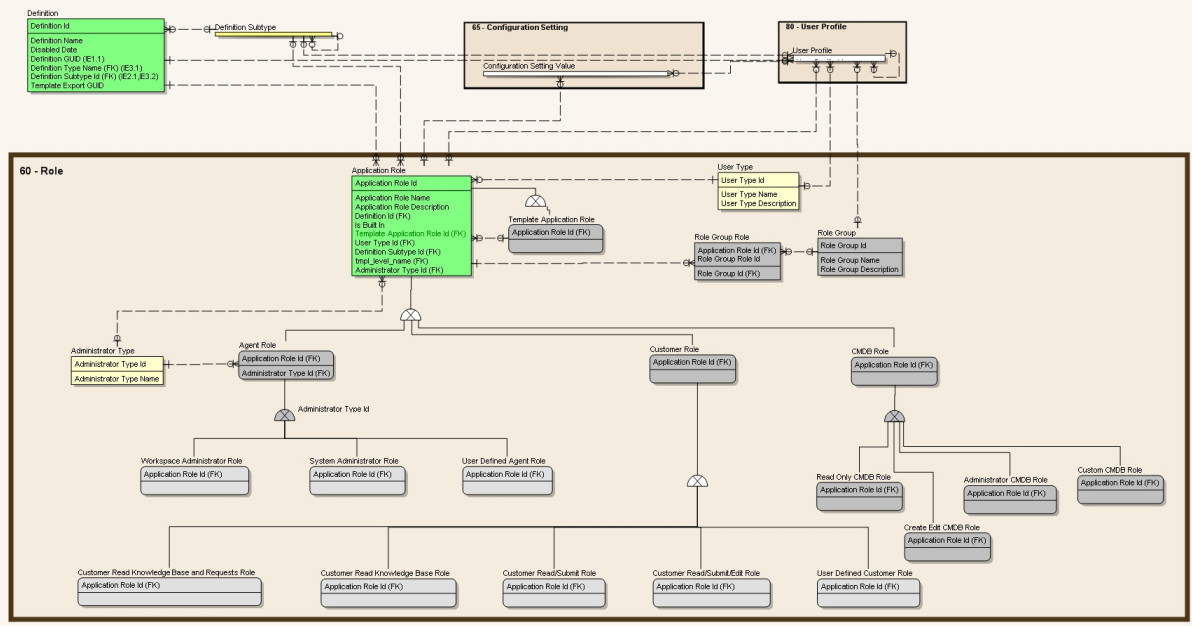


## 51 - Item Field Collection - Physical

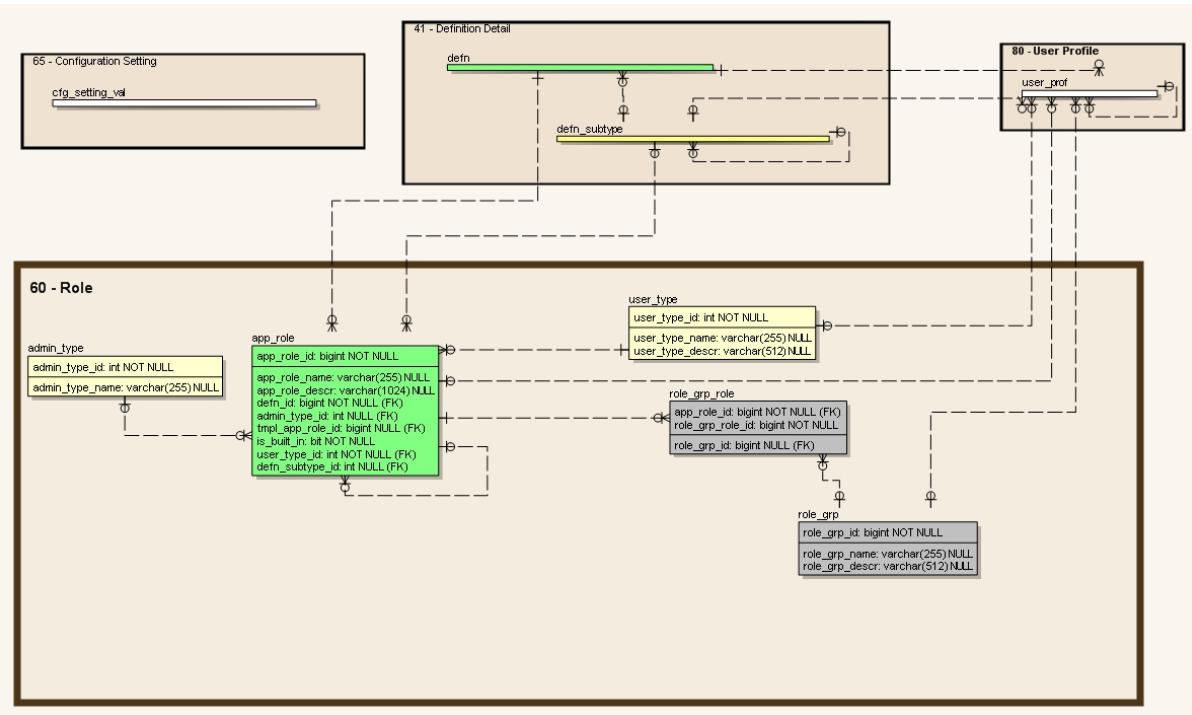
item\_field\_coll

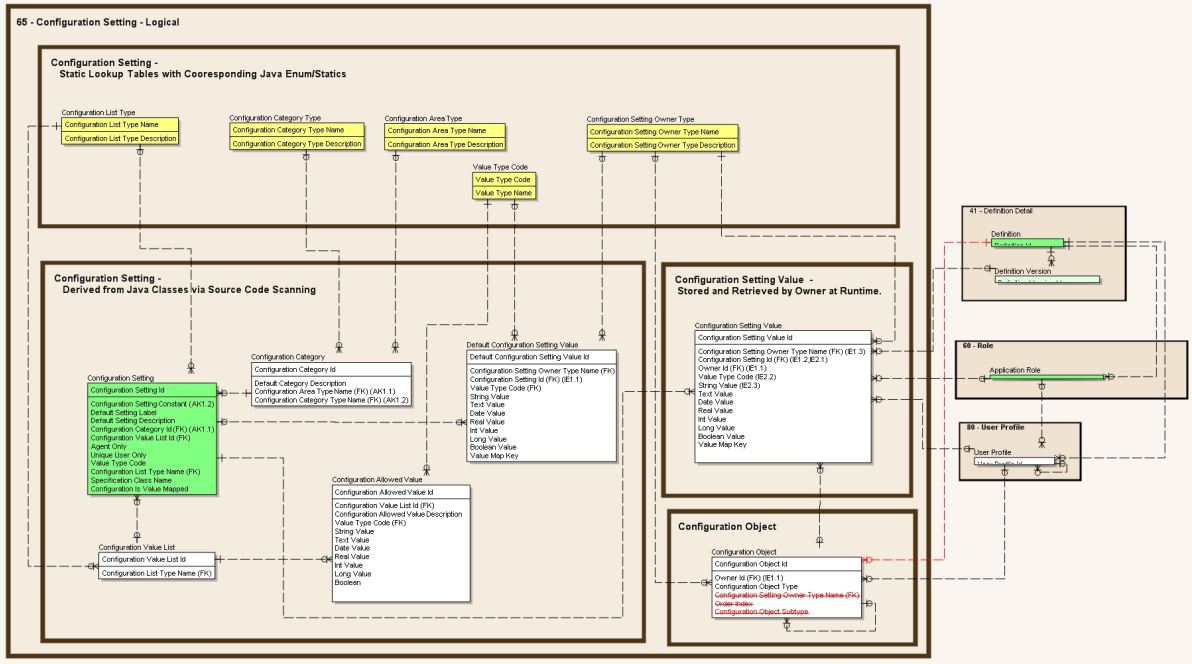
item_id: bigint (FK)
item_field_defn_id: bigint(FK)
string_val: varchar(255) (IE1.1)

## Logical

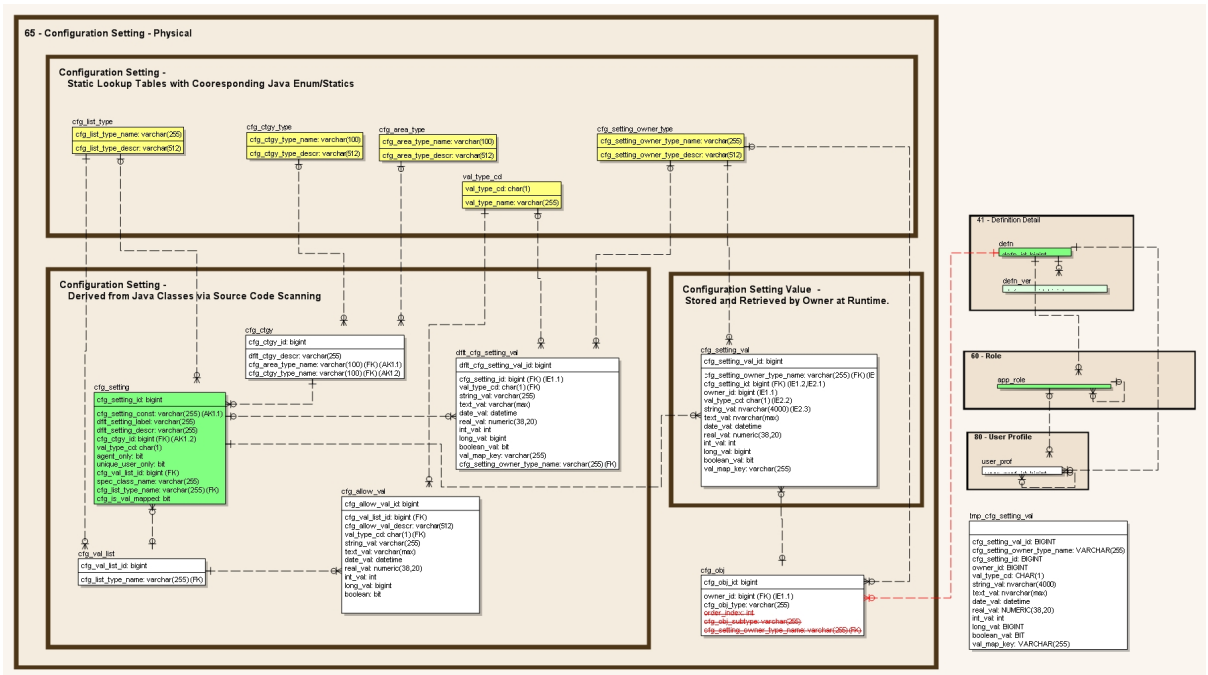


## Physical

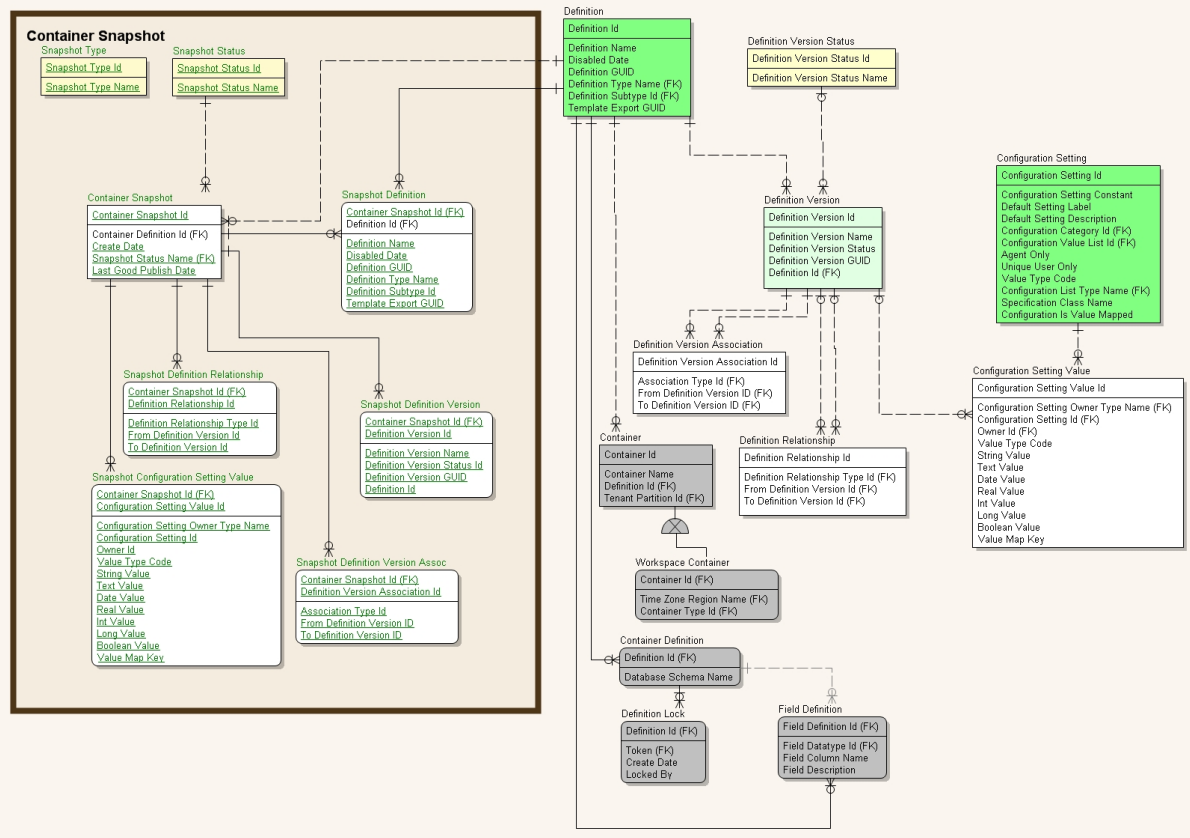




Physical	
----------	--

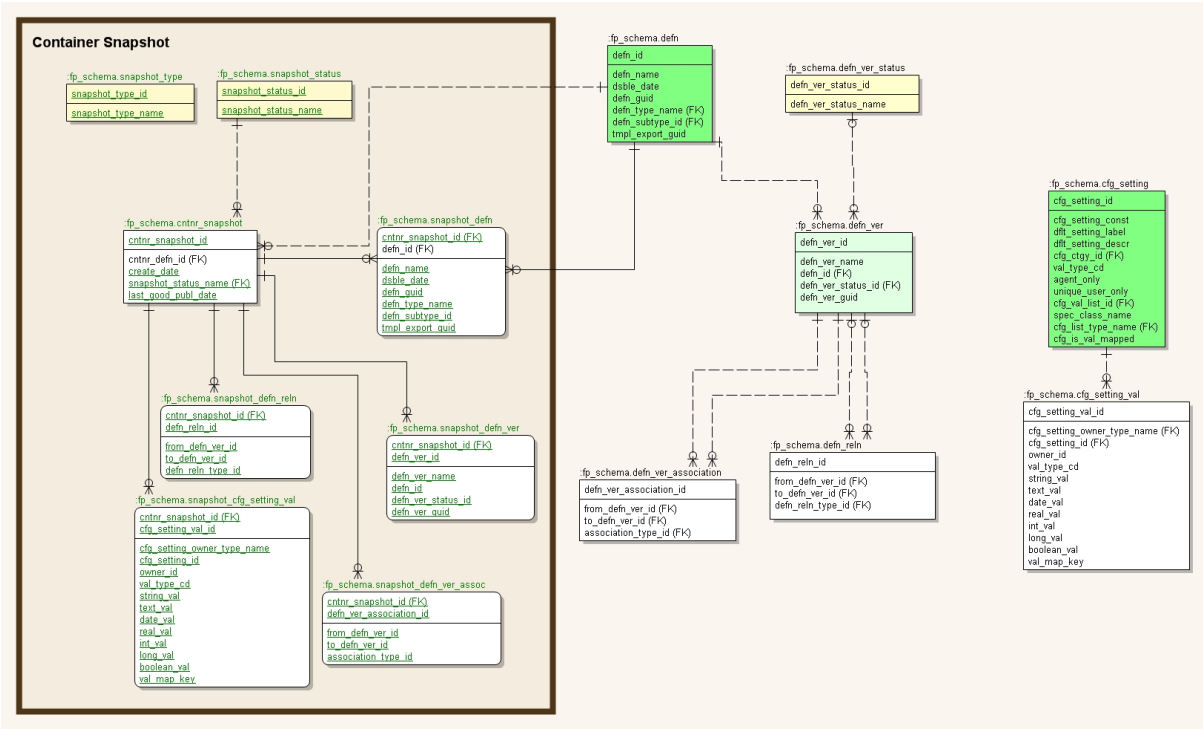


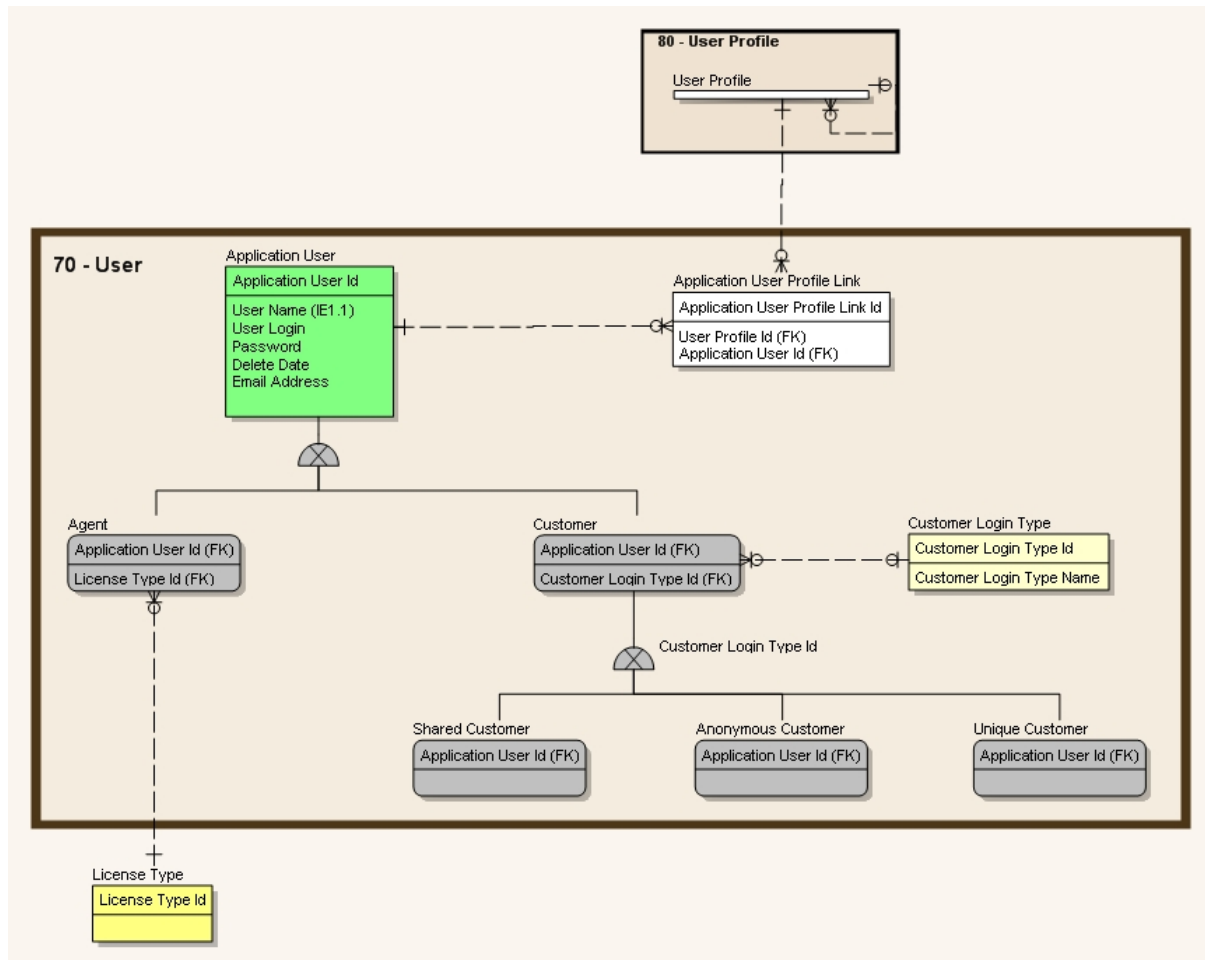
## Logical

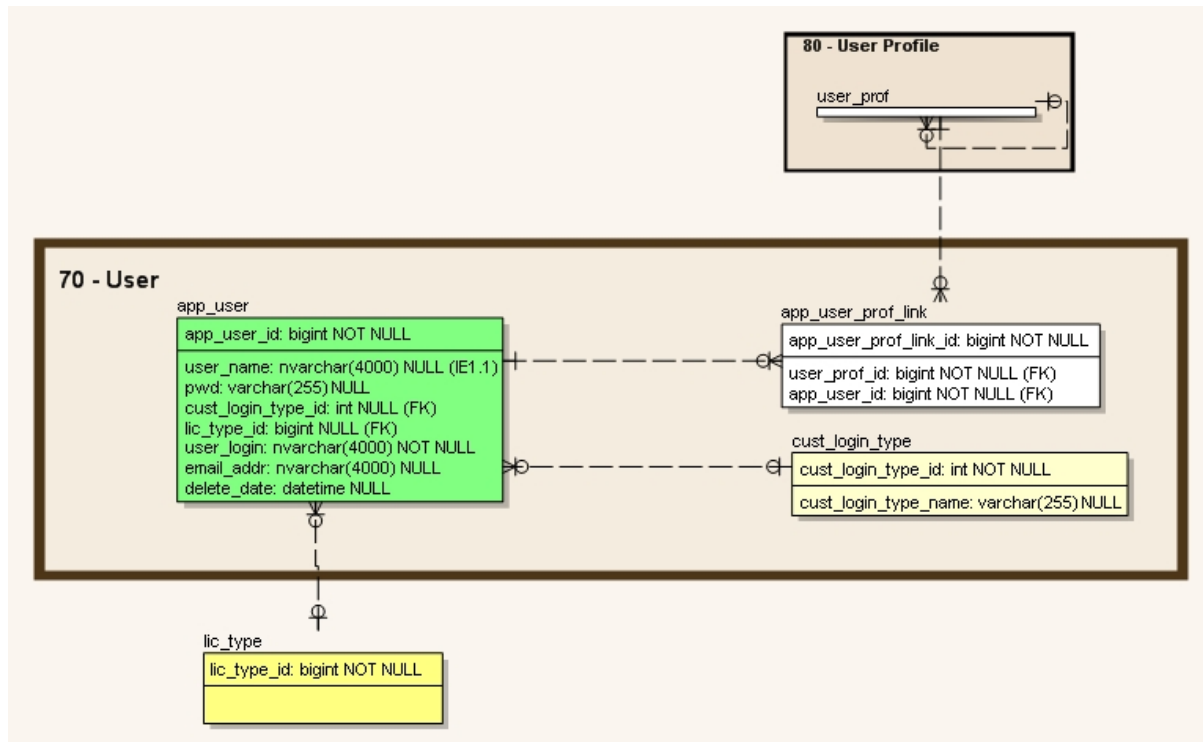




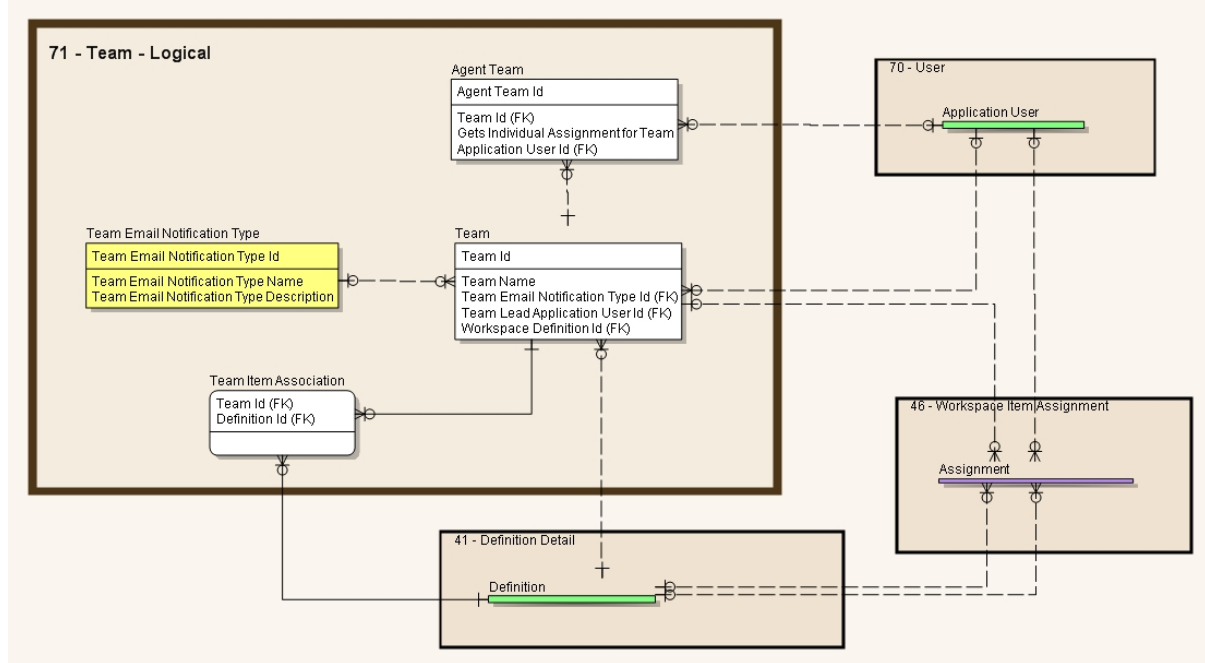
Physical
----------

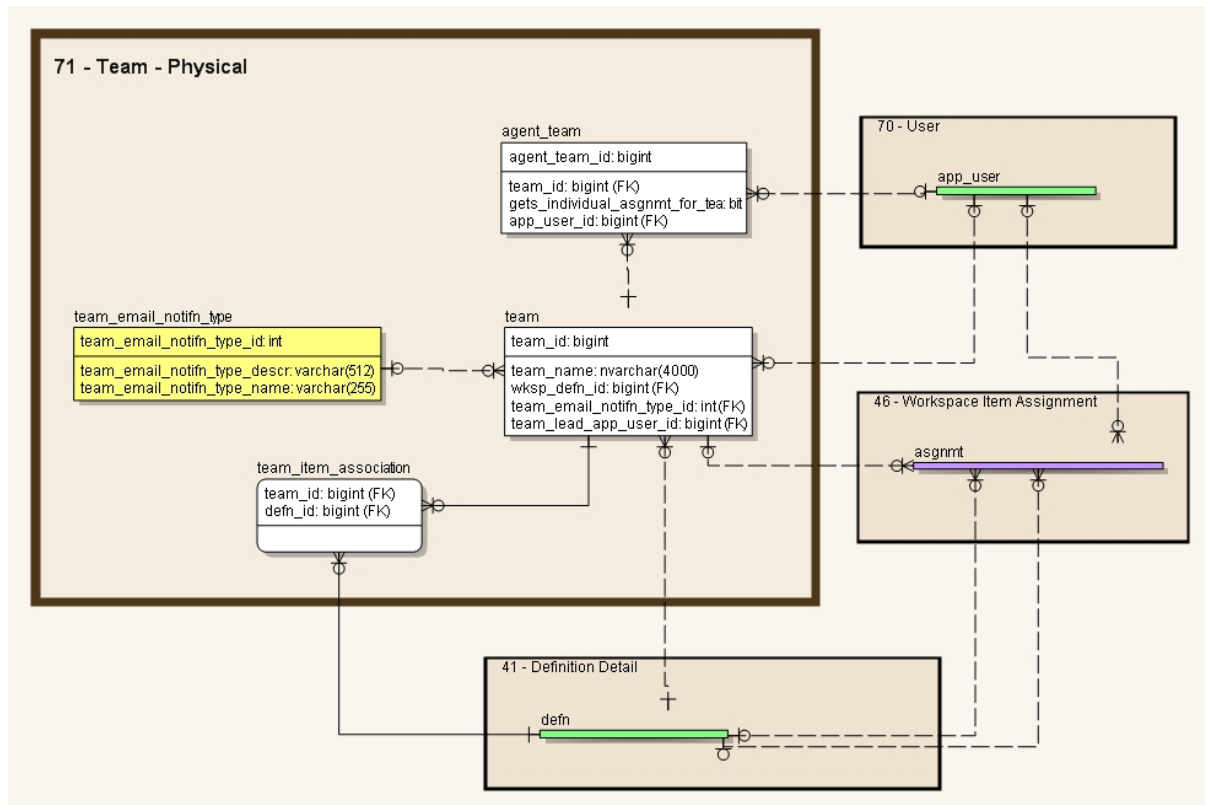




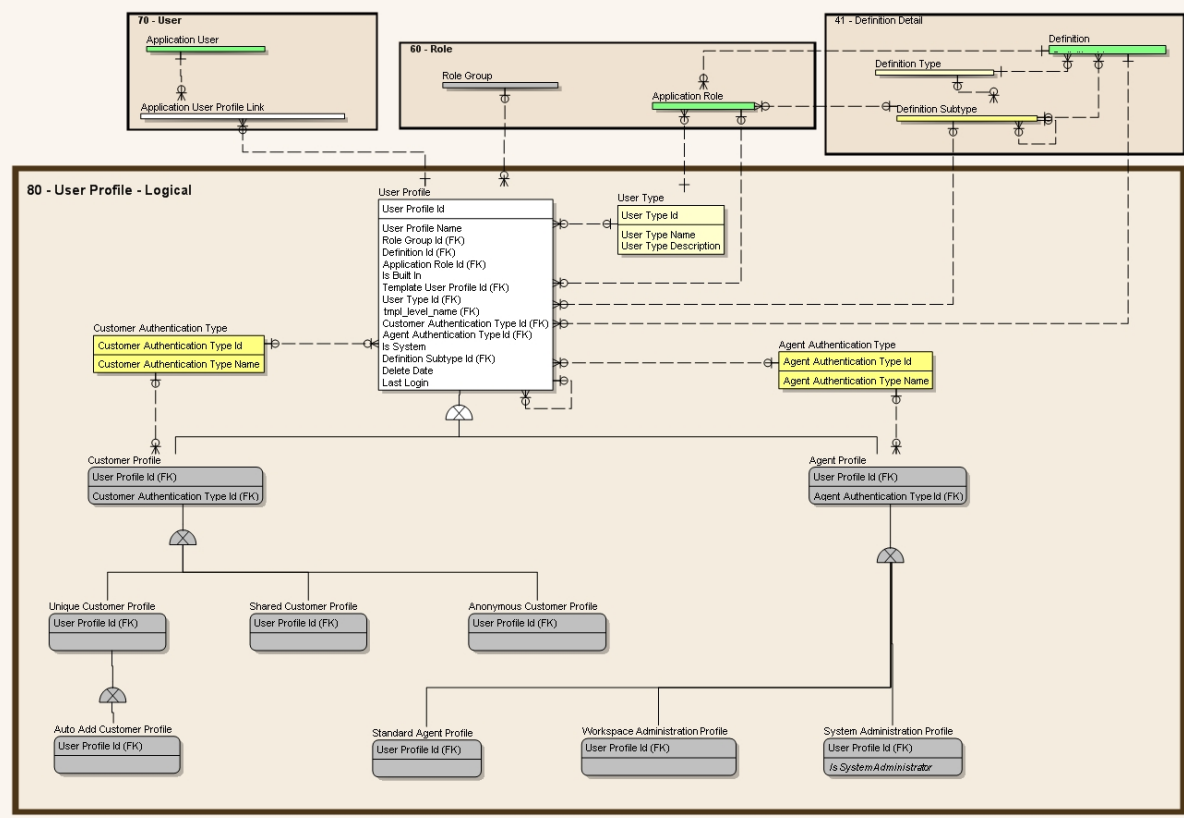


## Logical

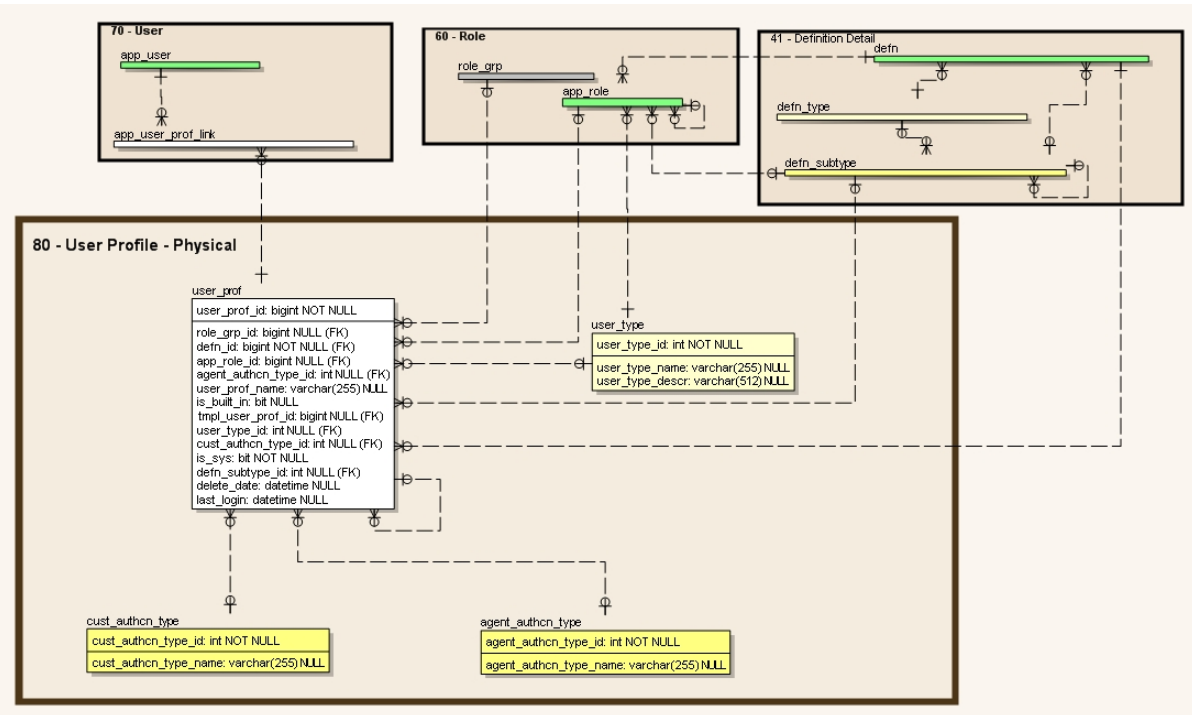




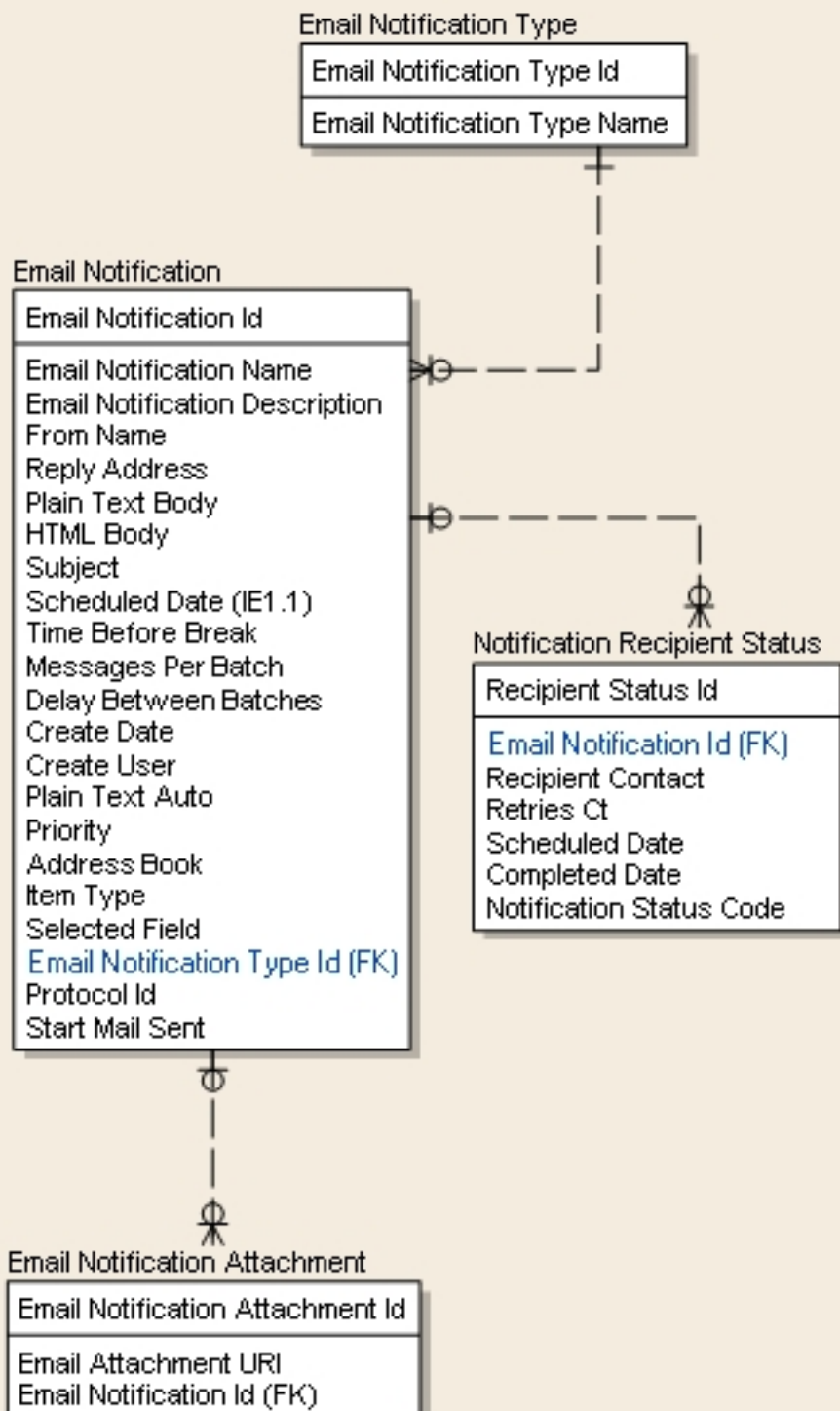
## Logical



## Physical

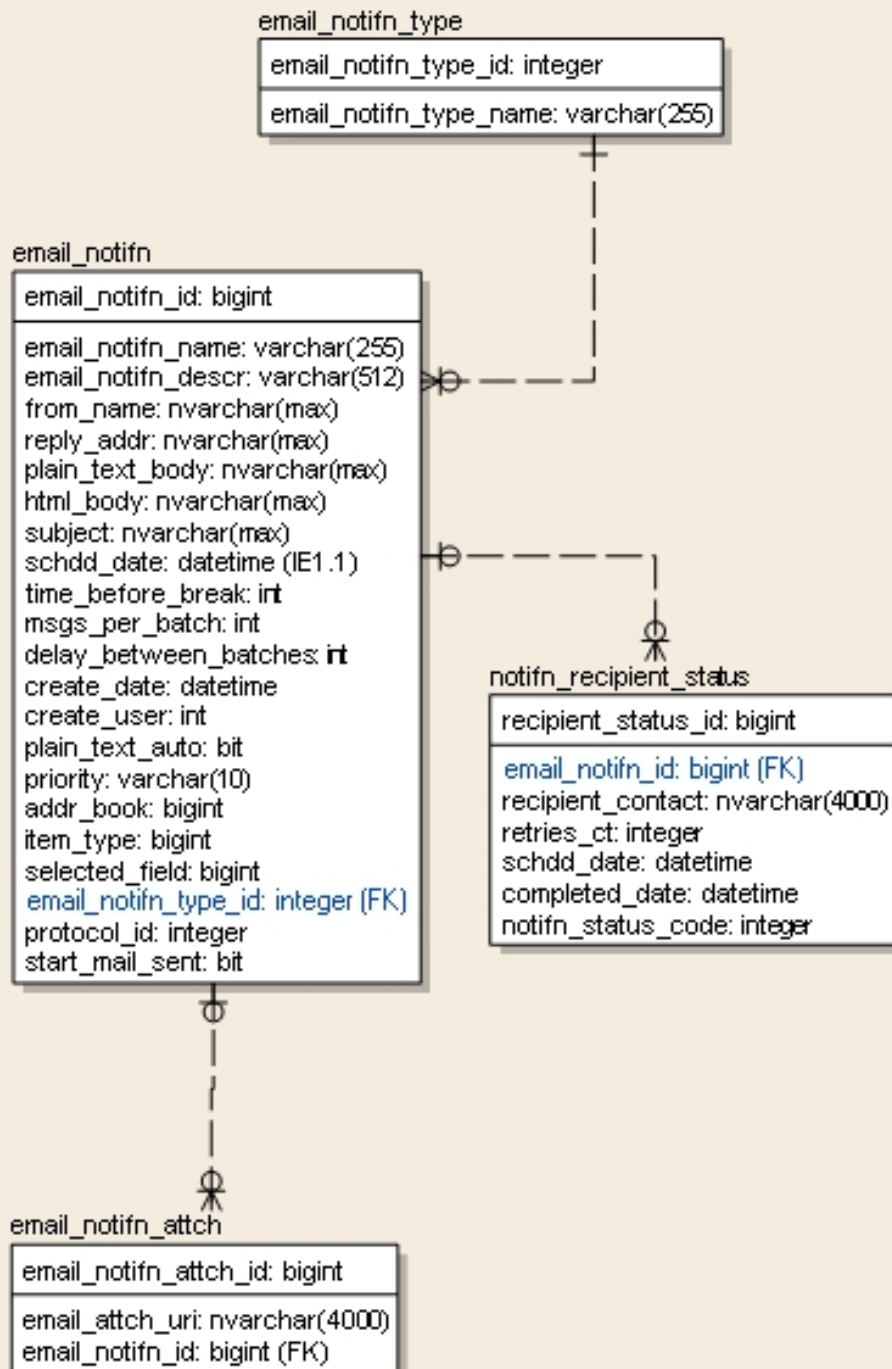


## 96 - Email Notification - Logical

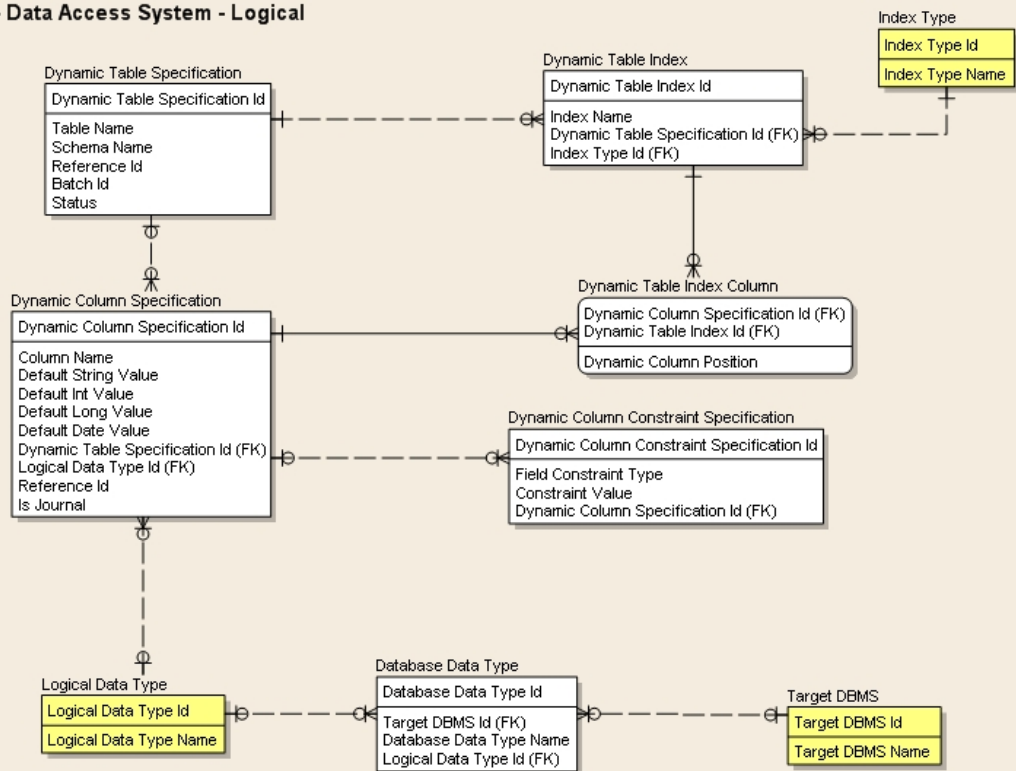




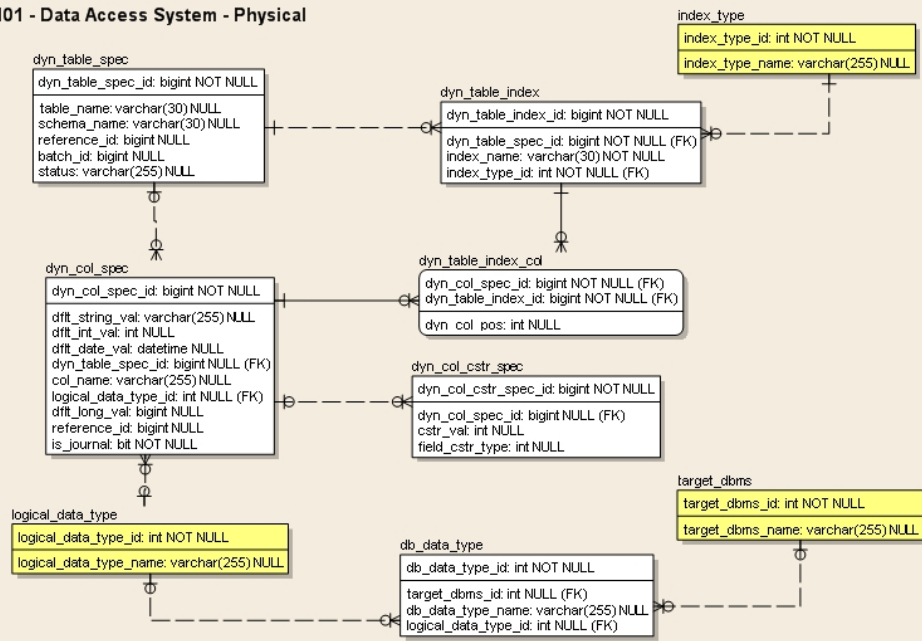
## 96 - Email Notification - Physical

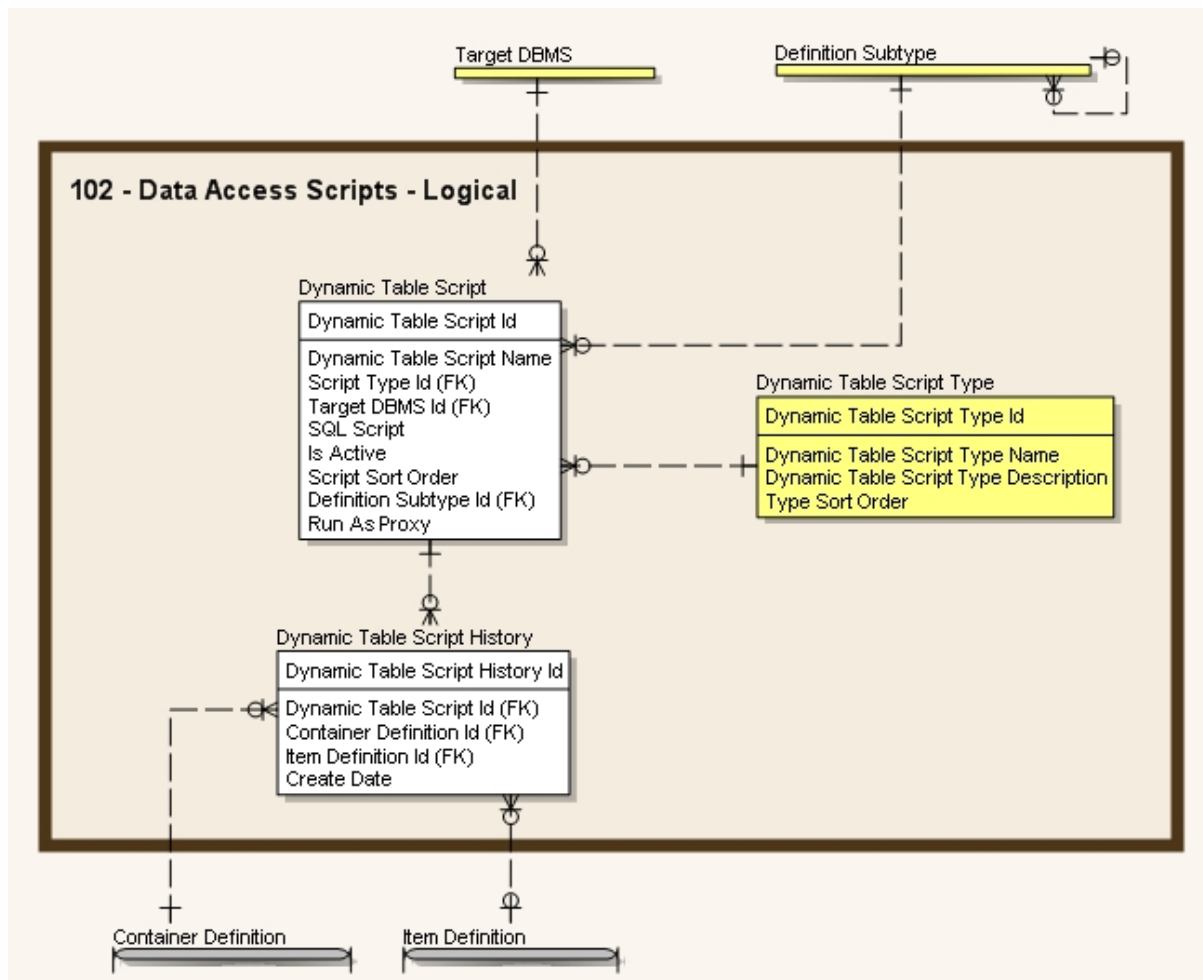


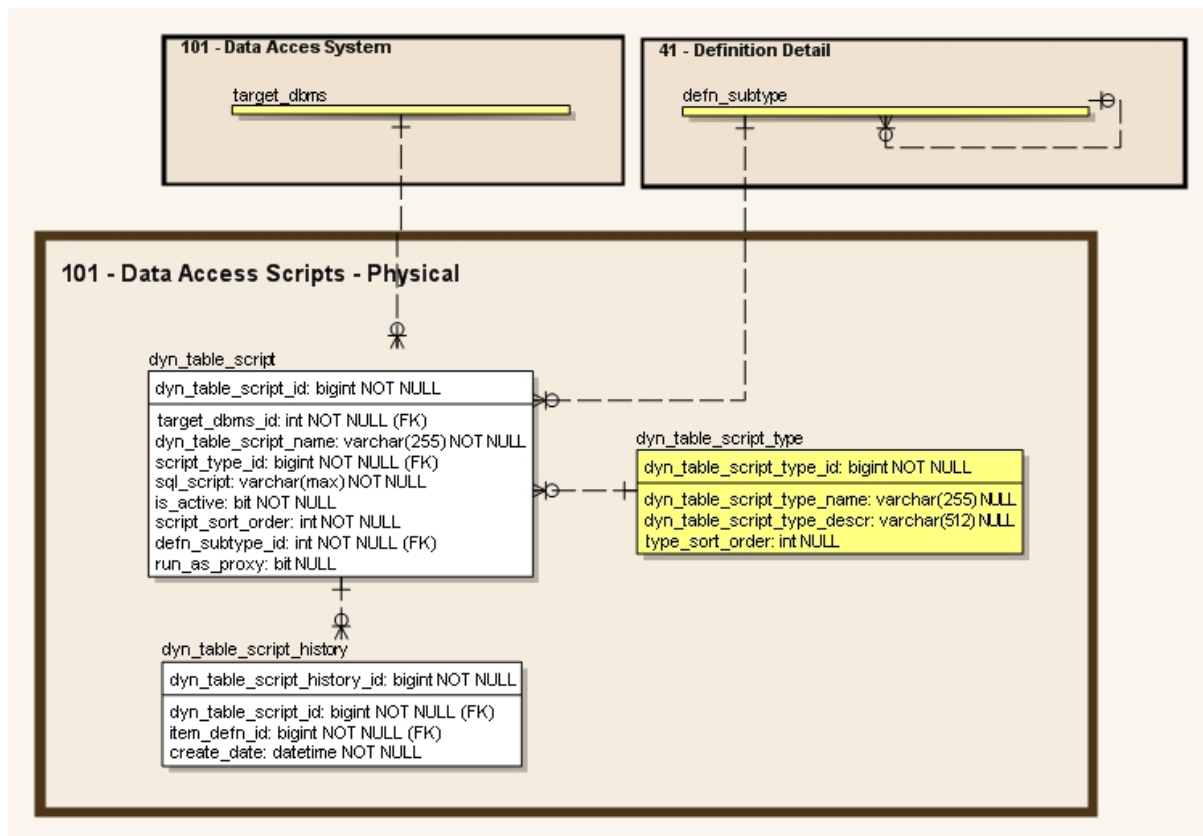
### 101 - Data Access System - Logical



## 101 - Data Access System - Physical







## 110 - Id Generation - Logical

Id Generator

Database Table Name
Last Id Value

## 110 - Id Generation - Physical

:fp\_schema.id\_generator

db\_table\_name: varchar(255) NOT NULL

last\_id\_val: bigint NULL

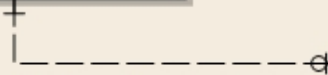
## 120 - Job Scheduler - Physical

Job Scheduler Configuration

Job Scheduler Configuration Id
Job Scheduler Instance Id (AK1.1)
Service Host Name
Service Port
Job Scheduler Default Interrupt
Job Scheduler default Maximum Wait

Job Configuration Detail

Job Configuration Detail Id
Job Scheduler Configuration Id (FK) (AK1.3)
Job Name (AK1.1)
Job Class
Group Name (AK1.2)
Trigger Type Name
Trigger Start Date
Trigger End Date
Trigger Hour
Trigger Minute
Trigger Frequency
Is Job Enabled
History Retain Days
History Retain Count
Job Parameters
Create Date
Create User
Cron
Repeat Count
Time Zone
Version ID (AK1.4)





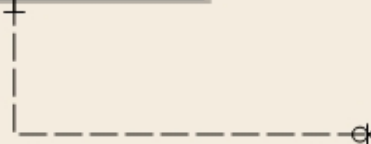
## 120 - Job Scheduler - Physical

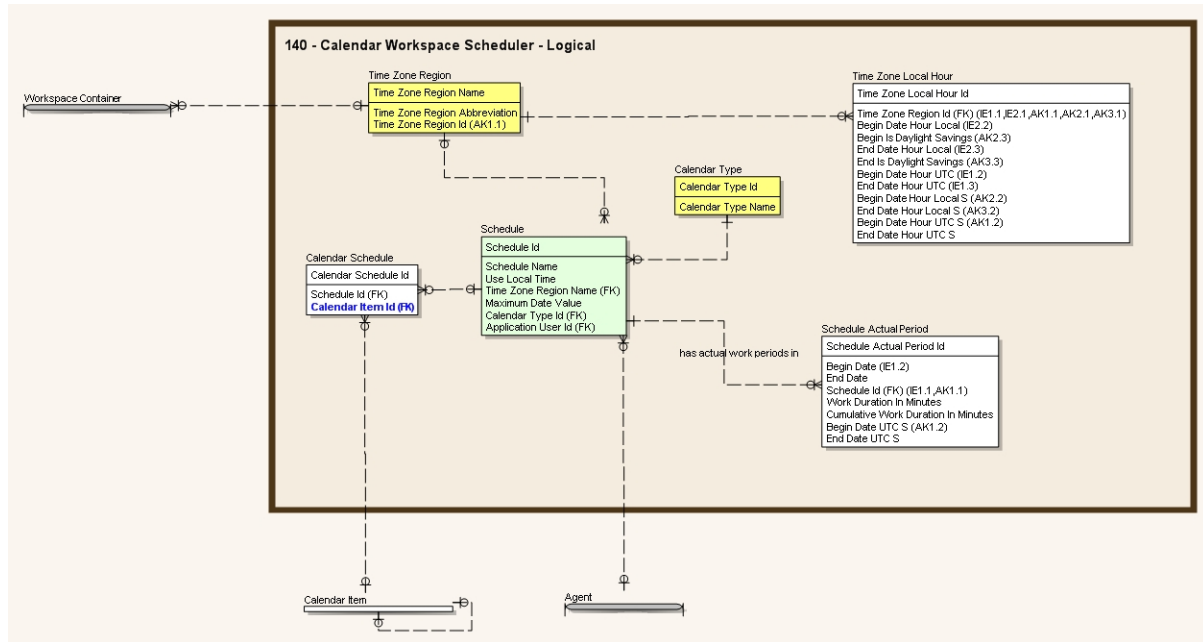
job\_schedlr\_cfg

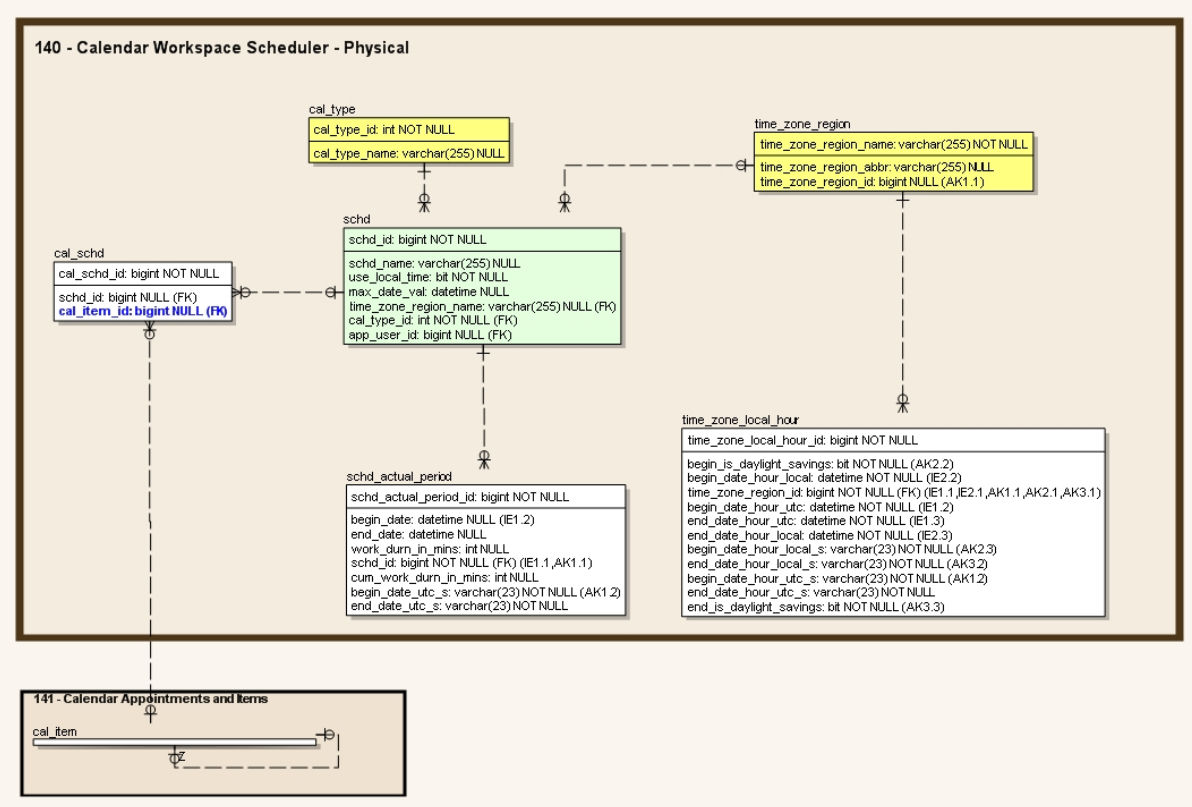
job_schedlr_cfg_id: bigint
svc_host_name: varchar(255)
svc_port: int
job_schedlr_instc_id: varchar(255) (AK1.1)
job_schedlr_dflt_interrupt: bit
job_schedlr_dflt_max_wait: bigint

job\_cfg\_dtl

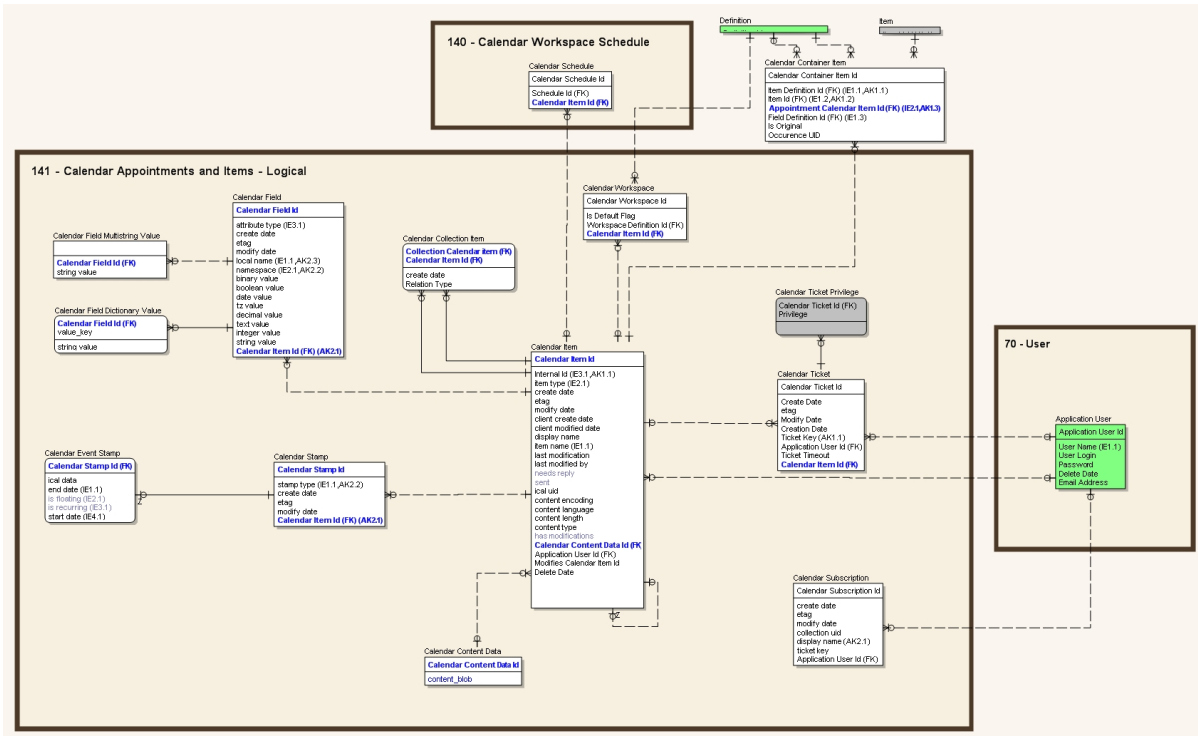
job_cfg_dtl_id: bigint
job_schedlr_cfg_id: bigint (FK) (AK1.3)
job_name: varchar(100) (AK1.1)
job_class: VARCHAR(150)
grp_name: varchar(100) (AK1.2)
trigger_type_name: varchar(255)
trigger_start_date: datetime
trigger_end_date: datetime
trigger_hour: int
trigger_minute: int
trigger_freq: int
is_job_enabled: bit
history_retain_days: int
history_retain_ct: int
job_parms: varchar(max)
create_date: datetime
create_user: varchar(255)
cron: varchar(255)
repeat_ct: int
time_zone: varchar(255)
ver_id: bigint (AK1.4)

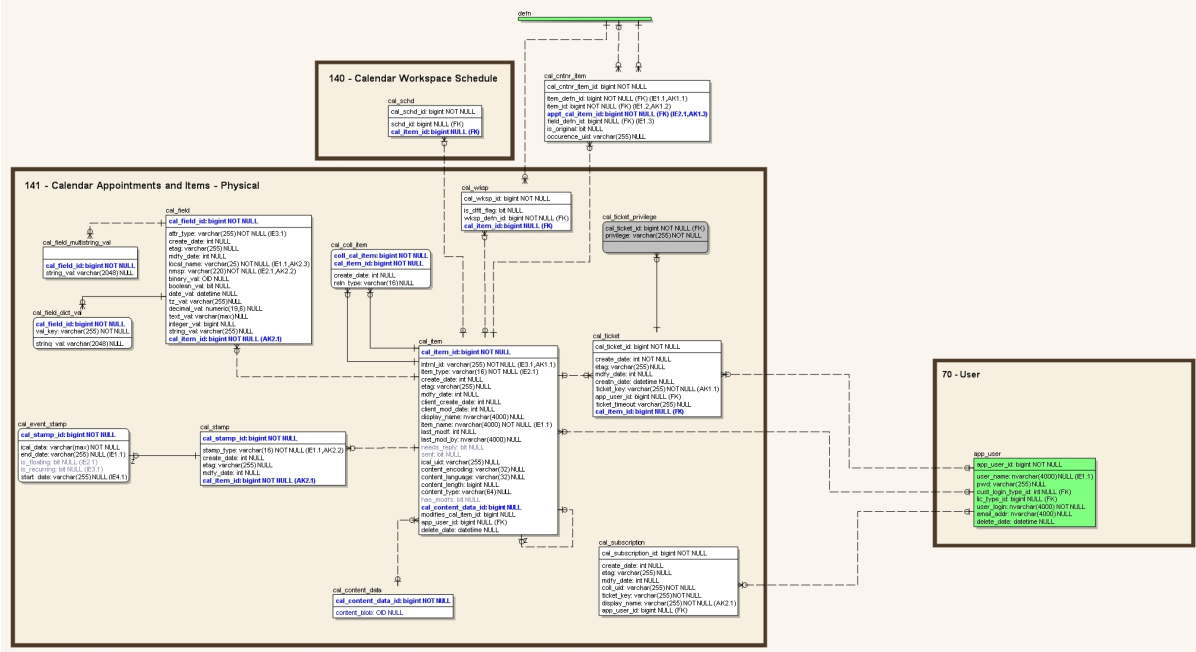


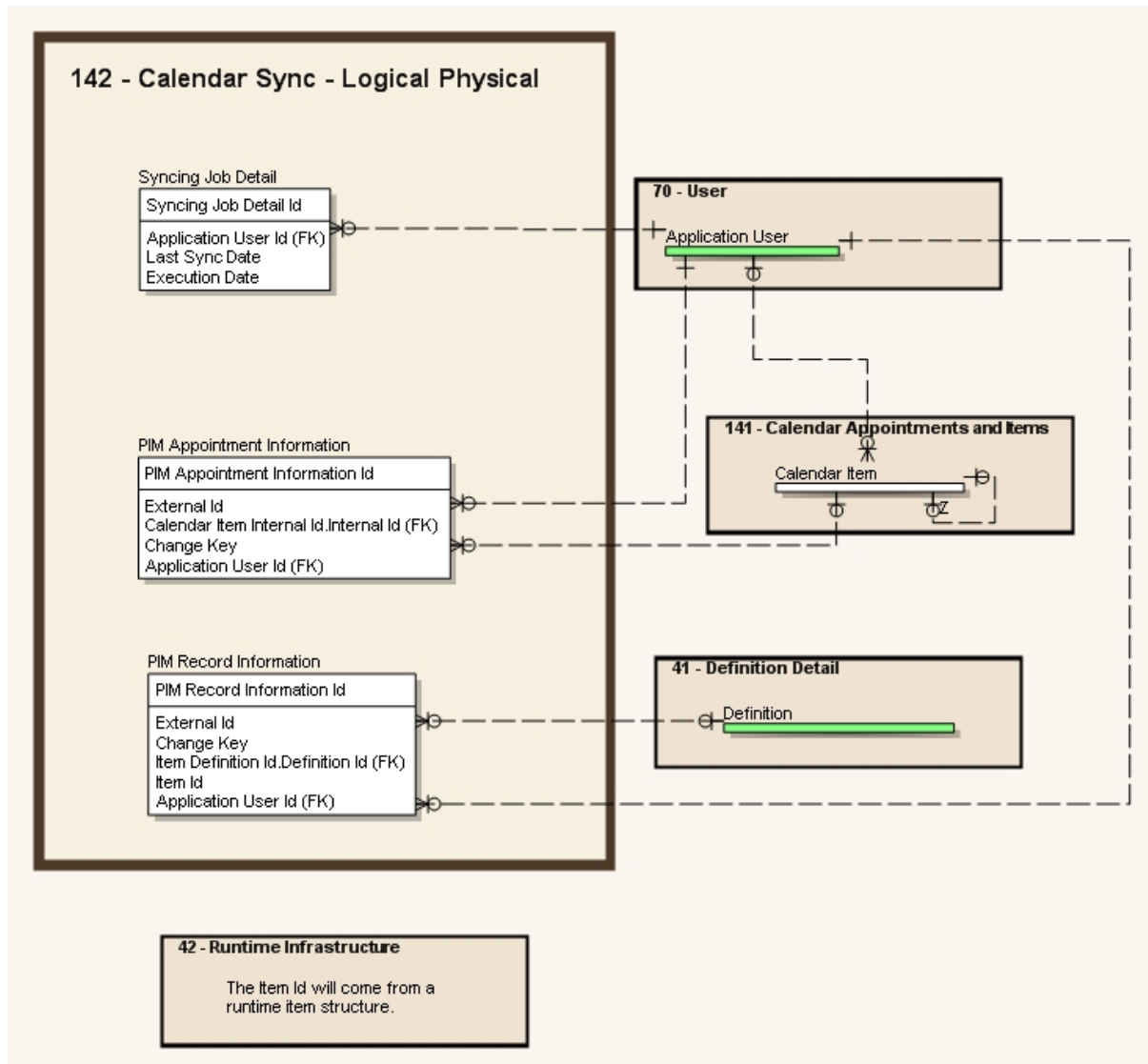


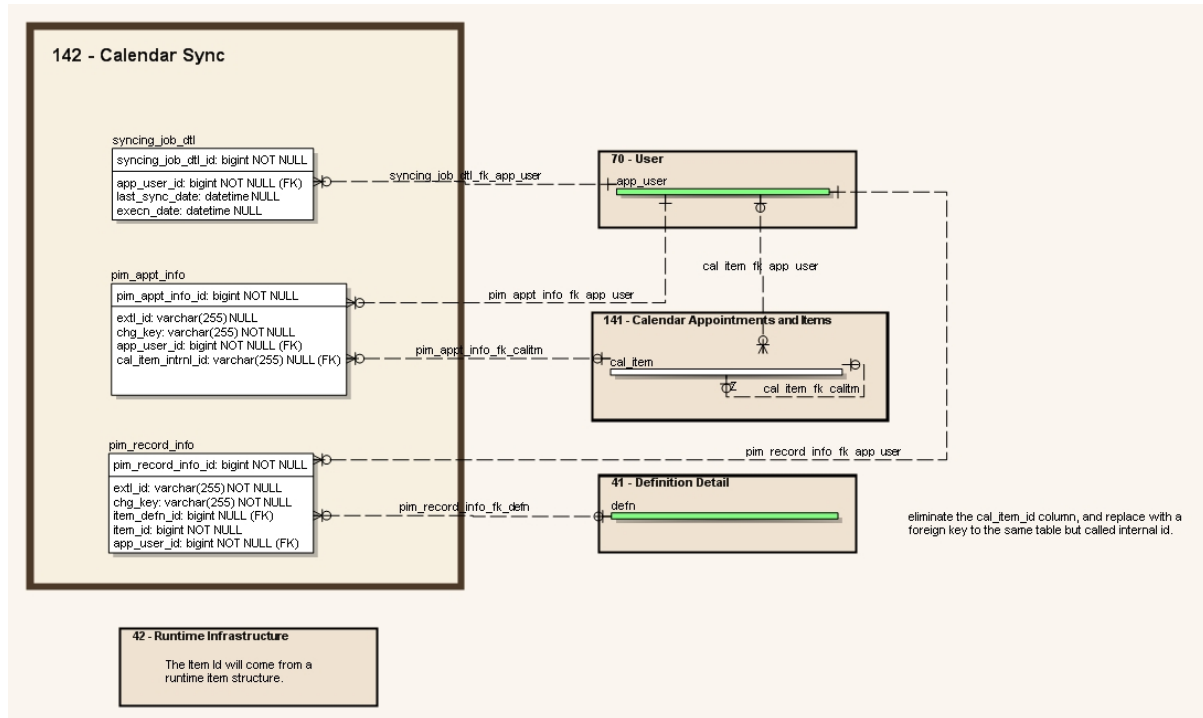


## Logical

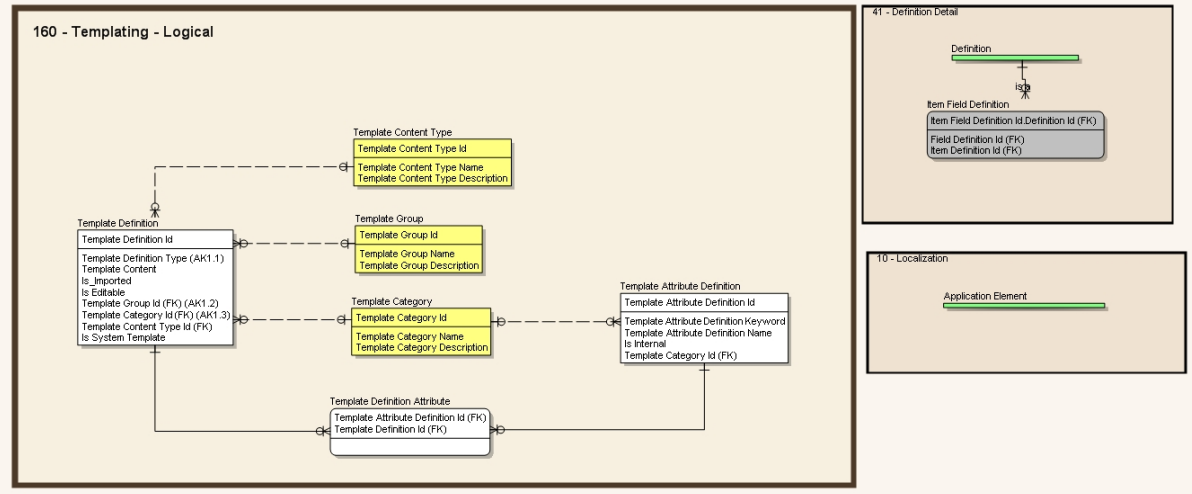




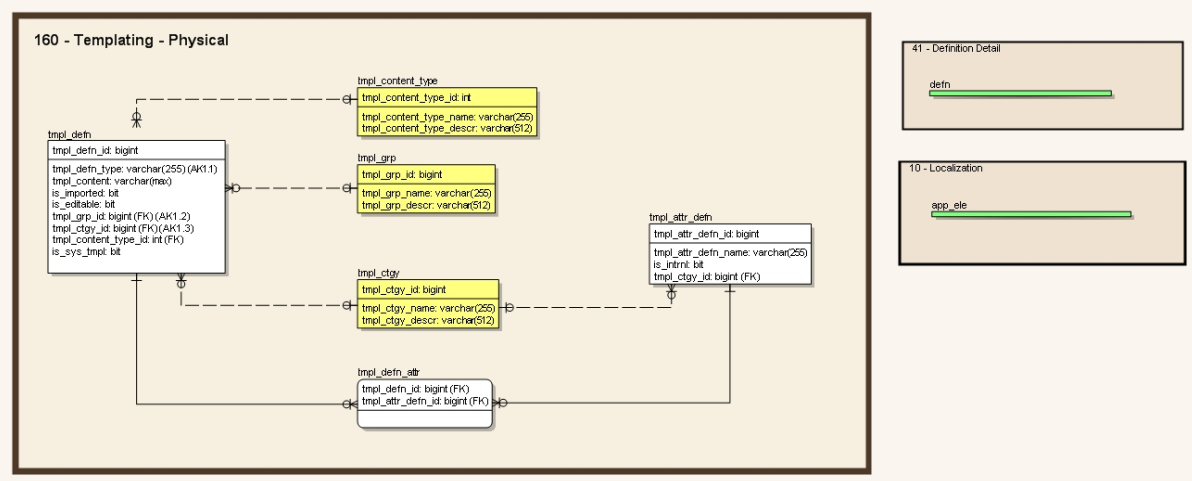




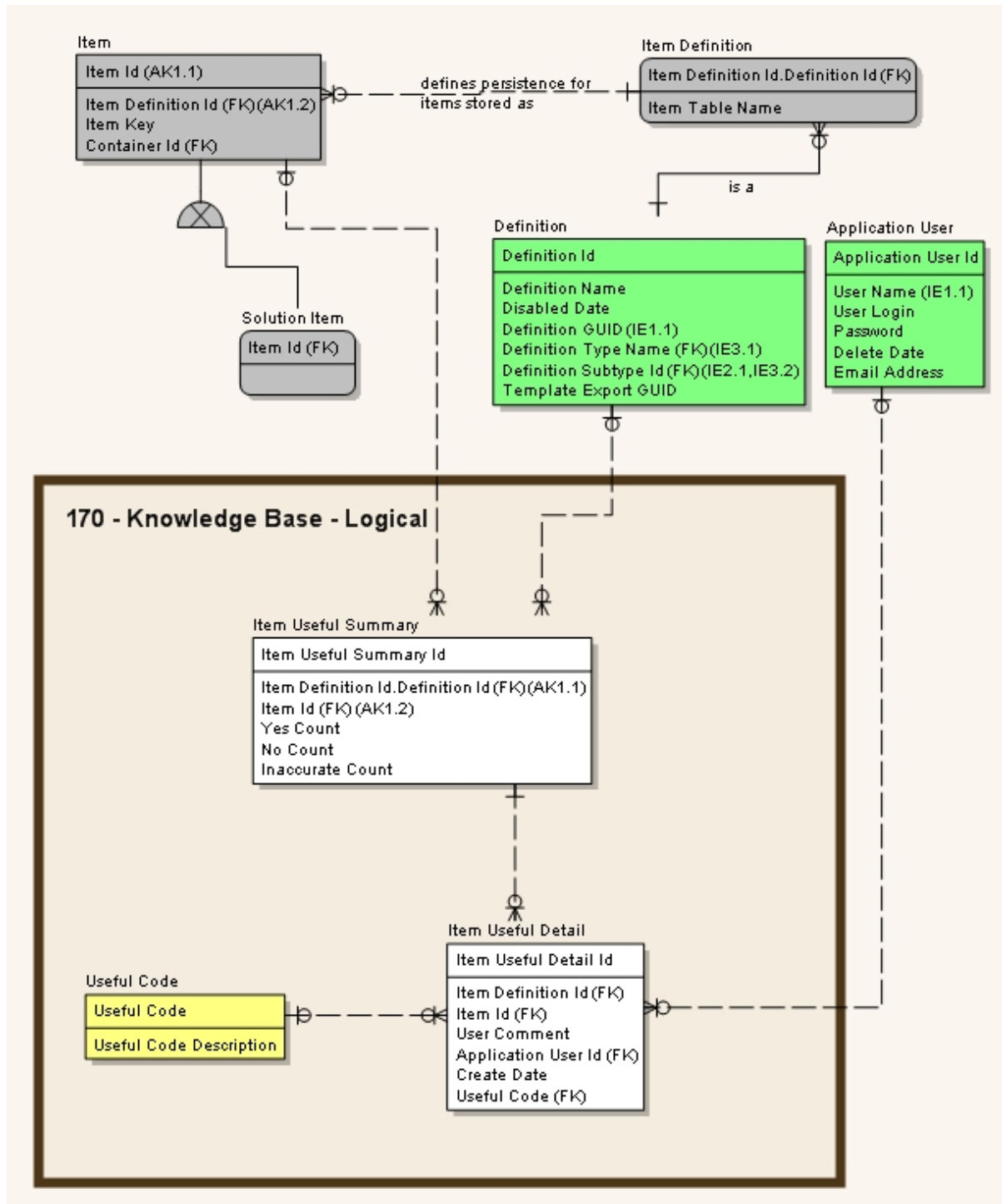
## Logical

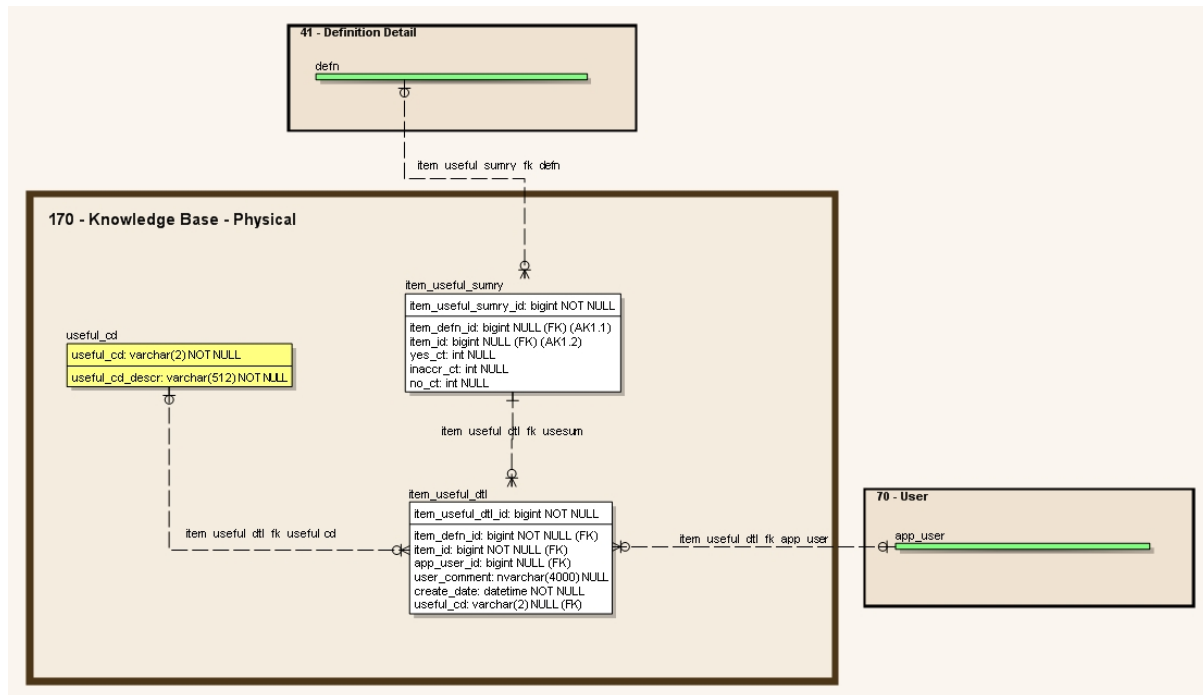


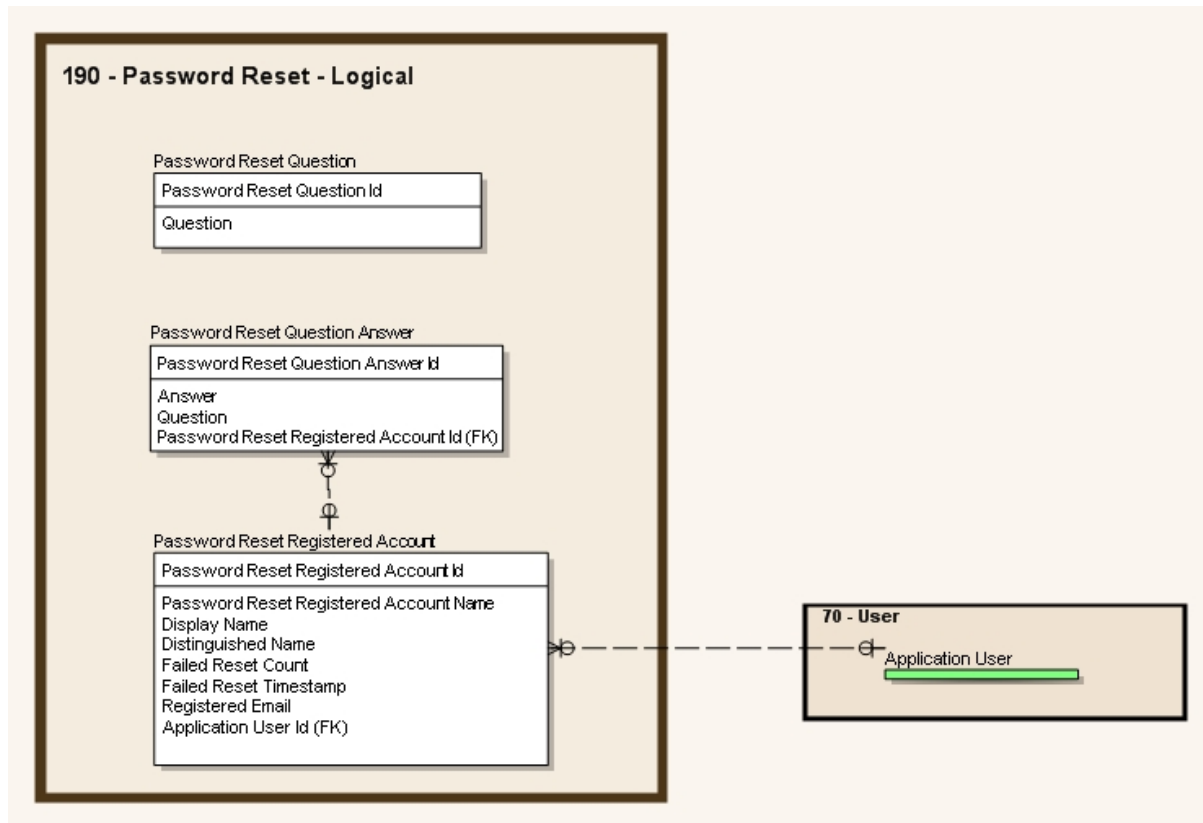


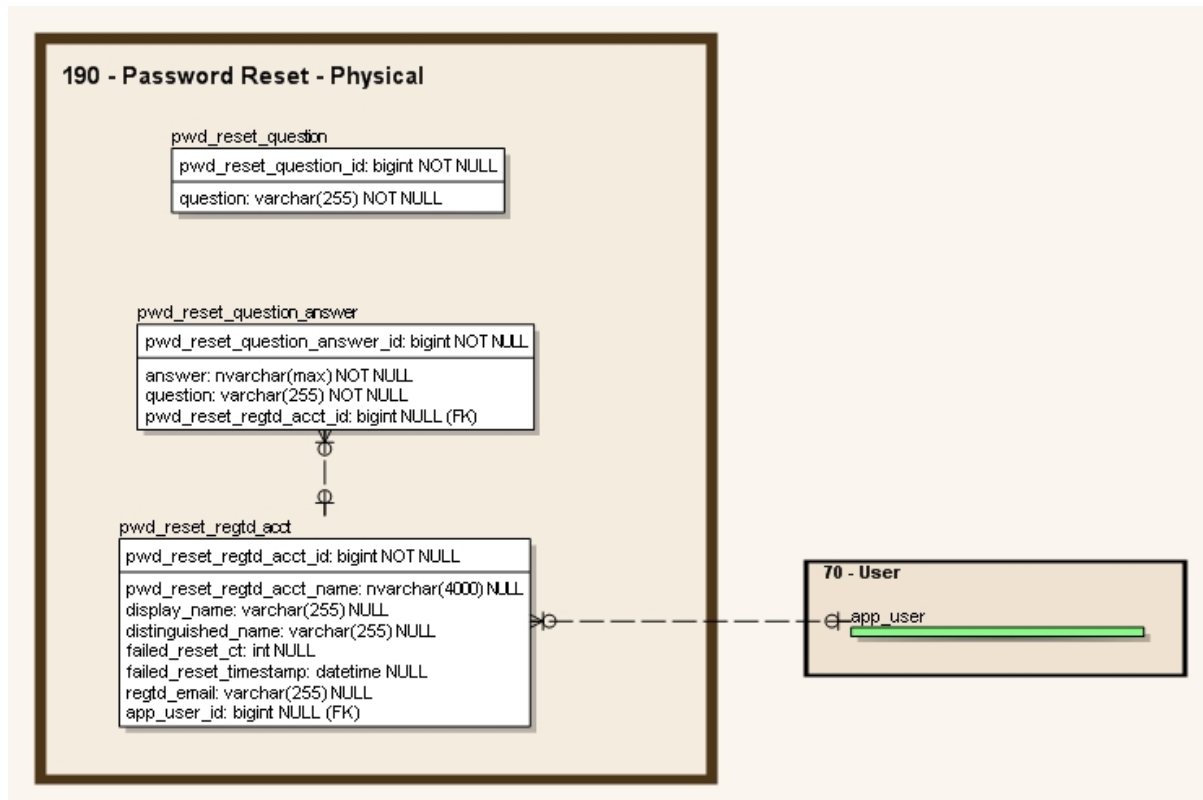


## Logical

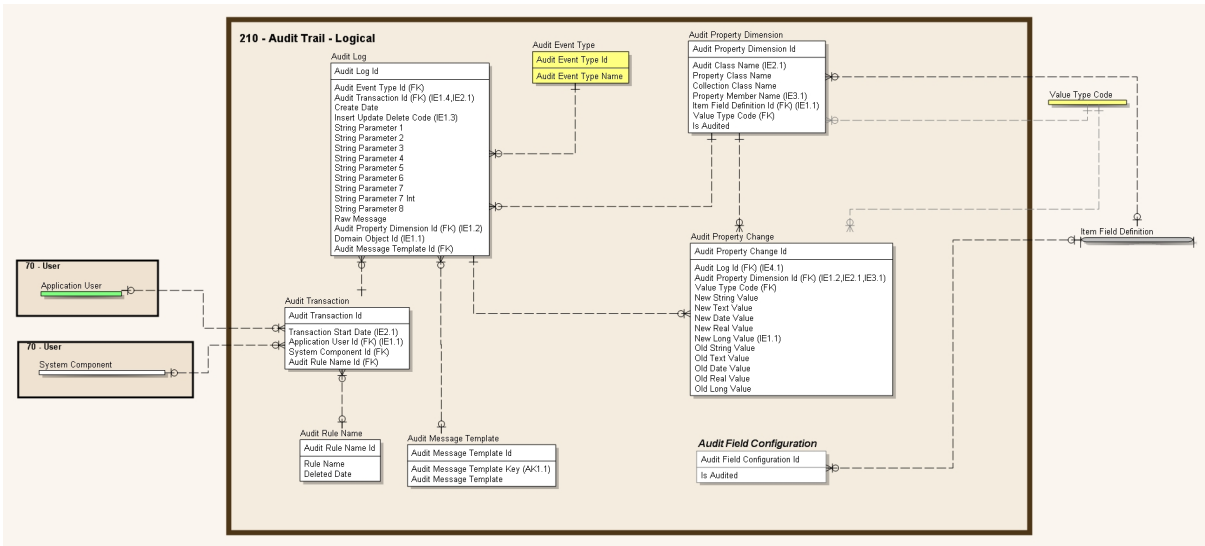




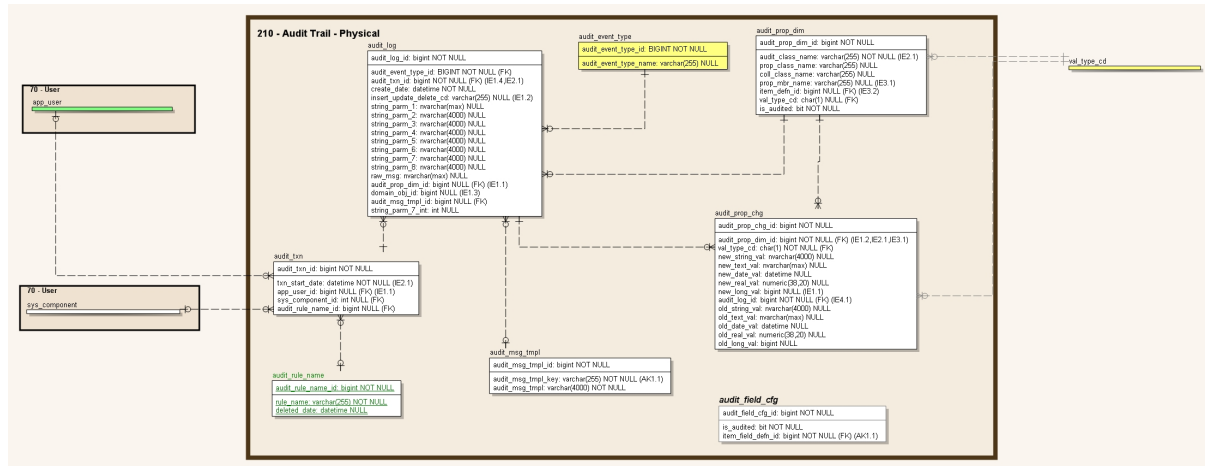




Logical
---------



## Physical



## 230 - Rules - Logical

### Rule Usage Statistic

Rule Usage Statistic Id
Definition UUID (AK1.1)
Last Hit Timestamp
Hit Count
Impact Count

Rules are stored in Definitions

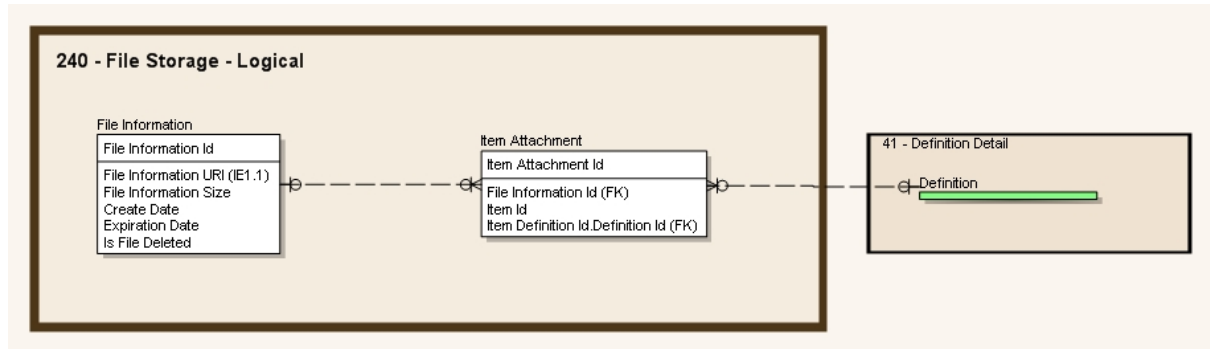


## 230 - Rules - Physical

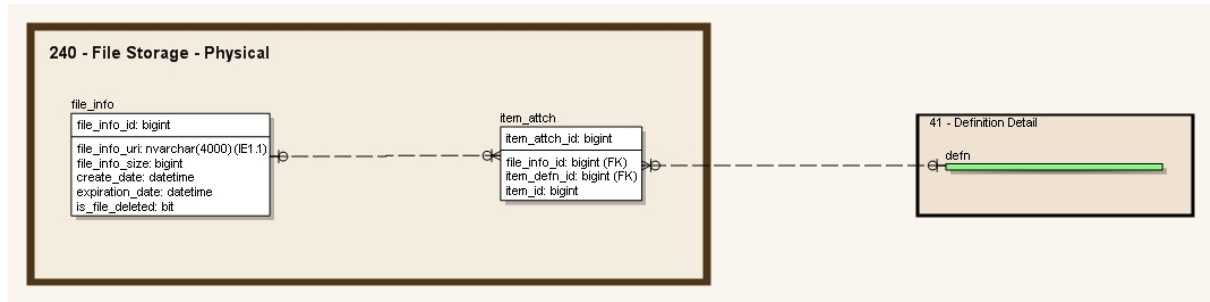
rule\_usage\_statistic

rule_usage_stat_id: bigint
defn_uuid: varchar(255) (AK1.1)
last_hit_timestamp: datetime
hit_ct: int
impact_ct: int

Rules are stored in Definitions



## Physical



## 250 - Apache Jack Rabbit JCR - Logical

FootPrints uses Apache Jackrabbit to name and store attachments.

Apache Jackrabbit is a fully featured Java Content Repository (JCR) that implements all of the JCR API.

A content repository is an information management system that provides various services for storing, accessing, and managing content. In addition to a hierarchically structured storage system, common services of a content repository are versioning, access control, full text searching, and event monitoring.

Jackrabbit creates and defines these tables for storage. They should not be accessed directly, but the JCR API should be used instead.

Default Bundle

Node Id Hi
Node Id Lo
bundle Data

Default Refs

Node Id Hi
Node Id Lo
Refs Data

Default Binval

Binval Id (AK1.1)
Binval Data

Default Names

id
name

Security Bundle

Node Id Hi
Node Id Lo
bundle Data

Security Refs

Node Id Hi
Node Id Lo
Refs Data

Security Binval

Binval Id (AK1.1)
Binval Data

Security Names

id
name

## 250 - Apache Jack Rabbit JCR - Physical

FootPrints uses Apache Jackrabbit to name and store attachments.

Apache Jackrabbit is a fully featured Java Content Repository (JCR) that implements all of the JCR API.

A content repository is an information management system that provides various services for storing, accessing, and managing content. In addition to a hierarchically structured storage system, common services of a content repository are versioning, access control, full text searching, and event monitoring.

:content\_repository.dflt\_bundle

node_id_hi: char(18)
node_id_lo: char(18)
bundle_data: char(18)

:content\_repository.dflt\_refs

node_id_hi: char(18)
node_id_lo: char(18)
refs_data: char(18)

:content\_repository.secuty\_bundle

node_id_hi: char(18)
node_id_lo: char(18)
bundle_data: char(18)

:content\_repository.secuty\_refs

node_id_hi: char(18)
node_id_lo: char(18)
refs_data: char(18)

:content\_repository.dflt\_binval

binval_id: char(18) (AK1.1)
binval_data: char(18)

:content\_repository.dflt\_names

id: char(18)
name: varchar(255)

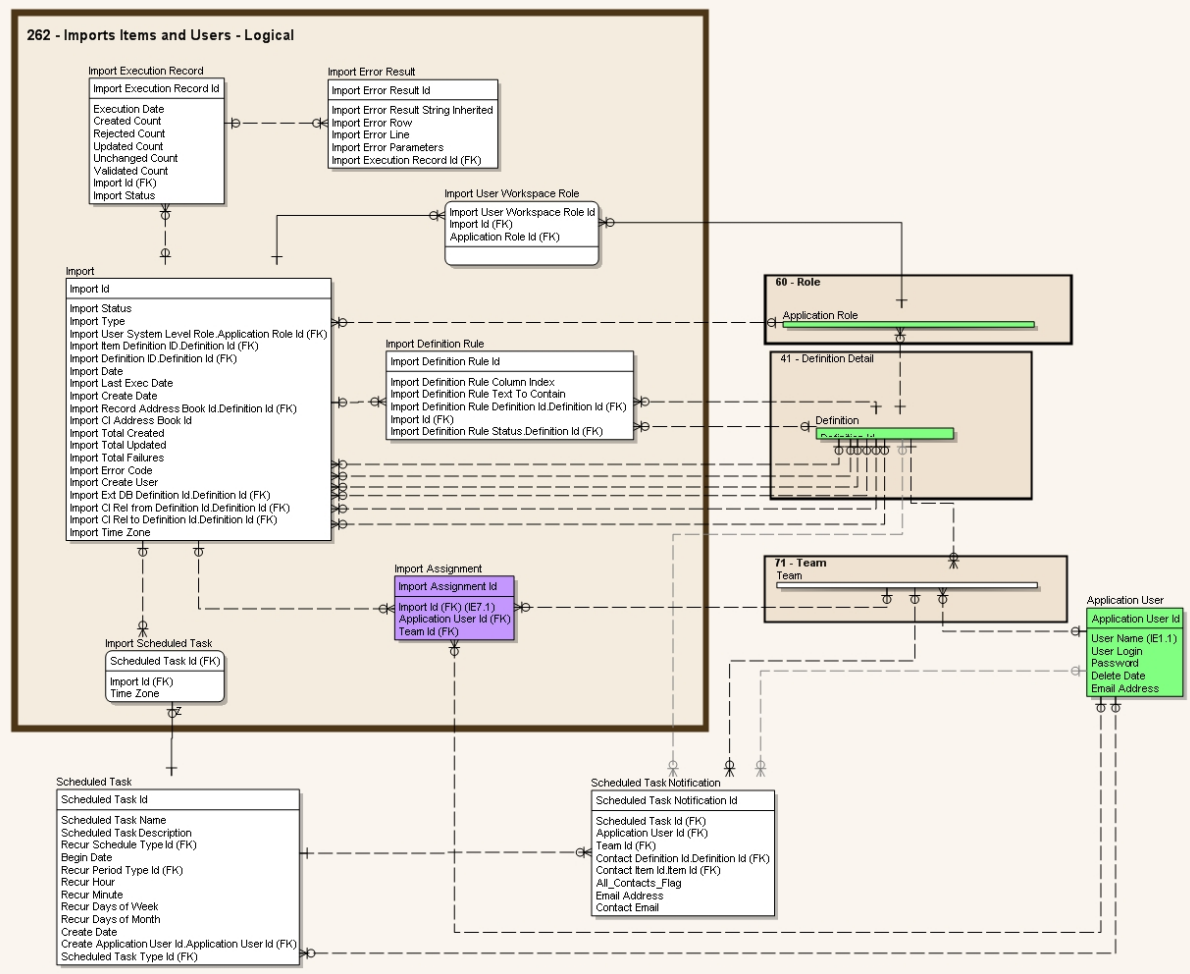
:content\_repository.secuty\_binval

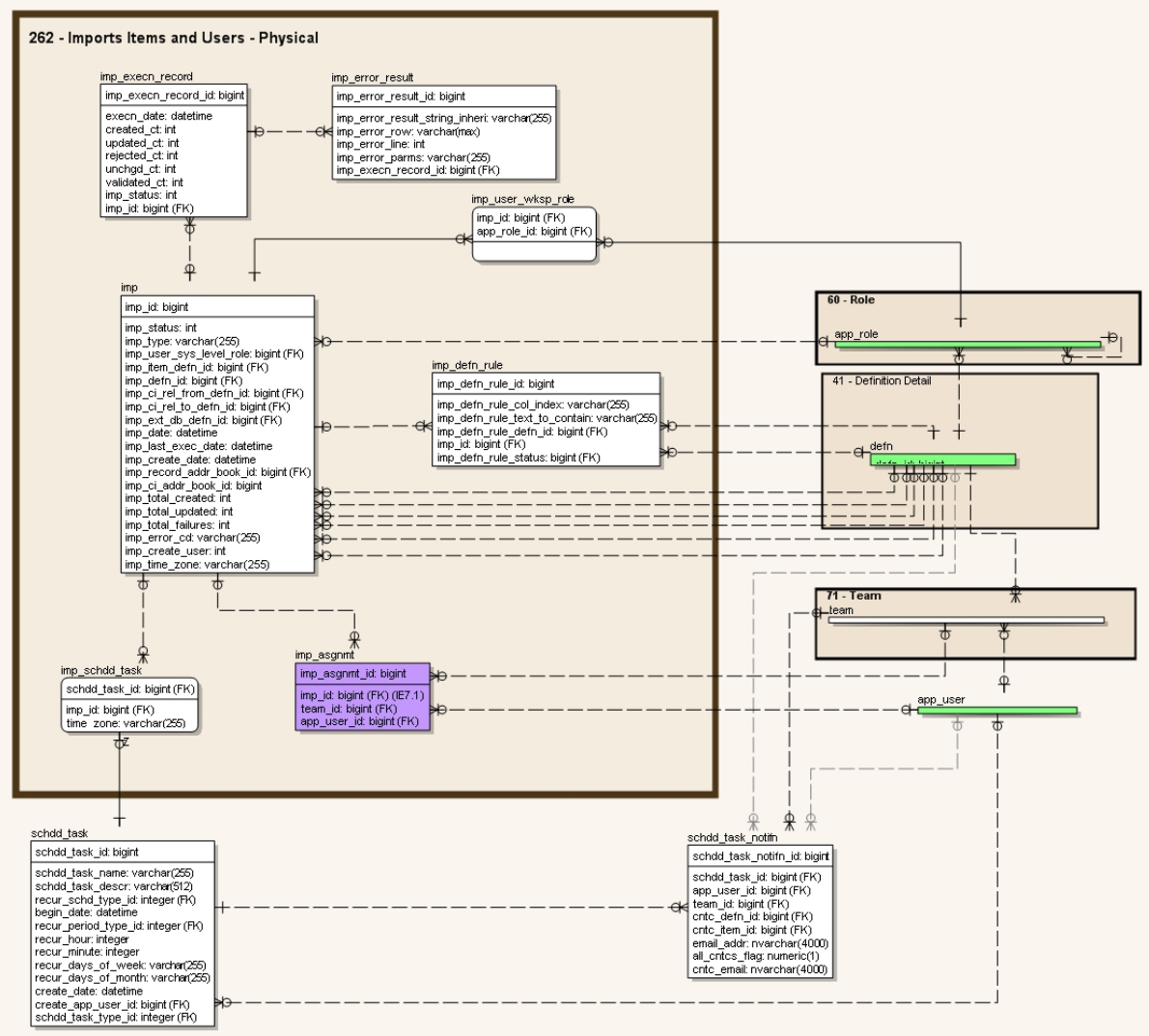
binval_id: char(18) (AK1.1)
binval_data: char(18)

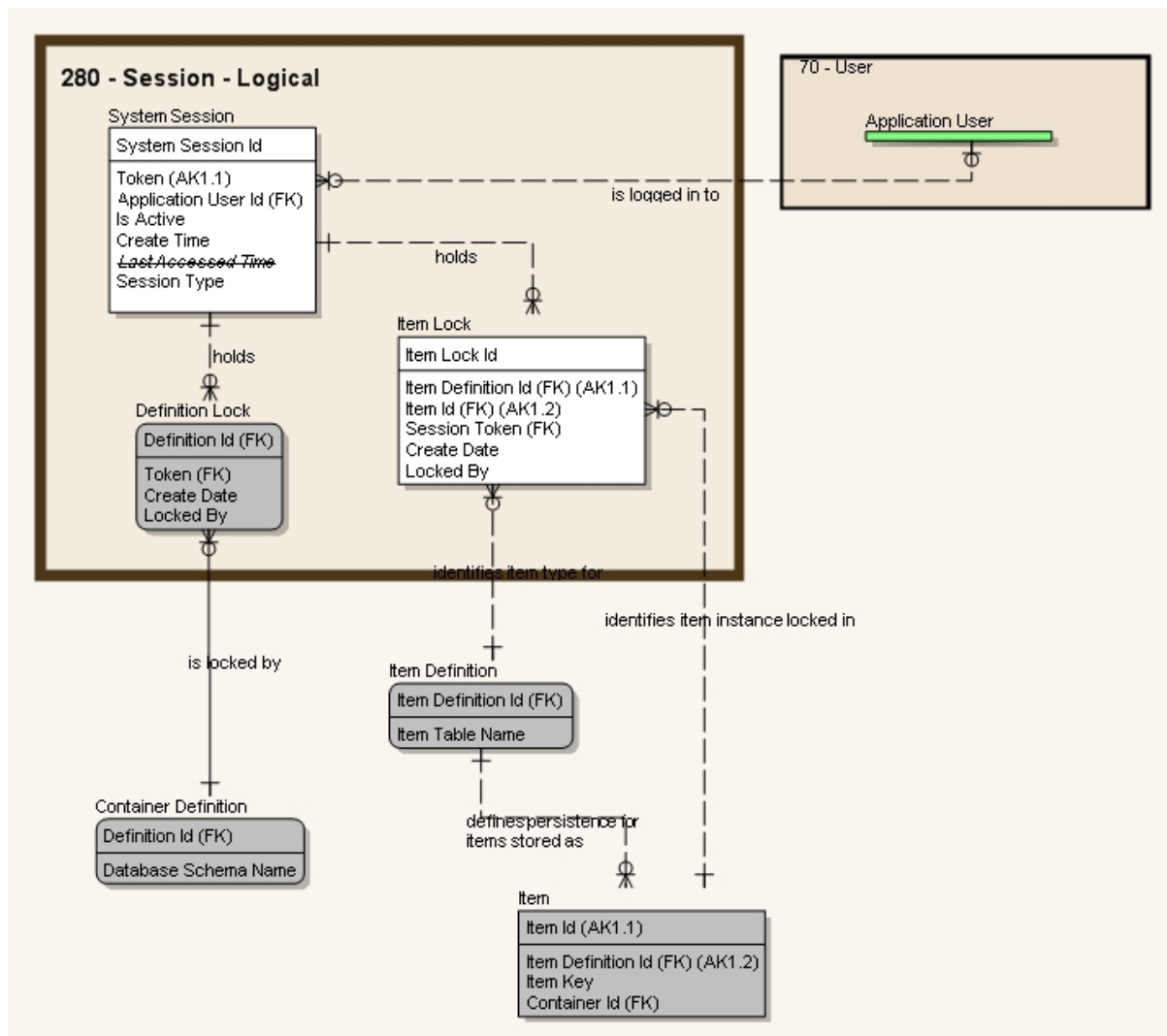
:content\_repository.secuty\_names

id: char(18)
name: varchar(255)

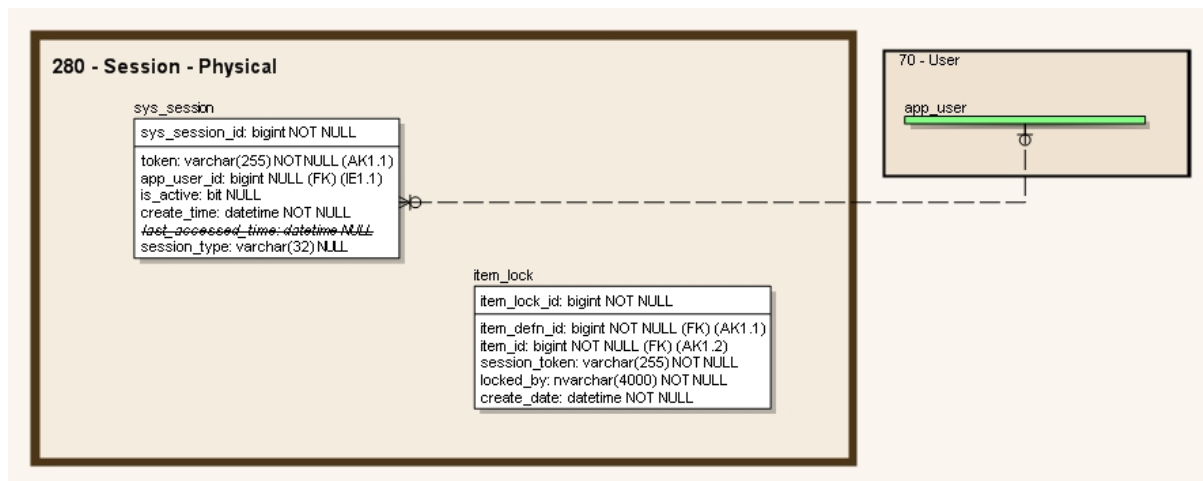
## Logical



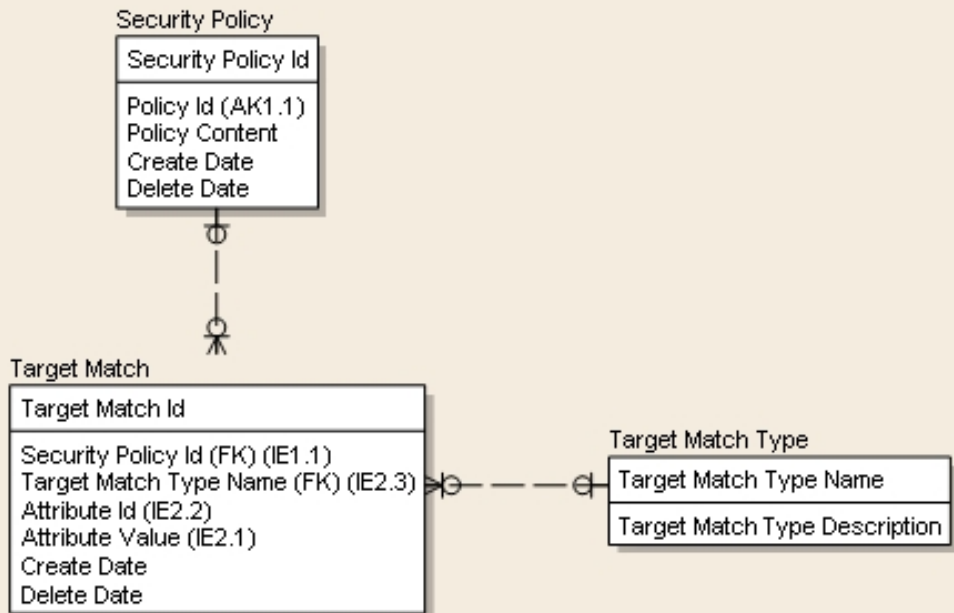




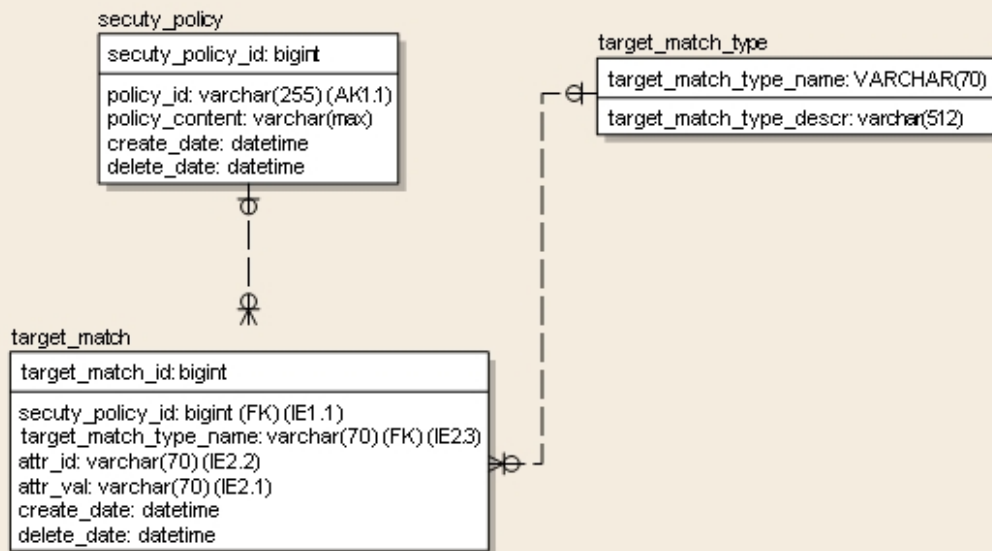




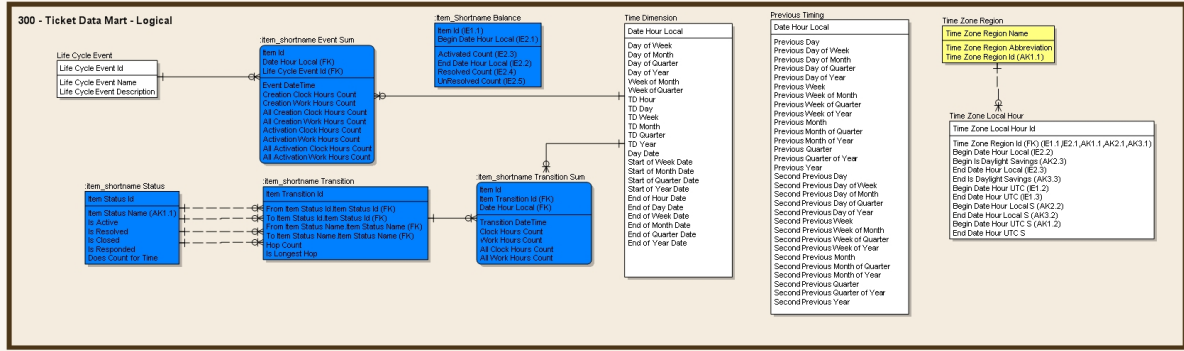
## 290 - Security - Logical



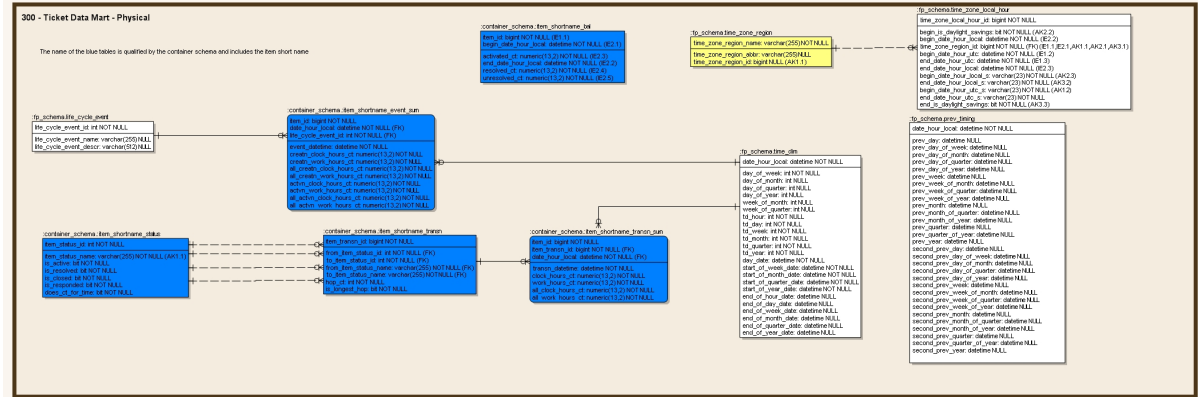
## 290 - Security - Physical



# Logical

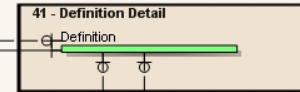
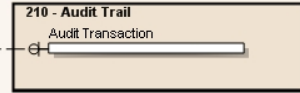
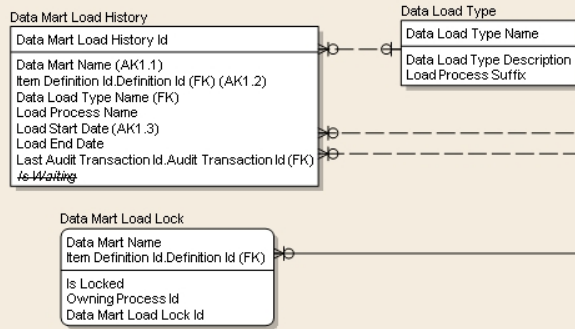


Physical	
----------	--

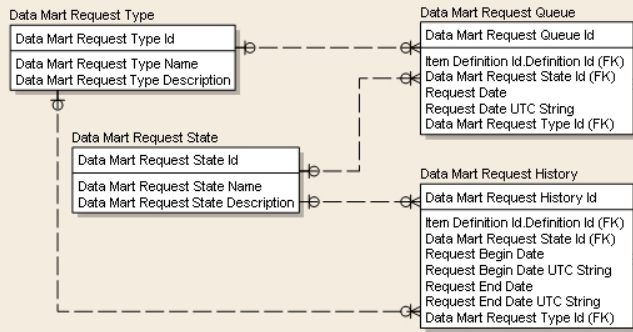


### 305 - Data Mart Load - Logical

#### Load Execution Management

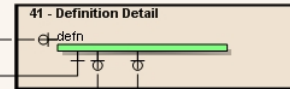
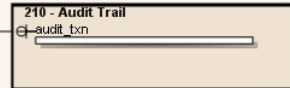
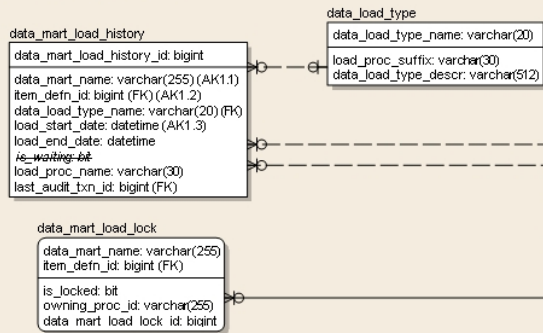


#### Load Request Management

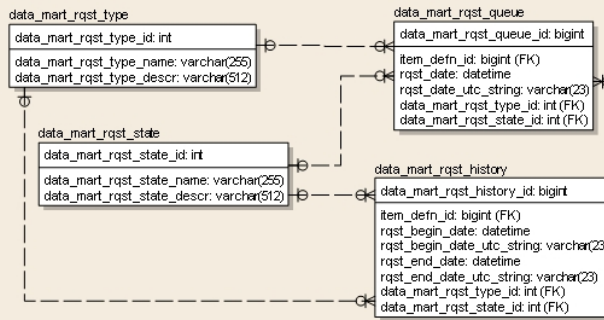


## 305 - Data Mart Load - Physical

## Load Execution Management



## Load Request Management



### 310 - Auxiliary Tables - Logical

#### Configuration File Repository

Configuration File Related Path
---------------------------------

Configuration File Content
----------------------------

#### External System Certificate

External System Certificate Id
--------------------------------

External System Certificate Name
Certificate



### 310 - Auxiliary Tables - Physical

cfg\_file\_repository

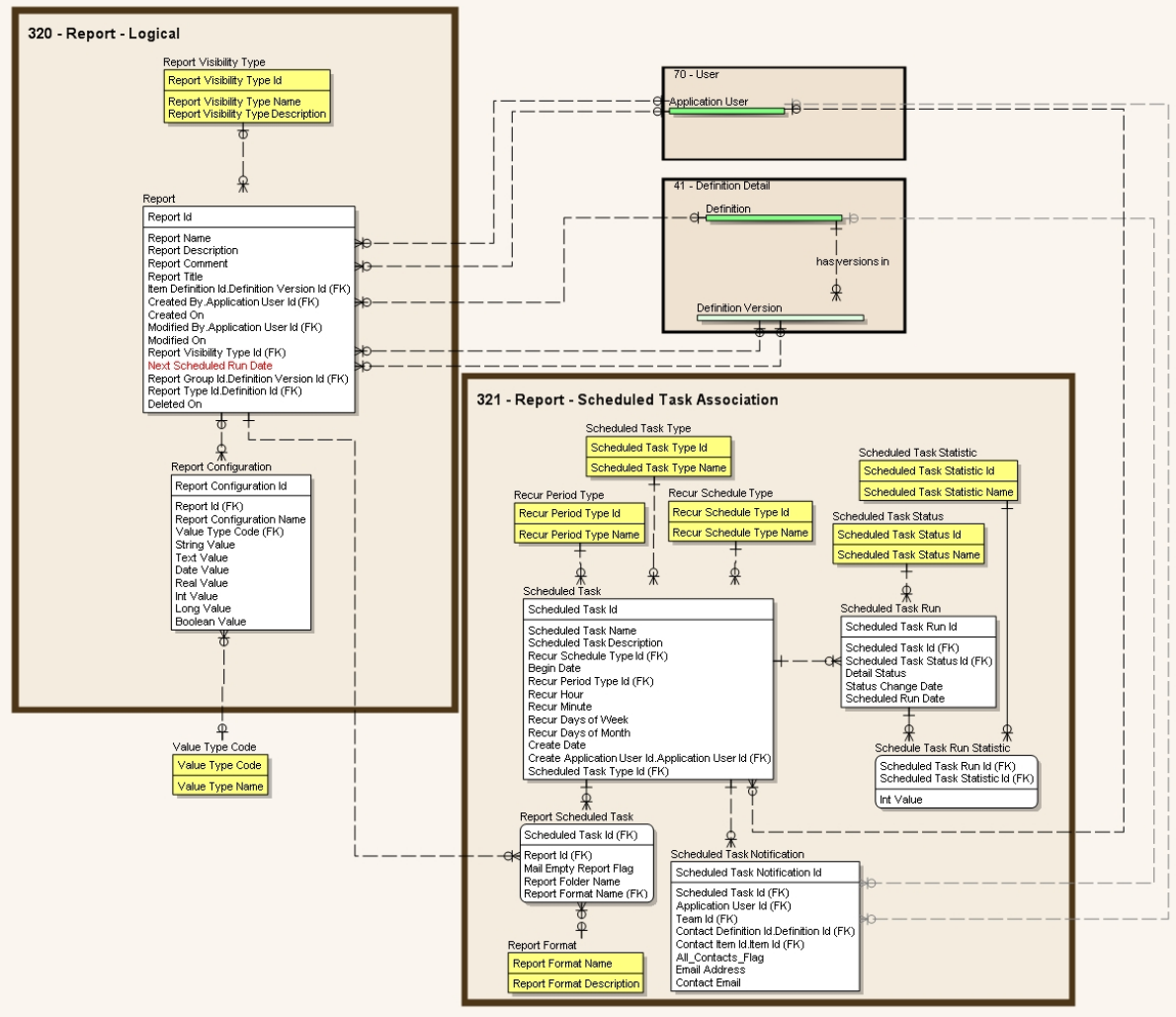
cfg_file_related_path: varchar(255)
-------------------------------------

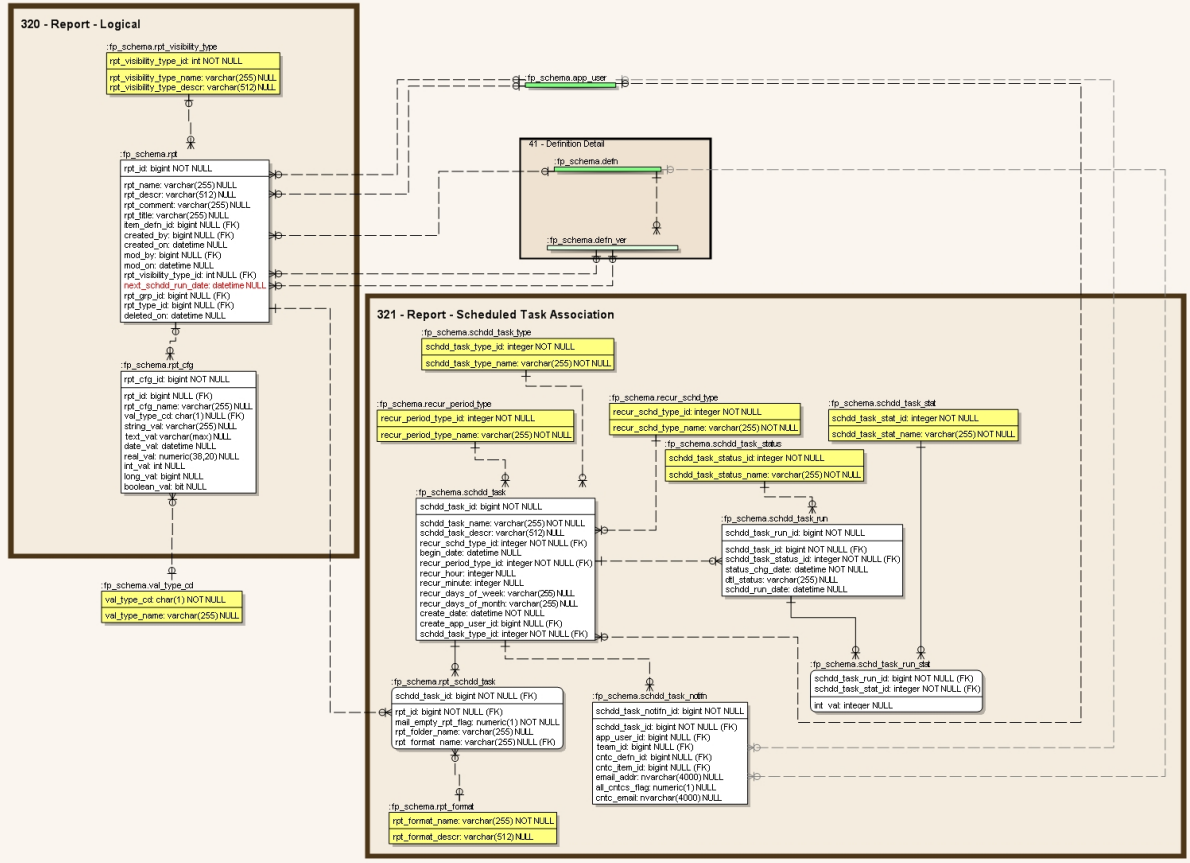
cfg_file_content: varbinary(max)
----------------------------------

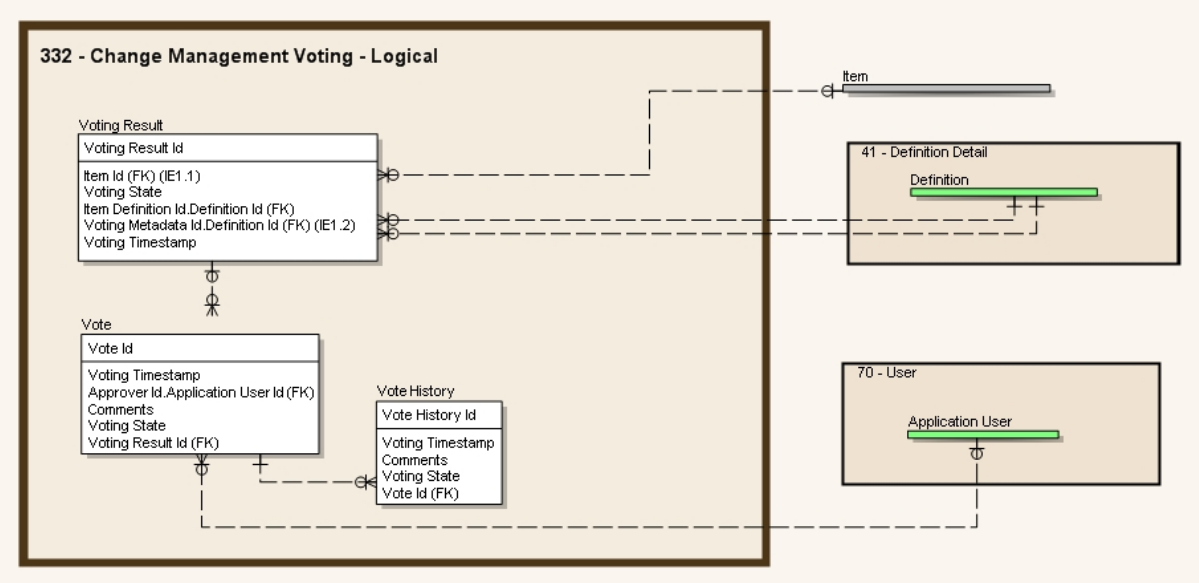
extl\_sys\_certificate

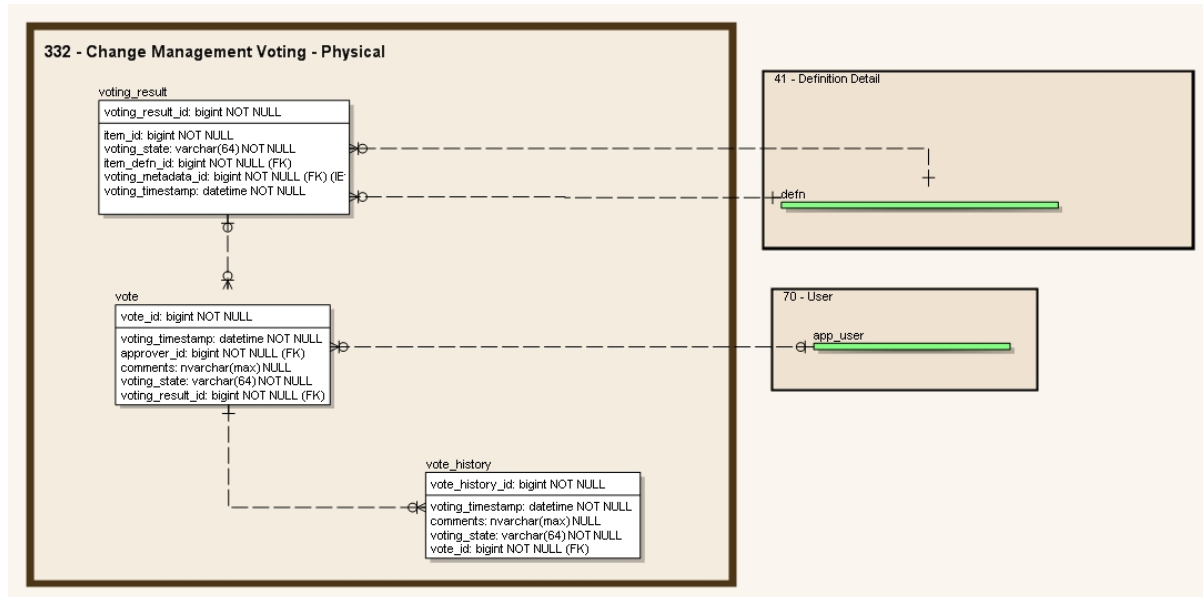
extl_sys_certificate_id: bigint
---------------------------------

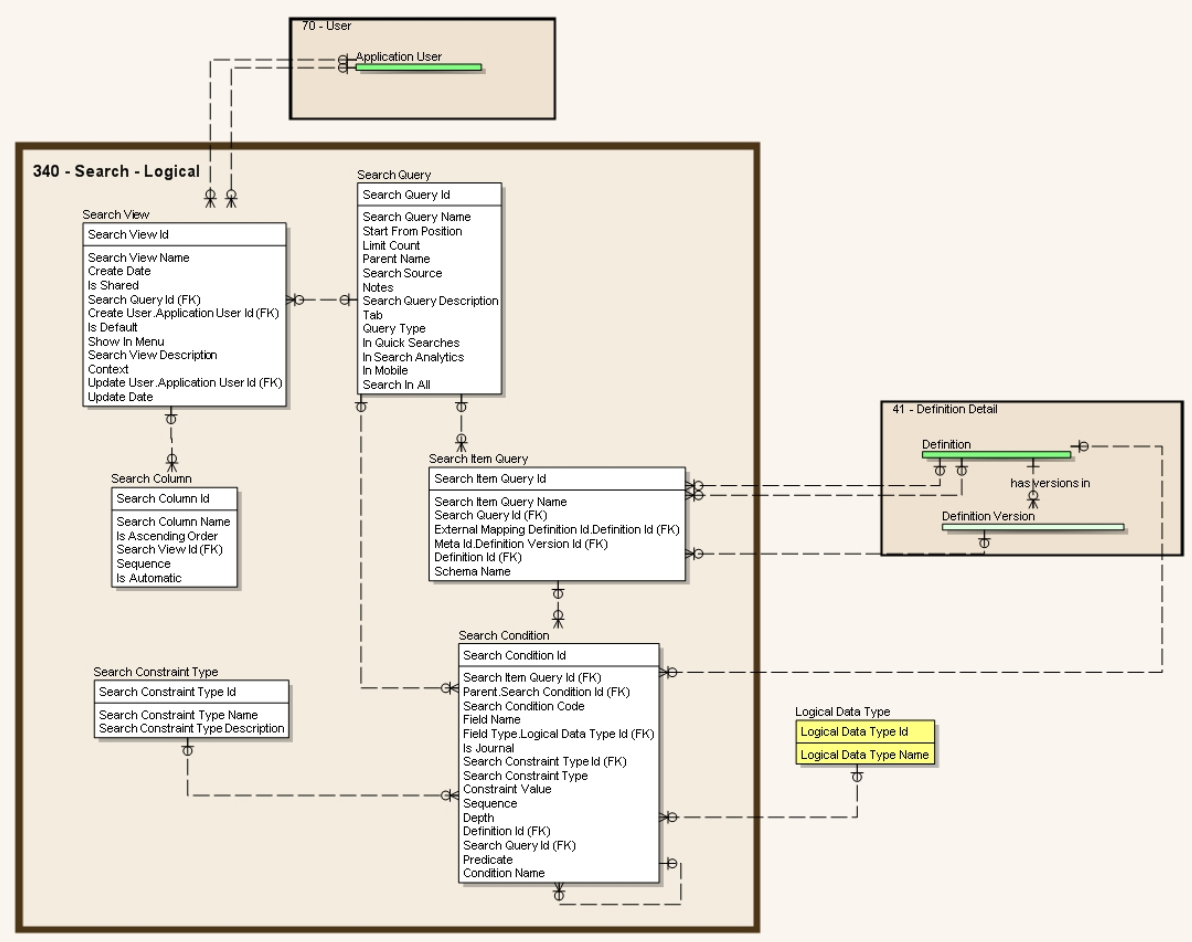
extl_sys_certificate_name: varchar(255)
certificate: varbinary(max)



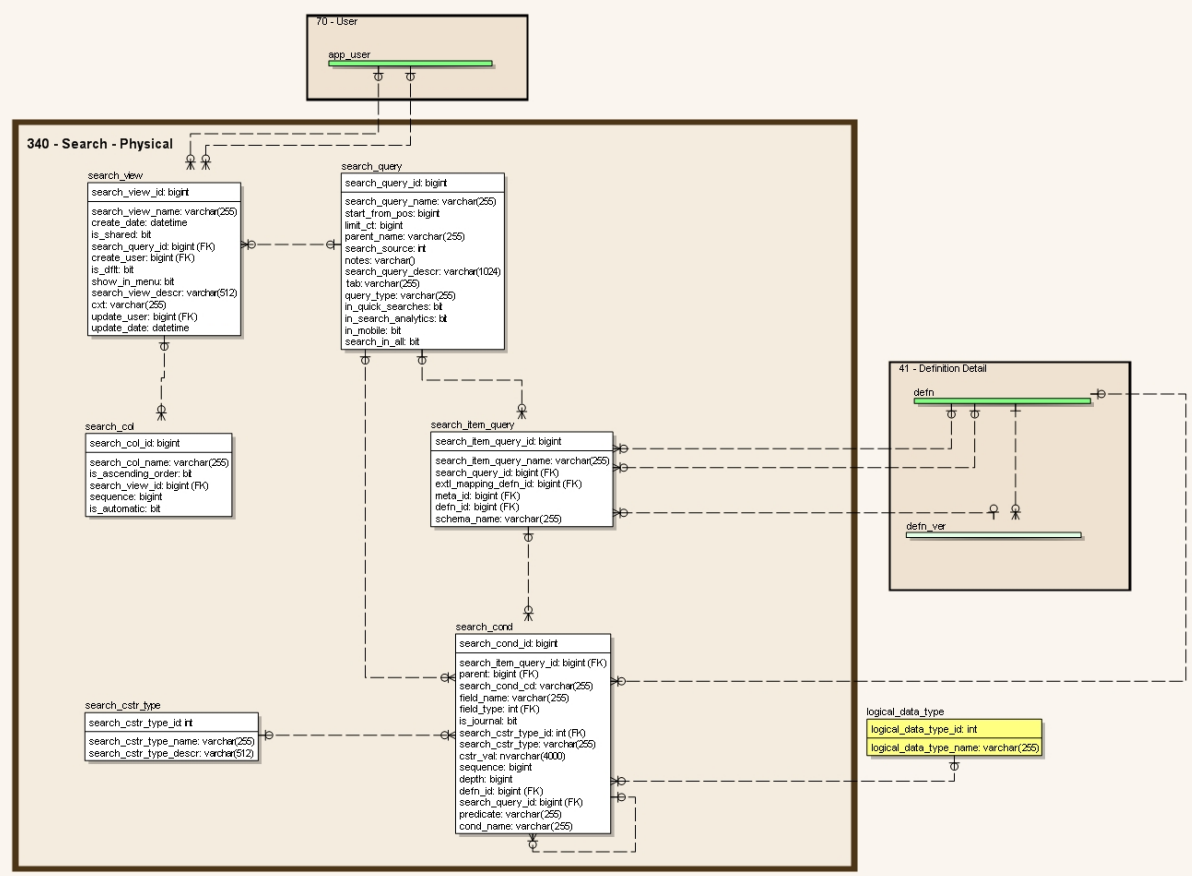


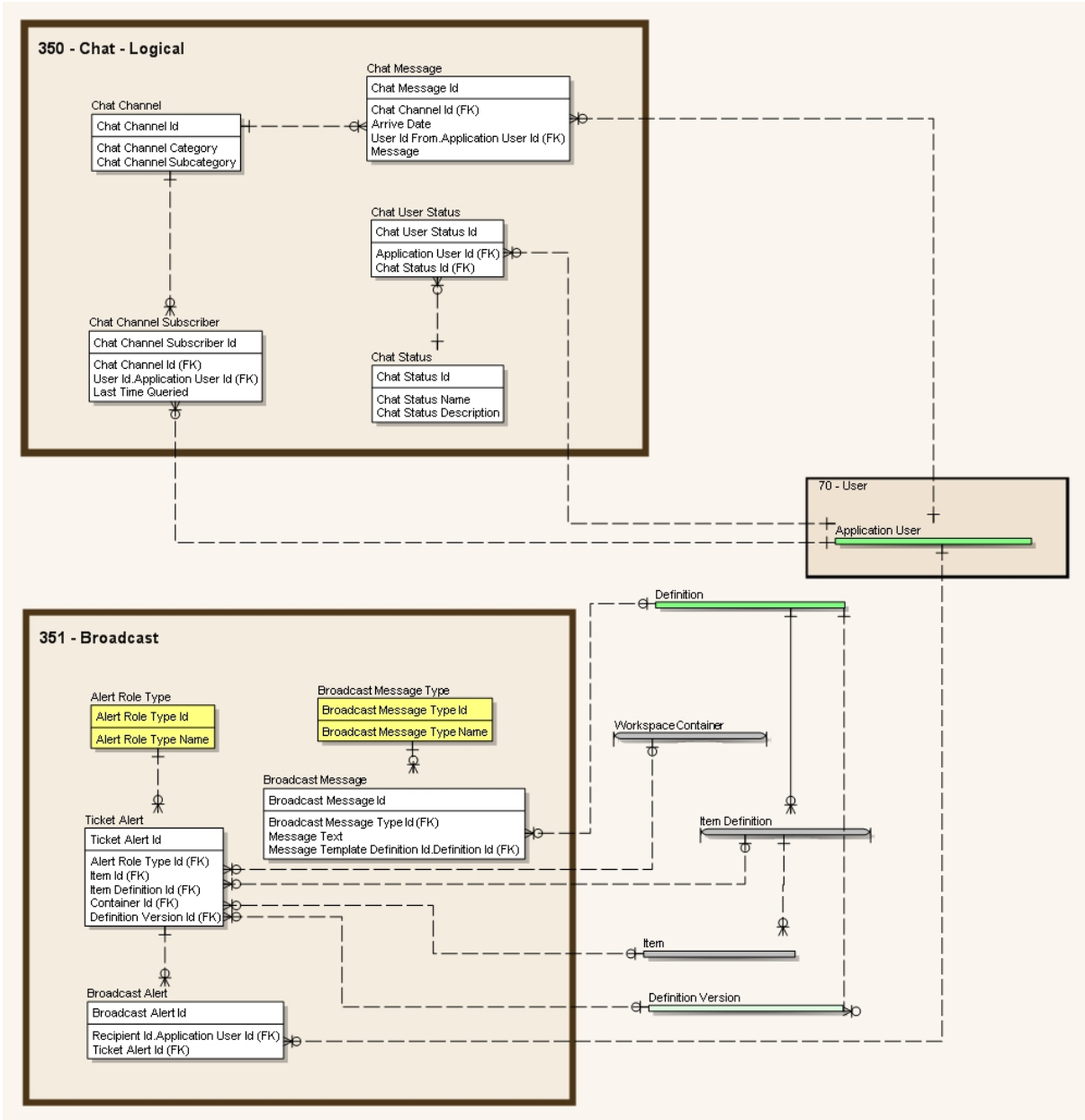






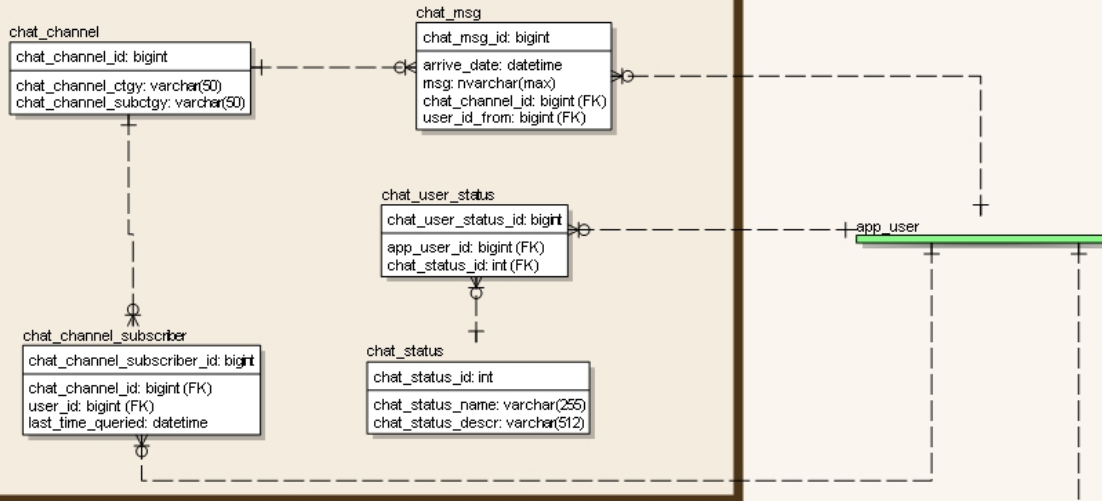
## Physical



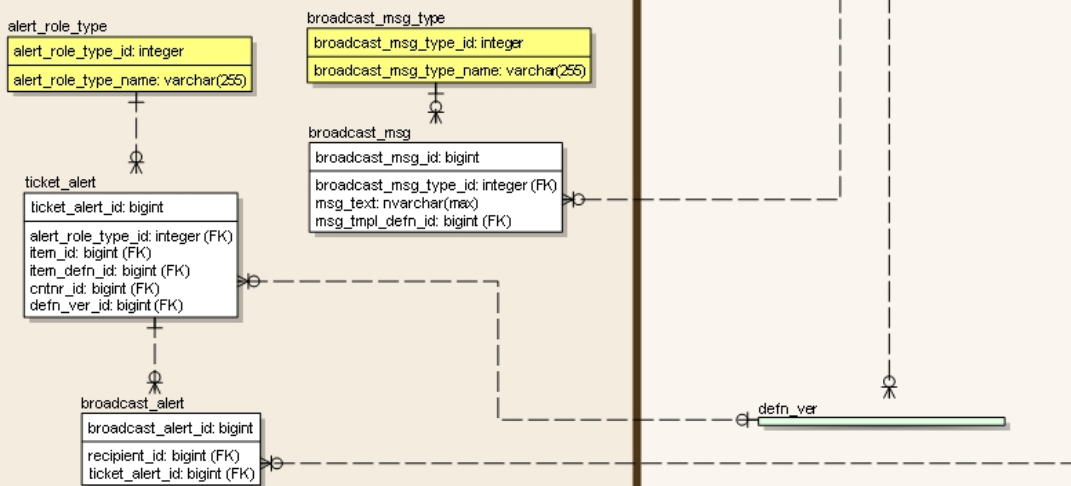


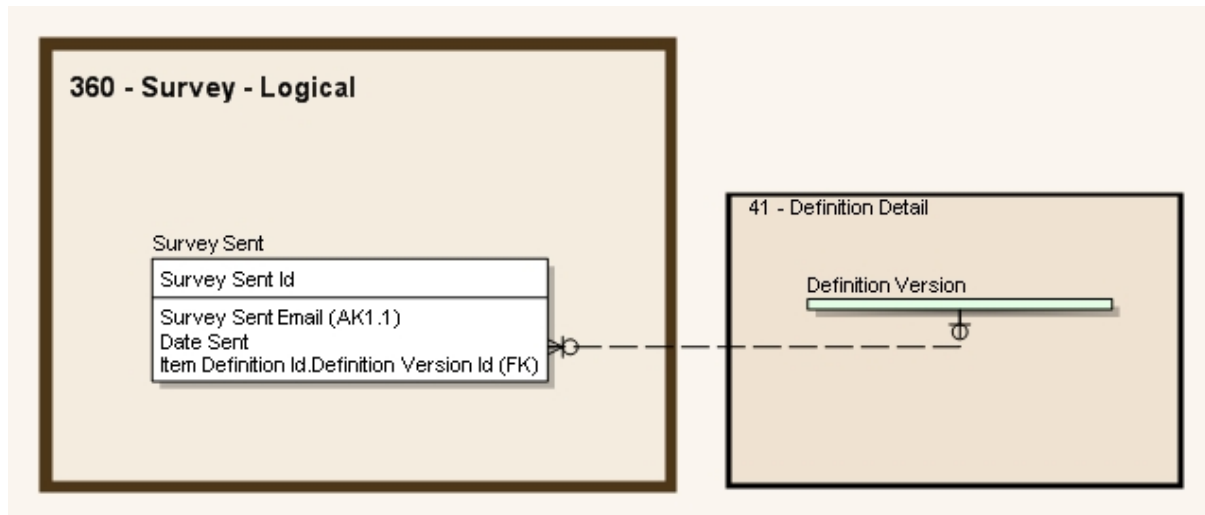


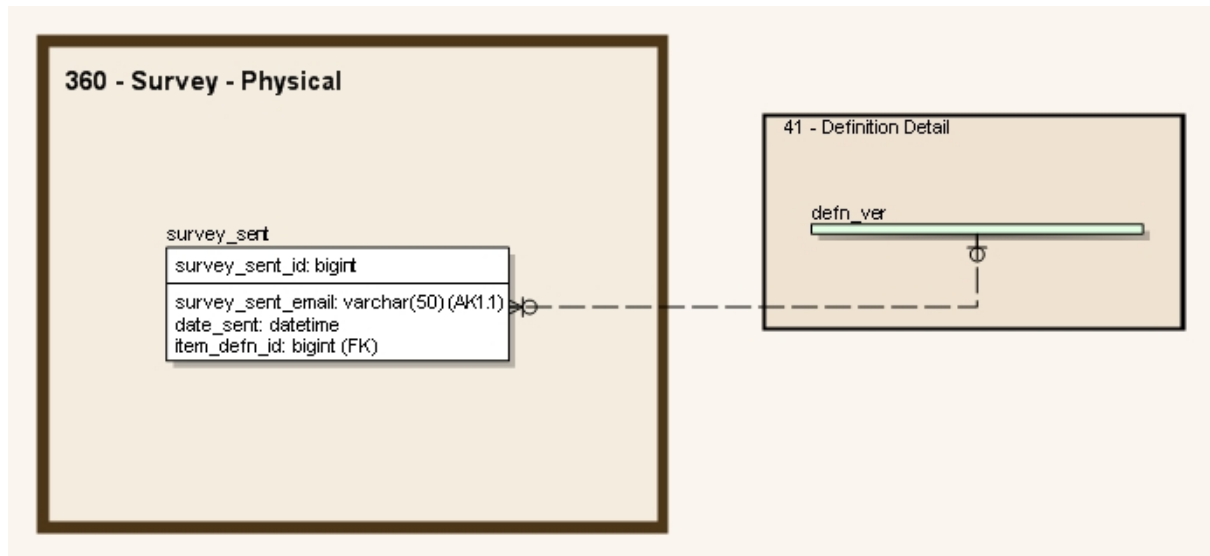
## 350 - Chat - Logical



## 351 - Broadcast







### 370 - Deployment - Logical

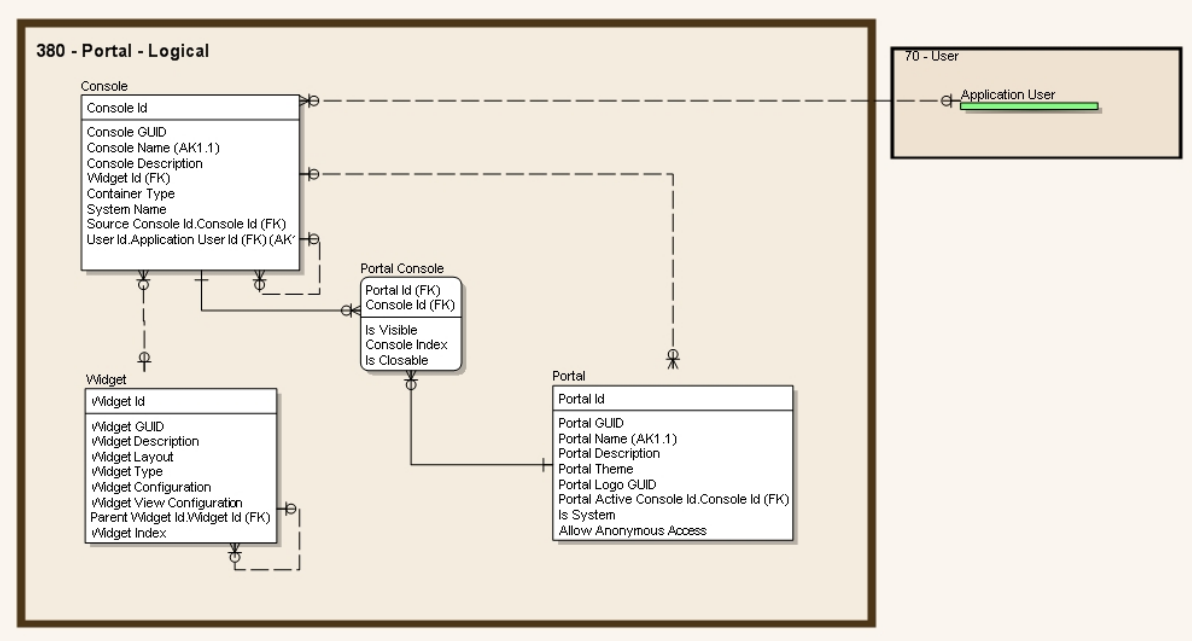
#### Deployment Update

Update Pack Id
From Version
To Version
Product Version
Build Number
Start Datetime
Finish Datetime
Success
Error

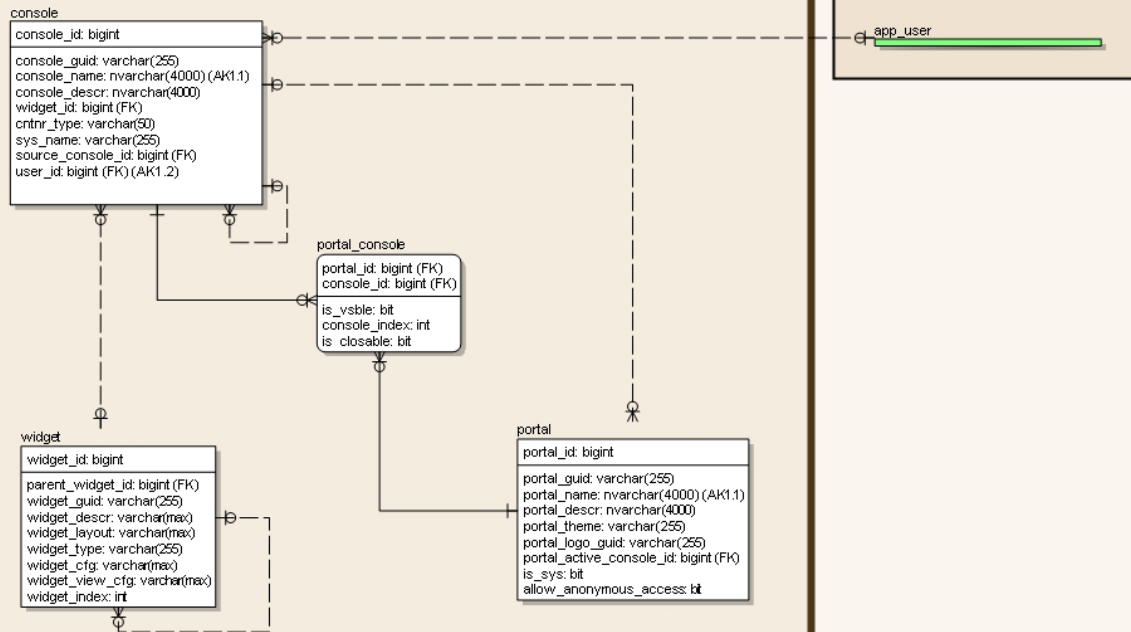
## 370 - Deployment - Physical

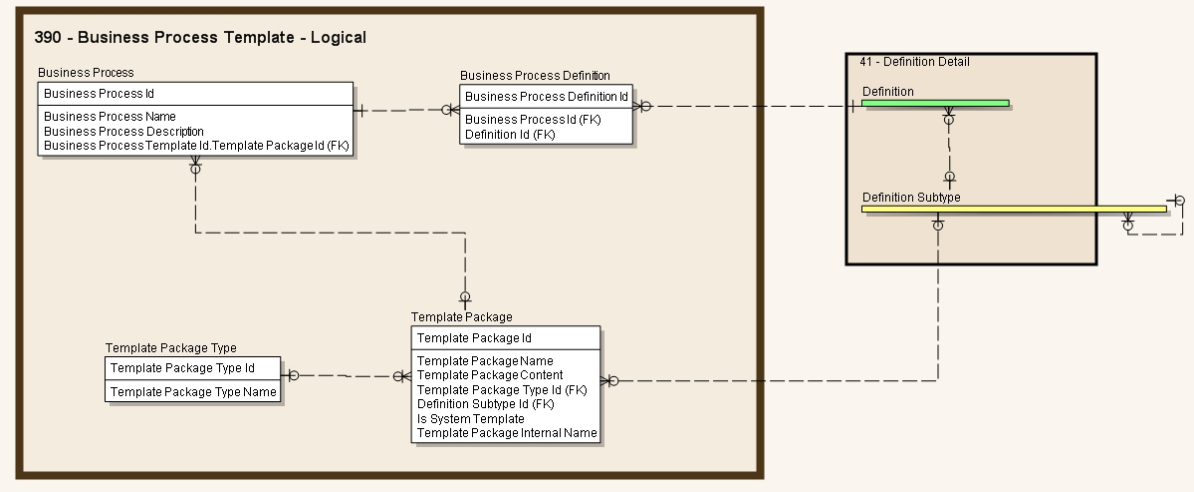
deployment\_update

update_pack_id: varchar(64)
from_ver: varchar(64)
to_ver: varchar(64)
prod_ver: varchar(64)
build_number: varchar(64)
start_datetime: datetime
finish_datetime: datetime
success: bit
error: varchar(4000)



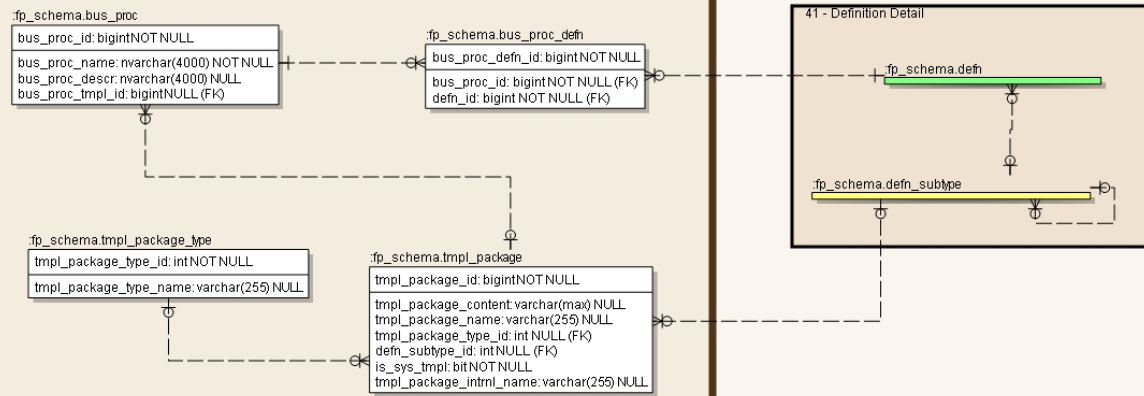
## 380 - Portal - Physical

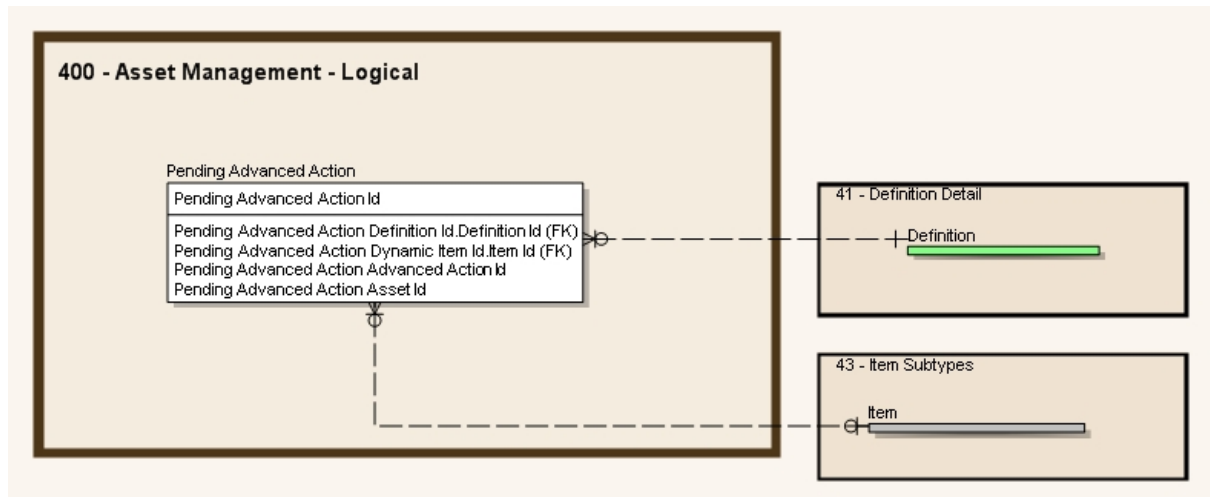


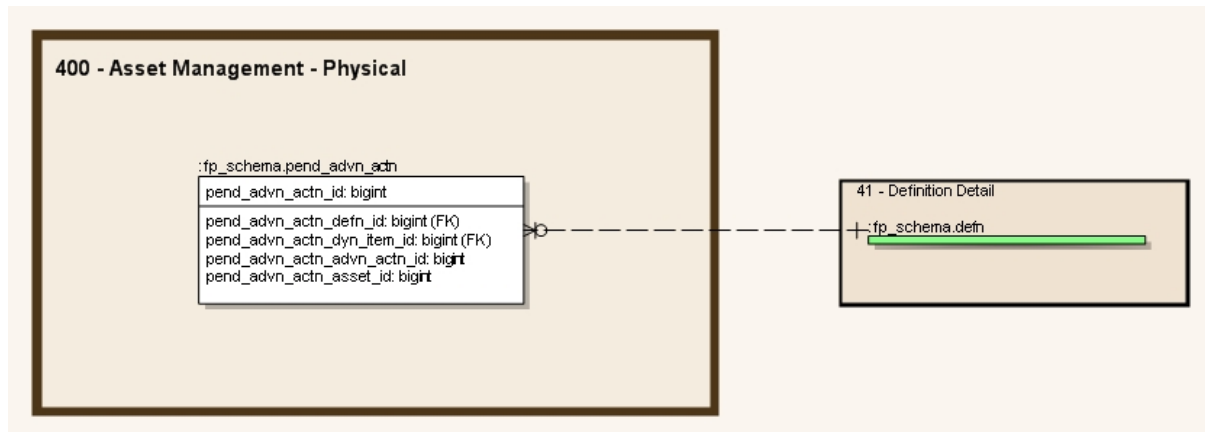


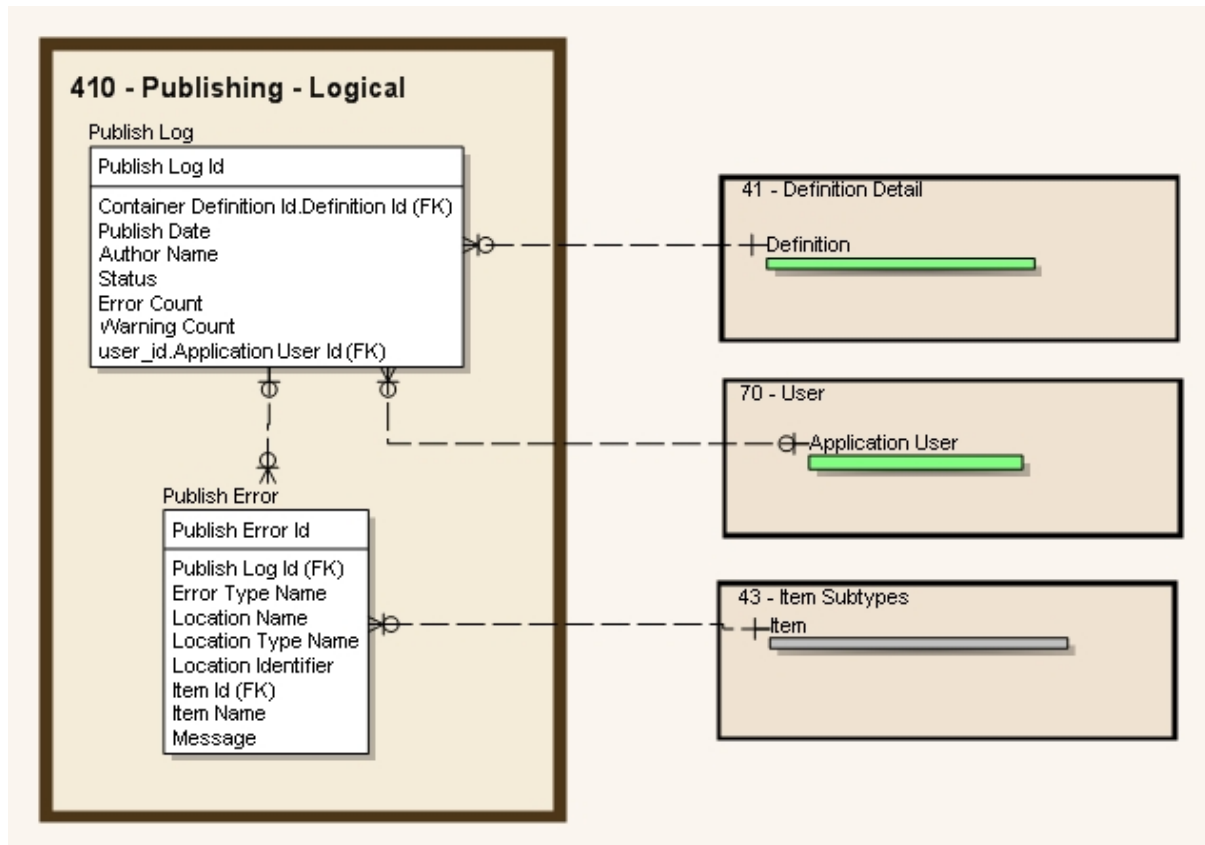


## 390 - Business Process Template - Physical









**410 - Publishing - Physical**

publ\_log

publ\_log\_id: bigint  
cntnr\_defn\_id: bigint (FK)  
publ\_date: datetime  
author\_name: nvarchar(4000)  
status: bit  
error\_ct: smallint  
warn\_ct: smallint  
user\_id: bigint (FK)



publ\_error

publ\_error\_id: bigint  
publ\_log\_id: bigint (FK)  
error\_type\_name: varchar(10)  
location\_name: nvarchar(max)  
location\_type\_name: varchar(255)  
location\_id: varchar(36)  
item\_id: bigint  
item\_name: nvarchar(max)  
msg: nvarchar(max)

41 - Definition Detail

defn

70 - User

app\_user

### 430 - UI Component State - Logical

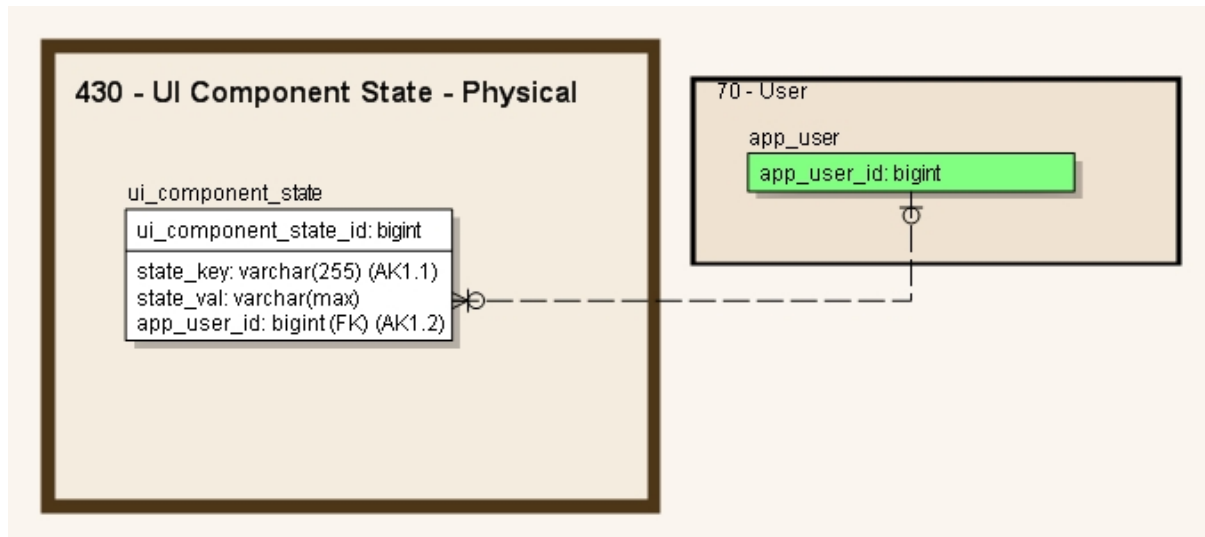
UI Component State

UI Component State Id
State Key (AK1.2)
State Value
Application User Id (FK) (AK1.1)

70 - User

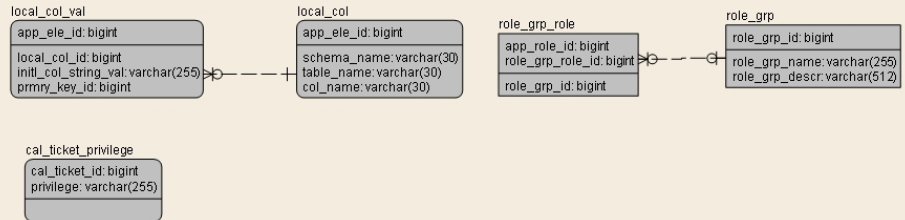
Application User



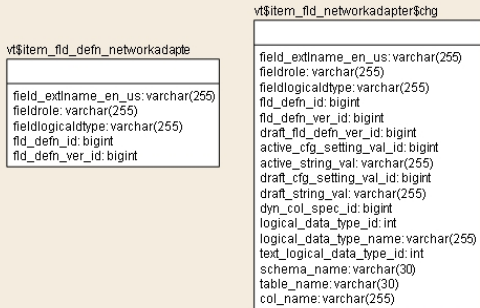


900 - To Be Deleted - Physical

Requires Java code removal before Table removal. But Java code is not used, so tables are empty. Ignore, do not use, but do not delete.



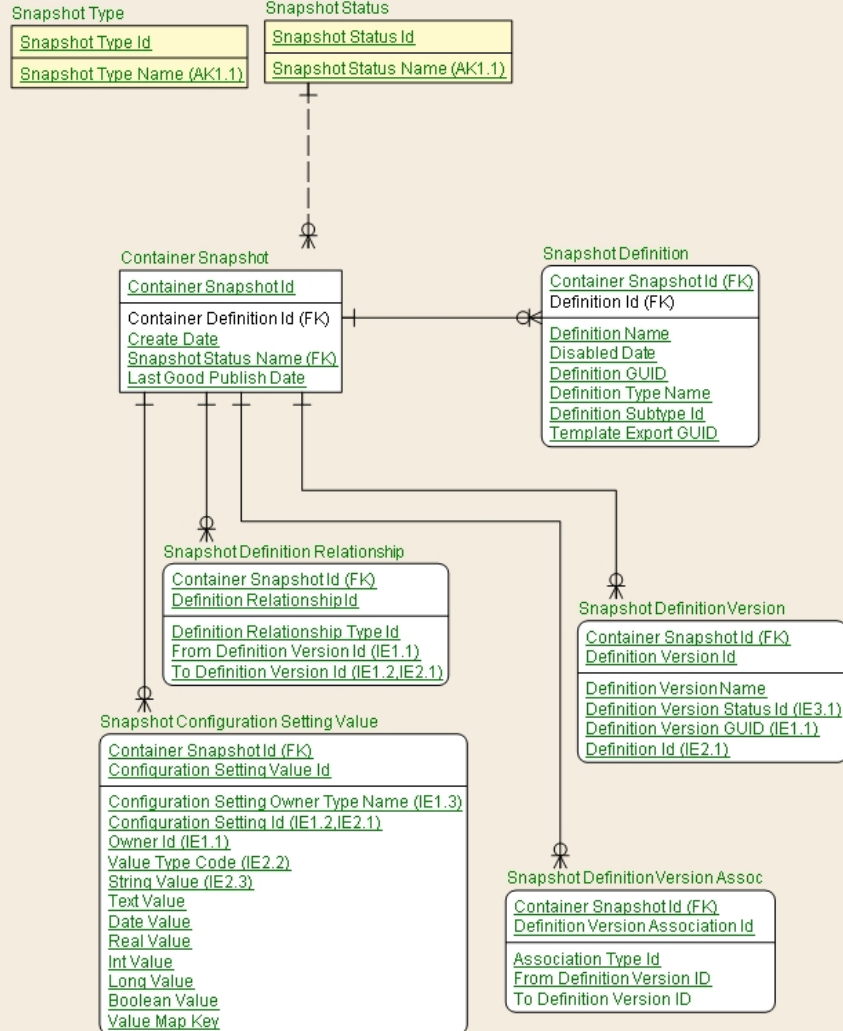
12.1.03 upgrade temporary tables used to expand size of network adapter column for BCM asset integration. They remain just for debugging purposes if needed. They will be deleted in the next release.





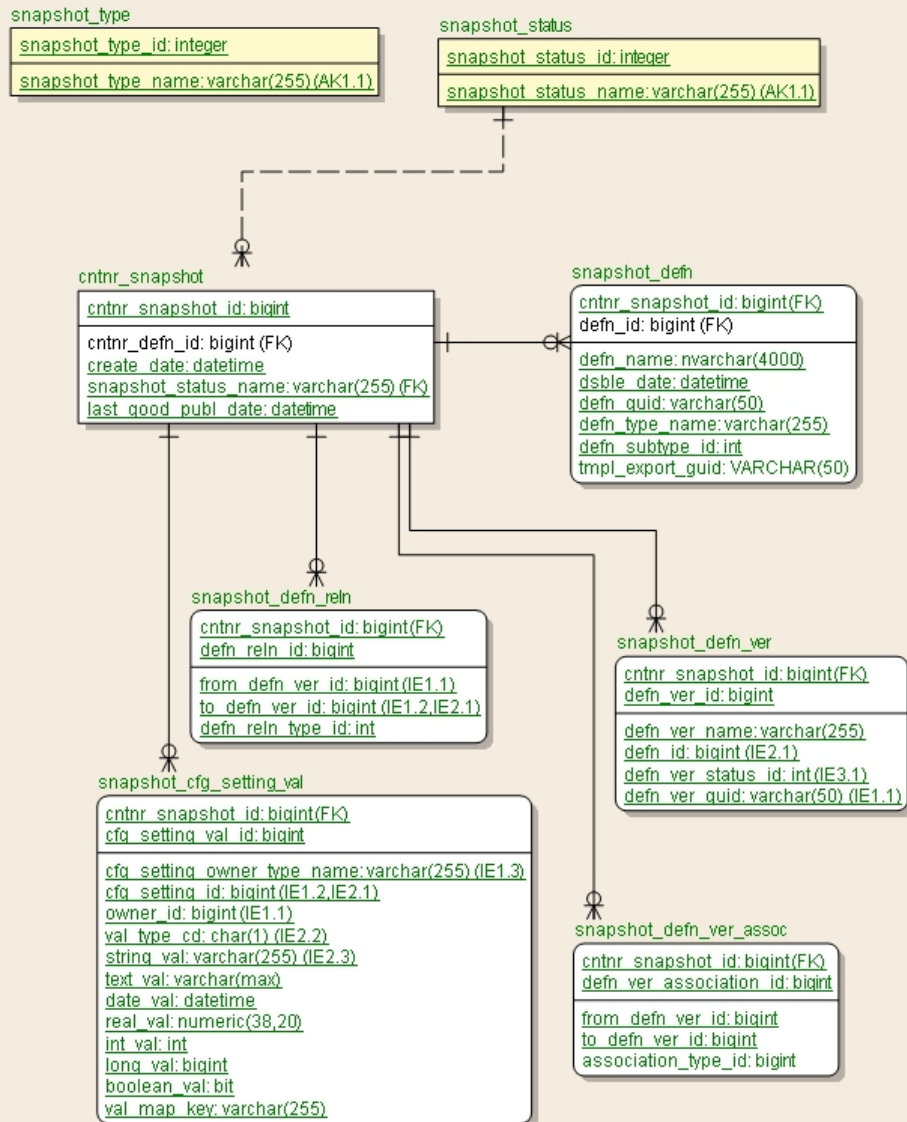
901 - Tables Added From FootPrints 12.1.08 to 12.1.09

066 - Container Snapshot



## 901 - Tables Added From FootPrints 12.1.08 to 12.1.09

## Container Snapshot



**902 - Columns Added From FootPrints 12.1.08 to 12.1.09**

**(none)**

**902 - Columns Added From FootPrints 12.1.08 to 12.1.09**

**(none)**

**903 - Tables Deleted From FootPrints 12.1.08 to 12.1.09**

**( no tables dropped )**

**903 - Tables Deleted From FootPrints 12.1.08 to 12.1.09**

**( no tables dropped )**

**904 - Columns Deleted From FootPrints 12.1.08 to 12.1.09**

**( no columns dropped )**

**904 - Columns Deleted From FootPrints 12.1.08 to 12.1.09**  
**( no columns dropped )**



## Legend

### Color Scheme

Green indicates that the entity is the main (or one of the main entities) in the subject area.

#### Configuration Setting

Configuration Setting Id  
Configuration Setting Constant (AK1.2)  
Default Setting Label  
Default Setting Description  
Configuration Category Id (FK) (AK1.1)  
Configuration Value List Id (FK)  
Agent Only  
Unique User Only  
Value Type Code  
Configuration List Type Name (FK)  
Specification Class Name  
Configuration Is Value Mapped

#### Configuration Setting Owner Type

Configuration Setting Owner Type Name  
Configuration Setting Owner Type Description

Yellow indicates that the entity represents a type (usually corresponding to an enumeration class in the code).

Grey indicates that the entity represents a logical construct that will not be represented physically in the final data tables. The entity in question is thought important enough to include for information and understanding purposes.

#### Item Definition

Item Definition Id (FK)  
Item Table Name

#### Delete Entry

Delete Entry Number Inherited  
Delete Entry Name  
Delete Entry Comment

Red indicates that the entity is being deleted from the model, and is most likely present because the code has not incorporated the deletion change yet.

### Version Information

BMC FootPrints Service Core Version 12.1.09 - SQL Server  
Diagram Last Updated January 26, 2018

### Add/Deletion Notation

#### New!

Added Table  
Added Table Id  
Added Column Name  
Added Column Description

Newly Added Tables are indicated with a Green underline font style.  
-- There is also a New Indicator

Deleted Tables are indicated with a Red strike through font style.

#### Deleted-Table

Deleted-Table-Id  
Deleted Table Name  
Deleted Table Description

#### Added Column

Added Column Id  
Added Column Name  
Added Column Description

Newly Added Columns are indicated with a Green underline font style.

Deleted Columns are indicated with a Red strike through font style.

#### Deleted Column

Deleted Column Id  
Deleted-Column-Name  
Deleted Column Description

### Erwin Information Engineering Notation

#### Dashed Relationship Line

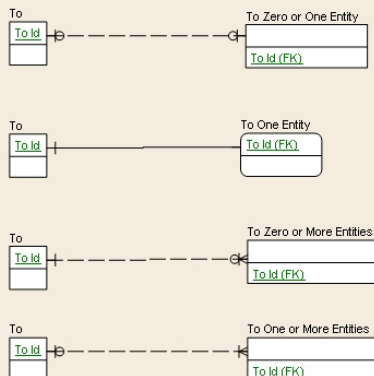
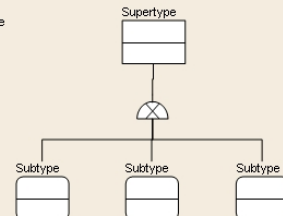
A non-identifying relationship is a relationship between two entities in which an instance of the child entity is not identified through its association with a parent entity, which means the child entity is not dependent on the parent entity for its identity and can exist without it. In a non-identifying relationship, one instance of the parent entity is related to multiple instances of the child.

Solid Relationship Line - Rounded Entity Corners

An identifying relationship is a relationship between two entities in which an instance of a child entity is identified through its association with a parent entity, which means the child entity is dependent on the parent entity for its identity and cannot exist without it. In an identifying relationship, one instance of the parent entity is related to multiple instances of the child.

#### Subtype Relationships

In an exclusive subtype relationship, an attribute in the supertype entity can derive its value from only one subtype entity. For example, if a model includes an EMPLOYEE supertype entity with FULL-TIME and PART-TIME subtype entities, the employee-status attribute value for each employee can be full-time or part-time, but not both.



Subject Area(s) of Model "Renoir Domain Model Detail"	
Name	Definition
000 - Very High Level Domain Model	A high-level Domain model indicating the logical concepts used to design both the Data Model and the persistent Object Model. The Definition is a key abstract class used to define dynamic objects that can be customized and published. Gray Objects indicate Logical Only concepts that do not correspond to unique tables but are stored in other tables. Child tables may be stored in a hierarchy, or template tables may be generated as multiple concrete tables.
001 - High Level Domain Model	A high level Domain model indicating the logical concepts used to design both the Data Model and the persistent Object Model. This diagram has all of the main concepts needed to describe FootPrints 12. It was prepared by aggregating and summarizing 50 Logical Data Models prepared from analysis of some 700+ FootPrints 11 screen shots and associated functional descriptions of the new system. It is refined into physical tables in this model.
010 - Translation	This model provides for translation of all strings necessary in FootPrints Java. Much of FootPrints is customizable, and much of that customization can be used internally in the form of application strings, delivered Data Templates, and customer data entry. The process of defining the need for a string and obtaining the string and not losing association with the string is important. This model uses the idea of namespaces and identifiers to uniquely identify strings. Much of the UI code uses the actual English string as the unique identifier.
040 - Container	The Container is an important abstraction of the FootPrints concepts of Workspace, Address Book, CMDB, and others. The idea is that the Container has specific settings that apply to the Items that can be defined to the Container. So the Container Type allows certain Item Types. It also means that new Container Types such as Knowledge Base, Survey can be defined without having to describe them as Workspaces.
041 - Definition Detail	A detailed look at the tables needed to store a definition. The idea is that all dynamic objects can share the functionality of having versions, being published, and participating in relationships with other definitions. A new definition type can be added to the Java code without affecting the database storage.
046 - Workspace Item Assignment	The concept of assignment of Workspace Item to Team or Application User or an external party.
047 - Workspace Time Tracking	Information about the time spent viewing and editing a workspace item.
048 - Item Field Journaling	Any field with journaling turned on will have a collection of historical values in this table, in addition to the current value in the dynamic item.
050 - Item Relationship	A generic table that stores a relationship between two dynamic items. Since the dynamic items might be in different containers, all relationships are stored in a single generic table. Partitioned indexes might be designed on this table if performance becomes a problem, using one or both definition ids as partition keys.
051 - Item Field Collection	This shows the table for multi-select fields.
060 - Role	Shows the different types of FootPrints Roles and the relationship to Application User types and Use Profiles. There is some redundancy of function in the Perl FootPrints security model that is the subject of design here.
065 - Configuration Setting	The intent of the model is to provide for documentation, implementation, localization, and business behavior for the large amount and variety of Configuration Data in FootPrints. The Configuration is broken into Categories which are independently designed and developed. The information needed for the Administration UI, the developer, and query access are all facilitated. The same Definition object can be used at Design Time, Install Time, and Runtime. The actual properties of the Definition objects are stored as Configuration Setting Values, with a reference to a Configuration Setting. The Configuration Setting names are defined in Java code with annotations, and are bound to the database by the same name. So all possible names for Configuration Settings are known and stored via code scanning in the Configuration Setting table. The goal of this pattern is to quickly allow new metadata properties to be defined in code, to have these properties existence and possible values known in the database, to enable query optimization by holding a cache of long strings vs. lengthy names in name/value pairs, and to make it impossible to have errors in the binding of database values and coding values. Database views can be defined that will simplify the display of these properties.
066 - Container Snapshot	
070 - User	The agent or customer application user and types.
071 - Team	Describe the Team of Agent Application Users and the Relationship to Assignment.
080 - User Profile	The user profile contains user preferences. This subject area shows the various types of profiles that are defined.
096 - Email Notification	The email notification system uses table to manage processing of items.
101 - Data Access System	These tables define the physical database structure of the dynamic items for all types of containers. The data access layer uses these tables for SQL generation.
102 - Data Access Scripts	Any component that requires the generation of SQL DDL scripts that are based on dynamic item definitions can use this facility. When a container or item is published, these tables are checked for scripts related to the item type and subtype and are used to generate schema and table specific views and stored procedures.
110 - Id Generation	The id generator table.
120 - Job Scheduler	The tables used to define the job scheduler instance and template for job configuration. Quartz is used to execute the background jobs.
140 - Workspace Calendar and Operation Schedule	The tables used to define the Workspace Calendar aka the Work Schedule. The work schedule is stored as an iCal availability item, and it can be shared across workspaces. These tables simplify the data for reporting.
141 - Calendar Appointments and Items	The tables used to define the Workspace Calendar and associated appointments as iCal items. These tables are accessed exclusively through the Calendar service API.
142 - Calendar Sync	Tables used to sync the calendar appointments with a device that supports a calendar.
160 - Templating	Tables for notification template messaging and substitution of item field values inside messages.
170 - Knowledge Base	Tables to add on to knowledge base items regarding user comments on usefulness.
190 - Password Reset	Tables to support self help Password Reset functionality.

210 - Audit Trail	Tables to support an audit trail of changes to the database and also audit messages formed with a template message and substitution arguments.
230 - Rules	Statistics on Rule usage.
240 - File Storage	Tables that describe attachments stored in the file system.
250 - Apache Jack Rabbit JCR	FootPrints uses Apache Jackrabbit to name and store attachments. Use the API instead of direct access.
262 - Import Items and Users	Tables for importing data to dynamic items.
280 - Session and Locking	Tables to support the FootPrints session and dynamic item pessimistic locking.
290 - Security	Security tables used to store XACML permissions.
300 - Ticket Data Mart	Tables that support the Ticket Data Mart used in reporting.
305 - Data Mart Load	Tables that control the Full and Incremental load process of the Data Mart.
310 - Auxiliary Tables	Internal use tables.
320 - Report	Tables that define the existence and configuration of Reports in the UI.
332 - Change Management Voting	Tables that control Change Management voting.
340 - Search	Tables that defined saved searches and help with query generation and results presentation.
350 - Chat	Tables that support the Chat communication feature.
360 - Survey	Additional tables used to support surveys.
370 - Deployment	Tables to control deployment to the database.
380 - Portal	Tables that define console, portal, and widget objects in the UI.
390 - Business Process Template	Tables that hold definition templates for one or more container that constitute a Business Process Template.
400 - Asset Management	Asset Management integration tables.
410 - Publishing	Tables that control container and item publishing.
430 - UI Component State	Tables that define the UI state of UI components so that they preserve user configurations and layout.
900 To Be Deleted	A list of tables that can be deleted because they are not used. Others are referenced in Java code that must be removed before table deletion.
901 - Tables Added	Tables added since the last release.
902 - Columns Added	Columns added since the last release.
903 - Tables Deleted	Tables dropped since the last release.
904 - Columns Deleted	Columns dropped since the last release.
Legend	

### ER Diagram(s) of Subject Area "000 - Very High Level Domain Model"

Name	Physical Display Level
Entity Level	Column
Table Level	Table

### Entity/Table(s) of Subject Area "000 - Very High Level Domain Model"

Physical Name	Name	Logical Only	Do Not Generate	Definition
addr_book	Address Book Container	true	true	<p>A list of contacts that can be used to define customers. An address book can be used to define customers in a single workspace, and it can be shared among other workspaces. The fields of the Address Book can be customized. Data for the address book can be imported or exported.</p> <p>This address book can be associated with multiple workspaces. These relationships will be shown inside the definition relationship tables.</p>
app_ele	Application Element		false	<p>The identity and name of an element in the application implementation. An element can be one of many types, and can contain other elements. Example Application Elements include: a screen, a form, a dialog box, a text box, the Home Page, an Architecture Component, the Item Status field, the system values of the Item Status field. The Application Element is not the full implementation of the element, but is a key and nametag for the element. In many cases, a well-known algorithm can be used to locate the correct Application Element from the Application Element implementation. In other cases, an association to the Application Element Id must be recorded as part of the implementation.</p>

app_ele_string	Application Element String		false	A string resource that is needed in English and other translations. The string is associated with an Application Element and a Usage, for example the Label, the Tooltip, etc.
app_role	Application Role		false	A User Security Role within Footprints. This includes Agent Users and Customer Users, and includes Role settings at the Instance and Container level. Workspace Role and CMDB Role may both be implemented as Container Role. The Role has associated Rights and Permissions, and is a way of granting Roles and Permissions to a User. There is some conceptual overlap between the User Profile and the Role. The User is assigned a single Role as part of the User Profile, and a User can only have one System Role and one Workspace Role. The User Profile can have property settings that override the Role Settings.
app_user	Application User		false	A user authorized to use the FootPrints system. The user is either a Customer User or an Agent User. The application user holds the minimum information needed to identify the user and associate the user with a user profile, where user settings are stored. A contact is not necessarily an application user. An application user entry is required in order for the user to access any online functions.
app_user_prof_link	Application User Profile Link		false	The link between the application user and the user profile.
appt	Appointment	true	true	A calendar appointment at a specific time for specific parties. This is logical, the actual appointment is stored in the Calendaring tables that support iCal interchange.
audit_trail	Audit Trail	true	true	A listing of all changes made to information stored in the FootPrints system. This includes but is not necessarily limited to database column changes. All tables potentially have changes to them recorded here, so those relationships via a generic foreign key are not shown.
audit_txn_audit_trail	Audit Transaction Audit Trail		false	
cmdb_cntr	CMDB Container	true	true	A Configuration Management Database (CMDB) class of container. The CMDB container implements ITIL CMDB special features beyond the capabilities of the generic Container. The CMDB acts as a repository of Configuration Items and supports population, synchronization, and referencing of items within it.
cfg_item	Configuration Item	true	true	A specialization of Item contained in a CMDB container. A Configuration Item (CI) in the Configuration Management Data Base (CMDB). A Configuration Item is a repository item representing a software, hardware, or service item. There are important relationships to other items both within the CMDB but also externally to the CMDB. The CI has the ability to store fields as defined based on its CI Type. Current Footprints creates a table for the CI type named with the CMDB sequence identifier and the CI Type sequence identifier.

cfg_setting	Configuration Setting		false	<p>A configuration variable that can be identified in programming code and which has an atomic value that affects the behavior of the application. Configuration values are either scalars, or are supported by value lists that also can be programmatically identified. They have localized labels and value choices, and are categorized to support organization and separation in programming namespaces and configuration screens. They support the concept of Built-In Values, Default Values, and Customized Values. Configuration should be supported by an Audit Trail, and should be able to determine if configuration has changed by level, category, and for the individual value in order to support cacheing. A configuration value has a definition to identify its usage, and an enum that identifies if a change will require a restart and what level of restart is required if the value changes. If a restart is required, both the currently running value and the value after restart should be accessible. This means that historical values of the Configuration Settings should be accessible.</p>
addr_book_entry	Contact Item	true	true	<p>A contact that belongs to the FootPrints address book. This could be implemented as a specialization of Item which has the needed features of flexible schema, and data mapping adapters.</p>
cntr	Container	true	true	<p>The highest abstract level of Runtime Object. The Runtime object has a corresponding definition that acts as a catalog of the Runtime Entity. There is a hierarchy or composition concept in the Runtime Entity, but it may not be necessary to actually have navigable properties.</p>
cntr_defn	Container Definition	true	true	<p>The definition of a Container and the associated Configuration Settings at the container level. This would include the collection of Item Definitions. The Container_Definition will have a Container Class identifier (Workspace, CMDB, Address Book) and may have specializations such as Workspace_Container_Definition. As much as possible, the Container Definition will suffice to describe any Container Type. There will be an association to a default Container Template, as well as more application specific Container Templates that can be chosen to initially populate the Container Template. There will also be Container Group Definition, and Container Group Templates. See the database views v_cntr_defn, v_cntr_defn_active.</p>
defn	Definition		false	<p>A definition holds the structural and behavioral information needed for an application element that is configurable by the system designer and/or the end user at definition time. Current Footprints uses a combination of directories, files, tables, and programming code to represent this information. The new FootPrints will provide all of the same information within the definition. The definition can be persisted in a structure that is isolated at the Instance Level. The data for the definition will be stored as Configuration Settings associated with the Definition Version. See the database views v_defn_ver_active, v_cntr_item_field.</p>

defn_reln	Definition Relationship		false	All relationships between definitions are stored in this table in order to make the model able to handle new relationships without changes to tables.
defn_ver	Definition Version		false	A version of a Definition with a version identifier that changes when data changes are made to the template. The definition version will suffice to store the current state of the FootPrints Configuration. A full copy of the definition tree is versioned when a container is published. See the database views v_defn_ver_active, v_cntrn_item_field.
field_defn	Field Definition	true	true	The definition of a Field, including persistence information and default UI information. Some fields will be defined as database fields, and other fields will be defined as calculated fields. The field will have data types and associations to business rule definitions. Some fields are built-in, some are defaults from templates, and some are custom fields. See the database view v_cntrn_item_field.
item	Item	true	true	The Item is the abstract class for Issue, Configuration Item, Address Book entry, and other information items that are the main information entities in FootPrints. The Item structure and behavior is defined by its Definition. The Item is namespaced, identified by, and logically stored in the single parent container. It can be referenced in the same manner both inside and outside the container. The Item is the storage model and the implementation for the Item Definition. There are many important specializations of the Item class, and there even more Template definitions of Container/Item that provide an end-user application experience to the End-User. The Item implementation must implement flexible schema in an easy to manage, well engineered, and high performing way. The item can store user-defined and user-named fields defined in its definition. The Item could implement the nice-to-have features such as an inheritance model for Item subtypes, and have the ability to represent collections of subordinate items. The Item data can be imported, exported, or synchronized with external data sources. The Item data can be linked and synchronized with other items in the same Instance.

item_defn	Item Definition	true	true	The definition of an Item, which includes the Item Name and localizations and the collection of Item Fields. It also includes customizations and localizations of string resources for Item Names, and Item Field names.
item_field_defn	Item Field Definition	true	true	Item Field Definition is an association between the Item Definition entity and the Field Definition entity. This association contains a distinct value for each of the intersection points between the two entities.
item_reln	Item Relationship		false	An Item Relationship is a relationship between instances contained in two Dynamic Object generated Tables. For example, the following. A Configuration Item Type ("Computer") related to an Workspace Item ("Incident"). An Address Book Entry ("Contact") related to another Address Book Entry ("Master Contact"). A Dynamic Link between an Incident in Workspace 1 ("Current Tickets") and Workspace 3 ("Archive Workspace"). A Dynamic Link between an Incident in Workspace 1 ("Current Tickets") and Workspace 5 ("Change Management"). There are no foreign keys defined.
item_reln_defn	Item Relationship Definition	true	true	The definition of a Relationship type within the Container Definition. This has important Configuration Settings and Application Element Identifier values that clarify the meaning of the Relationship to programming code.
locale	Locale		false	A language and locale used to identify a specific dialect. It is used to determine variable forms of words, idioms, numeric format, date format, images, colors, etc. "en-US" for example is United States English, "es-LA" for Latin American Spanish.
schd	Schedule	false	false	A personal or workspace calendar in Footprints. It is used to provide easy access to work hours and settings for appointment scheduling and metric reporting. The Calendar Service maintains the system of record tables for the schedule, but those tables are too difficult and time consuming to query for trend reporting.
string_rsrc_val	String Resource Value		false	The translated value of the resource.
tmpl	Template	true	true	A template for a Definition, consisting of a set of properties, values and Configuration Settings. The Template has a name and other properties, and contains a reference to a Definition.
tenant_parttn	Tenant Partition	true	true	A partition of the FootPrints product that is isolated from other partitions in both data security and design definition and configuration. The Tenant Partition is the global layer above the Container which hosts the Definitions and the implementations of the Containers, and hosts system level information such as the Users and Roles. In current Footprints, this is the root folder and the installed schema for the database tables.
wksp_item	Ticket Item	true	true	A specialization of Item contained in a Workspace container. Workspace Items are tracked for resolution and closure, and have parties that are awaiting resolution and closure, and responsible parties that perform work against them.
user_prof	User Profile		false	A container for Permissions and Settings (the composite parts are the associated Role and the User Preferences). The permissions for a User to access a particular container according to the capabilities defined. In most cases this will be more defined in a subcategory.  Example: Shared User Profile

				This profile will be shared between multiple users that access the system with the same User ID.
wksp_actvy_data_mart	Workspace Activity Data Mart	true	true	A summarized data store of Workspace Activity metric calculations used to support the Executive Dashboard Trend Reporting feature in an upcoming FootPrints release. There is a version of this for both the FootPrints Monet (Perl) and Renoir (Java) releases. The data mart tables conform to a very specific star-schema design and must be populated using history data and an incremental feed. The Star Schema structure must be able to use custom Workspace Item Fields, so the design must be coordinated with the implementation of Workspace Item.
wksp_cntnr	Workspace Container	true	true	A workspace is the principle architectural structure in Numara FootPrints for storing and tracking information. A Workspace is a separate sub-database or work area within the system with its own fields, forms, options, workflow, settings and users. Most of the behaviour of the Workspace comes from the Container Abstract class, and only things that are specific to the Workspace are implemented here. For Example, Current FootPrints allows a single CMDB to be referenced from a Workspace, and a single Address book, yet these are listed as features for improvement. So convenience methods that get and set Container Group Relationship might be appropriate.

#### Attribute/Column(s) of Entity/Table "Address Book Container"

Physical Name	Name	Definition	Physical Data Type
enable_master_cntc_record	Enable Master Contact Record	Indicates if Enable Master Contact Record.	bit
addr_book_cntnr_name	Address Book Container Name	The name of the Address Book Container.	varchar(255)
prmr_y_key	Primary Key	A Foreign Key to the Address Field table.	bigint
orgal_field	Organizational Field	A Foreign Key to the Address Field table.	bigint
cntnr_type_id	Container Type Id	A Foreign Key to the Container Type table.	int
addr_book_type_id	Address Book Type Id	A Foreign Key to the Address Book Type table.	int
cntnr_id	Container Id	A Foreign Key to the Container table.	bigint

#### Attribute/Column(s) of Entity/Table "Application Element"

Physical Name	Name	Definition	Physical Data Type
app_ele_id	Application Element Id	The automatically generated primary key of the app_ele table.	bigint
app_ele_name	Application Element Name	The name of the Application Element.	VARCHAR(2000)
intrnl_id	Internal Id	An alternate automatically generated key of the app_ele table.	int
is_intrnl_flag	Is Internal Flag	A boolean flag for indicating if the Application Element is defined internally by Engineering or by Partners or is defined by the customer.	bit
app_ele_type_id	Application Element Type Id	A Foreign Key to the Application Element Type table.	int
intrnl_id_type_id	Internal Identifier Type Id	A Foreign Key to the Internal Identifier Type table.	int
app_ele_nmosp_id	Application Element Namespace Id	A Foreign Key to the Application Element Namespace table.	bigint

#### Attribute/Column(s) of Entity/Table "Application Element String"

Physical Name	Name	Definition	Physical Data Type
app_ele_string_id	Application Element String Id	The automatically generated primary key of the app_ele_string table.	bigint



app_ele_id	Application Element Id	A Foreign Key to the Application Element table.	bigint
string_usage_type_id	String Usage Type Id	A Foreign Key to the String Usage Type table.	int
part_of_speech_id	Part Of Speech Id	The unique identifier for an instance of Part Of Speech. This identifier must be kept in sync with programming code that uses an enumeration or fixed constant to identify particular values.	int
needs_mplle_plural_forms_flag	Needs Multiple Plural Forms Flag	A boolean flag that indicates that one string per plural form must be specified. An example is "{N} Records Found.	numeric(1)

#### Attribute/Column(s) of Entity/Table "Application Role"

Physical Name	Name	Definition	Physical Data Type
app_role_id	Application Role Id	The automatically generated primary key of the app_role table.	bigint
tmpl_app_role_id	Template Application Role Id	A Foreign Key to the Application Role table. This indicates which Template Role was used to create a built-in role in case some features concerning new properties added to later release of the system will need to be added to existing templates.	bigint
app_role_name	Application Role Name	The name of the Application Role.	varchar(255)
defn_id	Definition Id	A Foreign Key to the Definition table.	bigint
admin_type_id	Administrator Type Id	A Foreign Key to the Administrator Type table.	int
is_built_in	Is Built In	Indicates that this instance of Application Role is a built-in instance that cannot be changed by the end user.	bit
app_role_descr	Application Role Description	A descriptive name of the Application Role. This descriptive name may be localized.	varchar(1024)
user_type_id	User Type Id	A Foreign Key to the User Type table.	int
defn_subtype_id	Definition Subtype Id	A Foreign Key to the Definition Subtype table.	int
tmpl_app_role_id	Template Application Role Id	A Foreign Key to the Application Role table.	bigint
tmpl_level_name	tmpl_level_name	The value of a Built_in_level, stored as a string.	varchar(255)

#### Attribute/Column(s) of Entity/Table "Application User"

Physical Name	Name	Definition	Physical Data Type
pwd	Password	This space intentionally left blank.	varchar(255)
user_name	User Name	The name of the Application User.	nvarchar(4000)
cust_login_type_id	Customer Login Type Id	A Foreign Key to the Customer Login Type table.	int
app_user_id	Application User Id	The automatically generated primary key of the app_user table.	bigint
lic_type_id	License Type Id	A Foreign Key to the License Type table.	bigint
user_login	User Login	The login name used to access the Application. This may not be filled in if external application security is used.	nvarchar(4000)
email_addr	Email Address	An email address for the Application User.	nvarchar(4000)
delete_date	Delete Date	The value of a Application User, stored as a date.	datetime

#### Attribute/Column(s) of Entity/Table "Application User Profile Link"

Physical Name	Name	Definition	Physical Data Type
user_prof_id	User Profile Id	A Foreign Key to the User Profile table.	bigint
app_user_prof_link_id	Application User Profile Link Id	The automatically generated primary key of the app_user_prof_link table.	bigint
app_user_id	Application User Id	A Foreign Key to the Application User table.	bigint

#### Attribute/Column(s) of Entity/Table "Audit Trail"

Physical Name	Name	Definition	Physical Data Type
item_id	Item Id	A Foreign Key to the Item table.	bigint

#### Attribute/Column(s) of Entity/Table "Audit Transaction Audit Trail"

Physical Name	Name	Definition	Physical Data Type
item_id	Item Id	A Foreign Key to the Item table.	bigint

#### Attribute/Column(s) of Entity/Table "CMDB Container"

Physical Name	Name	Definition	Physical Data Type
---------------	------	------------	--------------------

cntr_type_id	Container Type Id	A Foreign Key to the Container Type table.	int
cntr_id	Container Id	A Foreign Key to the Container table.	bigint

#### Attribute/Column(s) of Entity/Table "Configuration Item"

Physical Name	Name	Definition	Physical Data Type
item_id	Item Id	A Foreign Key to the Item table.	bigint

#### Attribute/Column(s) of Entity/Table "Configuration Setting"

Physical Name	Name	Definition	Physical Data Type
cfg_ctgy_id	Configuration Category Id	A Foreign Key to the Configuration Category table.	bigint
cfg_setting_const	Configuration Setting Constant	The constant name of the Configuration Setting Constant, which can be aligned with an enumeration or other compile time constant value for access in Java.	varchar(255)
dflt_setting_label	Default Setting Label	The name of the Configuration Setting.	varchar(255)
dflt_setting_descr	Default Setting Description	The description of the purpose and usage of the configuration setting. This description serves for documentation, and will be used to help develop UI strings for the Label, ToolTip, HelpText etc. of the UI for the configuration setting.	varchar(255)
cfg_setting_id	Configuration Setting Id	The automatically generated primary key of the cfg_setting table.	bigint
agent_only	Agent Only	Indicates if Agent Only.	bit
unique_user_only	Unique User Only	Indicates if Unique User Only.	bit
cfg_val_list_id	Configuration Value List Id	A Foreign Key to the Configuration Value List table.	bigint
val_type_cd	Value Type Code	The data type for a Configuration Setting. The choices are 'S' string, 'D' date, 'T' text, 'L' Long, 'R' real, and 'A' for any type.	char(1)
cfg_list_type_name	Configuration List Type Name	The unique string identifier for an instance of Configuration List Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)
spec_class_name	Specification Class Name	The value of a Configuration Setting, stored as a string.	varchar(255)
cfg_is_val_mapped	Configuration Is Value Mapped	Indicates if Configuration Is Value Mapped.	bit

#### Attribute/Column(s) of Entity/Table "Contact Item"

Physical Name	Name	Definition	Physical Data Type
item_id	Item Id	A Foreign Key to the Item table.	bigint
master_cntc_item_id	Master Contact Item Id	A Foreign Key to the Master Contact Item table.	BIGINT

#### Attribute/Column(s) of Entity/Table "Container"

Physical Name	Name	Definition	Physical Data Type
cntr_name	Container Name	The name of the Container.	varchar(255)
cntr_id	Container Id	The automatically generated primary key of the cntr table.	bigint
tenant_parttn_id	Tenant Partition Id	A Foreign Key to the Tenant Partition table.	bigint
defn_id	Definition Id	A Foreign Key to the Definition table.	bigint

#### Attribute/Column(s) of Entity/Table "Container Definition"

Physical Name	Name	Definition	Physical Data Type
db_schema_name	Database Schema Name	The name of the database schema. It is a database object name, limited to 30 characters in length.	varchar(30)
defn_id	Definition Id	A Foreign Key to the Definition table.	bigint

#### Attribute/Column(s) of Entity/Table "Definition"

Physical Name	Name	Definition	Physical Data Type
defn_id	Definition Id	The automatically generated primary key of the defn table.	bigint
defn_name	Definition Name	The name of the Definition.	nvarchar(4000)
dsble_date	Disabled Date	The date that this instance of Definition is logically deleted and disabled.	datetime

defn_guid	Definition GUID	A Global Unique Identifier search key for the Definition, composed of letters and numbers. It is computed by the client and typically used in unstructured data.	varchar(50)
defn_type_name	Definition Type Name	The unique string identifier for an instance of Definition Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)
defn_subtype_id	Definition Subtype Id	A Foreign Key to the Definition Subtype table.	int
tmpl_export_guid	Template Export GUID	The value of a Definition, stored as a string.	VARCHAR(50)

#### Attribute/Column(s) of Entity/Table "Definition Relationship"

Physical Name	Name	Definition	Physical Data Type
defn_reln_id	Definition Relationship Id	The automatically generated primary key of the defn_reln table.	bigint
from_defn_ver_id	From Definition Version Id	A Foreign Key to the Definition Version table.	bigint
to_defn_ver_id	To Definition Version Id	A Foreign Key to the Definition Version table.	bigint
defn_reln_type_id	Definition Relationship Type Id	A Foreign Key to the Definition Relationship Type table.	int

#### Attribute/Column(s) of Entity/Table "Definition Version"

Physical Name	Name	Definition	Physical Data Type
defn_id	Definition Id	A Foreign Key to the Definition table.	bigint
defn_ver_id	Definition Version Id	The automatically generated primary key of the defn_ver table.	bigint
defn_ver_name	Definition Version Name	The name of the Definition Version.	nvarchar(4000)
defn_ver_status_id	Definition Version Status Id	A Foreign Key to the Definition Version Status table.	int
defn_ver_guid	Definition Version GUID	A Global Unique Identifier search key for the Definition Version, composed of letters and numbers. It is computed by the client and typically used in unstructured data.	varchar(50)

#### Attribute/Column(s) of Entity/Table "Field Definition"

Physical Name	Name	Definition	Physical Data Type
field_dtype_id	Field Datatype Id	A Foreign Key to the Field Datatype table.	bigint
field_col_name	Field Column Name	The name of the DBMS_Object_Name. It is a database object name, limited to 30 characters in length.	varchar(30)
field_descr	Field Description	Field Description.	varchar(255)
field_defn_id	Field Definition Id	A Foreign Key to the Definition table.	bigint

#### Attribute/Column(s) of Entity/Table "Item"

Physical Name	Name	Definition	Physical Data Type
item_key	Item Key	A search key for the Item, composed of letters and numbers.	varchar(255)
item_id	Item Id	The automatically generated primary key of the item table.	bigint
cntnr_id	Container Id	A Foreign Key to the Container table.	bigint
item_defn_id	Item Definition Id	A Foreign Key to the Definition table for an instance of type Definition.	bigint

#### Attribute/Column(s) of Entity/Table "Item Definition"

Physical Name	Name	Definition	Physical Data Type
item_table_name	Item Table Name	The name of the DBMS_Object_Name. It is a database object name, limited to 30 characters in length.	varchar(30)
item_defn_id	Item Definition Id	A Foreign Key to the Definition table for an instance of type Definition.	bigint

#### Attribute/Column(s) of Entity/Table "Item Field Definition"

Physical Name	Name	Definition	Physical Data Type
item_field_defn_id	Item Field Definition Id	A Foreign Key to the Definition table for an instance of type Definition.	bigint

item_defn_id	Item Definition Id	A Foreign Key to the Definition table for an instance of type Definition.	bigint
field_defn_id	Field Definition Id	A Foreign Key to the Definition table.	bigint

#### Attribute/Column(s) of Entity/Table "Item Relationship"

Physical Name	Name	Definition	Physical Data Type
item_reln_id	Item Relationship Id	The automatically generated primary key of the item_reln table.	bigint
item_reln_defn_id	Item Relationship Definition Id	A Foreign Key to the Definition table for an instance of type Definition.	bigint
from_item_defn_id	From Item Definition Id	A Foreign Key to the Definition table for an instance of type Item Definition.	bigint
to_item_defn_id	To Item Definition Id	A Foreign Key to the Definition table for an instance of type Item Definition.	bigint
link_type_defn_ver_id	Link Type Definition Version Id	A Foreign Key to the Definition Version table for an instance of type Link Type Definition Version. The link type will hold the configurable properties for the relationship.	bigint
from_item_id	From Item Id	The From Item Id in the relationship.	bigint
to_item_id	To Item Id	The To Item Id in the relationship.	bigint

#### Attribute/Column(s) of Entity/Table "Item Relationship Definition"

Physical Name	Name	Definition	Physical Data Type
item_reln_defn_id	Item Relationship Definition Id	A Foreign Key to the Definition table for an instance of type Definition.	bigint
from_mbr_role_type_id	From Member Role Type Id		char(18)
to_mbr_role_type_id	To Member Role Type Id		char(18)
reln_type_id	Relationship Type Id	Relationship type	char(18)

#### Attribute/Column(s) of Entity/Table "Locale"

Physical Name	Name	Definition	Physical Data Type
locale_cd	Locale Code	A well-known string identifier used as a code to identify the Locale.	varchar(255)
locale_name	Locale Name	The name of the Locale.	varchar(255)
locale_id	Locale Id	The automatically generated primary key of the locale table.	bigint
is_sys_flag	Is System Flag	Indicates if Is System Flag.	bit

#### Attribute/Column(s) of Entity/Table "Schedule"

Physical Name	Name	Definition	Physical Data Type
schd_id	Schedule Id	The automatically generated primary key of the schd table.	bigint
schd_name	Schedule Name	The name of the Schedule.	varchar(255)
max_date_val	Maximum Date Value	The time spans are forecast out into the future as CalendarActualPeriod. This is the maximum date for which those periods are generated.	datetime
time_zone_region_name	Time Zone Region Name	The name of the Time Zone Region for the calendar, which derives from or overrides the value used for the system or workspace.	varchar(255)
cal_type_id	Calendar Type Id	A Foreign Key to the Calendar Type table.	int
app_user_id	Application User Id	A Foreign Key to the Application User table.	bigint
use_local_time	Use Local Time	Indicates that the workschedule is base on local time for the time zone region. If you want UTC time, use that as the region.	bit

#### Attribute/Column(s) of Entity/Table "String Resource Value"

Physical Name	Name	Definition	Physical Data Type
string_rsrc_val_id	String Resource Value Id	The automatically generated primary key of the string_rsrc_val table.	bigint
string_val	String Value	The default value for the string, which can be overridden by user customization.	nchar(2000)
app_ele_string_id	Application Element String Id	A Foreign Key to the Application Element String table.	bigint
cxt_phrase	Context Phrase	A free-form text field in the source language (usually english) which describes the context and has any additional guidance specific to the associated locale.	varchar(255)

prod_rlse_id	Product Release Id	A Foreign Key to the Product Release table.	bigint
locale_id	Locale Id	A Foreign Key to the Locale table.	bigint

#### Attribute/Column(s) of Entity/Table "Template"

Physical Name	Name	Definition	Physical Data Type
defn_id	Definition Id	A Foreign Key to the Definition table.	bigint

#### Attribute/Column(s) of Entity/Table "Tenant Partition"

Physical Name	Name	Definition	Physical Data Type
sys_env_id	System Environment Id	A Foreign Key to the System Environment table.	bigint
tenant_parttn_id	Tenant Partition Id	The automatically generated primary key of the tenant_parttn table.	bigint

#### Attribute/Column(s) of Entity/Table "Ticket Item"

Physical Name	Name	Definition	Physical Data Type
item_id	Item Id	A Foreign Key to the Item table.	bigint

#### Attribute/Column(s) of Entity/Table "User Profile"

Physical Name	Name	Definition	Physical Data Type
user_prof_id	User Profile Id	The automatically generated primary key of the user_prof table.	bigint
tmpl_user_prof_id	Template User Profile Id	A Foreign Key to the User Profile table. This will not be a navigable key in the object model. This indicates which Template User Profile was used to create a built-in template in case some features concerning new properties added to later release of the system will need to be added to existing templates.	bigint
role_grp_id	Role Group Id	A Foreign Key to the Role Group table. A system level user profile can have a role group, which will have the effect of assigning a collection of workspace, cmdb, and address book roles at the container role level.	bigint
defn_id	Definition Id	A Foreign Key to the Definition table. Optionally, the container that owns the profile.	bigint
app_role_id	Application Role Id	A Foreign Key to the Application Role table. The User Profile is the means for setting the default role at the system and workspace levels, and is also the role of a Unique User Profile tied to a application user.	bigint
agent_authcn_type_id	Agent Authentication Type Id	A Foreign Key to the Agent Authentication Type table.	int
user_prof_name	User Profile Name	The name of the User Profile.	varchar(255)
is_built_in	Is Built In	Indicates that this instance of User Profile is a built-in instance that cannot be changed by the end user.	bit
user_type_id	User Type Id	A Foreign Key to the User Type table.	int
cust_authcn_type_id	Customer Authentication Type Id	A Foreign Key to the Customer Authentication Type table.	int
is_sys	Is System	Indicates if this is a system level profile.	bit
defn_subtype_id	Definition Subtype Id	A Foreign Key to the Definition Subtype table.	int
delete_date	Delete Date	The value of a User Profile, stored as a date.	datetime
last_login	Last Login	date time of last login in UTC.	datetime
tmpl_level_name	tmpl_level_name	The value of a Built_in_level, stored as a string.	varchar(255)

#### Attribute/Column(s) of Entity/Table "Workspace Container"

Physical Name	Name	Definition	Physical Data Type
time_zone_region_name	Time Zone Region Name	The name of the Time Zone Region. This is 2. The 'primary key' of the java timezone is the string region identifier in English, example 'America/Phoenix'. There is no numeric identifier. As the JRE is updated, or specific classes are updated, the list could change to include new entries, and the detail data behind the regions could change.	varchar(255)

cntrnr_type_id	Container Type Id	A Foreign Key to the Container Type table.	int
cntrnr_id	Container Id	A Foreign Key to the Container table.	bigint

### ER Diagram(s) of Subject Area "001 - High Level Domain Model"

Name	Physical Display Level
Entity Level	Column
Table Level	Table

### Entity/Table(s) of Subject Area "001 - High Level Domain Model"

Physical Name	Name	Logical Only	Do Not Generate	Definition
actvy_defn	Activity Definition	true	true	The definition of a task that must take place in order to fulfill an operation contract.
addr_book	Address Book Container	true	true	<p>A list of contacts that can be used to define customers. An address book can be used to define customers in a single workspace, and it can be shared among other workspaces. The fields of the Address Book can be customized. Data for the address book can be imported or exported.</p> <p>This address book can be associated with multiple workspaces. These relationships will be shown inside the definition relationship tables.</p>
agent	Agent	true	true	a user authorized to use the FootPrints agent functions, which enable the agent to provide services to the customer. These include the ability to edit, escalate, and close tickets.
app_ele	Application Element		false	<p>The identity and name of an element in the application implementation. An element can be one of many types, and can contain other elements. Example Application Elements include: a screen, a form, a dialog box, a text box, the Home Page, an Architecture Component, the Item Status field, the system values of the Item Status field. The Application Element is not the full implementation of the element, but is a key and nametag for the element. In many cases, a well-known algorithm can be used to locate the correct Application Element from the Application Element implementation. In other cases, an association to the Application Element Id must be recorded as part of the implementation.</p>
app_ele_string	Application Element String		false	A string resource that is needed in English and other translations. The string is associated with an Application Element and a Usage, for example the Label, the Tooltip, etc.
app_role	Application Role		false	<p>A User Security Role within Footprints. This includes Agent Users and Customer Users, and includes Role settings at the Instance and Container level. Workspace Role and CMDB Role may both be implemented as Container Role. The Role has associated Rights and Permissions, and is a way of granting Roles and Permissions to a User. There is some conceptual overlap between the User Profile and the Role. The User is assigned a single Role as part of the User Profile, and a User can only have one System Role and one Workspace Role. The User Profile can have property settings that override the Role Settings.</p>

app_user	Application User		false	A user authorized to use the FootPrints system. The user is either a Customer User or an Agent User. The application user holds the minimum information needed to identify the user and associate the user with a user profile, where user settings are stored. A contact is not necessarily an application user. An application user entry is required in order for the user to access any online functions.
appt	Appointment	true	true	A calendar appointment at a specific time for specific parties. This is logical, the actual appointment is stored in the Calendaring tables that support iCal interchange.
asset_item	Asset Item	true	true	
attch	Attachment	true	true	A binary or text file attachment that can be related to various entities that accept attachments, such as Issue or Emails
attch_inbd_email	Attachment Inbound Email		false	
attch_outbd_email	Attachment Outbound Email		false	
audit_trail	Audit Trail	true	true	A listing of all changes made to information stored in the FootPrints system. This includes but is not necessarily limited to database column changes. All tables potentially have changes to them recorded here, so those relationships via a generic foreign key are not shown.
audit_txn	Audit Transaction	true	true	A change made to information stored in the FootPrints system. This includes but is not necessarily limited to database column changes. All table potentially have changes to them recorded
audit_txn_audit_trail	Audit Transaction Audit Trail		false	
bus_rule_defn	Business Rule Definition	true	true	A Business rule is a statement that defines or constrains some aspect of the business. It is intended to assert business structure or to control or influence the behavior of the business.
cmdb_cntr	CMDB Container	true	true	A Configuration Management Database (CMDB) class of container. The CMDB container implements ITIL CMDB special features beyond the capabilities of the generic Container. The CMDB acts as a repository of Configuration Items and supports population, synchronization, and referencing of items within it.
cfg_item	Configuration Item	true	true	A specialization of Item contained in a CMDB container. A Configuration Item (CI) in the Configuration Management Data Base (CMDB). A Configuration Item is a repository item representing a software, hardware, or service item. There are important relationships to other items both within the CMDB but also externally to the CMDB. The CI has the ability to store fields as defined based on its CI Type. Current Footprints creates a table for the CI type named with the CMDB sequence identifier and the CI Type sequence identifier.
cfg_setting	Configuration Setting		false	A configuration variable that can be identified in programming code and which has an atomic value that affects the behavior of the application. Configuration values are either scalars, or are supported by value lists that also can be programmatically identified. They have localized labels and value choices, and are categorized to support organization and separation in programming namespaces and configuration screens. They support the concept of Built-In Values, Default Values, and Customized Values. Configuration should be supported by an Audit Trail, and should be able to determine if configuration has changed by level, category, and for the individual value in order to support cacheing. A configuration value has a definition to identify its usage, and a enum that identifies if a

				change will require a restart and what level of restart is required if the value changes. If a restart is required, both the currently running value and the value after restart should be accessible. This means that historical values of the Configuration Settings should be accessible.
cfg_setting_val	Configuration Setting Value		false	Configuration Setting Value is an association between the Configuration Setting entity and a container entity (i.e. User Profile, Role, and Definition). This association contains a distinct value for each of the intersection points between the two entities.
addr_book_entry	Contact Item	true	true	A contact that belongs to the FootPrints address book. This could be implemented as a specialization of Item which has the needed features of flexible schema, and data mapping adapters.
cntr	Container	true	true	The highest abstract level of Runtime Object. The Runtime object has a corresponding definition that acts as a catalog of the Runtime Entity. There is a hierarchy or composition concept in the Runtime Entity, but it may not be necessary to actually have navigable properties.
cntr_defn	Container Definition	true	true	The definition of a Container and the associated Configuration Settings at the container level. This would include the collection of Item Definitions. The Container_Definition will have a Container Class identifier (Workspace, CMDB, Address Book) and may have specializations such as Workspace_Container_Definition. As much as possible, the Container Definition will suffice to describe any Container Type. There will be an association to a default Container Template, as well as more application specific Container Templates that can be chosen to initially populate the Container Template. There will also be Container Group Definition, and Container Group Templates. See the database views v_cntr_defn, v_cntr_defn_active.
cntr_grp_defn	Container Group Definition	true	true	The definition of a Group Container and the associated Configuration Settings at the container level. This would include the reason for the grouping and the types of containers in the group.
cntr_grp_reln	Container Group Relationship	true	true	The container group may exist in the runtime environment. It may suffice to have the Definition of the container group.
cntr_home_page	Container Home Page	true	true	The homepage is the URL or local file that automatically loads when a web browser starts or when the browser's "home" button is pressed.
cust	Customer	true	true	a user authorized to use the FootPrints customer functions. This includes the ability to enter requests, and interact with requests as they are worked by Agents.
db_obj	Database Object	true	true	Database objects are the components that save and query information. A database contains several objects that are used to provide storage and display large, relational data. The main objects of a database are tables, stored procedures, views, users and functions.
db_obj_defn	Database Object Definition	true	true	The database definition object that links to configuration settings and properties to determine the functionality of a database item. Database objects are the components that save and query information. A database contains several objects that are used to provide storage and display large, relational data. The main objects of a database are tables, stored procedures, views, users and functions.



dfit_rule	Default Rule	true	true	A default rule is a statement that defines or constrains some aspect of the business. It is intended to assert business structure or to control or influence the behavior of the business.
defn	Definition		false	A definition holds the structural and behavioral information needed for an application element that is configurable by the system designer and/or the end user at definition time. Current Footprints uses a combination of directories, files, tables, and programming code to represent this information. The new FootPrints will provide all of the same information within the definition. The definition can be persisted in a structure that is isolated at the Instance Level. The data for the definition will be stored as Configuration Settings associated with the Definition Version. See the database views v_defn_ver_active, v_cntrn_item_field.
defn_subtype	Definition Subtype		false	The subtype of the definition used as a discriminator for the class of definition. Each subtype will have different configuration settings and associated definitions. These include associated built-in field definitions for item subtypes. See the database view v_defn_subtype.
defn_type	Definition Type		false	The scope of the Template chosen from Instance, Container Group, Container, Item, Field.
defn_ver	Definition Version		false	A version of a Definition with a version identifier that changes when data changes are made to the template. The definition version will suffice to store the current state of the FootPrints Configuration. A full copy of the definition tree is versioned when a container is published. See the database views v_defn_ver_active, v_cntrn_item_field.
event	Event	true	true	An event is an action that is usually initiated outside the FootPrints application and that is handled by a piece of code inside the program.
event_wkflw_proc_defn	Event Workflow Process Definition		false	
extl_data_source_defn	External Data Source Definition	true	true	The abstraction of the structural and behavioral source information that is configurable by the system designer and/or the end user at definition time that specifies the file or other data source used for input or archiving.
extl_data_source_defn_db_obj_d	External Data Source Definition Database Object Definition		false	
field_dtype	Field Datatype	true	true	The type of field (i.e. a custom field that the users have defined).
field_defn	Field Definition	true	true	The definition of a Field, including persistence information and default UI information. Some fields will be defined as database fields, and other fields will be defined as calculated fields. The field will have data types and associations to business rule definitions. Some fields are built-in, some are defaults from templates, and some are custom fields. See the database view v_cntrn_item_field.
field_dpncy_rule	Field Dependency Rule	true	true	A field dependency rule constrains some aspect of the associated field. It is intended to assert business structure or to control or influence the behavior of the business.
inbd_email	Inbound Email	true	true	A record of an email received by Footprints via a configured account. Generally, the email is deleted from the email system after successful processing.
item	Item	true	true	The Item is the abstract class for Issue, Configuration Item, Address Book entry, and other information items that are the main information entities in FootPrints. The Item structure and behavior is defined by its Definition. The Item is

				namespaced, identified by, and logically stored in the single parent container. It can be referenced in the same manner both inside and outside the container. The Item is the storage model and the implementation for the Item Definition. There are many important specializations of the Item class, and there even more Template definitions of Container/Item that provide an end-user application experience to the End-User. The Item implementation must implement flexible schema in an easy to manage, well engineered, and high performing way. The item can store user-defined and user-named fields defined in its definition. The Item could implement the nice-to-have features such as an inheritance model for Item subtypes, and have the ability to represent collections of subordinate items. The Item data can be imported, exported, or synchronized with external data sources. The Item data can be linked and synchronized with other items in the same Instance.
item_defn	Item Definition	true	true	The definition of an Item, which includes the Item Name and localizations and the collection of Item Fields. It also includes customizations and localizations of string resources for Item Names, and Item Field names.
item_field	Item Field	true	true	A place related to a specific item that stores or displays data.
item_field_defn	Item Field Definition	true	true	Item Field Definition is an association between the Item Definition entity and the Field Definition entity. This association contains a distinct value for each of the intersection points between the two entities.
item_reln	Item Relationship		false	An Item Relationship is a relationship between instances contained in two Dynamic Object generated Tables. For example, the following. A Configuration Item Type ("Computer") related to an Workspace Item ("Incident"). An Address Book Entry ("Contact") related to another Address Book Entry ("Master Contact"). A Dynamic Link between an Incident in Workspace 1 ("Current Tickets") and Workspace 3 ("Archive Workspace"). A Dynamic Link between an Incident in Workspace 1 ("Current Tickets") and Workspace 5 ("Change Management"). There are no foreign keys defined.
item_vote	Item Vote	true	true	The record of a vote on a specific item. The current effective vote is maintained here, history of conflicting changes is in the Audit Trail.
knowledge_base_cntnr	Knowledge Base Container	true	true	A container for knowledge base information.
lic_type	License Type		false	A lookup table for the type of FootPrints license. Must be secure from tampering.
locale	Locale		false	A language and locale used to identify a specific dialect. It is used to determine variable forms of words, idioms, numeric format, date format, images, colors, etc. "en-US" for example is United States English, "es-LA" for Latin American Spanish.
local_col	Localized Column		false	(Not used, but referenced in Java code that must be removed before table deletion.) (Not currently used. Data localization is implemented in Configuration Setting Value. May be retired in the future). A Database Column name used in the Transparent Database Localization pattern. The data in the column is the original data for which a translated value can exist. In the case of English, this is a little strange because the English translation can be different than the original data.

local_col_val	Localized Column Value		false	(Not used, but referenced in Java code that must be removed before table deletion.) (Not currently used. Data localization is implemented in Configuration Setting Value. May be retired in the future). A data value from a column and row in a Localized table. The initial english value is obtained from the table, but may be modified for visual display purposes. The Primary Key identifier of the source table is used along with the table name to establish the row identity.
addr_book_entry	Master Contact Item	true	true	A contact that belongs to the FootPrints address book. This could be implemented as a specialization of Item which has the needed features of flexible schema, and data mapping adapters.
metric_defn	Metric Definition	true	true	The definition of the level of service in the service level agreement.
metric_threshold	Metric Threshold	true	true	The agreed upon levels of service.
notifn	Notification	true	true	An automated email sent as a result of system activity and notification settings.
op	Operation	true	true	Pertaining to a process or series of actions for achieving a result.
op_visibility	Operation Visibility	true	true	Whether or not a user can see the operation.
outbd_email	Outbound Email	true	true	A record of an email sent by Footprints via a configured account.
prsnnt_mode	Presentation Mode	true	true	A code representing the logical mode of UI presentation. The code is used to categorize UI Element size and capabilities. Examples include: "Full View", "Quick View", "Mobile View".
prod_ver	Product Version	true	true	Information on the exact software version of the Numara products installed on the system.
rqst_for_chg_item	Request For Change Item	true	true	An item that allows input or display of information regarding a request for change.
schd	Schedule	false	false	A personal or workspace calendar in Footprints. It is used to provide easy access to work hours and settings for appointment scheduling and metric reporting. The Calendar Service maintains the system of record tables for the schedule, but those tables are too difficult and time consuming to query for trend reporting.
secuty_permnsn	Security Permission	true	true	Authorization granted to access a system resource.
secuty_right	Security Right	true	true	Rights granted to access a system resource.
svc_rqst	Service Item	true	true	A Service Item in a Service Portfolio container. An item representing a provided or available software service or functionality.
svc_level_agrmt	Service Level Agreement	true	true	A service level agreement (frequently abbreviated as SLA) is a part of a service contract where the level of service is formally defined.
prob_item	Service Level Target Item	true	true	A Service Level Target Item in a Service Portfolio container.
svc_ctlg_cntr	Service Portfolio Container	true	true	A catalog of services that an IT user can use on service requests. This is a type of Configuration Item.
soln_item	Solution Item	true	true	An item that stores the solution information for a particular issue.
string_rsrc_val	String Resource Value		false	The translated value of the resource.
survey_item	Survey Item	true	true	An item that allows input or display of survey information.
sys_env	System Environment	true	true	The System Environment layer represents information about the hardware and software environment which hosts one or more Tenant Partitions. It would be used in Use Cases such as installation, moving Tenant Partitions, finding and connecting to an Tenant Partition, and performance monitoring and Tenant maintenance.

sys_log	System Log	true	true	Information Logged by various system components for the purpose of debugging. Generally, this information is not stored in the database because the database may not be available during error conditions.
sys_mfgr	System Manufacturer	true	true	The System Manufacturer is the term given to the Independent Software Vendor for the FootPrints software. There is one instance of this entity. We need a term that does not include our company name or our product name so that renaming is not necessary in the event of changes. Numara Software is always the System Manufacturer, so we don't need very information much at the Instance level other than localizations. The System Manufacturer it is relevant to interfaces with the Manufacturer's other systems.
sys_prvdr	System Provider	true	true	The System Provider is a business entity that provides the hosted FootPrints application to Tenants in a multi-tenant sense. The System Provider is in the business of providing FootPrints application setup and configuration services so that the Tenant can access a live working system.
tmpl	Template	true	true	A template for a Definition, consisting of a set of properties, values and Configuration Settings. The Template has a name and other properties, and contains a reference to a Definition.
tenant	Tenant	true	true	The Tenant is an owner of one or more FootPrints Tenant Partitions that are configured and maintained by a System Provider. A customer that purchase FootPrints is a tenant in the sense of multi-tenant architecture, even if the Tenant does not use Numara SaaS hosting and installs and maintains the system themselves.
tenant_parttn	Tenant Partition	true	true	A partition of the FootPrints product that is isolated from other partitions in both data security and design definition and configuration. The Tenant Partition is the global layer above the Container which hosts the Definitions and the implementations of the Containers, and hosts system level information such as the Users and Roles. In current Footprints, this is the root folder and the installed schema for the database tables.
tenant_parttn_defn	Tenant Partition Definition	true	true	The Root Node definition object that links to configuration settings and properties and the instance level, and also has the collection of Container Definitions below it.
wksp_item	Ticket Item	true	true	A specialization of Item contained in a Workspace container. Workspace Items are tracked for resolution and closure, and have parties that are awaiting resolution and closure, and responsible parties that perform work against them.
ui_ele	UI Element	true	true	The name and identity of an Application Element that appears in the User Interface.
ui_ele_defn	UI Element Definition	true	true	The UI element definition object that links to configuration settings and properties and the instance level to determine the functionality and display of a UI item.
ui_field	UI Field	true	true	The name and identity of an application field element that appears in the User Interface.
ui_field_defn	UI Field Definition	true	true	The field UI element definition object that links to configuration settings and properties and the instance level to determine it's functionality and display.
ui_form	UI Form	true	true	The name and identity of an application form element that appears in the User Interface.
ui_form_defn	UI Form Definition	true	true	The form UI element definition object that links to configuration settings and properties and the instance level to determine it's functionality and display.

ui_tab	UI Tab	true	true	The name and identity of an application tab element that appears in the User Interface.
ui_tab_defn	UI Tab Definition	true	true	The tab UI element definition object that links to configuration settings and properties and the instance level to determine its functionality and display.
user_prefc_defn	User Preference Definition	true	true	The abstraction of the structural and behavioral preference information that is configurable by the system designer and/or the end user at definition time.
user_prof	User Profile		false	A container for Permissions and Settings (the composite parts are the associated Role and the User Preferences). The permissions for a User to access a particular container according to the capabilities defined. In most cases this will be more defined in a subcategory.  Example: Shared User Profile This profile will be shared between multiple users that access the system with the same User ID.
valid_vals_rule	Valid Values Rule	true	true	A valid values rule constrains some aspect of the Footprints application to a set or range of values. It is intended to assert business structure or to control or influence the behavior of the business.
valdn_rule	Validation Rule	true	true	A validation rule constrains some aspect of the associated field. It is intended to assert business structure or to control or influence the behavior of the business.
wkflw_actn	Workflow Action	true	true	A workflow action that Footprints can take in response to Change Management activity or escalations or other system features that trigger actions. The change to a phase action for Approval, Disapproval, or Expiration can trigger any of these actions.
wkflw_proc_defn	Workflow Process Definition	true	true	The management of steps in a business processes. A workflow specifies what tasks need to be done, in what order (sometimes linearly, sometimes in parallel), and who has permission to perform each task. Most tasks are performed by humans but they can also be automated processes.
wkflw_proc_defn_wkflw_actn	Workflow Process Definition Workflow Action		false	
wksp_actvy_data_mart	Workspace Activity Data Mart	true	true	A summarized data store of Workspace Activity metric calculations used to support the Executive Dashboard Trend Reporting feature in an upcoming FootPrints release. There is a version of this for both the FootPrints Monet (Perl) and Renoir (Java) releases. The data mart tables conform to a very specific star-schema design and must be populated using history data and an incremental feed. The Star Schema structure must be able to use custom Workspace Item Fields, so the design must be coordinated with the implementation of Workspace Item.
wksp_cntnr	Workspace Container	true	true	A workspace is the principle architectural structure in Numara FootPrints for storing and tracking information. A Workspace is a separate sub-database or work area within the system with its own fields, forms, options, workflow, settings and users. Most of the behaviour of the Workspace comes from the Container Abstract class, and only things that are specific to the Workspace are implemented here. For Example, Current FootPrints allows a single CMDB to be referenced from a Workspace, and a single Address book, yet these are listed as features for improvement. So convenience methods that get and set Container Group Relationship might be appropriate.

prob_item	WorkTarget Item	true	true	A Work Target Item in a Service Portfolio container.
-----------	-----------------	------	------	--

#### Attribute/Column(s) of Entity/Table "Activity Definition"

Physical Name	Name	Definition	Physical Data Type
defn_id	Definition Id	A Foreign Key to the Definition table.	bigint

#### Attribute/Column(s) of Entity/Table "Address Book Container"

Physical Name	Name	Definition	Physical Data Type
enable_master_cntc_record	Enable Master Contact Record	Indicates if Enable Master Contact Record.	bit
addr_book_cntrn_name	Address Book Container Name	The name of the Address Book Container.	varchar(255)
prmr_y_key	Primary Key	A Foreign Key to the Address Field table.	bigint
orgal_field	Organizational Field	A Foreign Key to the Address Field table.	bigint
cntrn_type_id	Container Type Id	A Foreign Key to the Container Type table.	int
addr_book_type_id	Address Book Type Id	A Foreign Key to the Address Book Type table.	int
cntrn_id	Container Id	A Foreign Key to the Container table.	bigint

#### Attribute/Column(s) of Entity/Table "Agent"

Physical Name	Name	Definition	Physical Data Type
lic_type_id	License Type Id	A Foreign Key to the License Type table.	bigint
app_user_id	Application User Id	A Foreign Key to the Application User table.	bigint

#### Attribute/Column(s) of Entity/Table "Application Element"

Physical Name	Name	Definition	Physical Data Type
app_ele_id	Application Element Id	The automatically generated primary key of the app_ele table.	bigint
app_ele_name	Application Element Name	The name of the Application Element.	VARCHAR(2000)
intrnl_id	Internal Id	An alternate automatically generated key of the app_ele table.	int
is_intrnl_flag	Is Internal Flag	A boolean flag for indicating if the Application Element is defined internally by Engineering or by Partners or is defined by the customer.	bit
app_ele_type_id	Application Element Type Id	A Foreign Key to the Application Element Type table.	int
intrnl_id_type_id	Internal Identifier Type Id	A Foreign Key to the Internal Identifier Type table.	int
app_ele_nmosp_id	Application Element Namespace Id	A Foreign Key to the Application Element Namespace table.	bigint

#### Attribute/Column(s) of Entity/Table "Application Element String"

Physical Name	Name	Definition	Physical Data Type
app_ele_string_id	Application Element String Id	The automatically generated primary key of the app_ele_string table.	bigint
app_ele_id	Application Element Id	A Foreign Key to the Application Element table.	bigint
string_usage_type_id	String Usage Type Id	A Foreign Key to the String Usage Type table.	int
part_of_speech_id	Part Of Speech Id	The unique identifier for an instance of Part Of Speech. This identifier must be kept in sync with programming code that uses an enumeration or fixed constant to identify particular values.	int
needs_mplpe_plural_forms_flag	Needs Multiple Plural Forms Flag	A boolean flag that indicates that one string per plural form must be specified. An example is "{N} Records Found.	numeric(1)

#### Attribute/Column(s) of Entity/Table "Application Role"

Physical Name	Name	Definition	Physical Data Type
app_role_id	Application Role Id	The automatically generated primary key of the app_role table.	bigint
tmpl_app_role_id	Template Application Role Id	A Foreign Key to the Application Role table. This indicates which Template Role was used to create a built-in role in case some features concerning new properties added to later release of the system will need to be added to	bigint

		existing templates.	
app_role_name	Application Role Name	The name of the Application Role.	varchar(255)
defn_id	Definition Id	A Foreign Key to the Definition table.	bigint
admin_type_id	Administrator Type Id	A Foreign Key to the Administrator Type table.	int
is_built_in	Is Built In	Indicates that this instance of Application Role is a built-in instance that cannot be changed by the end user.	bit
app_role_descr	Application Role Description	A descriptive name of the Application Role. This descriptive name may be localized.	varchar(1024)
user_type_id	User Type Id	A Foreign Key to the User Type table.	int
defn_subtype_id	Definition Subtype Id	A Foreign Key to the Definition Subtype table.	int
tmpl_app_role_id	Template Application Role Id	A Foreign Key to the Application Role table.	bigint
tmpl_level_name	tmpl_level_name	The value of a Built_in_level, stored as a string.	varchar(255)

#### Attribute/Column(s) of Entity/Table "Application User"

Physical Name	Name	Definition	Physical Data Type
pwd	Password	This space intentionally left blank.	varchar(255)
user_name	User Name	The name of the Application User.	nvarchar(4000)
cust_login_type_id	Customer Login Type Id	A Foreign Key to the Customer Login Type table.	int
app_user_id	Application User Id	The automatically generated primary key of the app_user table.	bigint
lic_type_id	License Type Id	A Foreign Key to the License Type table.	bigint
user_login	User Login	The login name used to access the Application. This may not be filled in if external application security is used.	nvarchar(4000)
email_addr	Email Address	An email address for the Application User.	nvarchar(4000)
delete_date	Delete Date	The value of a Application User, stored as a date.	datetime

#### Attribute/Column(s) of Entity/Table "Asset Item"

Physical Name	Name	Definition	Physical Data Type
item_id	Item Id	A Foreign Key to the Item table.	bigint

#### Attribute/Column(s) of Entity/Table "Audit Trail"

Physical Name	Name	Definition	Physical Data Type
item_id	Item Id	A Foreign Key to the Item table.	bigint

#### Attribute/Column(s) of Entity/Table "Audit Transaction Audit Trail"

Physical Name	Name	Definition	Physical Data Type
item_id	Item Id	A Foreign Key to the Item table.	bigint

#### Attribute/Column(s) of Entity/Table "Business Rule Definition"

Physical Name	Name	Definition	Physical Data Type
defn_id	Definition Id	A Foreign Key to the Definition table.	bigint

#### Attribute/Column(s) of Entity/Table "CMDB Container"

Physical Name	Name	Definition	Physical Data Type
cntnr_type_id	Container Type Id	A Foreign Key to the Container Type table.	int
cntnr_id	Container Id	A Foreign Key to the Container table.	bigint

#### Attribute/Column(s) of Entity/Table "Configuration Item"

Physical Name	Name	Definition	Physical Data Type
item_id	Item Id	A Foreign Key to the Item table.	bigint

#### Attribute/Column(s) of Entity/Table "Configuration Setting"

Physical Name	Name	Definition	Physical Data Type
cfg_ctgy_id	Configuration Category Id	A Foreign Key to the Configuration Category table.	bigint

cfg_setting_const	Configuration Setting Constant	The constant name of the Configuration Setting Constant, which can be aligned with an enumeration or other compile time constant value for access in Java.	varchar(255)
dflt_setting_label	Default Setting Label	The name of the Configuration Setting.	varchar(255)
dflt_setting_descr	Default Setting Description	The description of the purpose and usage of the configuration setting. This description serves for documentation, and will be used to help develop UI strings for the Label, ToolTip, HelpText etc. of the UI for the configuration setting.	varchar(255)
cfg_setting_id	Configuration Setting Id	The automatically generated primary key of the cfg_setting table.	bigint
agent_only	Agent Only	Indicates if Agent Only.	bit
unique_user_only	Unique User Only	Indicates if Unique User Only.	bit
cfg_val_list_id	Configuration Value List Id	A Foreign Key to the Configuration Value List table.	bigint
val_type_cd	Value Type Code	The data type for a Configuration Setting. The choices are 'S' string, 'D' date, 'T' text, 'L' Long, 'R' real, and 'A' for any type.	char(1)
cfg_list_type_name	Configuration List Type Name	The unique string identifier for an instance of Configuration List Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)
spec_class_name	Specification Class Name	The value of a Configuration Setting, stored as a string.	varchar(255)
cfg_is_val_mapped	Configuration Is Value Mapped	Indicates if Configuration Is Value Mapped.	bit

#### Attribute/Column(s) of Entity/Table "Configuration Setting Value"

Physical Name	Name	Definition	Physical Data Type
cfg_setting_val_id	Configuration Setting Value Id	The automatically generated primary key of the cfg_setting_val table.	bigint
cfg_setting_id	Configuration Setting Id	A Foreign Key to the Configuration Setting table.	bigint
val_type_cd	Value Type Code	The data type for a Configuration Setting Value. The choices are 'S' string, 'D' date, 'T' text, 'I' integer.	char(1)
string_val	String Value	The value of a Configuration Setting Value, stored as a string.	nvarchar(4000)
text_val	Text Value	The value of a Configuration Setting Value, stored as a long text object.	nvarchar(max)
real_val	Real Value	The value of a Configuration Setting Value stored as a real.	numeric(38,20)
long_val	Long Value	The value of a Configuration Setting Value stored as Long.	bigint
date_val	Date Value	The value of a Configuration Setting Value, stored as a date.	datetime
boolean_val	Boolean Value	The value of a Configuration Setting Value stored as Boolean.	bit
int_val	Int Value	The value of a Configuration Setting Value stored as Int.	int
owner_id	Owner Id	A Foreign Key to the Owning instance in either the Definition, Definition Version, Role, or User Profile tables.	bigint
val_map_key	Value Map Key	The value of a Configuration Setting Value, stored as a string.	varchar(255)
cfg_setting_owner_type_name	Configuration Setting Owner Type Name	The unique string identifier for an instance of Configuration Setting Owner Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)

#### Attribute/Column(s) of Entity/Table "Contact Item"

Physical Name	Name	Definition	Physical Data Type
item_id	Item Id	A Foreign Key to the Item table.	bigint
master_cntc_item_id	Master Contact Item Id	A Foreign Key to the Master Contact Item table.	BIGINT

#### Attribute/Column(s) of Entity/Table "Container"

Physical Name	Name	Definition	Physical Data Type
cntrn_name	Container Name	The name of the Container.	varchar(255)



cntnr_id	Container Id	The automatically generated primary key of the cntnr table.	bigint
tenant_parttn_id	Tenant Partition Id	A Foreign Key to the Tenant Partition table.	bigint
defn_id	Definition Id	A Foreign Key to the Definition table.	bigint

#### Attribute/Column(s) of Entity/Table "Container Definition"

Physical Name	Name	Definition	Physical Data Type
db_schema_name	Database Schema Name	The name of the database schema. It is a database object name, limited to 30 characters in length.	varchar(30)
defn_id	Definition Id	A Foreign Key to the Definition table.	bigint

#### Attribute/Column(s) of Entity/Table "Container Group Definition"

Physical Name	Name	Definition	Physical Data Type
defn_id	Definition Id	A Foreign Key to the Definition table.	bigint

#### Attribute/Column(s) of Entity/Table "Container Group Relationship"

Physical Name	Name	Definition	Physical Data Type
cntnr_grp_reln_id	Container Group Relationship Id	The automatically generated primary key of the cntnr_grp_reln table.	bigint

#### Attribute/Column(s) of Entity/Table "Container Home Page"

Physical Name	Name	Definition	Physical Data Type
app_ele_id	Application Element Id	A Foreign Key to the Application Element table.	bigint

#### Attribute/Column(s) of Entity/Table "Customer"

Physical Name	Name	Definition	Physical Data Type
cust_login_type_id	Customer Login Type Id	A Foreign Key to the Customer Login Type table.	int
app_user_id	Application User Id	A Foreign Key to the Application User table.	bigint

#### Attribute/Column(s) of Entity/Table "Database Object Definition"

Physical Name	Name	Definition	Physical Data Type
defn_id	Definition Id	A Foreign Key to the Definition table.	bigint

#### Attribute/Column(s) of Entity/Table "Default Rule"

Physical Name	Name	Definition	Physical Data Type
defn_id	Definition Id	A Foreign Key to the Definition table.	bigint

#### Attribute/Column(s) of Entity/Table "Definition"

Physical Name	Name	Definition	Physical Data Type
defn_id	Definition Id	The automatically generated primary key of the defn table.	bigint
defn_name	Definition Name	The name of the Definition.	nvarchar(4000)
dsble_date	Disabled Date	The date that this instance of Definition is logically deleted and disabled.	datetime
defn_guid	Definition GUID	A Global Unique Identifier search key for the Definition, composed of letters and numbers. It is computed by the client and typically used in unstructured data.	varchar(50)
defn_type_name	Definition Type Name	The unique string identifier for an instance of Definition Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)
defn_subtype_id	Definition Subtype Id	A Foreign Key to the Definition Subtype table.	int
tmp_export_guid	Template Export GUID	The value of a Definition, stored as a string.	VARCHAR(50)

#### Attribute/Column(s) of Entity/Table "Definition Subtype"

Physical Name	Name	Definition	Physical Data Type
defn_subtype_name	Definition Subtype Name	The name of the Definition Subtype.	varchar(255)
defn_subtype_descr	Definition Subtype Description	A descriptive name of the Definition Subtype. This descriptive name may be localized.	varchar(512)

defn_subtype_id	Definition Subtype Id	The automatically generated primary key of the defn_subtype table.	int
defn_parent_subtype_id	definition parent subtype id	A Foreign Key to the Definition Subtype table.	int
defn_type_name	Definition Type Name	The unique string identifier for an instance of Definition Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)
max_in_parent	Maximum In Parent	A Foreign Key to the Definition Subtype table.	int

#### Attribute/Column(s) of Entity/Table "Definition Type"

Physical Name	Name	Definition	Physical Data Type
defn_type_name	Definition Type Name	The unique string identifier for an instance of Definition Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)
defn_type_descr	Definition Type Description	A descriptive name of the Definition Type. This descriptive name may be localized.	varchar(512)

#### Attribute/Column(s) of Entity/Table "Definition Version"

Physical Name	Name	Definition	Physical Data Type
defn_id	Definition Id	A Foreign Key to the Definition table.	bigint
defn_ver_id	Definition Version Id	The automatically generated primary key of the defn_ver table.	bigint
defn_ver_name	Definition Version Name	The name of the Definition Version.	nvarchar(4000)
defn_ver_status_id	Definition Version Status Id	A Foreign Key to the Definition Version Status table.	int
defn_ver_guid	Definition Version GUID	A Global Unique Identifier search key for the Definition Version, composed of letters and numbers. It is computed by the client and typically used in unstructured data.	varchar(50)

#### Attribute/Column(s) of Entity/Table "Event Workflow Process Definition"

Physical Name	Name	Definition	Physical Data Type
defn_id	Definition Id	A Foreign Key to the Definition table.	bigint

#### Attribute/Column(s) of Entity/Table "External Data Source Definition Database Object Definition"

Physical Name	Name	Definition	Physical Data Type
defn_id	Definition Id	A Foreign Key to the Definition table.	bigint

#### Attribute/Column(s) of Entity/Table "Field Datatype"

Physical Name	Name	Definition	Physical Data Type
field_dtype_id	Field Datatype Id	The automatically generated primary key of the field_dtype table.	bigint
field_dtype_name	Field Datatype Name	The name of the Field Datatype.	varchar(255)

#### Attribute/Column(s) of Entity/Table "Field Definition"

Physical Name	Name	Definition	Physical Data Type
field_dtype_id	Field Datatype Id	A Foreign Key to the Field Datatype table.	bigint
field_col_name	Field Column Name	The name of the DBMS_Object_Name. It is a database object name, limited to 30 characters in length.	varchar(30)
field_descr	Field Description	Field Description.	varchar(255)
field_defn_id	Field Definition Id	A Foreign Key to the Definition table.	bigint

#### Attribute/Column(s) of Entity/Table "Field Dependency Rule"

Physical Name	Name	Definition	Physical Data Type
defn_id	Definition Id	A Foreign Key to the Definition table.	bigint

#### Attribute/Column(s) of Entity/Table "Item"

Physical Name	Name	Definition	Physical Data Type
item_key	Item Key	A search key for the Item, composed of letters and numbers.	varchar(255)
item_id	Item Id	The automatically generated primary key of the item table.	bigint

cntr_id	Container Id	A Foreign Key to the Container table.	bigint
item_defn_id	Item Definition Id	A Foreign Key to the Definition table for an instance of type Definition.	bigint

#### Attribute/Column(s) of Entity/Table "Item Definition"

Physical Name	Name	Definition	Physical Data Type
item_table_name	Item Table Name	The name of the DBMS_Object_Name. It is a database object name, limited to 30 characters in length.	varchar(30)
item_defn_id	Item Definition Id	A Foreign Key to the Definition table for an instance of type Definition.	bigint

#### Attribute/Column(s) of Entity/Table "Item Field"

Physical Name	Name	Definition	Physical Data Type
item_id	Item Id	A Foreign Key to the Item table.	bigint

#### Attribute/Column(s) of Entity/Table "Item Field Definition"

Physical Name	Name	Definition	Physical Data Type
item_field_defn_id	Item Field Definition Id	A Foreign Key to the Definition table for an instance of type Definition.	bigint
item_defn_id	Item Definition Id	A Foreign Key to the Definition table for an instance of type Definition.	bigint
field_defn_id	Field Definition Id	A Foreign Key to the Definition table.	bigint

#### Attribute/Column(s) of Entity/Table "Item Relationship"

Physical Name	Name	Definition	Physical Data Type
item_reln_id	Item Relationship Id	The automatically generated primary key of the item_reln table.	bigint
item_reln_defn_id	Item Relationship Definition Id	A Foreign Key to the Definition table for an instance of type Definition.	bigint
from_item_defn_id	From Item Definition Id	A Foreign Key to the Definition table for an instance of type Item Definition.	bigint
to_item_defn_id	To Item Definition Id	A Foreign Key to the Definition table for an instance of type Item Definition.	bigint
link_type_defn_ver_id	Link Type Definition Version Id	A Foreign Key to the Definition Version table for an instance of type Link Type Definition Version. The link type will hold the configurable properties for the relationship.	bigint
from_item_id	From Item Id	The From Item Id in the relationship.	bigint
to_item_id	To Item Id	The To Item Id in the relationship.	bigint

#### Attribute/Column(s) of Entity/Table "Item Vote"

Physical Name	Name	Definition	Physical Data Type
item_id	Item Id	A Foreign Key to the Item table.	bigint

#### Attribute/Column(s) of Entity/Table "Knowledge Base Container"

Physical Name	Name	Definition	Physical Data Type
cntr_type_id	Container Type Id	A Foreign Key to the Container Type table.	int
cntr_id	Container Id	A Foreign Key to the Container table.	bigint

#### Attribute/Column(s) of Entity/Table "License Type"

Physical Name	Name	Definition	Physical Data Type
lic_type_id	License Type Id	The automatically generated primary key of the lic_type table.	bigint

#### Attribute/Column(s) of Entity/Table "Locale"

Physical Name	Name	Definition	Physical Data Type
locale_cd	Locale Code	A well-known string identifier used as a code to identify the Locale.	varchar(255)
locale_name	Locale Name	The name of the Locale.	varchar(255)
locale_id	Locale Id	The automatically generated primary key of the locale table.	bigint
is_sys_flag	Is System Flag	Indicates if Is System Flag.	bit

#### Attribute/Column(s) of Entity/Table "Localized Column"

Physical Name	Name	Definition	Physical Data Type
app_ele_id	Application Element Id	A Foreign Key to the Application Element table.	bigint

schema_name	Schema Name	The name of the Schema containing the Localized table. In Internal systems and in Definition level systems, this will be the logical name "FP", "FP Workspace", but in the Runtime system, this will be "FP 01", "FP 01 Workspace 001" for example.	varchar(30)
table_name	Table Name	The name of the Localized Table.	varchar(30)
col_name	Column Name	The name of the DBMS_Object_Name. It is a database object name, limited to 30 characters in length.	varchar(30)

#### Attribute/Column(s) of Entity/Table "Localized Column Value"

Physical Name	Name	Definition	Physical Data Type
app_ele_id	Application Element Id	A Foreign Key to the Application Element table.	bigint
local_col_id	Localized Column Id	A Foreign Key to the Application Element table.	bigint
initt_col_string_val	Initial Column String Value	Initial Column String Value.	varchar(255)
prmry_key_id	Primary Key Id	The integer primary key value of the row in the named Table that is being localized.	bigint

#### Attribute/Column(s) of Entity/Table "Master Contact Item"

Physical Name	Name	Definition	Physical Data Type
master_cntc_item_id	Master Contact Item Id	The automatically generated primary key of the addr_book_entry table.	BIGINT

#### Attribute/Column(s) of Entity/Table "Request For Change Item"

Physical Name	Name	Definition	Physical Data Type
item_id	Item Id	A Foreign Key to the Item table.	bigint

#### Attribute/Column(s) of Entity/Table "Schedule"

Physical Name	Name	Definition	Physical Data Type
schd_id	Schedule Id	The automatically generated primary key of the schd table.	bigint
schd_name	Schedule Name	The name of the Schedule.	varchar(255)
max_date_val	Maximum Date Value	The time spans are forecast out into the future as CalendarActualPeriod. This is the maximum date for which those periods are generated.	datetime
time_zone_region_name	Time Zone Region Name	The name of the Time Zone Region for the calendar, which derives from or overrides the value used for the system or workspace.	varchar(255)
cal_type_id	Calendar Type Id	A Foreign Key to the Calendar Type table.	int
app_user_id	Application User Id	A Foreign Key to the Application User table.	bigint
use_local_time	Use Local Time	Indicates that the workschedule is base on local time for the time zone region. If you want UTC time, use that as the region.	bit

#### Attribute/Column(s) of Entity/Table "Security Right"

Physical Name	Name	Definition	Physical Data Type
app_role_id	Application Role Id	A Foreign Key to the Application Role table.	bigint

#### Attribute/Column(s) of Entity/Table "Service Item"

Physical Name	Name	Definition	Physical Data Type
item_id	Item Id	A Foreign Key to the Item table.	bigint

#### Attribute/Column(s) of Entity/Table "Service Level Target Item"

Physical Name	Name	Definition	Physical Data Type
item_id	Item Id	A Foreign Key to the Item table.	bigint

#### Attribute/Column(s) of Entity/Table "Service Portfolio Container"

Physical Name	Name	Definition	Physical Data Type
cntnr_type_id	Container Type Id	A Foreign Key to the Container Type table.	int
cntnr_id	Container Id	A Foreign Key to the Container table.	bigint

#### Attribute/Column(s) of Entity/Table "Solution Item"

Physical Name	Name	Definition	Physical Data Type
---------------	------	------------	--------------------

item_id	Item Id	A Foreign Key to the Item table.	bigint
---------	---------	----------------------------------	--------

#### Attribute/Column(s) of Entity/Table "String Resource Value"

Physical Name	Name	Definition	Physical Data Type
string_rsrc_val_id	String Resource Value Id	The automatically generated primary key of the string_rsrc_val table.	bigint
string_val	String Value	The default value for the string, which can be overridden by user customization.	nchar(2000)
app_ele_string_id	Application Element String Id	A Foreign Key to the Application Element String table.	bigint
cxt_phrase	Context Phrase	A free-form text field in the source language (usually english) which describes the context and has any additional guidance specific to the associated locale.	varchar(255)
prod_rlse_id	Product Release Id	A Foreign Key to the Product Release table.	bigint
locale_id	Locale Id	A Foreign Key to the Locale table.	bigint

#### Attribute/Column(s) of Entity/Table "Survey Item"

Physical Name	Name	Definition	Physical Data Type
item_id	Item Id	A Foreign Key to the Item table.	bigint

#### Attribute/Column(s) of Entity/Table "System Environment"

Physical Name	Name	Definition	Physical Data Type
sys_env_id	System Environment Id	The automatically generated primary key of the sys_env table.	bigint
sys_env_name	System Environment Name	The name of the System Environment.	varchar(255)

#### Attribute/Column(s) of Entity/Table "System Log"

Physical Name	Name	Definition	Physical Data Type
tenant_parttn_id	Tenant Partition Id	A Foreign Key to the Tenant Partition table.	bigint

#### Attribute/Column(s) of Entity/Table "Template"

Physical Name	Name	Definition	Physical Data Type
defn_id	Definition Id	A Foreign Key to the Definition table.	bigint

#### Attribute/Column(s) of Entity/Table "Tenant Partition"

Physical Name	Name	Definition	Physical Data Type
sys_env_id	System Environment Id	A Foreign Key to the System Environment table.	bigint
tenant_parttn_id	Tenant Partition Id	The automatically generated primary key of the tenant_parttn table.	bigint

#### Attribute/Column(s) of Entity/Table "Tenant Partition Definition"

Physical Name	Name	Definition	Physical Data Type
db_schema_name	Database Schema Name	The name of the database schema. It is a database object name, limited to 30 characters in length.	varchar(30)
defn_id	Definition Id	A Foreign Key to the Definition table.	bigint

#### Attribute/Column(s) of Entity/Table "Ticket Item"

Physical Name	Name	Definition	Physical Data Type
item_id	Item Id	A Foreign Key to the Item table.	bigint

#### Attribute/Column(s) of Entity/Table "UI Element"

Physical Name	Name	Definition	Physical Data Type
app_ele_id	Application Element Id	A Foreign Key to the Application Element table.	bigint

#### Attribute/Column(s) of Entity/Table "UI Element Definition"

Physical Name	Name	Definition	Physical Data Type
defn_id	Definition Id	A Foreign Key to the Definition table.	bigint

#### Attribute/Column(s) of Entity/Table "UI Field"

Physical Name	Name	Definition	Physical Data Type
app_ele_id	Application Element Id	A Foreign Key to the Application Element table.	bigint

#### Attribute/Column(s) of Entity/Table "UI Field Definition"

Physical Name	Name	Definition	Physical Data Type
---------------	------	------------	--------------------

defn_id	Definition Id	A Foreign Key to the Definition table.	bigint
---------	---------------	--	--------

#### Attribute/Column(s) of Entity/Table "UI Form"

Physical Name	Name	Definition	Physical Data Type
app_ele_id	Application Element Id	A Foreign Key to the Application Element table.	bigint

#### Attribute/Column(s) of Entity/Table "UI Form Definition"

Physical Name	Name	Definition	Physical Data Type
defn_id	Definition Id	A Foreign Key to the Definition table.	bigint

#### Attribute/Column(s) of Entity/Table "UI Tab"

Physical Name	Name	Definition	Physical Data Type
app_ele_id	Application Element Id	A Foreign Key to the Application Element table.	bigint

#### Attribute/Column(s) of Entity/Table "UI Tab Definition"

Physical Name	Name	Definition	Physical Data Type
defn_id	Definition Id	A Foreign Key to the Definition table.	bigint

#### Attribute/Column(s) of Entity/Table "User Preference Definition"

Physical Name	Name	Definition	Physical Data Type
defn_id	Definition Id	A Foreign Key to the Definition table.	bigint

#### Attribute/Column(s) of Entity/Table "User Profile"

Physical Name	Name	Definition	Physical Data Type
user_prof_id	User Profile Id	The automatically generated primary key of the user_prof table.	bigint
tmpl_user_prof_id	Template User Profile Id	A Foreign Key to the User Profile table. This will not be a navigable key in the object model. This indicates which Template User Profile was used to create a built-in template in case some features concerning new properties added to later release of the system will need to be added to existing templates.	bigint
role_grp_id	Role Group Id	A Foreign Key to the Role Group table. A system level user profile can have a role group, which will have the effect of assigning a collection of workspace, cmdb, and address book roles at the container role level.	bigint
defn_id	Definition Id	A Foreign Key to the Definition table. Optionally, the container that owns the profile.	bigint
app_role_id	Application Role Id	A Foreign Key to the Application Role table. The User Profile is the means for setting the default role at the system and workspace levels, and is also the role of a Unique User Profile tied to a application user.	bigint
agent_authcn_type_id	Agent Authentication Type Id	A Foreign Key to the Agent Authentication Type table.	int
user_prof_name	User Profile Name	The name of the User Profile.	varchar(255)
is_built_in	Is Built In	Indicates that this instance of User Profile is a built-in instance that cannot be changed by the end user.	bit
user_type_id	User Type Id	A Foreign Key to the User Type table.	int
cust_authcn_type_id	Customer Authentication Type Id	A Foreign Key to the Customer Authentication Type table.	int
is_sys	Is System	Indicates if this is a system level profile.	bit
defn_subtype_id	Definition Subtype Id	A Foreign Key to the Definition Subtype table.	int
delete_date	Delete Date	The value of a User Profile, stored as a date.	datetime
last_login	Last Login	date time of last login in UTC.	datetime
tmpl_level_name	tmpl_level_name	The value of a Built_in_level, stored as a string.	varchar(255)

#### Attribute/Column(s) of Entity/Table "Valid Values Rule"

Physical Name	Name	Definition	Physical Data Type
defn_id	Definition Id	A Foreign Key to the Definition table.	bigint

#### Attribute/Column(s) of Entity/Table "Validation Rule"

Physical Name	Name	Definition	Physical Data Type
defn_id	Definition Id	A Foreign Key to the Definition table.	bigint

#### Attribute/Column(s) of Entity/Table "Workflow Process Definition"

Physical Name	Name	Definition	Physical Data Type
---------------	------	------------	--------------------

defn_id	Definition Id	A Foreign Key to the Definition table.	bigint
---------	---------------	--	--------

#### Attribute/Column(s) of Entity/Table "Workflow Process Definition Workflow Action"

Physical Name	Name	Definition	Physical Data Type
defn_id	Definition Id	A Foreign Key to the Definition table.	bigint

#### Attribute/Column(s) of Entity/Table "Workspace Container"

Physical Name	Name	Definition	Physical Data Type
time_zone_region_name	Time Zone Region Name	The name of the Time Zone Region. This is 2. The 'primary key' of the java timezone is the string region identifier in English, example 'America/Phoenix'. There is no numeric identifier. As the JRE is updated, or specific classes are updated, the list could change to include new entries, and the detail data behind the regions could change.	varchar(255)
cntr_type_id	Container Type Id	A Foreign Key to the Container Type table.	int
cntr_id	Container Id	A Foreign Key to the Container table.	bigint

#### Attribute/Column(s) of Entity/Table "WorkTarget Item"

Physical Name	Name	Definition	Physical Data Type
worktarget_item_id	WorkTarget Item Id	The automatically generated primary key of the prob_item table.	BIGINT
item_id	Item Id	A Foreign Key to the Item table.	bigint

#### ER Diagram(s) of Subject Area "010 - Translation"

Name	Physical Display Level
Logical	Column
Physical	Column

#### Entity/Table(s) of Subject Area "010 - Translation"

Physical Name	Name	Logical Only	Do Not Generate	Definition
app_ele	Application Element		false	The identity and name of an element in the application implementation. An element can be one of many types, and can contain other elements. Example Application Elements include: a screen, a form, a dialog box, a text box, the Home Page, an Architecture Component, the Item Status field, the system values of the Item Status field. The Application Element is not the full implementation of the element, but is a key and nametag for the element. In many cases, a well-known algorithm can be used to locate the correct Application Element from the Application Element implementation. In other cases, an association to the Application Element Id must be recorded as part of the implementation.
app_ele_nmosp	Application Element Namespace		false	A string that best describes the namespace of the Element Name. For example "FootPrints.UserInterface.Item.MaintenanceForm". These are maintained in a picklist so that consistency can be managed. There may be a hierarchy in the name, but the components of the hierarchy will not be enforced or maintained.
app_ele_string	Application Element String		false	A string resource that is needed in English and other translations. The string is associated with an Application Element and a Usage, for example the Label, the Tooltip, etc.
app_ele_string_usage	Application Element String Usage		false	The collection of String Usage types that are valid for use with an Application Element type. This is used for Internal systems to validate entry and control the edit interface.
app_ele_type	Application Element Type		false	The class subtype indicator for Application Element, used in an inheritance pattern. Types include UI Element, Error Code, Localized Column, etc. There may need to be further subtypes, for example of UI Element. The subtypes will share common Finder methods and key construction methods and metadata for identification.

intrnl_id_type	Internal Identifier Type		false	There are identifiers that are created by Numara Software that are used internally to uniquely identify elements. These are serial ids that will never change. There are also types of serial ids that may be used for configuration data, picklist data, etc. This Type provides a different domain of values for those keys. There may also be different domains of values provided by our partners or Professional Services. Primarily, Engineering will provide internal ids.
locale	Locale		false	A language and locale used to identify a specific dialect. It is used to determine variable forms of words, idioms, numeric format, date format, images, colors, etc. "en-US" for example is United States English, "es-LA" for Latin American Spanish.
string_rsrc_val	String Resource Value		false	The translated value of the resource.
string_usage_type	String Usage Type		false	The Usage Type of a String Resource by an Application Element, for example Label, ToolTip, Accessibility, Help Text, Data Display Value, etc. The types may be appropriate by Application Element Type.

#### Attribute/Column(s) of Entity/Table "Application Element"

Physical Name	Name	Definition	Physical Data Type
app_ele_id	Application Element Id	The automatically generated primary key of the app_ele table.	bigint
app_ele_name	Application Element Name	The name of the Application Element.	VARCHAR(2000)
intrnl_id	Internal Id	An alternate automatically generated key of the app_ele table.	int
is_intrnl_flag	Is Internal Flag	A boolean flag for indicating if the Application Element is defined internally by Engineering or by Partners or is defined by the customer.	bit
app_ele_type_id	Application Element Type Id	A Foreign Key to the Application Element Type table.	int
intrnl_id_type_id	Internal Identifier Type Id	A Foreign Key to the Internal Identifier Type table.	int
app_ele_nmsp_id	Application Element Namespace Id	A Foreign Key to the Application Element Namespace table.	bigint

#### Attribute/Column(s) of Entity/Table "Application Element Namespace"

Physical Name	Name	Definition	Physical Data Type
app_ele_nmsp_id	Application Element Namespace Id	The automatically generated primary key of the application_element_namespace table.	bigint
app_ele_nmsp_name	Application Element Namespace Name	The name of the Application Element Namespace.	varchar(255)

#### Attribute/Column(s) of Entity/Table "Application Element String"

Physical Name	Name	Definition	Physical Data Type
app_ele_string_id	Application Element String Id	The automatically generated primary key of the app_ele_string table.	bigint
app_ele_id	Application Element Id	A Foreign Key to the Application Element table.	bigint
string_usage_type_id	String Usage Type Id	A Foreign Key to the String Usage Type table.	int
part_of_speech_id	Part Of Speech Id	The unique identifier for an instance of Part Of Speech. This identifier must be kept in sync with programming code that uses an enumeration or fixed constant to identify particular values.	int
needs_mplpe_plural_forms_flag	Needs Multiple Plural Forms Flag	A boolean flag that indicates that one string per plural form must be specified. An example is "{N} Records Found.	numeric(1)

#### Attribute/Column(s) of Entity/Table "Application Element String Usage"

Physical Name	Name	Definition	Physical Data Type
app_ele_type_id	Application Element Type Id	A Foreign Key to the Application Element Type table.	int



string_usage_type_id	String Usage Type Id	A Foreign Key to the String Usage Type table.	int
----------------------	----------------------	---	-----

#### Attribute/Column(s) of Entity/Table "Application Element Type"

Physical Name	Name	Definition	Physical Data Type
app_ele_type_name	Application Element Type Name	The name of the Application Element Type.	varchar(255)
app_ele_type_id	Application Element Type Id	The automatically generated primary key of the app_ele_type table.	int

#### Attribute/Column(s) of Entity/Table "Internal Identifier Type"

Physical Name	Name	Definition	Physical Data Type
intrnl_id_type_id	Internal Identifier Type Id	The automatically generated primary key of the intrnl_id_type table.	int
intrnl_id_type_descr	Internal Identifier Type Description	A descriptive name of the Internal Identifier Type. This descriptive name may be localized.	varchar(512)

#### Attribute/Column(s) of Entity/Table "Locale"

Physical Name	Name	Definition	Physical Data Type
locale_cd	Locale Code	A well-known string identifier used as a code to identify the Locale.	varchar(255)
locale_name	Locale Name	The name of the Locale.	varchar(255)
locale_id	Locale Id	The automatically generated primary key of the locale table.	bigint
is_sys_flag	Is System Flag	Indicates if Is System Flag.	bit

#### Attribute/Column(s) of Entity/Table "String Resource Value"

Physical Name	Name	Definition	Physical Data Type
string_rsrc_val_id	String Resource Value Id	The automatically generated primary key of the string_rsrc_val table.	bigint
string_val	String Value	The default value for the string, which can be overridden by user customization.	nchar(2000)
app_ele_string_id	Application Element String Id	A Foreign Key to the Application Element String table.	bigint
cxt_phrase	Context Phrase	A free-form text field in the source language (usually english) which describes the context and has any additional guidance specific to the associated locale.	varchar(255)
prod_rlse_id	Product Release Id	A Foreign Key to the Product Release table.	bigint
locale_id	Locale Id	A Foreign Key to the Locale table.	bigint

#### Attribute/Column(s) of Entity/Table "String Usage Type"

Physical Name	Name	Definition	Physical Data Type
string_usage_type_name	String Usage Type Name	The name of the String Usage Type.	varchar(255)
string_usage_type_id	String Usage Type Id	The automatically generated primary key of the string_usage_type table.	int

#### ER Diagram(s) of Subject Area "040 - Container"

Name	Physical Display Level
Logical	Column

#### Entity/Table(s) of Subject Area "040 - Container"

Physical Name	Name	Logical Only	Do Not Generate	Definition
addr_book	Address Book Container	true	true	<p>A list of contacts that can be used to define customers. An address book can be used to define customers in a single workspace, and it can be shared among other workspaces. The fields of the Address Book can be customized. Data for the address book can be imported or exported.</p> <p>This address book can be associated with multiple workspaces. These relationships will be shown inside the definition relationship tables.</p>

cmdb_cntnr	CMDB Container	true	true	A Configuration Management Database (CMDB) class of container. The CMDB container implements ITIL CMDB special features beyond the capabilities of the generic Container. The CMDB acts as a repository of Configuration Items and supports population, synchronization, and referencing of items within it.
cntnr	Container	true	true	The highest abstract level of Runtime Object. The Runtime object has a corresponding definition that acts as a catalog of the Runtime Entity. There is a hierarchy or composition concept in the Runtime Entity, but it may not be necessary to actually have navigable properties.
cntnr_defn	Container Definition	true	true	The definition of a Container and the associated Configuration Settings at the container level. This would include the collection of Item Definitions. The Container_Definition will have a Container Class identifier (Workspace, CMDB, Address Book) and may have specializations such as Workspace_Container_Definition. As much as possible, the Container Definition will suffice to describe any Container Type. There will be an association to a default Container Template, as well as more application specific Container Templates that can be chosen to initially populate the Container Template. There will also be Container Group Definition, and Container Group Templates. See the database views v_cntnr_defn, v_cntnr_defn_active.
defn	Definition		false	A definition holds the structural and behavioral information needed for an application element that is configurable by the system designer and/or the end user at definition time. Current Footprints uses a combination of directories, files, tables, and programming code to represent this information. The new FootPrints will provide all of the same information within the definition. The definition can be persisted in a structure that is isolated at the Instance Level. The data for the definition will be stored as Configuration Settings associated with the Definition Version. See the database views v_defn_ver_active, v_cntnr_item_field.
field_defn	Field Definition	true	true	The definition of a Field, including persistence information and default UI information. Some fields will be defined as database fields, and other fields will be defined as calculated fields. The field will have data types and associations to business rule definitions. Some fields are built-in, some are defaults from templates, and some are custom fields. See the database view v_cntnr_item_field.
wksp_cntnr	Workspace Container	true	true	A workspace is the principle architectural structure in Numara FootPrints for storing and tracking information. A Workspace is a separate sub-database or work area within the system with its own fields, forms, options, workflow, settings and users. Most of the behaviour of the Workspace comes from the Container Abstract class, and only things that are specific to the Workspace are implemented here. For Example, Current FootPrints allows a single CMDB to be referenced from a Workspace, and a single Address book, yet these are listed as features for improvement. So convenience methods that get and set Container Group Relationship might be appropriate.

#### Attribute/Column(s) of Entity/Table "Address Book Container"

Physical Name	Name	Definition	Physical Data Type
---------------	------	------------	--------------------

enable_master_cntc_record	Enable Master Contact Record	Indicates if Enable Master Contact Record.	bit
addr_book_cntnr_name	Address Book Container Name	The name of the Address Book Container.	varchar(255)
prmr_y_key	Primary Key	A Foreign Key to the Address Field table.	bigint
orgal_field	Organizational Field	A Foreign Key to the Address Field table.	bigint
cntnr_type_id	Container Type Id	A Foreign Key to the Container Type table.	int
addr_book_type_id	Address Book Type Id	A Foreign Key to the Address Book Type table.	int
cntnr_id	Container Id	A Foreign Key to the Container table.	bigint

#### Attribute/Column(s) of Entity/Table "CMDB Container"

Physical Name	Name	Definition	Physical Data Type
cntnr_type_id	Container Type Id	A Foreign Key to the Container Type table.	int
cntnr_id	Container Id	A Foreign Key to the Container table.	bigint

#### Attribute/Column(s) of Entity/Table "Container"

Physical Name	Name	Definition	Physical Data Type
cntnr_name	Container Name	The name of the Container.	varchar(255)
cntnr_id	Container Id	The automatically generated primary key of the cntnr table.	bigint
tenant_parttn_id	Tenant Partition Id	A Foreign Key to the Tenant Partition table.	bigint
defn_id	Definition Id	A Foreign Key to the Definition table.	bigint

#### Attribute/Column(s) of Entity/Table "Container Definition"

Physical Name	Name	Definition	Physical Data Type
db_schema_name	Database Schema Name	The name of the database schema. It is a database object name, limited to 30 characters in length.	varchar(30)
defn_id	Definition Id	A Foreign Key to the Definition table.	bigint

#### Attribute/Column(s) of Entity/Table "Definition"

Physical Name	Name	Definition	Physical Data Type
defn_id	Definition Id	The automatically generated primary key of the defn table.	bigint
defn_name	Definition Name	The name of the Definition.	nvarchar(4000)
dsble_date	Disabled Date	The date that this instance of Definition is logically deleted and disabled.	datetime
defn_guid	Definition GUID	A Global Unique Identifier search key for the Definition, composed of letters and numbers. It is computed by the client and typically used in unstructured data.	varchar(50)
defn_type_name	Definition Type Name	The unique string identifier for an instance of Definition Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)
defn_subtype_id	Definition Subtype Id	A Foreign Key to the Definition Subtype table.	int
tmpl_export_guid	Template Export GUID	The value of a Definition, stored as a string.	VARCHAR(50)

#### Attribute/Column(s) of Entity/Table "Field Definition"

Physical Name	Name	Definition	Physical Data Type
field_dtype_id	Field Datatype Id	A Foreign Key to the Field Datatype table.	bigint
field_col_name	Field Column Name	The name of the DBMS_Object_Name. It is a database object name, limited to 30 characters in length.	varchar(30)
field_descr	Field Description	Field Description.	varchar(255)
field_defn_id	Field Definition Id	A Foreign Key to the Definition table.	bigint

#### Attribute/Column(s) of Entity/Table "Workspace Container"

Physical Name	Name	Definition	Physical Data Type
time_zone_region_name	Time Zone Region Name	The name of the Time Zone Region. This is 2. The 'primary key' of the java timezone is the string region identifier in English, example 'America/	varchar(255)

		Phoenix'. There is no numeric identifier. As the JRE is updated, or specific classes are updated, the list could change to include new entries, and the detail data behind the regions could change.	
cntnr_type_id	Container Type Id	A Foreign Key to the Container Type table.	int
cntnr_id	Container Id	A Foreign Key to the Container table.	bigint

### ER Diagram(s) of Subject Area "041 - Definition Detail"

Name	Physical Display Level
Logical	Column
Physical	Column

### Entity/Table(s) of Subject Area "041 - Definition Detail"

Physical Name	Name	Logical Only	Do Not Generate	Definition
association_type	Association Type		false	
cntnr_defn	Container Definition	true	true	The definition of a Container and the associated Configuration Settings at the container level. This would include the collection of Item Definitions. The Container Definition will have a Container Class identifier (Workspace, CMDB, Address Book) and may have specializations such as Workspace Container Definition. As much as possible, the Container Definition will suffice to describe any Container Type. There will be an association to a default Container Template, as well as more application specific Container Templates that can be chosen to initially populate the Container Template. There will also be Container Group Definition, and Container Group Templates. See the database views v_cntnr_defn, v_cntnr_defn_active.
defn	Definition		false	A definition holds the structural and behavioral information needed for an application element that is configurable by the system designer and/or the end user at definition time. Current Footprints uses a combination of directories, files, tables, and programming code to represent this information. The new FootPrints will provide all of the same information within the definition. The definition can be persisted in a structure that is isolated at the Instance Level. The data for the definition will be stored as Configuration Settings associated with the Definition Version. See the database views v_defn_ver_active, v_cntnr_item_field.
defn_reln	Definition Relationship		false	All relationships between definitions are stored in this table in order to make the model able to handle new relationships without changes to tables.
defn_reln_type	Definition Relationship Type		false	The Definition Relationship Type entity depicts a set of relationships that a Definition instance in the system can have with another Definition instance (i.e. parent_child indicates a hierarchical relationship). This specific relationship is further clarified when combined with the Definition Relationship Member Type entity.
defn_subtype	Definition Subtype		false	The subtype of the definition used as a discriminator for the class of definition. Each subtype will have different configuration settings and associated definitions. These include associated built-in field definitions for item subtypes. See the database view v_defn_subtype.
defn_type	Definition Type		false	The scope of the Template chosen from Instance, Container Group, Container, Item, Field.
defn_ver	Definition Version		false	A version of a Definition with a version identifier that changes when data changes are made to the template. The definition version will suffice to store the current state of the FootPrints Configuration. A full copy of the definition tree is versioned when a container is published. See

				the database views v_defn_ver_active, v_cntrn_item_field.
defn_ver_association	Definition Version Association		false	
defn_ver_status	Definition Version Status		false	<p>The attribute of the definition version entity that tells what state the definition instance is in. The word "Published" in this context means a change being committed or made ready for runtime. Currently the available status options are:</p> <p>Draft Dirty - the entity instance is being created or changed, but has not yet been published, so is not yet valid for runtime.</p> <p>Published Active - the entity instance has been published, and is available for runtime.</p> <p>Published Outdated - the entity instance has been previously published, but has been replaced with a newer version of itself in runtime.</p>
field_dtype	Field Datatype	true	true	The type of field (i.e. a custom field that the users have defined).
field_defn	Field Definition	true	true	The definition of a Field, including persistence information and default UI information. Some fields will be defined as database fields, and other fields will be defined as calculated fields. The field will have data types and associations to business rule definitions. Some fields are built-in, some are defaults from templates, and some are custom fields. See the database view v_cntrn_item_field.
item_defn	Item Definition	true	true	The definition of an Item, which includes the Item Name and localizations and the collection of Item Fields. It also includes customizations and localizations of string resources for Item Names, and Item Field names.
item_field_defn	Item Field Definition	true	true	Item Field Definition is an association between the Item Definition entity and the Field Definition entity. This association contains a distinct value for each of the intersection points between the two entities.
tenant_parttn_defn	Tenant Partition Definition	true	true	The Root Node definition object that links to configuration settings and properties and the instance level, and also has the collection of Container Definitions below it.

#### Attribute/Column(s) of Entity/Table "Association Type"

Physical Name	Name	Definition	Physical Data Type
association_type_id	Association Type Id	The automatically generated primary key of the association_type table.	bigint
association_type_name	Association Type Name	The name of the Association Type.	varchar(255)
association_type_descr	Association Type Description	A descriptive name of the Association Type. This descriptive name may be localized.	varchar(512)

#### Attribute/Column(s) of Entity/Table "Container Definition"

Physical Name	Name	Definition	Physical Data Type
db_schema_name	Database Schema Name	The name of the database schema. It is a database object name, limited to 30 characters in length.	varchar(30)
defn_id	Definition Id	A Foreign Key to the Definition table.	bigint

#### Attribute/Column(s) of Entity/Table "Definition"

Physical Name	Name	Definition	Physical Data Type
defn_id	Definition Id	The automatically generated primary key of the defn table.	bigint
defn_name	Definition Name	The name of the Definition.	nvarchar(4000)
dsble_date	Disabled Date	The date that this instance of Definition is logically deleted and disabled.	datetime
defn_guid	Definition GUID	A Global Unique Identifier search key for the Definition, composed of letters and numbers. It is computed by the client and typically used in unstructured data.	varchar(50)

defn_type_name	Definition Type Name	The unique string identifier for an instance of Definition Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)
defn_subtype_id	Definition Subtype Id	A Foreign Key to the Definition Subtype table.	int
tmpl_export_guid	Template Export GUID	The value of a Definition, stored as a string.	VARCHAR(50)

#### Attribute/Column(s) of Entity/Table "Definition Relationship"

Physical Name	Name	Definition	Physical Data Type
defn_reln_id	Definition Relationship Id	The automatically generated primary key of the defn_reln table.	bigint
from_defn_ver_id	From Definition Version Id	A Foreign Key to the Definition Version table.	bigint
to_defn_ver_id	To Definition Version Id	A Foreign Key to the Definition Version table.	bigint
defn_reln_type_id	Definition Relationship Type Id	A Foreign Key to the Definition Relationship Type table.	int

#### Attribute/Column(s) of Entity/Table "Definition Relationship Type"

Physical Name	Name	Definition	Physical Data Type
defn_reln_type_name	Definition Relationship Type Name	The name of the Definition Relationship Type.	varchar(255)
defn_reln_type_id	Definition Relationship Type Id	The automatically generated primary key of the defn_reln_type table.	int
defn_reln_type_descr	Definition Relationship Type Description	A descriptive name of the Definition Relationship Type. This descriptive name may be localized.	varchar(512)

#### Attribute/Column(s) of Entity/Table "Definition Subtype"

Physical Name	Name	Definition	Physical Data Type
defn_subtype_name	Definition Subtype Name	The name of the Definition Subtype.	varchar(255)
defn_subtype_descr	Definition Subtype Description	A descriptive name of the Definition Subtype. This descriptive name may be localized.	varchar(512)
defn_subtype_id	Definition Subtype Id	The automatically generated primary key of the defn_subtype table.	int
defn_parent_subtype_id	definition parent subtype id	A Foreign Key to the Definition Subtype table.	int
defn_type_name	Definition Type Name	The unique string identifier for an instance of Definition Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)
max_in_parent	Maximum In Parent	A Foreign Key to the Definition Subtype table.	int

#### Attribute/Column(s) of Entity/Table "Definition Type"

Physical Name	Name	Definition	Physical Data Type
defn_type_name	Definition Type Name	The unique string identifier for an instance of Definition Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)
defn_type_descr	Definition Type Description	A descriptive name of the Definition Type. This descriptive name may be localized.	varchar(512)

#### Attribute/Column(s) of Entity/Table "Definition Version"

Physical Name	Name	Definition	Physical Data Type
defn_id	Definition Id	A Foreign Key to the Definition table.	bigint
defn_ver_id	Definition Version Id	The automatically generated primary key of the defn_ver table.	bigint
defn_ver_name	Definition Version Name	The name of the Definition Version.	nvarchar(4000)
defn_ver_status_id	Definition Version Status Id	A Foreign Key to the Definition Version Status table.	int
defn_ver_guid	Definition Version GUID	A Global Unique Identifier search key for the Definition Version, composed of letters and numbers. It is computed by the client and typically used in unstructured data.	varchar(50)

#### Attribute/Column(s) of Entity/Table "Definition Version Association"

Physical Name	Name	Definition	Physical Data Type
---------------	------	------------	--------------------

defn_ver_association_id	Definition Version Association Id	The automatically generated primary key of the defn_ver_association table.	bigint
from_defn_ver_id	From Definition Version ID	A Foreign Key to the Definition Version table.	bigint
to_defn_ver_id	To Definition Version ID	A Foreign Key to the Definition Version table.	bigint
association_type_id	Association Type Id	A Foreign Key to the Association Type table.	bigint

#### Attribute/Column(s) of Entity/Table "Definition Version Status"

Physical Name	Name	Definition	Physical Data Type
defn_ver_status_name	Definition Version Status Name	The name of the Definition Version Status.	varchar(255)
defn_ver_status_id	Definition Version Status Id	The automatically generated primary key of the defn_ver_status table.	int

#### Attribute/Column(s) of Entity/Table "Field Datatype"

Physical Name	Name	Definition	Physical Data Type
field_dtype_id	Field Datatype Id	The automatically generated primary key of the field_dtype table.	bigint
field_dtype_name	Field Datatype Name	The name of the Field Datatype.	varchar(255)

#### Attribute/Column(s) of Entity/Table "Field Definition"

Physical Name	Name	Definition	Physical Data Type
field_dtype_id	Field Datatype Id	A Foreign Key to the Field Datatype table.	bigint
field_col_name	Field Column Name	The name of the DBMS_Object_Name. It is a database object name, limited to 30 characters in length.	varchar(30)
field_descr	Field Description	Field Description.	varchar(255)
field_defn_id	Field Definition Id	A Foreign Key to the Definition table.	bigint

#### Attribute/Column(s) of Entity/Table "Item Definition"

Physical Name	Name	Definition	Physical Data Type
item_table_name	Item Table Name	The name of the DBMS_Object_Name. It is a database object name, limited to 30 characters in length.	varchar(30)
item_defn_id	Item Definition Id	A Foreign Key to the Definition table for an instance of type Definition.	bigint

#### Attribute/Column(s) of Entity/Table "Item Field Definition"

Physical Name	Name	Definition	Physical Data Type
item_field_defn_id	Item Field Definition Id	A Foreign Key to the Definition table for an instance of type Definition.	bigint
item_defn_id	Item Definition Id	A Foreign Key to the Definition table for an instance of type Definition.	bigint
field_defn_id	Field Definition Id	A Foreign Key to the Definition table.	bigint

#### Attribute/Column(s) of Entity/Table "Tenant Partition Definition"

Physical Name	Name	Definition	Physical Data Type
db_schema_name	Database Schema Name	The name of the database schema. It is a database object name, limited to 30 characters in length.	varchar(30)
defn_id	Definition Id	A Foreign Key to the Definition table.	bigint

#### ER Diagram(s) of Subject Area "046 - Workspace Item Assignment"

Name	Physical Display Level
Logical	Column
Physical	Column

#### Entity/Table(s) of Subject Area "046 - Workspace Item Assignment"

Physical Name	Name	Logical Only	Do Not Generate	Definition
app_user	Application User		false	A user authorized to use the FootPrints system. The user is either a Customer User or an Agent User. The application user holds the minimum information needed to identify the user and associate the user with a user profile, where user settings are stored. A contact is not necessarily an application user. An application user entry is required in order for the user to access any online functions.

asgnmt	Assignment		false	The current state and historical record of responsibility for a ticket. Regular assignments involve all parties that share responsibility for the ticket and are interested potentially in notifications. Primary Assignments additionally indicate that one and only one userid is responsible for the ticket for a span of time. A userid will have a span of Primary Assignment that is within the boundaries of his regular non-Primary Assignment. It is expected that an individual would have a single large span of time, possibly the life of the ticket, as a regular assignment. The individual may have zero, one, or more periods of PrimaryAssignment, stored as additional records.
asgnmt_type	Assignment Type		false	The assignment type indicates the notification level of the assignment. The current types consist of the following: Regular Assignment, CC Assignment, Primary Assignment .
cfg_item	Configuration Item	true	true	A specialization of Item contained in a CMDB container. A Configuration Item (CI) in the Configuration Management Data Base (CMDB). A Configuration Item is a repository item representing a software, hardware, or service item. There are important relationships to other items both within the CMDB but also externally to the CMDB. The CI has the ability to store fields as defined based on its CI Type. Current Footprints creates a table for the CI type named with the CMDB sequence identifier and the CI Type sequence identifier.
addr_book_entry	Contact Item	true	true	A contact that belongs to the FootPrints address book. This could be implemented as a specialization of Item which has the needed features of flexible schema, and data mapping adapters.
defn	Definition		false	A definition holds the structural and behavioral information needed for an application element that is configurable by the system designer and/or the end user at definition time. Current Footprints uses a combination of directories, files, tables, and programming code to represent this information. The new FootPrints will provide all of the same information within the definition. The definition can be persisted in a structure that is isolated at the Instance Level. The data for the definition will be stored as Configuration Settings associated with the Definition Version. See the database views v_defn_ver_active, v_cntrn_item_field.
item	Item	true	true	The Item is the abstract class for Issue, Configuration Item, Address Book entry, and other information items that are the main information entities in FootPrints. The Item structure and behavior is defined by its Definition. The Item is namespaced, identified by, and logically stored in the single parent container. It can be referenced in the same manner both inside and outside the container. The Item is the storage model and the implementation for the Item Definition. There are many important specializations of the Item class, and there even more Template definitions of Container/Item that provide an end-user application experience to the End-User. The Item implementation must implement flexible schema in an easy to manage, well engineered, and high performing way. The item can store user-defined and user-named fields defined in its definition. The Item could implement the nice-to-have features such as an inheritance model for Item subtypes, and



				have the ability to represent collections of subordinate items. The Item data can be imported, exported, or synchronized with external data sources. The Item data can be linked and synchronized with other items in the same Instance.
item_field	Item Field	true	true	A place related to a specific item that stores or displays data.
team	Team		false	A team is a group of application users that can be used in Team level assignments.
wksp_item	Ticket Item	true	true	A specialization of Item contained in a Workspace container. Workspace Items are tracked for resolution and closure, and have parties that are awaiting resolution and closure, and responsible parties that perform work against them.

#### Attribute/Column(s) of Entity/Table "Application User"

Physical Name	Name	Definition	Physical Data Type
pwd	Password	This space intentionally left blank.	varchar(255)
user_name	User Name	The name of the Application User.	nvarchar(4000)
cust_login_type_id	Customer Login Type Id	A Foreign Key to the Customer Login Type table.	int
app_user_id	Application User Id	The automatically generated primary key of the app_user table.	bigint
lic_type_id	License Type Id	A Foreign Key to the License Type table.	bigint
user_login	User Login	The login name used to access the Application. This may not be filled in if external application security is used.	nvarchar(4000)
email_addr	Email Address	An email address for the Application User.	nvarchar(4000)
delete_date	Delete Date	The value of a Application User, stored as a date.	datetime

#### Attribute/Column(s) of Entity/Table "Assignment"

Physical Name	Name	Definition	Physical Data Type
asnmt_id	Assignment Id	The automatically generated primary key of the asgnmt table.	bigint
item_defn_id	Item Definition Id	A Foreign Key to the Definition table.	bigint
team_id	Team Id	A Foreign Key to the Team table.	bigint
begin_date	Begin Date	The value of a Assignment, stored as a date.	datetime
end_date	End Date	The value of a Assignment, stored as a date.	datetime
asnmt_type_id	Assignment Type Id	A Foreign Key to the Assignment Type table.	int
app_user_id	Application User Id	A Foreign Key to the Application User table.	bigint
cc_email_addr	CC Email Address	Cannot be used with TeamId, App_user_id, can only be used with CC Assignment	nvarchar(4000)
sort_order	Sort Order	The value of a Assignment stored as boolean.	int
item_id	Item Id	A Foreign Key to the Item table.	bigint
is_team	Is Team	Special clarifying field of "Is Team Level Assignment" to state that the Team does not have to be immediately resolved to a Team+Individual.	bit
delete_txn_id	Delete Transaction Id	A unique incremental transaction identifier that indicates a soft delete. The id is generated via id generation, and may or may not correspond to the audit trail transaction id. This method of marking the soft delete allows for a unique index to be created on the Workspace Definition Id, Team Name, and Delete Transaction Id to enforce no duplicates in the live copies, and it allows multiple soft deleted rows with the same team name.	bigint

field_id	Field Id	A Foreign Key to the Definition table.	bigint
----------	----------	--	--------

#### Attribute/Column(s) of Entity/Table "Assignment Type"

Physical Name	Name	Definition	Physical Data Type
asgnmt_type_id	Assignment Type Id	The automatically generated primary key of the asgnmt_type table.	int
asgnmt_type_name	Assignment Type Name	The name of the Assignment Type.	varchar(255)
asgnmt_type_descr	Assignment Type Description	A descriptive name of the Assignment Type. This descriptive name may be localized.	varchar(512)

#### Attribute/Column(s) of Entity/Table "Configuration Item"

Physical Name	Name	Definition	Physical Data Type
item_id	Item Id	A Foreign Key to the Item table.	bigint

#### Attribute/Column(s) of Entity/Table "Contact Item"

Physical Name	Name	Definition	Physical Data Type
item_id	Item Id	A Foreign Key to the Item table.	bigint
master_cntc_item_id	Master Contact Item Id	A Foreign Key to the Master Contact Item table.	BIGINT

#### Attribute/Column(s) of Entity/Table "Definition"

Physical Name	Name	Definition	Physical Data Type
defn_id	Definition Id	The automatically generated primary key of the defn table.	bigint
defn_name	Definition Name	The name of the Definition.	nvarchar(4000)
dsble_date	Disabled Date	The date that this instance of Definition is logically deleted and disabled.	datetime
defn_guid	Definition GUID	A Global Unique Identifier search key for the Definition, composed of letters and numbers. It is computed by the client and typically used in unstructured data.	varchar(50)
defn_type_name	Definition Type Name	The unique string identifier for an instance of Definition Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)
defn_subtype_id	Definition Subtype Id	A Foreign Key to the Definition Subtype table.	int
tmpl_export_guid	Template Export GUID	The value of a Definition, stored as a string.	VARCHAR(50)

#### Attribute/Column(s) of Entity/Table "Item"

Physical Name	Name	Definition	Physical Data Type
item_key	Item Key	A search key for the Item, composed of letters and numbers.	varchar(255)
item_id	Item Id	The automatically generated primary key of the item table.	bigint
cntnr_id	Container Id	A Foreign Key to the Container table.	bigint
item_defn_id	Item Definition Id	A Foreign Key to the Definition table for an instance of type Definition.	bigint

#### Attribute/Column(s) of Entity/Table "Item Field"

Physical Name	Name	Definition	Physical Data Type
item_id	Item Id	A Foreign Key to the Item table.	bigint

#### Attribute/Column(s) of Entity/Table "Team"

Physical Name	Name	Definition	Physical Data Type
team_id	Team Id	The automatically generated primary key of the team table.	bigint
team_name	Team Name	The name of the Team.	nvarchar(4000)
wksp_defn_id	Workspace Definition Id	A Foreign Key to the Definition table.	bigint
team_email_notifn_type_id	Team Email Notification Type Id	A Foreign Key to the Team Email Notification Type table.	int
team_lead_app_user_id	Team Lead Application User Id	A Foreign Key to the Application User table.	bigint

#### Attribute/Column(s) of Entity/Table "Ticket Item"

Physical Name	Name	Definition	Physical Data Type
---------------	------	------------	--------------------

item_id	Item Id	A Foreign Key to the Item table.	bigint
---------	---------	----------------------------------	--------

### ER Diagram(s) of Subject Area "047 - Workspace Time Tracking"

Name	Physical Display Level
Logical	Column
Physical	Column

### Entity/Table(s) of Subject Area "047 - Workspace Time Tracking"

Physical Name	Name	Logical Only	Do Not Generate	Definition
app_user	Application User		false	A user authorized to use the FootPrints system. The user is either a Customer User or an Agent User. The application user holds the minimum information needed to identify the user and associate the user with a user profile, where user settings are stored. A contact is not necessarily an application user. An application user entry is required in order for the user to access any online functions.
defn	Definition		false	A definition holds the structural and behavioral information needed for an application element that is configurable by the system designer and/or the end user at definition time. Current Footprints uses a combination of directories, files, tables, and programming code to represent this information. The new FootPrints will provide all of the same information within the definition. The definition can be persisted in a structure that is isolated at the Instance Level. The data for the definition will be stored as Configuration Settings associated with the Definition Version. See the database views v_defn_ver_active, v_cntrn_item_field.
item	Item	true	true	The Item is the abstract class for Issue, Configuration Item, Address Book entry, and other information items that are the main information entities in FootPrints. The Item structure and behavior is defined by its Definition. The Item is namespaced, identified by, and logically stored in the single parent container. It can be referenced in the same manner both inside and outside the container. The Item is the storage model and the implementation for the Item Definition. There are many important specializations of the Item class, and there even more Template definitions of Container/Item that provide an end-user application experience to the End-User. The Item implementation must implement flexible schema in an easy to manage, well engineered, and high performing way. The item can store user-defined and user-named fields defined in its definition. The Item could implement the nice-to-have features such as an inheritance model for Item subtypes, and have the ability to represent collections of subordinate items. The Item data can be imported, exported, or synchronized with external data sources. The Item data can be linked and synchronized with other items in the same Instance.

time_tracking	Time Tracking		false	A table of time tracking entries used to accumulate a log of time spent on ticket editing and viewing.
time_tracking_type	Time Tracking Type		false	The type of time tracking entry. Values include AUTOMATIC, MANUAL_DURATION, MANUAL_RANGE.

#### Attribute/Column(s) of Entity/Table "Application User"

Physical Name	Name	Definition	Physical Data Type
pwd	Password	This space intentionally left blank.	varchar(255)
user_name	User Name	The name of the Application User.	nvarchar(4000)
cust_login_type_id	Customer Login Type Id	A Foreign Key to the Customer Login Type table.	int
app_user_id	Application User Id	The automatically generated primary key of the app_user table.	bigint
lic_type_id	License Type Id	A Foreign Key to the License Type table.	bigint
user_login	User Login	The login name used to access the Application. This may not be filled in if external application security is used.	nvarchar(4000)
email_addr	Email Address	An email address for the Application User.	nvarchar(4000)
delete_date	Delete Date	The value of a Application User, stored as a date.	datetime

#### Attribute/Column(s) of Entity/Table "Definition"

Physical Name	Name	Definition	Physical Data Type
defn_id	Definition Id	The automatically generated primary key of the defn table.	bigint
defn_name	Definition Name	The name of the Definition.	nvarchar(4000)
dsble_date	Disabled Date	The date that this instance of Definition is logically deleted and disabled.	datetime
defn_guid	Definition GUID	A Global Unique Identifier search key for the Definition, composed of letters and numbers. It is computed by the client and typically used in unstructured data.	varchar(50)
defn_type_name	Definition Type Name	The unique string identifier for an instance of Definition Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)
defn_subtype_id	Definition Subtype Id	A Foreign Key to the Definition Subtype table.	int
tmpl_export_guid	Template Export GUID	The value of a Definition, stored as a string.	VARCHAR(50)

#### Attribute/Column(s) of Entity/Table "Item"

Physical Name	Name	Definition	Physical Data Type
item_key	Item Key	A search key for the Item, composed of letters and numbers.	varchar(255)
item_id	Item Id	The automatically generated primary key of the item table.	bigint
cntnr_id	Container Id	A Foreign Key to the Container table.	bigint
item_defn_id	Item Definition Id	A Foreign Key to the Definition table for an instance of type Definition.	bigint

#### Attribute/Column(s) of Entity/Table "Time Tracking"

Physical Name	Name	Definition	Physical Data Type
time_tracking_id	Time Tracking Id	The automatically generated primary key of the time_tracking table.	bigint
time_tracking_type_id	Time Tracking Type Id	A Foreign Key to the Time Tracking Type table.	int
app_user_id	Application User Id	A Foreign Key to the Application User table.	bigint
start_date_time	Start Date Time	The value of a Time Tracking, stored as a date.	datetime
end_date_time	End Date Time	The value of a Time Tracking, stored as a date.	datetime
time_tracking_descr	Time Tracking Description	A descriptive name of the Time Tracking. This descriptive name may be localized.	nvarchar(4000)

rate_per_hour	Rate Per Hour	The value of a Time Tracking stored as real.	numeric(38,20)
rate_descr	Rate Description	A descriptive name of the Time Tracking. This descriptive name may be localized.	varchar(512)
durn_mins	Duration Minutes	The value of a Time Tracking stored as Long.	bigint
item_id	item_id		BIGINT
item_defn_id	Item Definition Id	A Foreign Key to the Definition table.	bigint
item_id	Item Id	A Foreign Key to the Item table.	bigint

#### Attribute/Column(s) of Entity/Table "Time Tracking Type"

Physical Name	Name	Definition	Physical Data Type
time_tracking_type_name	Time Tracking Type Name	The name of the Time Tracking Type.	varchar(255)
time_tracking_type_id	Time Tracking Type Id	The automatically generated primary key of the time_tracking_type table.	int
time_tracking_type_descr	Time Tracking Type Description	A descriptive name of the Time Tracking Type. This descriptive name may be localized.	varchar(512)

#### ER Diagram(s) of Subject Area "048 - Item Field Journaling"

Name	Physical Display Level
Logical	Column
Physical	Column

#### Entity/Table(s) of Subject Area "048 - Item Field Journaling"

Physical Name	Name	Logical Only	Do Not Generate	Definition
app_user	Application User		false	A user authorized to use the FootPrints system. The user is either a Customer User or an Agent User. The application user holds the minimum information needed to identify the user and associate the user with a user profile, where user settings are stored. A contact is not necessarily an application user. An application user entry is required in order for the user to access any online functions.
item	Item	true	true	The Item is the abstract class for Issue, Configuration Item, Address Book entry, and other information items that are the main information entities in FootPrints. The Item structure and behavior is defined by its Definition. The Item is namespaced, identified by, and logically stored in the single parent container. It can be referenced in the same manner both inside and outside the container. The Item is the storage model and the implementation for the Item Definition. There are many important specializations of the Item class, and there even more Template definitions of Container/Item that provide an end-user application experience to the End-User. The Item implementation must implement flexible schema in an easy to manage, well engineered, and high performing way. The item can store user-defined and user-named fields defined in its definition. The Item could implement the nice-to-have features such as an inheritance model for Item subtypes, and have the ability to represent collections of subordinate items. The Item data can be imported, exported, or synchronized with external data sources. The Item data can be linked and synchronized with other items in the same Instance.

item_defn	Item Definition	true	true	The definition of an Item, which includes the Item Name and localizations and the collection of Item Fields. It also includes customizations and localizations of string resources for Item Names, and Item Field names.
item_field	Item Field	true	true	A place related to a specific item that stores or displays data.
item_field_defn	Item Field Definition	true	true	Item Field Definition is an association between the Item Definition entity and the Field Definition entity. This association contains a distinct value for each of the intersection points between the two entities.
item_field_journal	Item Field Journal		false	Holds a collection of field values for strings, each of which has create information for the current state of the field. Used when an Item has a multi-line edit field with the journaling feature turned on.
wksp_item	Ticket Item	true	true	A specialization of Item contained in a Workspace container. Workspace Items are tracked for resolution and closure, and have parties that are awaiting resolution and closure, and responsible parties that perform work against them.

#### Attribute/Column(s) of Entity/Table "Application User"

Physical Name	Name	Definition	Physical Data Type
pwd	Password	This space intentionally left blank.	varchar(255)
user_name	User Name	The name of the Application User.	nvarchar(4000)
cust_login_type_id	Customer Login Type Id	A Foreign Key to the Customer Login Type table.	int
app_user_id	Application User Id	The automatically generated primary key of the app_user table.	bigint
lic_type_id	License Type Id	A Foreign Key to the License Type table.	bigint
user_login	User Login	The login name used to access the Application. This may not be filled in if external application security is used.	nvarchar(4000)
email_addr	Email Address	An email address for the Application User.	nvarchar(4000)
delete_date	Delete Date	The value of a Application User, stored as a date.	datetime

#### Attribute/Column(s) of Entity/Table "Item"

Physical Name	Name	Definition	Physical Data Type
item_key	Item Key	A search key for the Item, composed of letters and numbers.	varchar(255)
item_id	Item Id	The automatically generated primary key of the item table.	bigint
cntr_id	Container Id	A Foreign Key to the Container table.	bigint
item_defn_id	Item Definition Id	A Foreign Key to the Definition table for an instance of type Definition.	bigint

#### Attribute/Column(s) of Entity/Table "Item Definition"

Physical Name	Name	Definition	Physical Data Type
item_table_name	Item Table Name	The name of the DBMS_Object_Name. It is a database object name, limited to 30 characters in length.	varchar(30)
item_defn_id	Item Definition Id	A Foreign Key to the Definition table for an instance of type Definition.	bigint

#### Attribute/Column(s) of Entity/Table "Item Field"

Physical Name	Name	Definition	Physical Data Type
item_id	Item Id	A Foreign Key to the Item table.	bigint

#### Attribute/Column(s) of Entity/Table "Item Field Definition"

Physical Name	Name	Definition	Physical Data Type
item_field_defn_id	Item Field Definition Id	A Foreign Key to the Definition table for an instance of type Definition.	bigint
item_defn_id	Item Definition Id	A Foreign Key to the Definition table for an instance of type Definition.	bigint

field_defn_id	Field Definition Id	A Foreign Key to the Definition table.	bigint
---------------	---------------------	--	--------

#### Attribute/Column(s) of Entity/Table "Item Field Journal"

Physical Name	Name	Definition	Physical Data Type
id	id	The automatically generated primary key of the item_field_journal table.	bigint
app_user_id	Application User Id	A Foreign Key to the Application User table.	bigint
soft_delete_id	Soft Delete Id	A transaction id that is null for active data and filled in if the item_field_journal row is soft deleted. This allows unique indexes to be defined that include the soft delete id.	int
string_val	String Value	The value of a Item Field Journal, stored as a string.	varchar(255)
create_date	Create Date	The date that the Item Field Journal was created.	datetime
item_field_defn_id	Item Field Definition Id	A Foreign Key to the Definition table for an instance of type Definition.	bigint
item_id	Item Id	A Foreign Key to the Item table.	bigint

#### Attribute/Column(s) of Entity/Table "Ticket Item"

Physical Name	Name	Definition	Physical Data Type
item_id	Item Id	A Foreign Key to the Item table.	bigint

#### ER Diagram(s) of Subject Area "050 - Item Relationship"

Name	Physical Display Level
Logical	Column
Physical	Column

#### Entity/Table(s) of Subject Area "050 - Item Relationship"

Physical Name	Name	Logical Only	Do Not Generate	Definition
item	Item	true	true	The Item is the abstract class for Issue, Configuration Item, Address Book entry, and other information items that are the main information entities in FootPrints. The Item structure and behavior is defined by its Definition. The Item is namespaced, identified by, and logically stored in the single parent container. It can be referenced in the same manner both inside and outside the container. The Item is the storage model and the implementation for the Item Definition. There are many important specializations of the Item class, and there even more Template definitions of Container/Item that provide an end-user application experience to the End-User. The Item implementation must implement flexible schema in an easy to manage, well engineered, and high performing way. The Item can store user-defined and user-named fields defined in its definition. The Item could implement the nice-to-have features such as an inheritance model for Item subtypes, and have the ability to represent collections of subordinate items. The Item data can be imported, exported, or synchronized with external data sources. The Item data can be linked and synchronized with other items in the same Instance.

item_defn	Item Definition	true	true	The definition of an Item, which includes the Item Name and localizations and the collection of Item Fields. It also includes customizations and localizations of string resources for Item Names, and Item Field names.
item_reln	Item Relationship		false	An Item Relationship is a relationship between instances contained in two Dynamic Object generated Tables. For example, the following. A Configuration Item Type ("Computer") related to an Workspace Item ("Incident"). An Address Book Entry ("Contact") related to another Address Book Entry ("Master Contact"). A Dynamic Link between an Incident in Workspace 1 ("Current Tickets") and Workspace 3 ("Archive Workspace"). A Dynamic Link between an Incident in Workspace 1 ("Current Tickets") and Workspace 5 ("Change Management"). There are no foreign keys defined.
item_reln_defn	Item Relationship Definition	true	true	The definition of a Relationship type within the Container Definition. This has important Configuration Settings and Application Element Identifier values that clarify the meaning of the Relationship to programming code.
link_type_defn_ver	Link Type Definition Version	true	true	A definition version instance that describes the link properties. This design allows the link properties to be dynamically defined by configuration settings. That way links can add behavior in the future without affecting existing data.

#### Attribute/Column(s) of Entity/Table "Item"

Physical Name	Name	Definition	Physical Data Type
item_key	Item Key	A search key for the Item, composed of letters and numbers.	varchar(255)
item_id	Item Id	The automatically generated primary key of the item table.	bigint
cntnr_id	Container Id	A Foreign Key to the Container table.	bigint
item_defn_id	Item Definition Id	A Foreign Key to the Definition table for an instance of type Definition.	bigint

#### Attribute/Column(s) of Entity/Table "Item Definition"

Physical Name	Name	Definition	Physical Data Type
item_table_name	Item Table Name	The name of the DBMS_Object_Name. It is a database object name, limited to 30 characters in length.	varchar(30)
item_defn_id	Item Definition Id	A Foreign Key to the Definition table for an instance of type Definition.	bigint

#### Attribute/Column(s) of Entity/Table "Item Relationship"

Physical Name	Name	Definition	Physical Data Type
item_reln_id	Item Relationship Id	The automatically generated primary key of the item_reln table.	bigint
item_reln_defn_id	Item Relationship Definition Id	A Foreign Key to the Definition table for an instance of type Definition.	bigint
from_item_defn_id	From Item Definition Id	A Foreign Key to the Definition table for an instance of type Item Definition.	bigint
to_item_defn_id	To Item Definition Id	A Foreign Key to the Definition table for an instance of type Item Definition.	bigint
link_type_defn_ver_id	Link Type Definition Version Id	A Foreign Key to the Definition Version table for an instance of type Link Type Definition Version. The link type will hold the configurable properties for the relationship.	bigint
from_item_id	From Item Id	The From Item Id in the relationship.	bigint
to_item_id	To Item Id	The To Item Id in the relationship.	bigint

#### Attribute/Column(s) of Entity/Table "Item Relationship Definition"

Physical Name	Name	Definition	Physical Data Type
item_reln_defn_id	Item Relationship Definition Id	A Foreign Key to the Definition table for an instance of type Definition.	bigint



from_mbr_role_type_id	From Member Role Type Id		char(18)
to_mbr_role_type_id	To Member Role Type Id		char(18)
reln_type_id	Relationship Type Id	Relationship type	char(18)

#### Attribute/Column(s) of Entity/Table "Link Type Definition Version"

Physical Name	Name	Definition	Physical Data Type
link_type_defn_ver_id	Link Type Definition Version Id	The automatically generated primary key of the Definition Version table for a an instance of type link_type_defn_ver.	bigint
cardinality_node_2	Cardinality Node 2	The value of a Cardinality Node 2 stored as a Configuration Setting owned by the Link Type Definition Version. See the meta data stored in the Configuration Setting table where dflt_setting_label = 'Cardinality Node 2'.	char(18)
color_val_setting	Color Value Setting	The value of a Color Value Setting stored as a Configuration Setting owned by the Link Type Definition Version. See the meta data stored in the Configuration Setting table where dflt_setting_label = 'Color Value Setting'.	char(18)
item_type_node_1	Item type Node 1	The value of a Item type Node 1 stored as a Configuration Setting owned by the Link Type Definition Version. See the meta data stored in the Configuration Setting table where dflt_setting_label = 'Item type Node 1'.	char(18)
cardinality_node_1	Cardinality Node 1	The value of a Cardinality Node 1 stored as a Configuration Setting owned by the Link Type Definition Version. See the meta data stored in the Configuration Setting table where dflt_setting_label = 'Cardinality Node 1'.	char(18)
item_type_node_2	Item type Node 2	The value of a Item type Node 2 stored as a Configuration Setting owned by the Link Type Definition Version. See the meta data stored in the Configuration Setting table where dflt_setting_label = 'Item type Node 2'.	char(18)
node_1_name	Node 1 Name	The value of a Node 1 Name stored as a Configuration Setting owned by the Link Type Definition Version. See the meta data stored in the Configuration Setting table where dflt_setting_label = 'Node 1 Name'.	char(18)
node_2_name	Node 2 Name	The value of a Node 2 Name stored as a Configuration Setting owned by the Link Type Definition Version. See the meta data stored in the Configuration Setting table where dflt_setting_label = 'Node 2 Name'.	char(18)
symmetric_link	Symmetric Link	The value of a Symmetric Link stored as a Configuration Setting owned by the Link Type Definition Version. See the meta data stored in the Configuration Setting table where dflt_setting_label = 'Symmetric Link'.	char(18)

#### ER Diagram(s) of Subject Area "051 - Item Field Collection"

Name	Physical Display Level
Logical	Column
Physical	Column

#### Entity/Table(s) of Subject Area "051 - Item Field Collection"

Physical Name	Name	Logical Only	Do Not Generate	Definition
item	Item	true	true	The Item is the abstract class for Issue, Configuration Item, Address Book entry, and other information items that are the main information entities in FootPrints. The Item structure and behavior is defined by its Definition. The Item is namespaced, identified by, and logically stored in the single parent container. It can be referenced in the same manner both inside and outside the container. The Item is the storage model and the implementation for the Item Definition. There are many important specializations of the Item class, and there even more Template definitions of Container/Item that provide an end-user application experience to the End-User. The Item implementation must implement flexible schema in an easy to manage, well engineered, and high performing way. The item can store user-defined and user-named fields defined in its definition. The Item could implement the nice-to-have features such as an inheritance model for Item subtypes, and have the ability to represent collections of subordinate items.

				The Item data can be imported, exported, or synchronized with external data sources. The Item data can be linked and synchronized with other items in the same Instance.
item_defn	Item Definition	true	true	The definition of an Item, which includes the Item Name and localizations and the collection of Item Fields. It also includes customizations and localizations of string resources for Item Names, and Item Field names.
item_field	Item Field	true	true	A place related to a specific item that stores or displays data.
item_field_coll	Item Field Collection		false	Holds an unordered collection of string field values for an Item. Used for picklists and multiple selections. All multiselect fields are stored here. This table can be implemented as a global system table, container-level table, or item-level table. Currently it is a system level table.
item_field_defn	Item Field Definition	true	true	Item Field Definition is an association between the Item Definition entity and the Field Definition entity. This association contains a distinct value for each of the intersection points between the two entities.

#### Attribute/Column(s) of Entity/Table "Item"

Physical Name	Name	Definition	Physical Data Type
item_key	Item Key	A search key for the Item, composed of letters and numbers.	varchar(255)
item_id	Item Id	The automatically generated primary key of the item table.	bigint
cntrn_id	Container Id	A Foreign Key to the Container table.	bigint
item_defn_id	Item Definition Id	A Foreign Key to the Definition table for an instance of type Definition.	bigint

#### Attribute/Column(s) of Entity/Table "Item Definition"

Physical Name	Name	Definition	Physical Data Type
item_table_name	Item Table Name	The name of the DBMS Object Name. It is a database object name, limited to 30 characters in length.	varchar(30)
item_defn_id	Item Definition Id	A Foreign Key to the Definition table for an instance of type Definition.	bigint

#### Attribute/Column(s) of Entity/Table "Item Field"

Physical Name	Name	Definition	Physical Data Type
item_id	Item Id	A Foreign Key to the Item table.	bigint

#### Attribute/Column(s) of Entity/Table "Item Field Collection"

Physical Name	Name	Definition	Physical Data Type
item_id	Item Id	A Foreign Key to the Item table.	bigint
string_val	String Value	The value of a Item Field Collection, stored as a string.	varchar(255)
item_field_defn_id	Item Field Definition Id	A Foreign Key to the Definition table for an instance of type Item Field Definition.	bigint

#### Attribute/Column(s) of Entity/Table "Item Field Definition"

Physical Name	Name	Definition	Physical Data Type
item_field_defn_id	Item Field Definition Id	A Foreign Key to the Definition table for an instance of type Definition.	bigint
item_defn_id	Item Definition Id	A Foreign Key to the Definition table for an instance of type Definition.	bigint
field_defn_id	Field Definition Id	A Foreign Key to the Definition table.	bigint

#### ER Diagram(s) of Subject Area "060 - Role"

Name	Physical Display Level
Logical	Column
Physical	Column

#### Entity/Table(s) of Subject Area "060 - Role"

Physical Name	Name	Logical Only	Do Not Generate	Definition
---------------	------	--------------	-----------------	------------

admin	Administrator CMDB Role	true	true	An Administrator CMDB Role.
admin_type	Administrator Type		false	The administrator type indicates whether the instance is a system role or not. This is important, since certain preferences cannot be changed on the system administrator role. Values include SYSTEM, WORKSPACE, AGENT.
agent_role	Agent Role	true	true	A role that identifies a user as an agent, and also provides more specific information about the application behavior for the agent.
app_role	Application Role		false	A User Security Role within Footprints. This includes Agent Users and Customer Users, and includes Role settings at the Instance and Container level. Workspace Role and CMDB Role may both be implemented as Container Role. The Role has associated Rights and Permissions, and is a way of granting Roles and Permissions to a User. There is some conceptual overlap between the User Profile and the Role. The User is assigned a single Role as part of the User Profile, and a User can only have one System Role and one Workspace Role. The User Profile can have property settings that override the Role Settings.
cmdb_role	CMDB Role	true	true	A role in the CMDB container
cfg_setting_val	Configuration Setting Value		false	Configuration Setting Value is an association between the Configuration Setting entity and a container entity (i.e. User Profile, Role, and Definition). This association contains a distinct value for each of the intersection points between the two entities.
create_edit	Create Edit CMDB Role	true	true	A Create Edit CMDB Role
custom	Custom CMDB Role	true	true	A custom CMDB role.
cust_read_knowledge_base_and_r	Customer Read Knowledge Base and Requests Role	true	true	A built in role that identifies a user as a customer, and also provides more specific information about the application behavior for the customer. Generally it specifically grants authorization for certain functions.
cust_read_knowledge_base_role	Customer Read Knowledge Base Role	true	true	A built in role that identifies a user as a customer, and also provides more specific information about the application behavior for the customer. Generally it specifically grants authorization for certain functions.
cust_read_submit_role	Customer Read/Submit Role	true	true	A built in role that identifies a user as a customer, and also provides more specific information about the application behavior for the customer. Generally it specifically grants authorization for certain functions.
cust_read_submit_edit_role	Customer Read/Submit/Edit Role	true	true	A built in role that identifies a user as a customer, and also provides more specific information about the application behavior for the customer. Generally it specifically grants authorization for certain functions.
cust_role	Customer Role	true	true	A role that identifies a user as a customer, and also provides more specific information about the application behavior for the customer.
defn	Definition		false	A definition holds the structural and behavioral information needed for an application element that is configurable by the system designer and/or the end user at definition time. Current Footprints uses a combination of directories, files, tables, and programming code to represent this information. The new FootPrints will provide all of the same information within the definition. The definition can be persisted in a structure that is isolated at the Instance Level. The data for the definition will be stored as Configuration Settings associated with the Definition Version. See the database views v_defn_ver_active, v_cntrn_item_field.

defn_subtype	Definition Subtype		false	The subtype of the definition used as a discriminator for the class of definition. Each subtype will have different configuration settings and associated definitions. These include associated built-in field definitions for item subtypes. See the database view <code>v_defn_subtype</code> .
read_only	Read Only CMDB Role	true	true	A Read Only CMDB Role.
role_grp	Role Group		false	(Not used, but referenced in Java code that must be removed before table deletion.) The role group will contain the roles that the user has for each of the workspaces that the user has access. The user will always have a role group even if the administrator chooses the Perl Footprints functionality.
role_grp_role	Role Group Role		false	(Not used, but referenced in Java code that must be removed before table deletion.) an association table that resolves the many to many relationship between the Role Group and the Role entities.
sys_admin_role	System Administrator Role	true	true	A built in role that identifies a user as a system administrator, and also provides more specific information about the application behavior for the agent.
tmpl_app_role	Template Application Role	true	true	The role template will function as a system predefined role that can then be linked to workspace roles. Changes made to the system role will be reflected at the workspace level until the connection is broken. This system role's connection to the workspace role will be in effect until any change is made to the workspace role. It then gets copied over as a stand alone role. 12/8/2011 The Role Template will be implemented as a subtype of role.
user_defined_agent_role	User Defined Agent Role	true	true	A role that identifies a user as an agent in the FootPrints system, and also provides more specific information about the application behavior for the agent. This is configured by the user or is a copy of a pre-existing role.
user_defined_cust_role	User Defined Customer Role	true	true	A role that identifies a user as a customer in the FootPrints system, and also provides more specific information about the application behavior for the agent. This is configured by the user or is a copy of a pre-existing customer role.
user_prof	User Profile		false	A container for Permissions and Settings (the composite parts are the associated Role and the User Preferences). The permissions for a User to access a particular container according to the capabilities defined. In most cases this will be more defined in a subcategory.  Example: Shared User Profile This profile will be shared between multiple users that access the system with the same User ID.
user_type	User Type		false	Indicates whether this entity is of an agent type or a customer type. Also CMDB and GUEST.
wksp_admin_role	Workspace Administrator Role	true	true	A built in role that identifies a user as a workspace administrator, and also provides more specific information about the application behavior for the agent.

#### Attribute/Column(s) of Entity/Table "Administrator CMDB Role"

Physical Name	Name	Definition	Physical Data Type
app_role_id	Application Role Id	A Foreign Key to the Application Role table.	bigint

#### Attribute/Column(s) of Entity/Table "Administrator Type"

Physical Name	Name	Definition	Physical Data Type
admin_type_name	Administrator Type Name	The name of the Administrator Type.	varchar(255)
admin_type_id	Administrator Type Id	The automatically generated primary key of the admin_type table.	int

#### Attribute/Column(s) of Entity/Table "Agent Role"

Physical Name	Name	Definition	Physical Data Type
---------------	------	------------	--------------------

admin_type_id	Administrator Type Id	A Foreign Key to the Administrator Type table.	int
app_role_id	Application Role Id	A Foreign Key to the Application Role table.	bigint

#### Attribute/Column(s) of Entity/Table "Application Role"

Physical Name	Name	Definition	Physical Data Type
app_role_id	Application Role Id	The automatically generated primary key of the app_role table.	bigint
tmpl_app_role_id	Template Application Role Id	A Foreign Key to the Application Role table. This indicates which Template Role was used to create a built-in role in case some features concerning new properties added to later release of the system will need to be added to existing templates.	bigint
app_role_name	Application Role Name	The name of the Application Role.	varchar(255)
defn_id	Definition Id	A Foreign Key to the Definition table.	bigint
admin_type_id	Administrator Type Id	A Foreign Key to the Administrator Type table.	int
is_built_in	Is Built In	Indicates that this instance of Application Role is a built-in instance that cannot be changed by the end user.	bit
app_role_descr	Application Role Description	A descriptive name of the Application Role. This descriptive name may be localized.	varchar(1024)
user_type_id	User Type Id	A Foreign Key to the User Type table.	int
defn_subtype_id	Definition Subtype Id	A Foreign Key to the Definition Subtype table.	int
tmpl_app_role_id	Template Application Role Id	A Foreign Key to the Application Role table.	bigint
tmpl_level_name	tmpl_level_name	The value of a Built_in_level, stored as a string.	varchar(255)

#### Attribute/Column(s) of Entity/Table "CMDB Role"

Physical Name	Name	Definition	Physical Data Type
app_role_id	Application Role Id	A Foreign Key to the Application Role table.	bigint

#### Attribute/Column(s) of Entity/Table "Configuration Setting Value"

Physical Name	Name	Definition	Physical Data Type
cfg_setting_val_id	Configuration Setting Value Id	The automatically generated primary key of the cfg_setting_val table.	bigint
cfg_setting_id	Configuration Setting Id	A Foreign Key to the Configuration Setting table.	bigint
val_type_cd	Value Type Code	The data type for a Configuration Setting Value. The choices are 'S' string, 'D' date, 'T' text, 'I' integer.	char(1)
string_val	String Value	The value of a Configuration Setting Value, stored as a string.	nvarchar(4000)
text_val	Text Value	The value of a Configuration Setting Value, stored as a long text object.	nvarchar(max)
real_val	Real Value	The value of a Configuration Setting Value stored as a real.	numeric(38,20)
long_val	Long Value	The value of a Configuration Setting Value stored as Long.	bigint
date_val	Date Value	The value of a Configuration Setting Value, stored as a date.	datetime
boolean_val	Boolean Value	The value of a Configuration Setting Value stored as Boolean.	bit
int_val	Int Value	The value of a Configuration Setting Value stored as Int.	int
owner_id	Owner Id	A Foreign Key to the Owning instance in either the Definition, Definition Version, Role, or User Profile tables.	bigint
val_map_key	Value Map Key	The value of a Configuration Setting Value, stored as a string.	varchar(255)
cfg_setting_owner_type_name	Configuration Setting Owner Type Name	The unique string identifier for an instance of Configuration Setting Owner Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)

#### Attribute/Column(s) of Entity/Table "Create Edit CMDB Role"

Physical Name	Name	Definition	Physical Data Type
---------------	------	------------	--------------------

app_role_id	Application Role Id	A Foreign Key to the Application Role table.	bigint
-------------	---------------------	--	--------

#### Attribute/Column(s) of Entity/Table "Custom CMDB Role"

Physical Name	Name	Definition	Physical Data Type
app_role_id	Application Role Id	A Foreign Key to the Application Role table.	bigint

#### Attribute/Column(s) of Entity/Table "Customer Read Knowledge Base and Requests Role"

Physical Name	Name	Definition	Physical Data Type
app_role_id	Application Role Id	A Foreign Key to the Application Role table.	bigint

#### Attribute/Column(s) of Entity/Table "Customer Read Knowledge Base Role"

Physical Name	Name	Definition	Physical Data Type
app_role_id	Application Role Id	A Foreign Key to the Application Role table.	bigint

#### Attribute/Column(s) of Entity/Table "Customer Read/Submit Role"

Physical Name	Name	Definition	Physical Data Type
app_role_id	Application Role Id	A Foreign Key to the Application Role table.	bigint

#### Attribute/Column(s) of Entity/Table "Customer Read/Submit/Edit Role"

Physical Name	Name	Definition	Physical Data Type
app_role_id	Application Role Id	A Foreign Key to the Application Role table.	bigint

#### Attribute/Column(s) of Entity/Table "Customer Role"

Physical Name	Name	Definition	Physical Data Type
app_role_id	Application Role Id	A Foreign Key to the Application Role table.	bigint

#### Attribute/Column(s) of Entity/Table "Definition"

Physical Name	Name	Definition	Physical Data Type
defn_id	Definition Id	The automatically generated primary key of the defn table.	bigint
defn_name	Definition Name	The name of the Definition.	nvarchar(4000)
dsble_date	Disabled Date	The date that this instance of Definition is logically deleted and disabled.	datetime
defn_guid	Definition GUID	A Global Unique Identifier search key for the Definition, composed of letters and numbers. It is computed by the client and typically used in unstructured data.	varchar(50)
defn_type_name	Definition Type Name	The unique string identifier for an instance of Definition Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)
defn_subtype_id	Definition Subtype Id	A Foreign Key to the Definition Subtype table.	int
tmpl_export_guid	Template Export GUID	The value of a Definition, stored as a string.	VARCHAR(50)

#### Attribute/Column(s) of Entity/Table "Definition Subtype"

Physical Name	Name	Definition	Physical Data Type
defn_subtype_name	Definition Subtype Name	The name of the Definition Subtype.	varchar(255)
defn_subtype_descr	Definition Subtype Description	A descriptive name of the Definition Subtype. This descriptive name may be localized.	varchar(512)
defn_subtype_id	Definition Subtype Id	The automatically generated primary key of the defn_subtype table.	int
defn_parent_subtype_id	definition parent subtype id	A Foreign Key to the Definition Subtype table.	int
defn_type_name	Definition Type Name	The unique string identifier for an instance of Definition Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)

max_in_parent	Maximum In Parent	A Foreign Key to the Definition Subtype table.	int
---------------	-------------------	--	-----

#### Attribute/Column(s) of Entity/Table "Read Only CMDB Role"

Physical Name	Name	Definition	Physical Data Type
app_role_id	Application Role Id	A Foreign Key to the Application Role table.	bigint

#### Attribute/Column(s) of Entity/Table "Role Group"

Physical Name	Name	Definition	Physical Data Type
role_grp_id	Role Group Id	The automatically generated primary key of the role_grp table.	bigint
role_grp_name	Role Group Name	The name of the Role Group.	varchar(255)
role_grp_descr	Role Group Description	A descriptive name of the Role Group. This descriptive name may be localized.	varchar(512)

#### Attribute/Column(s) of Entity/Table "Role Group Role"

Physical Name	Name	Definition	Physical Data Type
app_role_id	Application Role Id	A Foreign Key to the Application Role table.	bigint
role_grp_role_id	Role Group Role Id	The automatically generated primary key of the role_grp_role table.	bigint
role_grp_id	Role Group Id	A Foreign Key to the Role Group table.	bigint

#### Attribute/Column(s) of Entity/Table "System Administrator Role"

Physical Name	Name	Definition	Physical Data Type
app_role_id	Application Role Id	A Foreign Key to the Application Role table.	bigint

#### Attribute/Column(s) of Entity/Table "Template Application Role"

Physical Name	Name	Definition	Physical Data Type
app_role_id	Application Role Id	A Foreign Key to the Application Role table.	bigint

#### Attribute/Column(s) of Entity/Table "User Defined Agent Role"

Physical Name	Name	Definition	Physical Data Type
app_role_id	Application Role Id	A Foreign Key to the Application Role table.	bigint

#### Attribute/Column(s) of Entity/Table "User Defined Customer Role"

Physical Name	Name	Definition	Physical Data Type
app_role_id	Application Role Id	A Foreign Key to the Application Role table.	bigint

#### Attribute/Column(s) of Entity/Table "User Profile"

Physical Name	Name	Definition	Physical Data Type
user_prof_id	User Profile Id	The automatically generated primary key of the user_prof table.	bigint
tmpl_user_prof_id	Template User Profile Id	A Foreign Key to the User Profile table. This will not be a navigable key in the object model. This indicates which Template User Profile was used to create a built-in template in case some features concerning new properties added to later release of the system will need to be added to existing templates.	bigint
role_grp_id	Role Group Id	A Foreign Key to the Role Group table. A system level user profile can have a role group, which will have the effect of assigning a collection of workspace, cmdb, and address book roles at the container role level.	bigint
defn_id	Definition Id	A Foreign Key to the Definition table. Optionally, the container that owns the profile.	bigint
app_role_id	Application Role Id	A Foreign Key to the Application Role table. The User Profile is the means for setting the default role at the system and workspace levels, and is also the role of a Unique User Profile tied to a application user.	bigint
agent_authcn_type_id	Agent Authentication Type Id	A Foreign Key to the Agent Authentication Type table.	int
user_prof_name	User Profile Name	The name of the User Profile.	varchar(255)
is_built_in	Is Built In	Indicates that this instance of User Profile is a built-in instance that cannot be changed by the end user.	bit

user_type_id	User Type Id	A Foreign Key to the User Type table.	int
cust_authcn_type_id	Customer Authentication Type Id	A Foreign Key to the Customer Authentication Type table.	int
is_sys	Is System	Indicates if this is a system level profile.	bit
defn_subtype_id	Definition Subtype Id	A Foreign Key to the Definition Subtype table.	int
delete_date	Delete Date	The value of a User Profile, stored as a date.	datetime
last_login	Last Login	date time of last login in UTC.	datetime
tmpl_level_name	tmpl_level_name	The value of a Built_in_level, stored as a string.	varchar(255)

#### Attribute/Column(s) of Entity/Table "User Type"

Physical Name	Name	Definition	Physical Data Type
user_type_name	User Type Name	The name of the User Type.	varchar(255)
user_type_id	User Type Id	The automatically generated primary key of the user_type table.	int
user_type_descr	User Type Description	A descriptive name of the User Type. This descriptive name may be localized.	varchar(512)

#### Attribute/Column(s) of Entity/Table "Workspace Administrator Role"

Physical Name	Name	Definition	Physical Data Type
app_role_id	Application Role Id	A Foreign Key to the Application Role table.	bigint

#### ER Diagram(s) of Subject Area "065 - Configuration Setting"

Name	Physical Display Level
Logical	Column
Physical	Column

#### Entity/Table(s) of Subject Area "065 - Configuration Setting"

Physical Name	Name	Logical Only	Do Not Generate	Definition
app_role	Application Role		false	A User Security Role within Footprints. This includes Agent Users and Customer Users, and includes Role settings at the Instance and Container level. Workspace Role and CMDB Role may both be implemented as Container Role. The Role has associated Rights and Permissions, and is a way of granting Roles and Permissions to a User. There is some conceptual overlap between the User Profile and the Role. The User is assigned a single Role as part of the User Profile, and a User can only have one System Role and one Workspace Role. The User Profile can have property settings that override the Role Settings.
cfg_allow_val	Configuration Allowed Value		false	A Configuration Allowed Value represents a member of a list of possible values.
cfg_area_type	Configuration Area Type		false	The lookup table for the high level area of Configuration. Values include APPLICATION, BUSINESS, CORE, DEPLOYMENT, INFRASTRUCTURE, TOOL
cfg_ctgy	Configuration Category		false	A Category of Configuration, which corresponds to the namespace and name of an object that defines and contains the associated Configuration Settings. The intent is that the Category provides both a logical and physical grouping of configuration data so that developers working on different components do not have to compete with each other to check out the same files to define configuration, and do not have to merge changes.



cfg_ctgy_type	Configuration Category Type		false	The category of configuration lookup table. The category is like a namespace, it identifies where a setting is defined, and can be used to retrieve groups of settings via the API. The categories are defined in the Java code, but they are replicated to the Database for data integrity of the settings. They also serve to identify all possible categories, whether or not configuration settings are stored for them. Examples of the 70+ values include APPLICATION_HELP, BUSINESS_ADMIN_PIM_SYNCING, BUSINESS_ANALYTICS.
cfg_list_type	Configuration List Type		false	The type of list used for valid values and value lists of Configuration Settings. A single valued list, a list with X dimension, and a list with XY dimension are possible values. The Dimensions are used to construct the dynamic list based on dynamic key parts such as the Item Field Name, Item Status, Template Name, etc.
cfg_obj	Configuration Object		false	A configuration object holds configuration settings but is not versioned and published like a definition and definition version.
cfg_setting	Configuration Setting		false	A configuration variable that can be identified in programming code and which has an atomic value that affects the behavior of the application. Configuration values are either scalars, or are supported by value lists that also can be programmatically identified. They have localized labels and value choices, and are categorized to support organization and separation in programming namespaces and configuration screens. They support the concept of Built-In Values, Default Values, and Customized Values. Configuration should be supported by an Audit Trail, and should be able to determine if configuration has changed by level, category, and for the individual value in order to support cacheing. A configuration value has a definition to identify its usage, and a enum that identifies if a change will require a restart and what level of restart is required if the value changes. If a restart is required, both the currently running value and the value after restart should be accessible. This means that historical values of the Configuration Settings should be accessible.
cfg_setting_owner_type	Configuration Setting Owner Type		false	A lookup table for Owner Types of Configuration Setting Values. Values include: APPLICATION_USER, DEFINITION_IMPORTS_ROLE, SIMPLE_CONFIGURABLE_OBJECT, USER_PROFILE.
cfg_setting_val	Configuration Setting Value		false	Configuration Setting Value is an association between the Configuration Setting entity and a container entity (i.e. User Profile, Role, and Definition). This association contains a distinct value for each of the intersection points between the two entities.
cfg_val_list	Configuration Value List	false	false	The Configuration Value List is a class that represents a set of values. It is meant to be inherited by a sub category with the structure of the set of values more concretely defined in the lower entity.  Example: Configuration Dynamic List The Configuration Dynamic List inherits from the Configuration Value List but maps the values into a linear list so that the values can be identified by their position in the overall list of values.

dfit_cfg_setting_val	Default Configuration Setting Value		false	
defn	Definition		false	A definition holds the structural and behavioral information needed for an application element that is configurable by the system designer and/or the end user at definition time. Current Footprints uses a combination of directories, files, tables, and programming code to represent this information. The new FootPrints will provide all of the same information within the definition. The definition can be persisted in a structure that is isolated at the Instance Level. The data for the definition will be stored as Configuration Settings associated with the Definition Version. See the database views v_defn_ver_active, v_cntrn_item_field.
defn_ver	Definition Version		false	A version of a Definition with a version identifier that changes when data changes are made to the template. The definition version will suffice to store the current state of the FootPrints Configuration. A full copy of the definition tree is versioned when a container is published. See the database views v_defn_ver_active, v_cntrn_item_field.
tmp_cfg_setting_val	tmp_cfg_setting_val	false	false	
user_prof	User Profile		false	A container for Permissions and Settings (the composite parts are the associated Role and the User Preferences). The permissions for a User to access a particular container according to the capabilities defined. In most cases this will be more defined in a subcategory.  Example: Shared User Profile This profile will be shared between multiple users that access the system with the same User ID.
val_type_cd	Value Type Code		false	A lookup table for the data value type discriminator. Many tables share this discriminator for name value pair datatypes. Values include: B Boolean D Timestamp - Date + Time I Integer L Long R Real Number S String T Text - long string

#### Attribute/Column(s) of Entity/Table "Application Role"

Physical Name	Name	Definition	Physical Data Type
app_role_id	Application Role Id	The automatically generated primary key of the app_role table.	bigint
tmpl_app_role_id	Template Application Role Id	A Foreign Key to the Application Role table. This indicates which Template Role was used to create a built-in role in case some features concerning new properties added to later release of the system will need to be added to existing templates.	bigint
app_role_name	Application Role Name	The name of the Application Role.	varchar(255)
defn_id	Definition Id	A Foreign Key to the Definition table.	bigint
admin_type_id	Administrator Type Id	A Foreign Key to the Administrator Type table.	int
is_built_in	Is Built In	Indicates that this instance of Application Role is a built-in instance that cannot be changed by the end user.	bit
app_role_descr	Application Role Description	A descriptive name of the Application Role. This descriptive name may be localized.	varchar(1024)
user_type_id	User Type Id	A Foreign Key to the User Type table.	int
defn_subtype_id	Definition Subtype Id	A Foreign Key to the Definition Subtype table.	int
tmpl_app_role_id	Template Application Role Id	A Foreign Key to the Application Role table.	bigint
tmpl_level_name	tmpl_level_name	The value of a Built_in_level, stored as a string.	varchar(255)

#### Attribute/Column(s) of Entity/Table "Configuration Allowed Value"

Physical Name	Name	Definition	Physical Data Type
---------------	------	------------	--------------------

cfg_allow_val_id	Configuration Allowed Value Id	The automatically generated primary key of the cfg_allow_val table.	bigint
cfg_val_list_id	Configuration Value List Id	A Foreign Key to the Configuration Value List table.	bigint
string_val	String Value	The value of a Configuration Allowed Value, stored as a string.	varchar(255)
text_val	Text Value	The value of a Configuration Allowed Value, stored as a long text object.	varchar(max)
val_type_cd	Value Type Code	The data type for a Value Type Code. The choices are 'S' string, 'D' date, 'I' text, 'I' integer.	char(1)
real_val	Real Value	The value of a Configuration Allowed Value stored as real.	numeric(38,20)
long_val	Long Value	The value of a Configuration Allowed Value stored as Long.	bigint
cfg_allow_val_descr	Configuration Allowed Value Description	A descriptive name of the Configuration Allowed Value. This descriptive name may be localized.	varchar(512)
date_val	Date Value	The value of a Configuration Allowed Value, stored as a date.	datetime
boolean	Boolean	The value of a Configuration Allowed Value stored as boolean.	bit
int_val	Int Value	The value of a Configuration Allowed Value stored as boolean.	int

#### Attribute/Column(s) of Entity/Table "Configuration Area Type"

Physical Name	Name	Definition	Physical Data Type
cfg_area_type_name	Configuration Area Type Name	The unique string identifier for an instance of Configuration Area Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(100)
cfg_area_type_descr	Configuration Area Type Description	A descriptive name of the Configuration Area Type. This descriptive name may be localized.	varchar(512)

#### Attribute/Column(s) of Entity/Table "Configuration Category"

Physical Name	Name	Definition	Physical Data Type
cfg_ctgy_id	Configuration Category Id	The automatically generated primary key of the cfg_ctgy table.	bigint
dflt_ctgy_descr	Default Category Description	A description of the category.	varchar(255)
cfg_area_type_name	Configuration Area Type Name	The unique string identifier for an instance of Configuration Area Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(100)
cfg_ctgy_type_name	Configuration Category Type Name	The unique string identifier for an instance of Configuration Category Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(100)

#### Attribute/Column(s) of Entity/Table "Configuration Category Type"

Physical Name	Name	Definition	Physical Data Type
cfg_ctgy_type_name	Configuration Category Type Name	The unique string identifier for an instance of Configuration Category Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(100)
cfg_ctgy_type_descr	Configuration Category Type Description	A descriptive name of the Configuration Category Type. This descriptive name may be localized.	varchar(512)

#### Attribute/Column(s) of Entity/Table "Configuration List Type"

Physical Name	Name	Definition	Physical Data Type
cfg_list_type_name	Configuration List Type Name	The unique string identifier for an instance of Configuration List Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)

cfg_list_type_descr	Configuration List Type Description	A descriptive name of the Configuration List Type. This descriptive name may be localized.	varchar(512)
---------------------	-------------------------------------	--	--------------

#### Attribute/Column(s) of Entity/Table "Configuration Object"

Physical Name	Name	Definition	Physical Data Type
cfg_obj_id	Configuration Object Id	The automatically generated primary key of the cfg_obj table.	bigint
cfg_obj_type	Configuration Object Type	The type of a Configuration Object, stored as a string. Values include: AUTHENTICATION_METHOD, AUTHENTICATION_METHOD, SYSTEM_PREFERENCES_CONFIGURATION, FILE_STORAGE_CONFIGURATION, LICENSING_CONFIGURATION, RULE_CONFIGURATION.	varchar(255)
order_index	Order Index	deprecated and null. The value of a Configuration Object. The value is a string but it can be conformed to the datatype listed on the Configuration Setting.	int
cfg_obj_subtype	Configuration Object Subtype	deprecated and null. The value of a Configuration Object, stored as a string.	varchar(255)
owner_id	Owner Id	A Foreign Key to the Owning instance in either the Definition, Configuration Object, or User Profile tables.	bigint
cfg_setting_owner_type_name	Configuration Setting Owner Type Name	deprecated and null. The unique string identifier for an instance of Configuration Setting Owner Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)

#### Attribute/Column(s) of Entity/Table "Configuration Setting"

Physical Name	Name	Definition	Physical Data Type
cfg_ctgy_id	Configuration Category Id	A Foreign Key to the Configuration Category table.	bigint
cfg_setting_const	Configuration Setting Constant	The constant name of the Configuration Setting Constant, which can be aligned with an enumeration or other compile time constant value for access in Java.	varchar(255)
dflt_setting_label	Default Setting Label	The name of the Configuration Setting.	varchar(255)
dflt_setting_descr	Default Setting Description	The description of the purpose and usage of the configuration setting. This description serves for documentation, and will be used to help develop UI strings for the Label, ToolTip, HelpText etc. of the UI for the configuration setting.	varchar(255)
cfg_setting_id	Configuration Setting Id	The automatically generated primary key of the cfg_setting table.	bigint
agent_only	Agent Only	Indicates if Agent Only.	bit
unique_user_only	Unique User Only	Indicates if Unique User Only.	bit
cfg_val_list_id	Configuration Value List Id	A Foreign Key to the Configuration Value List table.	bigint
val_type_cd	Value Type Code	The data type for a Configuration Setting. The choices are 'S' string, 'D' date, 'T' text, 'L' Long, 'R' real, and 'A' for any type.	char(1)
cfg_list_type_name	Configuration List Type Name	The unique string identifier for an instance of Configuration List Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)
spec_class_name	Specification Class Name	The value of a Configuration Setting, stored as a string.	varchar(255)
cfg_is_val_mapped	Configuration Is Value Mapped	Indicates if Configuration Is Value Mapped.	bit

#### Attribute/Column(s) of Entity/Table "Configuration Setting Owner Type"

Physical Name	Name	Definition	Physical Data Type
cfg_setting_owner_type_descr	Configuration Setting Owner Type Description	A descriptive name of the Configuration Setting Owner Type. This descriptive name may be localized.	varchar(512)
cfg_setting_owner_type_name	Configuration Setting Owner Type Name	The unique string identifier for an instance of Configuration Setting Owner Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)

#### Attribute/Column(s) of Entity/Table "Configuration Setting Value"

Physical Name	Name	Definition	Physical Data Type
---------------	------	------------	--------------------

cfg_setting_val_id	Configuration Setting Value Id	The automatically generated primary key of the cfg_setting_val table.	bigint
cfg_setting_id	Configuration Setting Id	A Foreign Key to the Configuration Setting table.	bigint
val_type_cd	Value Type Code	The data type for a Configuration Setting Value. The choices are 'S' string, 'D' date, 'T' text, 'I' integer.	char(1)
string_val	String Value	The value of a Configuration Setting Value, stored as a string.	nvarchar(4000)
text_val	Text Value	The value of a Configuration Setting Value, stored as a long text object.	nvarchar(max)
real_val	Real Value	The value of a Configuration Setting Value stored as a real.	numeric(38,20)
long_val	Long Value	The value of a Configuration Setting Value stored as Long.	bigint
date_val	Date Value	The value of a Configuration Setting Value, stored as a date.	datetime
boolean_val	Boolean Value	The value of a Configuration Setting Value stored as Boolean.	bit
int_val	Int Value	The value of a Configuration Setting Value stored as Int.	int
owner_id	Owner Id	A Foreign Key to the Owning instance in either the Definition, Definition Version, Role, or User Profile tables.	bigint
val_map_key	Value Map Key	The value of a Configuration Setting Value, stored as a string.	varchar(255)
cfg_setting_owner_type_name	Configuration Setting Owner Type Name	The unique string identifier for an instance of Configuration Setting Owner Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)

#### Attribute/Column(s) of Entity/Table "Configuration Value List"

Physical Name	Name	Definition	Physical Data Type
cfg_val_list_id	Configuration Value List Id	The automatically generated primary key of the cfg_val_list table.	bigint
cfg_list_type_name	Configuration List Type Name	The unique string identifier for an instance of Configuration List Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)

#### Attribute/Column(s) of Entity/Table "Default Configuration Setting Value"

Physical Name	Name	Definition	Physical Data Type
dflt_cfg_setting_val_id	Default Configuration Setting Value Id	The automatically generated primary key of the dflt_cfg_setting_val table.	bigint
cfg_setting_id	Configuration Setting Id	A Foreign Key to the Configuration Setting table.	bigint
val_type_cd	Value Type Code	The data type for a Value Type Code. The choices are 'S' string, 'D' date, 'T' text, 'I' integer.	char(1)
string_val	String Value	The value of a Default Configuration Setting Value, stored as a string.	varchar(255)
text_val	Text Value	The value of a Default Configuration Setting Value, stored as a long text object.	varchar(max)
date_val	Date Value	The value of a Default Configuration Setting Value, stored as a date.	datetime
real_val	Real Value	The value of a Default Configuration Setting Value stored as real.	numeric(38,20)
int_val	Int Value	The value of a Default Configuration Setting Value stored as boolean.	int
long_val	Long Value	The value of a Default Configuration Setting Value stored as Long.	bigint
boolean_val	Boolean Value	The value of a Configuration Setting Value stored as boolean.	bit
val_map_key	Value Map Key	The value of a Default Configuration Setting Value, stored as a string.	varchar(255)
cfg_setting_owner_type_name	Configuration Setting Owner Type Name	The unique string identifier for an instance of Configuration Setting Owner Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)

#### Attribute/Column(s) of Entity/Table "Definition"

Physical Name	Name	Definition	Physical Data Type
---------------	------	------------	--------------------

defn_id	Definition Id	The automatically generated primary key of the defn table.	bigint
defn_name	Definition Name	The name of the Definition.	nvarchar(4000)
dsble_date	Disabled Date	The date that this instance of Definition is logically deleted and disabled.	datetime
defn_guid	Definition GUID	A Global Unique Identifier search key for the Definition, composed of letters and numbers. It is computed by the client and typically used in unstructured data.	varchar(50)
defn_type_name	Definition Type Name	The unique string identifier for an instance of Definition Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)
defn_subtype_id	Definition Subtype Id	A Foreign Key to the Definition Subtype table.	int
tmpl_export_guid	Template Export GUID	The value of a Definition, stored as a string.	VARCHAR(50)

#### Attribute/Column(s) of Entity/Table "Definition Version"

Physical Name	Name	Definition	Physical Data Type
defn_id	Definition Id	A Foreign Key to the Definition table.	bigint
defn_ver_id	Definition Version Id	The automatically generated primary key of the defn_ver table.	bigint
defn_ver_name	Definition Version Name	The name of the Definition Version.	nvarchar(4000)
defn_ver_status_id	Definition Version Status Id	A Foreign Key to the Definition Version Status table.	int
defn_ver_guid	Definition Version GUID	A Global Unique Identifier search key for the Definition Version, composed of letters and numbers. It is computed by the client and typically used in unstructured data.	varchar(50)

#### Attribute/Column(s) of Entity/Table "tmp\_cfg\_setting\_val"

Physical Name	Name	Definition	Physical Data Type
cfg_setting_val_id	cfg_setting_val_id		BIGINT
cfg_setting_owner_type_name	cfg_setting_owner_type_name	cfg_setting_owner_type_name.	VARCHAR(255)
cfg_setting_id	cfg_setting_id		BIGINT
owner_id	owner_id		BIGINT
val_type_cd	val_type_cd	val_type_cd.	CHAR(1)
string_val	string_val	string_val.	nvarchar(4000)
text_val	text_val	text_val.	nvarchar(max)
date_val	date_val		datetime
real_val	real_val		NUMERIC(38,20)
int_val	int_val		int
long_val	long_val		BIGINT
boolean_val	boolean_val		BIT
val_map_key	val_map_key	val_map_key.	VARCHAR(255)

#### Attribute/Column(s) of Entity/Table "User Profile"

Physical Name	Name	Definition	Physical Data Type
user_prof_id	User Profile Id	The automatically generated primary key of the user_prof table.	bigint
tmpl_user_prof_id	Template User Profile Id	A Foreign Key to the User Profile table. This will not be a navigable key in the object model. This indicates which Template User Profile was used to create a built-in template in case some features concerning new properties added to later release of the system will need to be added to existing templates.	bigint
role_grp_id	Role Group Id	A Foreign Key to the Role Group table. A system level user profile can have a role group, which will have the effect of assigning a collection of workspace, cmdb, and address book roles at the container role level.	bigint
defn_id	Definition Id	A Foreign Key to the Definition table. Optionally, the container that owns the profile.	bigint
app_role_id	Application Role Id	A Foreign Key to the Application Role table. The User Profile is the means for setting the default role at the system and workspace levels, and is also the role of a Unique User Profile tied to a application user.	bigint
agent_authcn_type_id	Agent Authentication Type Id	A Foreign Key to the Agent Authentication Type table.	int
user_prof_name	User Profile Name	The name of the User Profile.	varchar(255)
is_built_in	Is Built In	Indicates that this instance of User Profile is a built-in instance that cannot be changed by the end user.	bit

user_type_id	User Type Id	A Foreign Key to the User Type table.	int
cust_authcn_type_id	Customer Authentication Type Id	A Foreign Key to the Customer Authentication Type table.	int
is_sys	Is System	Indicates if this is a system level profile.	bit
defn_subtype_id	Definition Subtype Id	A Foreign Key to the Definition Subtype table.	int
delete_date	Delete Date	The value of a User Profile, stored as a date.	datetime
last_login	Last Login	date time of last login in UTC.	datetime
tmpl_level_name	tmpl_level_name	The value of a Built_in_level, stored as a string.	varchar(255)

#### Attribute/Column(s) of Entity/Table "Value Type Code"

Physical Name	Name	Definition	Physical Data Type
val_type_cd	Value Type Code	The data type for a Value Type Code. The choices are 'S' string, 'D' date, 'T' text, 'I' integer.	char(1)
val_type_name	Value Type Name	The name of the Value Type Code.	varchar(255)

#### ER Diagram(s) of Subject Area "066 - Container Snapshot"

Name	Physical Display Level
Logical	Column
Physical	Column

#### Entity/Table(s) of Subject Area "066 - Container Snapshot"

Physical Name	Name	Logical Only	Do Not Generate	Definition
cfg_setting	Configuration Setting		false	A configuration variable that can be identified in programming code and which has an atomic value that affects the behavior of the application. Configuration values are either scalars, or are supported by value lists that also can be programmatically identified. They have localized labels and value choices, and are categorized to support organization and separation in programming namespaces and configuration screens. They support the concept of Built-In Values, Default Values, and Customized Values. Configuration should be supported by an Audit Trail, and should be able to determine if configuration has changed by level, category, and for the individual value in order to support cacheing. A configuration value has a definition to identify its usage, and an enum that identifies if a change will require a restart and what level of restart is required if the value changes. If a restart is required, both the currently running value and the value after restart should be accessible. This means that historical values of the Configuration Settings should be accessible.

cfg_setting_val	Configuration Setting Value		false	Configuration Setting Value is an association between the Configuration Setting entity and a container entity (i.e. User Profile, Role, and Definition). This association contains a distinct value for each of the intersection points between the two entities.
cntnr	Container	true	true	The highest abstract level of Runtime Object. The Runtime object has a corresponding definition that acts as a catalog of the Runtime Entity. There is a hierarchy or composition concept in the Runtime Entity, but it may not be necessary to actually have navigable properties.
cntnr_defn	Container Definition	true	true	The definition of a Container and the associated Configuration Settings at the container level. This would include the collection of Item Definitions. The Container Definition will have a Container Class identifier (Workspace, CMDB, Address Book) and may have specializations such as Workspace Container Definition. As much as possible, the Container Definition will suffice to describe any Container Type. There will be an association to a default Container Template, as well as more application specific Container Templates that can be chosen to initially populate the Container Template. There will also be Container Group Definition, and Container Group Templates. See the database views v_cntnr_defn, v_cntnr_defn_active.
cntnr_snapshot	Container Snapshot		false	
defn	Definition		false	A definition holds the structural and behavioral information needed for an application element that is configurable by the system designer and/or the end user at definition time. Current Footprints uses a combination of directories, files, tables, and programming code to represent this information. The new FootPrints will provide all of the same information within the definition. The definition can be persisted in a structure that is isolated at the Instance Level. The data for the definition will be stored as Configuration Settings associated with the Definition Version. See the database views v_defn_ver_active, v_cntnr_item_field.
defn_lock	Definition Lock	true	true	A pessimistic lock of a definition. The lock helps implement single session editing. It is intended to be used for container definitions, but may be used by other definitions as well. This is not a table it is implemented with configuration settings.
defn_reln	Definition Relationship		false	All relationships between definitions are stored in this table in order to make the model able to handle new relationships without changes to tables.
defn_reln_type	Definition Relationship Type		false	The Definition Relationship Type entity depicts a set of relationships that a Definition instance in the system can have with another Definition instance (i.e. parent_child indicates a hierarchical relationship). This specific relationship is further clarified when combined with the Definition Relationship Member Type entity.
defn_ver	Definition Version		false	A version of a Definition with a version identifier that changes when data changes are made to the template. The definition version will suffice to store the current state of the FootPrints Configuration. A full copy of the definition tree is versioned when a container is published. See the database views v_defn_ver_active, v_cntnr_item_field.



defn_ver_association	Definition Version Association		false	
defn_ver_status	Definition Version Status		false	<p>The attribute of the definition version entity that tells what state the definition instance is in. The word "Published" in this context means a change being committed or made ready for runtime. Currently the available status options are:</p> <p>Draft Dirty - the entity instance is being created or changed, but has not yet been published, so is not yet valid for runtime.</p> <p>Published Active - the entity instance has been published, and is available for runtime.</p> <p>Published Outdated - the entity instance has been previously published, but has been replaced with a newer version of itself in runtime.</p>
dyn_table_script_history	Dynamic Table Script History		false	Records the successful execution of the Dynamic Table Script for a Container and/or Dynamic Item. This is used to control re-execution of scripts when the script version changes, or when a new script is added.
field_defn	Field Definition	true	true	The definition of a Field, including persistence information and default UI information. Some fields will be defined as database fields, and other fields will be defined as calculated fields. The field will have data types and associations to business rule definitions. Some fields are built-in, some are defaults from templates, and some are custom fields. See the database view v_cntrn_item_field.
snapshot_cfg_setting_val	Snapshot Configuration Setting Value		false	Configuration Setting Value is an association between the Configuration Setting entity and a container entity (i.e. User Profile, Role, and Definition). This association contains a distinct value for each of the intersection points between the two entities.
snapshot_defn	Snapshot Definition		false	
snapshot_defn_reln	Snapshot Definition Relationship		false	All relationships between definitions are stored in this table in order to make the model able to handle new relationships without changes to tables.
snapshot_defn_ver	Snapshot Definition Version		false	A version of a Definition with a version identifier that changes when data changes are made to the template. The definition version will suffice to store the current state of the FootPrints Configuration. A full copy of the definition tree is versioned when a container is published. See the database views v_defn_ver_active, v_cntrn_item_field.
snapshot_defn_ver_assoc	Snapshot Definition Version Assoc		false	
snapshot_status	Snapshot Status		false	
snapshot_type	Snapshot Type		false	
wksp_cntrn	Workspace Container	true	true	A workspace is the principle architectural structure in Numara FootPrints for storing and tracking information. A Workspace is a separate sub-database or work area within the system with its own fields, forms, options, workflow, settings and users. Most of the behaviour of the Workspace comes from the Container Abstract class, and only things that are specific to the Workspace are implemented here. For Example, Current FootPrints allows a single CMDDB to be referenced from a Workspace, and a single Address book, yet these are listed as features for improvement. So convenience methods that get and set Container Group Relationship might be appropriate.

#### Attribute/Column(s) of Entity/Table "Configuration Setting"

Physical Name	Name	Definition	Physical Data Type
---------------	------	------------	--------------------

cfg_ctgy_id	Configuration Category Id	A Foreign Key to the Configuration Category table.	bigint
cfg_setting_const	Configuration Setting Constant	The constant name of the Configuration Setting Constant, which can be aligned with an enumeration or other compile time constant value for access in Java.	varchar(255)
dflt_setting_label	Default Setting Label	The name of the Configuration Setting.	varchar(255)
dflt_setting_descr	Default Setting Description	The description of the purpose and usage of the configuration setting. This description serves for documentation, and will be used to help develop UI strings for the Label, ToolTip, HelpText etc. of the UI for the configuration setting.	varchar(255)
cfg_setting_id	Configuration Setting Id	The automatically generated primary key of the cfg_setting table.	bigint
agent_only	Agent Only	Indicates if Agent Only.	bit
unique_user_only	Unique User Only	Indicates if Unique User Only.	bit
cfg_val_list_id	Configuration Value List Id	A Foreign Key to the Configuration Value List table.	bigint
val_type_cd	Value Type Code	The data type for a Configuration Setting. The choices are 'S' string, 'D' date, 'T' text, 'L' Long, 'R' real, and 'A' for any type.	char(1)
cfg_list_type_name	Configuration List Type Name	The unique string identifier for an instance of Configuration List Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)
spec_class_name	Specification Class Name	The value of a Configuration Setting, stored as a string.	varchar(255)
cfg_is_val_mapped	Configuration Is Value Mapped	Indicates if Configuration Is Value Mapped.	bit

#### Attribute/Column(s) of Entity/Table "Configuration Setting Value"

Physical Name	Name	Definition	Physical Data Type
cfg_setting_val_id	Configuration Setting Value Id	The automatically generated primary key of the cfg_setting_val table.	bigint
cfg_setting_id	Configuration Setting Id	A Foreign Key to the Configuration Setting table.	bigint
val_type_cd	Value Type Code	The data type for a Configuration Setting Value. The choices are 'S' string, 'D' date, 'T' text, 'I' integer.	char(1)
string_val	String Value	The value of a Configuration Setting Value, stored as a string.	nvarchar(4000)
text_val	Text Value	The value of a Configuration Setting Value, stored as a long text object.	nvarchar(max)
real_val	Real Value	The value of a Configuration Setting Value stored as a real.	numeric(38,20)
long_val	Long Value	The value of a Configuration Setting Value stored as Long.	bigint
date_val	Date Value	The value of a Configuration Setting Value, stored as a date.	datetime
boolean_val	Boolean Value	The value of a Configuration Setting Value stored as Boolean.	bit
int_val	Int Value	The value of a Configuration Setting Value stored as Int.	int
owner_id	Owner Id	A Foreign Key to the Owning instance in either the Definition, Definition Version, Role, or User Profile tables.	bigint
val_map_key	Value Map Key	The value of a Configuration Setting Value, stored as a string.	varchar(255)
cfg_setting_owner_type_name	Configuration Setting Owner Type Name	The unique string identifier for an instance of Configuration Setting Owner Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)

#### Attribute/Column(s) of Entity/Table "Container"

Physical Name	Name	Definition	Physical Data Type
cntnr_name	Container Name	The name of the Container.	varchar(255)
cntnr_id	Container Id	The automatically generated primary key of the cntnr table.	bigint
tenant_parttn_id	Tenant Partition Id	A Foreign Key to the Tenant Partition table.	bigint

defn_id	Definition Id	A Foreign Key to the Definition table.	bigint
---------	---------------	--	--------

#### Attribute/Column(s) of Entity/Table "Container Definition"

Physical Name	Name	Definition	Physical Data Type
db_schema_name	Database Schema Name	The name of the database schema. It is a database object name, limited to 30 characters in length.	varchar(30)
defn_id	Definition Id	A Foreign Key to the Definition table.	bigint

#### Attribute/Column(s) of Entity/Table "Container Snapshot"

Physical Name	Name	Definition	Physical Data Type
cntnr_snapshot_id	Container Snapshot Id	The automatically generated primary key of the cntnr_snapshot table.	bigint
cntnr_defn_id	Container Definition Id	A Foreign Key to the Definition table.	bigint
create_date	Create Date	The date that the Container Snapshot was created.	datetime
snapshot_status_name	Snapshot Status Name	The unique string identifier for an instance of Snapshot Status. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)
last_good_publ_date	Last Good Publish Date		datetime

#### Attribute/Column(s) of Entity/Table "Definition"

Physical Name	Name	Definition	Physical Data Type
defn_id	Definition Id	The automatically generated primary key of the defn table.	bigint
defn_name	Definition Name	The name of the Definition.	nvarchar(4000)
dsble_date	Disabled Date	The date that this instance of Definition is logically deleted and disabled.	datetime
defn_guid	Definition GUID	A Global Unique Identifier search key for the Definition, composed of letters and numbers. It is computed by the client and typically used in unstructured data.	varchar(50)
defn_type_name	Definition Type Name	The unique string identifier for an instance of Definition Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)
defn_subtype_id	Definition Subtype Id	A Foreign Key to the Definition Subtype table.	int
tmpl_export_guid	Template Export GUID	The value of a Definition, stored as a string.	VARCHAR(50)

#### Attribute/Column(s) of Entity/Table "Definition Lock"

Physical Name	Name	Definition	Physical Data Type
locked_by	Locked By	a string description suitable for display of a session user or component that owns a lock.	varchar(255)
create_date	Create Date	The date that the Definition Lock was created.	datetime
defn_id	Definition Id	A Foreign Key to the Definition table.	bigint
token	Token	A Global Unique Identifier search key for the System Session, composed of letters and numbers. The session table is not persisted to the database for performance and integrity reasons. Locks will be cleared by the application in normal use, and may be cleared if they are orphaned.	varchar(255)

#### Attribute/Column(s) of Entity/Table "Definition Relationship"

Physical Name	Name	Definition	Physical Data Type
defn_reln_id	Definition Relationship Id	The automatically generated primary key of the defn_reln table.	bigint
from_defn_ver_id	From Definition Version Id	A Foreign Key to the Definition Version table.	bigint
to_defn_ver_id	To Definition Version Id	A Foreign Key to the Definition Version table.	bigint
defn_reln_type_id	Definition Relationship Type Id	A Foreign Key to the Definition Relationship Type table.	int

#### Attribute/Column(s) of Entity/Table "Definition Relationship Type"

Physical Name	Name	Definition	Physical Data Type
---------------	------	------------	--------------------

defn_reln_type_name	Definition Relationship Type Name	The name of the Definition Relationship Type.	varchar(255)
defn_reln_type_id	Definition Relationship Type Id	The automatically generated primary key of the defn_reln_type table.	int
defn_reln_type_descr	Definition Relationship Type Description	A descriptive name of the Definition Relationship Type. This descriptive name may be localized.	varchar(512)

#### Attribute/Column(s) of Entity/Table "Definition Version"

Physical Name	Name	Definition	Physical Data Type
defn_id	Definition Id	A Foreign Key to the Definition table.	bigint
defn_ver_id	Definition Version Id	The automatically generated primary key of the defn_ver table.	bigint
defn_ver_name	Definition Version Name	The name of the Definition Version.	nvarchar(4000)
defn_ver_status_id	Definition Version Status Id	A Foreign Key to the Definition Version Status table.	int
defn_ver_guid	Definition Version GUID	A Global Unique Identifier search key for the Definition Version, composed of letters and numbers. It is computed by the client and typically used in unstructured data.	varchar(50)

#### Attribute/Column(s) of Entity/Table "Definition Version Association"

Physical Name	Name	Definition	Physical Data Type
defn_ver_association_id	Definition Version Association Id	The automatically generated primary key of the defn_ver_association table.	bigint
from_defn_ver_id	From Definition Version ID	A Foreign Key to the Definition Version table.	bigint
to_defn_ver_id	To Definition Version ID	A Foreign Key to the Definition Version table.	bigint
association_type_id	Association Type Id	A Foreign Key to the Association Type table.	bigint

#### Attribute/Column(s) of Entity/Table "Definition Version Status"

Physical Name	Name	Definition	Physical Data Type
defn_ver_status_name	Definition Version Status Name	The name of the Definition Version Status.	varchar(255)
defn_ver_status_id	Definition Version Status Id	The automatically generated primary key of the defn_ver_status table.	int

#### Attribute/Column(s) of Entity/Table "Dynamic Table Script History"

Physical Name	Name	Definition	Physical Data Type
dyn_table_script_id	Dynamic Table Script Id	A Foreign Key to the Dynamic Table Script table.	bigint
dyn_table_script_history_id	Dynamic Table Script History Id	The automatically generated primary key of the dyn_table_script_history table.	bigint
item_defn_id	Item Definition Id	A Foreign Key to the Definition table for an instance of type Definition.	bigint
create_date	Create Date	The date that the Dynamic Table Script History was created.	datetime
cntr_defn_id	Container Definition Id	A Foreign Key to the Definition table.	bigint

#### Attribute/Column(s) of Entity/Table "Field Definition"

Physical Name	Name	Definition	Physical Data Type
field_dtype_id	Field Datatype Id	A Foreign Key to the Field Datatype table.	bigint
field_col_name	Field Column Name	The name of the DBMS_Object_Name. It is a database object name, limited to 30 characters in length.	varchar(30)
field_descr	Field Description	Field Description.	varchar(255)
field_defn_id	Field Definition Id	A Foreign Key to the Definition table.	bigint

#### Attribute/Column(s) of Entity/Table "Snapshot Configuration Setting Value"

Physical Name	Name	Definition	Physical Data Type
cfg_setting_val_id	Configuration Setting Value Id	The automatically generated primary key of the snapshot_cfg_setting_val table.	bigint
cfg_setting_id	Configuration Setting Id	A Foreign Key to the Snapshot Configuration Setting Value table.	bigint
val_type_cd	Value Type Code	The data type for a Snapshot Configuration Setting Value. The choices are 'S' string, 'D' date, 'T' text, 'I' integer.	char(1)

string_val	String Value	The value of a Snapshot Configuration Setting Value, stored as a string.	varchar(255)
text_val	Text Value	The value of a Snapshot Configuration Setting Value, stored as a long text object.	varchar(max)
real_val	Real Value	The value of a Configuration Setting Value stored as a real.	numeric(38,20)
long_val	Long Value	The value of a Configuration Setting Value stored as Long.	bigint
date_val	Date Value	The value of a Snapshot Configuration Setting Value, stored as a date.	datetime
boolean_val	Boolean Value	The value of a Configuration Setting Value stored as Boolean.	bit
int_val	Int Value	The value of a Configuration Setting Value stored as Int.	int
owner_id	Owner Id	A Foreign Key to the Owning instance in either the Definition, Definition Version, Role, or User Profile tables.	bigint
val_map_key	Value Map Key	The value of a Snapshot Configuration Setting Value, stored as a string.	varchar(255)
cfg_setting_owner_type_name	Configuration Setting Owner Type Name	The unique string identifier for an instance of Snapshot Configuration Setting Value. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)
cntnr_snapshot_id	Container Snapshot Id	A Foreign Key to the Container Snapshot table.	bigint

#### Attribute/Column(s) of Entity/Table "Snapshot Definition"

Physical Name	Name	Definition	Physical Data Type
cntnr_snapshot_id	Container Snapshot Id	A Foreign Key to the Container Snapshot table.	bigint
defn_id	Definition Id	A Foreign Key to the Definition table.	bigint
defn_name	Definition Name	The name of the Snapshot Definition.	nvarchar(4000)
dsble_date	Disabled Date	The date that this instance of Snapshot Definition is logically deleted and disabled.	datetime
defn_guid	Definition GUID	A Global Unique Identifier search key for the Snapshot Definition, composed of letters and numbers. It is computed by the client and typically used in unstructured data.	varchar(50)
defn_type_name	Definition Type Name	The unique string identifier for an instance of Snapshot Definition. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)
defn_subtype_id	Definition Subtype Id	A Foreign Key to the Snapshot Definition table.	int
tmpl_export_guid	Template Export GUID	The value of a Snapshot Definition, stored as a string.	VARCHAR(50)

#### Attribute/Column(s) of Entity/Table "Snapshot Definition Relationship"

Physical Name	Name	Definition	Physical Data Type
defn_reln_id	Definition Relationship Id	The automatically generated primary key of the snapshot_definition_relationsh table.	bigint
from_defn_ver_id	From Definition Version Id	A Foreign Key to the Snapshot Definition Relationship table.	bigint
to_defn_ver_id	To Definition Version Id	A Foreign Key to the Snapshot Definition Relationship table.	bigint
defn_reln_type_id	Definition Relationship Type Id	A Foreign Key to the Snapshot Definition Relationship table.	int
cntnr_snapshot_id	Container Snapshot Id	A Foreign Key to the Container Snapshot table.	bigint

#### Attribute/Column(s) of Entity/Table "Snapshot Definition Version"

Physical Name	Name	Definition	Physical Data Type
defn_id	Definition Id	A Foreign Key to the Snapshot Definition Version table.	bigint
defn_ver_id	Definition Version Id	The automatically generated primary key of the snapshot_definition_version table.	bigint
defn_ver_name	Definition Version Name	The name of the Snapshot Definition Version.	varchar(255)

defn_ver_status_id	Definition Version Status Id	A Foreign Key to the Snapshot Definition Version table.	int
defn_ver_guid	Definition Version GUID	A Global Unique Identifier search key for the Snapshot Definition Version, composed of letters and numbers. It is computed by the client and typically used in unstructured data.	varchar(50)
cntr_snapshot_id	Container Snapshot Id	A Foreign Key to the Container Snapshot table.	bigint

#### Attribute/Column(s) of Entity/Table "Snapshot Definition Version Assoc"

Physical Name	Name	Definition	Physical Data Type
defn_ver_association_id	Definition Version Association Id	The automatically generated primary key of the snapshot_definition_version_as table.	bigint
from_defn_ver_id	From Definition Version ID	A Foreign Key to the Snapshot Definition Version Assoc table.	bigint
to_defn_ver_id	To Definition Version ID	A Foreign Key to the Snapshot Definition Version Assoc table.	bigint
association_type_id	Association Type Id	A Foreign Key to the Snapshot Definition Version Assoc table.	bigint
cntr_snapshot_id	Container Snapshot Id	A Foreign Key to the Container Snapshot table.	bigint

#### Attribute/Column(s) of Entity/Table "Snapshot Status"

Physical Name	Name	Definition	Physical Data Type
snapshot_status_id	Snapshot Status Id	The unique identifier for an instance of Snapshot Status. This identifier must be kept in sync with programming code that uses an enumeration or fixed constant to identify particular values.	integer
snapshot_status_name	Snapshot Status Name	The unique string identifier for an instance of Snapshot Status. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)

#### Attribute/Column(s) of Entity/Table "Snapshot Type"

Physical Name	Name	Definition	Physical Data Type
snapshot_type_id	Snapshot Type Id	The unique identifier for an instance of Snapshot Type. This identifier must be kept in sync with programming code that uses an enumeration or fixed constant to identify particular values.	integer
snapshot_type_name	Snapshot Type Name	The unique string identifier for an instance of Snapshot Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)

#### Attribute/Column(s) of Entity/Table "Workspace Container"

Physical Name	Name	Definition	Physical Data Type
time_zone_region_name	Time Zone Region Name	The name of the Time Zone Region. This is 2. The 'primary key' of the java timezone is the string region identifier in English, example 'America/Phoenix'. There is no numeric identifier. As the JRE is updated, or specific classes are updated, the list could change to include new entries, and the detail data behind the regions could change.	varchar(255)
cntr_type_id	Container Type Id	A Foreign Key to the Container Type table.	int
cntr_id	Container Id	A Foreign Key to the Container table.	bigint

#### ER Diagram(s) of Subject Area "070 - User"

Name	Physical Display Level
Logical	Column
Physical	Column

#### Entity/Table(s) of Subject Area "070 - User"

Physical Name	Name	Logical Only	Do Not Generate	Definition
agent	Agent	true	true	a user authorized to use the FootPrints agent functions, which enable the agent to provide services to the customer. These include the ability to edit, escalate, and close tickets.

anonymous_cust	Anonymous Customer	true	true	a generic guest user account for users that do not have a unique identity in the system. Typically, this user has read only access to browse the knowledge base.
app_user	Application User		false	A user authorized to use the FootPrints system. The user is either a Customer User or an Agent User. The application user holds the minimum information needed to identify the user and associate the user with a user profile, where user settings are stored. A contact is not necessarily an application user. An application user entry is required in order for the user to access any online functions.
app_user_prof_link	Application User Profile Link		false	The link between the application user and the user profile.
cust	Customer	true	true	a user authorized to use the FootPrints customer functions. This includes the ability to enter requests, and interact with requests as they are worked by Agents.
cust_login_type	Customer Login Type		false	A lookup table for the customer login type. It identifies the method of authenticating customer users.
lic_type	License Type		false	A lookup table for the type of FootPrints license. Must be secure from tampering.
shared_cust	Shared Customer	true	true	A shared customer does not have a unique identity at login, but uses a customer account with permissions according to this shared customer identity.
unique_cust	Unique Customer	true	true	A unique customer has a unique identity in the system either before identification or once authentication is performed.
user_prof	User Profile		false	A container for Permissions and Settings (the composite parts are the associated Role and the User Preferences). The permissions for a User to access a particular container according to the capabilities defined. In most cases this will be more defined in a subcategory.  Example: Shared User Profile This profile will be shared between multiple users that access the system with the same User ID.

#### Attribute/Column(s) of Entity/Table "Agent"

Physical Name	Name	Definition	Physical Data Type
lic_type_id	License Type Id	A Foreign Key to the License Type table.	bigint
app_user_id	Application User Id	A Foreign Key to the Application User table.	bigint

#### Attribute/Column(s) of Entity/Table "Anonymous Customer"

Physical Name	Name	Definition	Physical Data Type
app_user_id	Application User Id	A Foreign Key to the Application User table.	bigint

#### Attribute/Column(s) of Entity/Table "Application User"

Physical Name	Name	Definition	Physical Data Type
pwd	Password	This space intentionally left blank.	varchar(255)
user_name	User Name	The name of the Application User.	nvarchar(4000)
cust_login_type_id	Customer Login Type Id	A Foreign Key to the Customer Login Type table.	int
app_user_id	Application User Id	The automatically generated primary key of the app_user table.	bigint
lic_type_id	License Type Id	A Foreign Key to the License Type table.	bigint
user_login	User Login	The login name used to access the Application. This may not be filled in if external application security is used.	nvarchar(4000)
email_addr	Email Address	An email address for the Application User.	nvarchar(4000)
delete_date	Delete Date	The value of a Application User, stored as a date.	datetime

#### Attribute/Column(s) of Entity/Table "Application User Profile Link"

Physical Name	Name	Definition	Physical Data Type
---------------	------	------------	--------------------

user_prof_id	User Profile Id	A Foreign Key to the User Profile table.	bigint
app_user_prof_link_id	Application User Profile Link Id	The automatically generated primary key of the app_user_prof_link table.	bigint
app_user_id	Application User Id	A Foreign Key to the Application User table.	bigint

#### Attribute/Column(s) of Entity/Table "Customer"

Physical Name	Name	Definition	Physical Data Type
cust_login_type_id	Customer Login Type Id	A Foreign Key to the Customer Login Type table.	int
app_user_id	Application User Id	A Foreign Key to the Application User table.	bigint

#### Attribute/Column(s) of Entity/Table "Customer Login Type"

Physical Name	Name	Definition	Physical Data Type
cust_login_type_name	Customer Login Type Name	The name of the Customer Login Type.	varchar(255)
cust_login_type_id	Customer Login Type Id	The automatically generated primary key of the cust_login_type table.	int

#### Attribute/Column(s) of Entity/Table "License Type"

Physical Name	Name	Definition	Physical Data Type
lic_type_id	License Type Id	The automatically generated primary key of the lic_type table.	bigint

#### Attribute/Column(s) of Entity/Table "Shared Customer"

Physical Name	Name	Definition	Physical Data Type
app_user_id	Application User Id	A Foreign Key to the Application User table.	bigint

#### Attribute/Column(s) of Entity/Table "Unique Customer"

Physical Name	Name	Definition	Physical Data Type
app_user_id	Application User Id	A Foreign Key to the Application User table.	bigint

#### Attribute/Column(s) of Entity/Table "User Profile"

Physical Name	Name	Definition	Physical Data Type
user_prof_id	User Profile Id	The automatically generated primary key of the user_prof table.	bigint
tmpl_user_prof_id	Template User Profile Id	A Foreign Key to the User Profile table. This will not be a navigable key in the object model. This indicates which Template User Profile was used to create a built-in template in case some features concerning new properties added to later release of the system will need to be added to existing templates.	bigint
role_grp_id	Role Group Id	A Foreign Key to the Role Group table. A system level user profile can have a role group, which will have the effect of assigning a collection of workspace, cmdb, and address book roles at the container role level.	bigint
defn_id	Definition Id	A Foreign Key to the Definition table. Optionally, the container that owns the profile.	bigint
app_role_id	Application Role Id	A Foreign Key to the Application Role table. The User Profile is the means for setting the default role at the system and workspace levels, and is also the role of a Unique User Profile tied to a application user.	bigint
agent_authcn_type_id	Agent Authentication Type Id	A Foreign Key to the Agent Authentication Type table.	int
user_prof_name	User Profile Name	The name of the User Profile.	varchar(255)
is_built_in	Is Built In	Indicates that this instance of User Profile is a built-in instance that cannot be changed by the end user.	bit
user_type_id	User Type Id	A Foreign Key to the User Type table.	int
cust_authcn_type_id	Customer Authentication Type Id	A Foreign Key to the Customer Authentication Type table.	int
is_sys	Is System	Indicates if this is a system level profile.	bit
defn_subtype_id	Definition Subtype Id	A Foreign Key to the Definition Subtype table.	int
delete_date	Delete Date	The value of a User Profile, stored as a date.	datetime



last_login	Last Login	date time of last login in UTC.	datetime
tmpl_level_name	tmpl_level_name	The value of a Built_in_level, stored as a string.	varchar(255)

### ER Diagram(s) of Subject Area "071 - Team"

Name	Physical Display Level
Logical	Column
Physical	Column

### Entity/Table(s) of Subject Area "071 - Team"

Physical Name	Name	Logical Only	Do Not Generate	Definition
agent_team	Agent Team		false	The list of Agent team members for a given team.
app_user	Application User		false	A user authorized to use the FootPrints system. The user is either a Customer User or an Agent User. The application user holds the minimum information needed to identify the user and associate the user with a user profile, where user settings are stored. A contact is not necessarily an application user. An application user entry is required in order for the user to access any online functions.
asgnmt	Assignment		false	The current state and historical record of responsibility for a ticket. Regular assignments involve all parties that share responsibility for the ticket and are interested potentially in notifications. Primary Assignments additionally indicate that one and only one userid is responsible for the ticket for a span of time. A userid will have a span of Primary Assignment that is within the boundaries of his regular non-Primary Assignment. It is expected that an individual would have a single large span of time, possibly the life of the ticket, as a regular assignment. The individual may have zero, one, or more periods of PrimaryAssignment, stored as additional records.
asgnmt_type	Assignment Type		false	The assignment type indicates the notification level of the assignment. The current types consist of the following: Regular Assignment, CC Assignment, Primary Assignment .
defn	Definition		false	A definition holds the structural and behavioral information needed for an application element that is configurable by the system designer and/or the end user at definition time. Current Footprints uses a combination of directones, files, tables, and programming code to represent this information. The new FootPrints will provide all of the same information within the definition. The definition can be persisted in a structure that is isolated at the Instance Level. The data for the definition will be stored as Configuration Settings associated with the Definition Version. See the database version v_defn_ver_active, v_cntrn_item_field.
item	Item	true	true	The Item is the abstract class for Issue, Configuration Item, Address Book entry, and other information items that are the main information entities in FootPrints. The Item structure and behavior is defined by its Definition. The Item is namespaced, identified by, and logically stored in the single parent container. It can be referenced in the same manner both inside and outside the container. The Item is the storage model and the implementation for the Item Definition. There are many important specializations of the Item class, and there even more Template definitions of Container/Item that provide an end-user application experience to the End-User. The Item implementation must implement flexible schema in an easy to

				manage, well engineered, and high performing way. The item can store user-defined and user-named fields defined in its definition. The Item could implement the nice-to-have features such as an inheritance model for Item subtypes, and have the ability to represent collections of subordinate items. The Item data can be imported, exported, or synchronized with external data sources. The Item data can be linked and synchronized with other items in the same Instance.
team	Team		false	A team is a group of application users that can be used in Team level assignments.
team_email_notifn_type	Team Email Notification Type		false	When Issues are assigned to this team, all members of the team will  Receive email notifications for all activity on the Issue: Always Only when no individual agents are assigned Never  (Note that email to members will follow the workspace preferences for assignees defined on the Mail Preferences page. Also, any individual assignees will receive mail according to the workspace preferences regardless of what teams are assigned.)
team_item_association	Team Item Association		false	This contains the association between the team, the team assignment method and the item types.  This is new functionality, as opposed to the current application that sets the assignment method directly on the team. The change was due to the new concept of multiple items in a workspace.

#### Attribute/Column(s) of Entity/Table "Agent Team"

Physical Name	Name	Definition	Physical Data Type
team_id	Team Id	A Foreign Key to the Team table.	bigint
agent_team_id	Agent Team Id	The automatically generated primary key of the agent_team table.	bigint
gets_individual_asgnmt_for_tea	Gets Individual Assignment for Team	Indicates if this Agent Team() row is logically deleted.  Igor S asked that this be nullable, and has not been able to find a place in the current Java code where it is being set at all. If the column will be used in the future a null can mean false, or the code can be changed to explicitly set it to false. dnoble 12-4-12	bit
app_user_id	Application User Id	A Foreign Key to the Application User table.	bigint

#### Attribute/Column(s) of Entity/Table "Application User"

Physical Name	Name	Definition	Physical Data Type
pwd	Password	This space intentionally left blank.	varchar(255)
user_name	User Name	The name of the Application User.	nvarchar(4000)
cust_login_type_id	Customer Login Type Id	A Foreign Key to the Customer Login Type table.	int
app_user_id	Application User Id	The automatically generated primary key of the app_user table.	bigint
lic_type_id	License Type Id	A Foreign Key to the License Type table.	bigint
user_login	User Login	The login name used to access the Application. This may not be filled in if external application security is used.	nvarchar(4000)
email_addr	Email Address	An email address for the Application User.	nvarchar(4000)
delete_date	Delete Date	The value of a Application User, stored as a date.	datetime

#### Attribute/Column(s) of Entity/Table "Assignment"

Physical Name	Name	Definition	Physical Data Type
asgnmt_id	Assignment Id	The automatically generated primary key of the asgnmt table.	bigint

item_defn_id	Item Definition Id	A Foreign Key to the Definition table.	bigint
team_id	Team Id	A Foreign Key to the Team table.	bigint
begin_date	Begin Date	The value of a Assignment, stored as a date.	datetime
end_date	End Date	The value of a Assignment, stored as a date.	datetime
asgnmt_type_id	Assignment Type Id	A Foreign Key to the Assignment Type table.	int
app_user_id	Application User Id	A Foreign Key to the Application User table.	bigint
cc_email_addr	CC Email Address	Cannot be used with TeamId, App_user_id, can only be used with CC Assignment	nvarchar(4000)
sort_order	Sort Order	The value of a Assignment stored as boolean.	int
item_id	Item Id	A Foreign Key to the Item table.	bigint
is_team	Is Team	Special clarifying field of "Is Team Level Assignment" to state that the Team does not have to be immediately resolved to a Team+Individual.	bit
delete_txn_id	Delete Transaction Id	A unique incremental transaction identifier that indicates a soft delete. The id is generated via id generation, and may or may not correspond to the audit trail transaction id. This method of marking the soft delete allows for a unique index to be created on the Workspace Definition Id, Team Name, and Delete Transaction Id to enforce no duplicates in the live copies, and it allows multiple soft deleted rows with the same team name.	bigint
field_id	Field Id	A Foreign Key to the Definition table.	bigint

#### Attribute/Column(s) of Entity/Table "Assignment Type"

Physical Name	Name	Definition	Physical Data Type
asgnmt_type_id	Assignment Type Id	The automatically generated primary key of the asgnmt_type table.	int
asgnmt_type_name	Assignment Type Name	The name of the Assignment Type.	varchar(255)
asgnmt_type_descr	Assignment Type Description	A descriptive name of the Assignment Type. This descriptive name may be localized.	varchar(512)

#### Attribute/Column(s) of Entity/Table "Definition"

Physical Name	Name	Definition	Physical Data Type
defn_id	Definition Id	The automatically generated primary key of the defn table.	bigint
defn_name	Definition Name	The name of the Definition.	nvarchar(4000)
dsble_date	Disabled Date	The date that this instance of Definition is logically deleted and disabled.	datetime
defn_guid	Definition GUID	A Global Unique Identifier search key for the Definition, composed of letters and numbers. It is computed by the client and typically used in unstructured data.	varchar(50)
defn_type_name	Definition Type Name	The unique string identifier for an instance of Definition Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)
defn_subtype_id	Definition Subtype Id	A Foreign Key to the Definition Subtype table.	int
tmpl_export_guid	Template Export GUID	The value of a Definition, stored as a string.	VARCHAR(50)

#### Attribute/Column(s) of Entity/Table "Item"

Physical Name	Name	Definition	Physical Data Type
item_key	Item Key	A search key for the Item, composed of letters and numbers.	varchar(255)
item_id	Item Id	The automatically generated primary key of the item table.	bigint
cntrn_id	Container Id	A Foreign Key to the Container table.	bigint
item_defn_id	Item Definition Id	A Foreign Key to the Definition table for an instance of type Definition.	bigint

#### Attribute/Column(s) of Entity/Table "Team"

Physical Name	Name	Definition	Physical Data Type
---------------	------	------------	--------------------

team_id	Team Id	The automatically generated primary key of the team table.	bigint
team_name	Team Name	The name of the Team.	nvarchar(4000)
wksp_defn_id	Workspace Definition Id	A Foreign Key to the Definition table.	bigint
team_email_notifn_type_id	Team Email Notification Type Id	A Foreign Key to the Team Email Notification Type table.	int
team_lead_app_user_id	Team Lead Application User Id	A Foreign Key to the Application User table.	bigint

#### Attribute/Column(s) of Entity/Table "Team Email Notification Type"

Physical Name	Name	Definition	Physical Data Type
team_email_notifn_type_id	Team Email Notification Type Id	The automatically generated primary key of the team_email_notifn_type table.	int
team_email_notifn_type_descr	Team Email Notification Type Description	A descriptive name of the Team Email Notification Type. This descriptive name may be localized.	varchar(512)
team_email_notifn_type_name	Team Email Notification Type Name	The name of the Team Email Notification Type.	varchar(255)

#### Attribute/Column(s) of Entity/Table "Team Item Association"

Physical Name	Name	Definition	Physical Data Type
team_id	Team Id	A Foreign Key to the Team table.	bigint
defn_id	Definition Id	A Foreign Key to the Definition table.	bigint

#### ER Diagram(s) of Subject Area "080 - User Profile"

Name	Physical Display Level
Logical	Column
Physical	Column

#### Entity/Table(s) of Subject Area "080 - User Profile"

Physical Name	Name	Logical Only	Do Not Generate	Definition
agent_authcn_type	Agent Authentication Type		false	An indicator to distinguish whether this profile is for an agent (including the workspace administrator) or a system administrator.
agent_prof	Agent Profile	true	true	Agent profiles are different from the customer profiles by the choice of the preference and role that is chosen.
anonymous_cust_prof	Anonymous Customer Profile	true	true	A type of workspace user profile that is used for the Anonymous Customer feature. Customers have read only access to the Knowledgebase and Solutions via the customer portal. This is required, but can be an empty profile specifying no anonymous access.
app_role	Application Role		false	A User Security Role within Footprints. This includes Agent Users and Customer Users, and includes Role settings at the Instance and Container level. Workspace Role and CMDB Role may both be implemented as Container Role. The Role has associated Rights and Permissions, and is a way of granting Roles and Permissions to a User. There is some conceptual overlap between the User Profile and the Role. The User is assigned a single Role as part of the User Profile, and a User can only have one System Role and one Workspace Role. The User Profile can have property settings that override the Role Settings.

app_user	Application User		false	A user authorized to use the FootPrints system. The user is either a Customer User or an Agent User. The application user holds the minimum information needed to identify the user and associate the user with a user profile, where user settings are stored. A contact is not necessarily an application user. An application user entry is required in order for the user to access any online functions.
app_user_prof_link	Application User Profile Link		false	The link between the application user and the user profile.
auto_add_cust_prof	Auto Add Customer Profile	true	true	A type of user profile that is used for the Auto Add Customer feature. An assignment of roles to workspaces that applies to a customer based on administrator assignment or criteria available from the Customer Authentication method (such as LDAP).
cust_authcn_type	Customer Authentication Type	false	false	An indicator to distinguish whether this profile is for a unique customer profile (including the specific customer profile) or if it requires a special authentication technique (i.e. Anonymous Customer Profile has read only access to the knowledge base without having to be authenticated).
cust_prof	Customer Profile	true	true	Customer profiles are different from the agent profiles by the choice of the preference and role that is chosen.
defn	Definition		false	A definition holds the structural and behavioral information needed for an application element that is configurable by the system designer and/or the end user at definition time. Current Footprints uses a combination of directories, files, tables, and programming code to represent this information. The new FootPrints will provide all of the same information within the definition. The definition can be persisted in a structure that is isolated at the Instance Level. The data for the definition will be stored as Configuration Settings associated with the Definition Version. See the database views v_defn_ver_active, v_cntrn_item_field.
defn_subtype	Definition Subtype		false	The subtype of the definition used as a discriminator for the class of definition. Each subtype will have different configuration settings and associated definitions. These include associated built-in field definitions for item subtypes. See the database view v_defn_subtype.
defn_type	Definition Type		false	The scope of the Template chosen from Instance, Container Group, Container, Item, Field.
role_grp	Role Group		false	(Not used, but referenced in Java code that must be removed before table deletion.) The role group will contain the roles that the user has for each of the workspaces that the user has access. The user will always have a role group even if the administrator chooses the Perl Footprints functionality.
shared_cust_prof	Shared Customer Profile	true	true	A user profile that is shared among a group of many users. It is used for shared customer authentication.
standard_agent_prof	Standard Agent Profile	true	true	Permissions for a User to access a workspace according to the capabilities defined in the role. Also, configuration settings for the User in that workspace. This is no more than a default built in profile with the name of "Workspace Administrator" which can be changed later by the System Administrator.

sys_admin_prof	System Administration Profile	true	true	Permissions for a User to access the FootPrints system according to the capabilities defined in the role. Also, configuration settings for the User.
unique_cust_prof	Unique Customer Profile	true	true	A user profile for a single unique user. This profile can be created by an administrator for agents and customers. It also can be created via auto-add for customers only.
user_prof	User Profile		false	A container for Permissions and Settings (the composite parts are the associated Role and the User Preferences). The permissions for a User to access a particular container according to the capabilities defined. In most cases this will be more defined in a subcategory.  Example: Shared User Profile This profile will be shared between multiple users that access the system with the same User ID.
user_type	User Type		false	Indicates whether this entity is of an agent type or a customer type. Also CMDB and GUEST.
wksp_admin_prof	Workspace Administration Profile	true	true	Permissions for a User to access a workspace according to the capabilities defined in the role. Also, configuration settings for the User in that workspace. This is no more than a default built in profile with the name of "Workspace Administrator" which can be changed later by the System Administrator.

#### Attribute/Column(s) of Entity/Table "Agent Authentication Type"

Physical Name	Name	Definition	Physical Data Type
agent_authcn_type_name	Agent Authentication Type Name	The name of the Agent Authentication Type.	varchar(255)
agent_authcn_type_id	Agent Authentication Type Id	The automatically generated primary key of the agent_authcn_type table.	int

#### Attribute/Column(s) of Entity/Table "Agent Profile"

Physical Name	Name	Definition	Physical Data Type
user_prof_id	User Profile Id	A Foreign Key to the User Profile table.	bigint
agent_authcn_type_id	Agent Authentication Type Id	A Foreign Key to the Agent Authentication Type table.	int

#### Attribute/Column(s) of Entity/Table "Anonymous Customer Profile"

Physical Name	Name	Definition	Physical Data Type
user_prof_id	User Profile Id	A Foreign Key to the User Profile table.	bigint

#### Attribute/Column(s) of Entity/Table "Application Role"

Physical Name	Name	Definition	Physical Data Type
app_role_id	Application Role Id	The automatically generated primary key of the app_role table.	bigint
tmpl_app_role_id	Template Application Role Id	A Foreign Key to the Application Role table. This indicates which Template Role was used to create a built-in role in case some features concerning new properties added to later release of the system will need to be added to existing templates.	bigint
app_role_name	Application Role Name	The name of the Application Role.	varchar(255)
defn_id	Definition Id	A Foreign Key to the Definition table.	bigint
admin_type_id	Administrator Type Id	A Foreign Key to the Administrator Type table.	int
is_built_in	Is Built In	Indicates that this instance of Application Role is a built-in instance that cannot be changed by the end user.	bit
app_role_descr	Application Role Description	A descriptive name of the Application Role. This descriptive name may be localized.	varchar(1024)
user_type_id	User Type Id	A Foreign Key to the User Type table.	int
defn_subtype_id	Definition Subtype Id	A Foreign Key to the Definition Subtype table.	int
tmpl_app_role_id	Template Application Role Id	A Foreign Key to the Application Role table.	bigint

tmpl_level_name	tmpl_level_name	The value of a Built_in_level, stored as a string.	varchar(255)
-----------------	-----------------	--	--------------

#### Attribute/Column(s) of Entity/Table "Application User"

Physical Name	Name	Definition	Physical Data Type
pwd	Password	This space intentionally left blank.	varchar(255)
user_name	User Name	The name of the Application User.	nvarchar(4000)
cust_login_type_id	Customer Login Type Id	A Foreign Key to the Customer Login Type table.	int
app_user_id	Application User Id	The automatically generated primary key of the app_user table.	bigint
lic_type_id	License Type Id	A Foreign Key to the License Type table.	bigint
user_login	User Login	The login name used to access the Application. This may not be filled in if external application security is used.	nvarchar(4000)
email_addr	Email Address	An email address for the Application User.	nvarchar(4000)
delete_date	Delete Date	The value of a Application User, stored as a date.	datetime

#### Attribute/Column(s) of Entity/Table "Application User Profile Link"

Physical Name	Name	Definition	Physical Data Type
user_prof_id	User Profile Id	A Foreign Key to the User Profile table.	bigint
app_user_prof_link_id	Application User Profile Link Id	The automatically generated primary key of the app_user_prof_link table.	bigint
app_user_id	Application User Id	A Foreign Key to the Application User table.	bigint

#### Attribute/Column(s) of Entity/Table "Auto Add Customer Profile"

Physical Name	Name	Definition	Physical Data Type
user_prof_id	User Profile Id	A Foreign Key to the User Profile table.	bigint

#### Attribute/Column(s) of Entity/Table "Customer Authentication Type"

Physical Name	Name	Definition	Physical Data Type
cust_authcn_type_id	Customer Authentication Type Id	The automatically generated primary key of the cust_authcn_type table.	int
cust_authcn_type_name	Customer Authentication Type Name	The name of the Customer Authentication Type.	varchar(255)

#### Attribute/Column(s) of Entity/Table "Customer Profile"

Physical Name	Name	Definition	Physical Data Type
user_prof_id	User Profile Id	A Foreign Key to the User Profile table.	bigint
cust_authcn_type_id	Customer Authentication Type Id	A Foreign Key to the Customer Authentication Type table.	int

#### Attribute/Column(s) of Entity/Table "Definition"

Physical Name	Name	Definition	Physical Data Type
defn_id	Definition Id	The automatically generated primary key of the defn table.	bigint
defn_name	Definition Name	The name of the Definition.	nvarchar(4000)
dsble_date	Disabled Date	The date that this instance of Definition is logically deleted and disabled.	datetime
defn_guid	Definition GUID	A Global Unique Identifier search key for the Definition, composed of letters and numbers. It is computed by the client and typically used in unstructured data.	varchar(50)
defn_type_name	Definition Type Name	The unique string identifier for an instance of Definition Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)
defn_subtype_id	Definition Subtype Id	A Foreign Key to the Definition Subtype table.	int
tmpl_export_guid	Template Export GUID	The value of a Definition, stored as a string.	VARCHAR(50)

#### Attribute/Column(s) of Entity/Table "Definition Subtype"

Physical Name	Name	Definition	Physical Data Type
---------------	------	------------	--------------------

defn_subtype_name	Definition Subtype Name	The name of the Definition Subtype.	varchar(255)
defn_subtype_descr	Definition Subtype Description	A descriptive name of the Definition Subtype. This descriptive name may be localized.	varchar(512)
defn_subtype_id	Definition Subtype Id	The automatically generated primary key of the defn_subtype table.	int
defn_parent_subtype_id	definition parent subtype id	A Foreign Key to the Definition Subtype table.	int
defn_type_name	Definition Type Name	The unique string identifier for an instance of Definition Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)
max_in_parent	Maximum In Parent	A Foreign Key to the Definition Subtype table.	int

#### Attribute/Column(s) of Entity/Table "Definition Type"

Physical Name	Name	Definition	Physical Data Type
defn_type_name	Definition Type Name	The unique string identifier for an instance of Definition Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)
defn_type_descr	Definition Type Description	A descriptive name of the Definition Type. This descriptive name may be localized.	varchar(512)

#### Attribute/Column(s) of Entity/Table "Role Group"

Physical Name	Name	Definition	Physical Data Type
role_grp_id	Role Group Id	The automatically generated primary key of the role_grp table.	bigint
role_grp_name	Role Group Name	The name of the Role Group.	varchar(255)
role_grp_descr	Role Group Description	A descriptive name of the Role Group. This descriptive name may be localized.	varchar(512)

#### Attribute/Column(s) of Entity/Table "Shared Customer Profile"

Physical Name	Name	Definition	Physical Data Type
user_prof_id	User Profile Id	A Foreign Key to the User Profile table.	bigint

#### Attribute/Column(s) of Entity/Table "Standard Agent Profile"

Physical Name	Name	Definition	Physical Data Type
user_prof_id	User Profile Id	A Foreign Key to the User Profile table.	bigint

#### Attribute/Column(s) of Entity/Table "System Administration Profile"

Physical Name	Name	Definition	Physical Data Type
user_prof_id	User Profile Id	A Foreign Key to the User Profile table.	bigint
is_sys_admin	Is System Administrator	Indicates if Is System Administrator.	char(1)

#### Attribute/Column(s) of Entity/Table "Unique Customer Profile"

Physical Name	Name	Definition	Physical Data Type
user_prof_id	User Profile Id	A Foreign Key to the User Profile table.	bigint

#### Attribute/Column(s) of Entity/Table "User Profile"

Physical Name	Name	Definition	Physical Data Type
user_prof_id	User Profile Id	The automatically generated primary key of the user_prof table.	bigint
tmpl_user_prof_id	Template User Profile Id	A Foreign Key to the User Profile table. This will not be a navigable key in the object model. This indicates which Template User Profile was used to create a built-in template in case some features concerning new properties added to later release of the system will need to be added to existing templates.	bigint
role_grp_id	Role Group Id	A Foreign Key to the Role Group table. A system level user profile can have a role group, which will have the effect of assigning a collection of workspace, cmdb, and address book roles at the container role level.	bigint
defn_id	Definition Id	A Foreign Key to the Definition table. Optionally, the container that owns the profile.	bigint
app_role_id	Application Role Id	A Foreign Key to the Application Role table. The User Profile is the means for setting the default role at the system and workspace levels, and is also the role of a Unique User Profile tied to a	bigint



		application user.	
agent_authcn_type_id	Agent Authentication Type Id	A Foreign Key to the Agent Authentication Type table.	int
user_prof_name	User Profile Name	The name of the User Profile.	varchar(255)
is_built_in	Is Built In	Indicates that this instance of User Profile is a built-in instance that cannot be changed by the end user.	bit
user_type_id	User Type Id	A Foreign Key to the User Type table.	int
cust_authcn_type_id	Customer Authentication Type Id	A Foreign Key to the Customer Authentication Type table.	int
is_sys	Is System	Indicates if this is a system level profile.	bit
defn_subtype_id	Definition Subtype Id	A Foreign Key to the Definition Subtype table.	int
delete_date	Delete Date	The value of a User Profile, stored as a date.	datetime
last_login	Last Login	date time of last login in UTC.	datetime
tmpl_level_name	tmpl_level_name	The value of a Built_in_level, stored as a string.	varchar(255)

#### Attribute/Column(s) of Entity/Table "User Type"

Physical Name	Name	Definition	Physical Data Type
user_type_name	User Type Name	The name of the User Type.	varchar(255)
user_type_id	User Type Id	The automatically generated primary key of the user_type table.	int
user_type_descr	User Type Description	A descriptive name of the User Type. This descriptive name may be localized.	varchar(512)

#### Attribute/Column(s) of Entity/Table "Workspace Administration Profile"

Physical Name	Name	Definition	Physical Data Type
user_prof_id	User Profile Id	A Foreign Key to the User Profile table.	bigint

#### ER Diagram(s) of Subject Area "096 - Email Notification"

Name	Physical Display Level
Logical	Column
Physical	Column

#### Entity/Table(s) of Subject Area "096 - Email Notification"

Physical Name	Name	Logical Only	Do Not Generate	Definition
email_notifn	Email Notification		false	An email notification entry.
email_notifn_atrch	Email Notification Attachment		false	A link to an email attachment used for a notification email.
email_notifn_type	Email Notification Type		false	
notifn_recipient_status	Notification Recipient Status		false	Manages the sent and received status of the distribution list of contacts for a notification.

#### Attribute/Column(s) of Entity/Table "Email Notification"

Physical Name	Name	Definition	Physical Data Type
email_notifn_id	Email Notification Id	The automatically generated primary key of the email_notifn table.	bigint
from_name	From Name	From Name.	nvarchar(max)
reply_addr	Reply Address	Reply Address	nvarchar(max)
plain_text_body	Plain Text Body	Plain Text Body	nvarchar(max)
html_body	HTML Body	HTML Body	nvarchar(max)
subject	Subject	Subject	nvarchar(max)
schdd_date	Scheduled Date	Scheduled Date of the mass email.	datetime
email_notifn_name	Email Notification Name	The name of the Email Notification.	varchar(255)
email_notifn_descr	Email Notification Description	A descriptive name of the Email Notification. This descriptive name may be localized.	varchar(512)
time_before_break	Time Before Break	Whether to send all emails at once, or send them at specified intervals.	int
msgs_per_batch	Messages Per Batch	How many messages to send Per Batch.	int
delay_between_batches	Delay Between Batches	Delay Between Batches	int
create_date	Create Date	Create Date	datetime
create_user	Create User	Create User	int
plain_text_auto	Plain Text Auto	Plain Text Auto	bit
priority	Priority	Email importance set in the rule ( low-5, normal-3, high-1)	varchar(10)

addr_book	Address Book	Address Book id used for mass emails.	bigint
item_type	Item Type	The item definition id, used for mass emails.	bigint
selected_field	Selected Field	The Selected Field for id lookup, used for mass emails.	bigint
email_notifn_type_id	Email Notification Type Id	The unique identifier for an instance of Email Notification Type. This identifier must be kept in sync with programming code that uses an enumeration or fixed constant to identify particular values.	integer
protocol_id	Protocol Id	The unique identifier for an instance of Email Notification. This identifier must be kept in sync with programming code that uses an enumeration or fixed constant to identify particular values.	integer
start_mail_sent	Start Mail Sent	Indicates if Start Mail Sent.	bit

#### Attribute/Column(s) of Entity/Table "Email Notification Attachment"

Physical Name	Name	Definition	Physical Data Type
email_notifn_attach_id	Email Notification Attachment Id	The automatically generated primary key of the email_notifn_attach table.	bigint
email_attach_uri	Email Attachment URI	The value of a Email Notification Attachment, stored as a string.	nvarchar(4000)
email_notifn_id	Email Notification Id	The automatically generated primary key of the email_notifn table.	bigint

#### Attribute/Column(s) of Entity/Table "Email Notification Type"

Physical Name	Name	Definition	Physical Data Type
email_notifn_type_id	Email Notification Type Id	The unique identifier for an instance of Email Notification Type. This identifier must be kept in sync with programming code that uses an enumeration or fixed constant to identify particular values.	integer
email_notifn_type_name	Email Notification Type Name	The unique string identifier for an instance of Email Notification Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)

#### Attribute/Column(s) of Entity/Table "Notification Recipient Status"

Physical Name	Name	Definition	Physical Data Type
recipient_status_id	Recipient Status Id	The automatically generated primary key of the notification_recipient_status table. The name is shortened for convenience.	bigint
email_notifn_id	Email Notification Id	A Foreign Key to the Email Notification table.	bigint
recipient_contact	Recipient Contact	The value of a Notification Recipient Status, stored as a string.	nvarchar(4000)
retries_ct	Retries Ct	The value of a Notification Recipient Status stored as boolean.	integer
schdd_date	Scheduled Date	The date that the Notification Recipient Status was created.	datetime
completed_date	Completed Date	Completed Date.	datetime
notifn_status_code	Notification Status Code	The unique identifier for an instance of Notification Recipient Status. This identifier must be kept in sync with programming code that uses an enumeration or fixed constant to identify particular values.	integer

#### ER Diagram(s) of Subject Area "101 - Data Access System"

Name	Physical Display Level
Logical	Column
Physical	Column

#### Entity/Table(s) of Subject Area "101 - Data Access System"

Physical Name	Name	Logical Only	Do Not Generate	Definition
db_data_type	Database Data Type		false	The physical database types associated with a logical database type, used in DDL generation and also database DML access.
dyn_col_cstr_spec	Dynamic Column Constraint Specification		false	A database constraint associated with a column.
dyn_col_spec	Dynamic Column Specification		false	The list of column specifications for the Dynamic Table Specification. A column will be added or modified to align with the column specification.

dyn_table_index	Dynamic Table Index	false	false	A physical database index associated with a dynamically generated table. Different DBMS can have different number and types of these indexes. An index specification will be generated to the database.
dyn_table_index_col	Dynamic Table Index Column	false	false	The collection of index column specifications for an index.
dyn_table_spec	Dynamic Table Specification		false	The table specifications for a physical database table associated with a dynamic item. A table will be created or altered to match this specification.
index_type	Index Type	false	false	The type of index. Values include: Primary, Unique, Full-Text, Regular, Reverse Key, Computed Reverse Key
logical_data_type	Logical Data Type		false	A reference table for logical data type. The logical type maps to the database type in DDL generation.
target_dbms	Target DBMS		false	A lookup table for the Target Database Management System types. Values include: SQLServer, PostgreSQL, MySQL, Oracle.

#### Attribute/Column(s) of Entity/Table "Database Data Type"

Physical Name	Name	Definition	Physical Data Type
target_dbms_id	Target DBMS Id	The unique identifier for an instance of Target DBMS. This identifier must be kept in sync with programming code that uses an enumeration or fixed constant to identify particular values.	int
db_data_type_name	Database Data Type Name	The name of the Database Data Type.	varchar(255)
logical_data_type_id	Logical Data Type Id	A Foreign Key to the Logical Data Type table.	int
db_data_type_id	Database Data Type Id	The automatically generated primary key of the db_data_type table.	int

#### Attribute/Column(s) of Entity/Table "Dynamic Column Constraint Specification"

Physical Name	Name	Definition	Physical Data Type
dyn_col_cstr_spec_id	Dynamic Column Constraint Specification Id	The automatically generated primary key of the dyn_col_cstr_spec table.	bigint
dyn_col_spec_id	Dynamic Column Specification Id	A Foreign Key to the Dynamic Column Specification table.	bigint
cstr_val	Constraint Value	The value of a Dynamic Column Constraint Specification stored as boolean.	int
field_cstr_type	Field Constraint Type	A Foreign Key to the Dynamic Column Constraint Specification table.	int

#### Attribute/Column(s) of Entity/Table "Dynamic Column Specification"

Physical Name	Name	Definition	Physical Data Type
dyn_col_spec_id	Dynamic Column Specification Id	The automatically generated primary key of the dyn_col_spec table.	bigint
dflt_string_val	Default String Value	The value of a database default. Not used.	varchar(255)
dflt_int_val	Default Int Value	The value of a database default. Not used.	int
dflt_date_val	Default Date Value	The value of a database default. Not used.	datetime
dyn_table_spec_id	Dynamic Table Specification Id	A Foreign Key to the Dynamic Table Specification table.	bigint
col_name	Column Name	The name of the Dynamic Column Specification.	varchar(255)
logical_data_type_id	Logical Data Type Id	A Foreign Key to the Logical Data Type table.	int
dflt_long_val	Default Long Value	The value of a database default. Not used.	bigint
reference_id	Reference Id	The value of a Dynamic Column Specification stored as Long.	bigint
is_journal	Is Journal	Indicates if this column supports the journaling storage feature.	bit

#### Attribute/Column(s) of Entity/Table "Dynamic Table Index"

Physical Name	Name	Definition	Physical Data Type
dyn_table_spec_id	Dynamic Table Specification Id	A Foreign Key to the Dynamic Table Specification table.	bigint

index_name	Index Name	The name of the DBMS_Object_Name. It is a database object name, limited to 30 characters in length.	varchar(30)
dyn_table_index_id	Dynamic Table Index Id	The automatically generated primary key of the dyn_table_index table.	bigint
index_type_id	Index Type Id	The unique identifier for an instance of Index Type. This identifier must be kept in sync with programming code that uses an enumeration or fixed constant to identify particular values.	int

#### Attribute/Column(s) of Entity/Table "Dynamic Table Index Column"

Physical Name	Name	Definition	Physical Data Type
dyn_col_spec_id	Dynamic Column Specification Id	A Foreign Key to the Dynamic Column Specification table.	bigint
dyn_table_index_id	Dynamic Table Index Id	A Foreign Key to the Dynamic Table Index table.	bigint
dyn_col_pos	Dynamic Column Position	The value of a Dynamic Table Index Column stored as boolean.	int

#### Attribute/Column(s) of Entity/Table "Dynamic Table Specification"

Physical Name	Name	Definition	Physical Data Type
dyn_table_spec_id	Dynamic Table Specification Id	The automatically generated primary key of the dyn_table_spec table.	bigint
table_name	Table Name	The name of the Database Table Name. It is a database object name, limited to 30 characters in length.	varchar(30)
schema_name	Schema Name	The name of the database schema. It is a database object name, limited to 30 characters in length.	varchar(30)
reference_id	Reference Id	A Foreign Key to the Dynamic Table Specification table.	bigint
batch_id	Batch Id	Publishing batch id, indicates things that are DDL generated together.	bigint
status	Status	The publishing status of the column. Helps indicate how to treat changes to information.	varchar(255)

#### Attribute/Column(s) of Entity/Table "Index Type"

Physical Name	Name	Definition	Physical Data Type
index_type_id	Index Type Id	The unique identifier for an instance of Index Type. This identifier must be kept in sync with programming code that uses an enumeration or fixed constant to identify particular values.	int
index_type_name	Index Type Name	The unique string identifier for an instance of Index Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)

#### Attribute/Column(s) of Entity/Table "Logical Data Type"

Physical Name	Name	Definition	Physical Data Type
logical_data_type_name	Logical Data Type Name	The name of the Logical Data Type.	varchar(255)
logical_data_type_id	Logical Data Type Id	The automatically generated primary key of the logical_data_type table.	int

#### Attribute/Column(s) of Entity/Table "Target DBMS"

Physical Name	Name	Definition	Physical Data Type
target_dbms_id	Target DBMS Id	The unique identifier for an instance of Target DBMS. This identifier must be kept in sync with programming code that uses an enumeration or fixed constant to identify particular values.	int
target_dbms_name	Target DBMS Name	The name of the Target DBMS.	varchar(255)

#### ER Diagram(s) of Subject Area "102 - Data Access Scripts"

Name	Physical Display Level
Logical	Column
Physical	Column

#### Entity/Table(s) of Subject Area "102 - Data Access Scripts"

Physical Name	Name	Logical Only	Do Not Generate	Definition
cntrn_defn	Container Definition	true	true	The definition of a Container and the associated Configuration_Settings at the container level. This would include the collection of Item Definitions. The Container_Definition will have a Container_Class identifier (Workspace, CMDB, Address

				Book) and may have specializations such as Workspace_Container_Definition. As much as possible, the Container Definition will suffice to describe any Container Type. There will be an association to a default Container Template, as well as more application specific Container Templates that can be chosen to initially populate the Container Template. There will also be Container Group Definition, and Container Group Templates. See the database views v_cntrn_defn, v_cntrn_defn_active.
defn_subtype	Definition Subtype		false	The subtype of the definition used as a discriminator for the class of definition. Each subtype will have different configuration settings and associated definitions. These include associated built-in field definitions for item subtypes. See the database view v_defn_subtype.
dyn_table_script	Dynamic Table Script		false	A DDL script with substitution parameters that is executed by Data Access during the publish process. It contains data structures that must be created along with the tables that are generated for the dynamic item. Parameters may include: a) schema name b) item name prefix (derived from the item name, possibly exactly equal to it) c) container id d) item definition Id Generally, the scripts can be run post level (container, item) creation using the Type Order and then the Script Order. If the script is changed in the directory then it is reloaded and gets a new identifier. This will cause it to re-run. Scripts must be fundamentally re-runnable.
dyn_table_script_history	Dynamic Table Script History		false	Records the successful execution of the Dynamic Table Script for a Container and/or Dynamic Item. This is used to control re-execution of scripts when the script version changes, or when a new script is added.
dyn_table_script_type	Dynamic Table Script Type		false	The type of Dynamic Table script, which is important for locating and re-running the script if need be. Examples include: Activity Data Mart Creation, Status Change Data Mart Creation. Generally, the scripts can be run post level (container, item) creation using the Type Order and then the Script Order.
item_defn	Item Definition	true	true	The definition of an Item, which includes the Item Name and localizations and the collection of Item Fields. It also includes customizations and localizations of string resources for Item Names, and Item Field names.
target_dbms	Target DBMS		false	A lookup table for the Target Database Management System types. Values include: SQLServer, PostgreSQL, MySQL, Oracle.

#### Attribute/Column(s) of Entity/Table "Container Definition"

Physical Name	Name	Definition	Physical Data Type
db_schema_name	Database Schema Name	The name of the database schema. It is a database object name, limited to 30 characters in length.	varchar(30)
defn_id	Definition Id	A Foreign Key to the Definition table.	bigint

#### Attribute/Column(s) of Entity/Table "Definition Subtype"

Physical Name	Name	Definition	Physical Data Type
defn_subtype_name	Definition Subtype Name	The name of the Definition Subtype.	varchar(255)
defn_subtype_descr	Definition Subtype Description	A descriptive name of the Definition Subtype. This descriptive name may be localized.	varchar(512)
defn_subtype_id	Definition Subtype Id	The automatically generated primary key of the defn_subtype table.	int
defn_parent_subtype_id	definition parent subtype id	A Foreign Key to the Definition Subtype table.	int
defn_type_name	Definition Type Name	The unique string identifier for an instance of Definition Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is	varchar(255)

		generally used to populate this table.	
max_in_parent	Maximum In Parent	A Foreign Key to the Definition Subtype table.	int

#### Attribute/Column(s) of Entity/Table "Dynamic Table Script"

Physical Name	Name	Definition	Physical Data Type
target_dbms_id	Target DBMS Id	The unique identifier for an instance of Target DBMS. This identifier must be kept in sync with programming code that uses an enumeration or fixed constant to identify particular values.	int
dyn_table_script_id	Dynamic Table Script Id	The automatically generated primary key of the dyn_table_script table.	bigint
dyn_table_script_name	Dynamic Table Script Name	The name of the Dynamic Table Script.	varchar(255)
script_type_id	Script Type Id	A Foreign Key to the Dynamic Table Script Type table.	bigint
is_active	Is Active	Indicates if the script is the active version, or if it is a superceded version. When deployment loads the script, the SQL text of the script is compared with any existing active row. If it is equal, the script is not loaded. If unequal, the existing script is marked inactive, and a new active script is loaded.	bit
script_sort_order	Script Sort Order	A sort number used to order a list of Dynamic Table Script.	int
sql_script	SQL Script	The value of a Dynamic Table Script, stored as a long text object.	varchar(max)
defn_subtype_id	Definition Subtype Id	A Foreign Key to the Definition Subtype table.	int
run_as_proxy	Run As Proxy	Used for Oracle, indicating if a proxy connection to impersonate the schema account is needed.	bit

#### Attribute/Column(s) of Entity/Table "Dynamic Table Script History"

Physical Name	Name	Definition	Physical Data Type
dyn_table_script_id	Dynamic Table Script Id	A Foreign Key to the Dynamic Table Script table.	bigint
dyn_table_script_history_id	Dynamic Table Script History Id	The automatically generated primary key of the dyn_table_script_history table.	bigint
item_defn_id	Item Definition Id	A Foreign Key to the Definition table for an instance of type Definition.	bigint
create_date	Create Date	The date that the Dynamic Table Script History was created.	datetime
cntr_defn_id	Container Definition Id	A Foreign Key to the Definition table.	bigint

#### Attribute/Column(s) of Entity/Table "Dynamic Table Script Type"

Physical Name	Name	Definition	Physical Data Type
dyn_table_script_type_id	Dynamic Table Script Type Id	The automatically generated primary key of the dyn_table_script_type table.	bigint
dyn_table_script_type_name	Dynamic Table Script Type Name	The unique string identifier for an instance of Dynamic Table Script Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)
dyn_table_script_type_descr	Dynamic Table Script Type Description	A descriptive name of the Dynamic Table Script Type. This descriptive name may be localized.	varchar(512)
type_sort_order	Type Sort Order	A sort number used to order a list of Dynamic Table Script Type.	int

#### Attribute/Column(s) of Entity/Table "Item Definition"

Physical Name	Name	Definition	Physical Data Type
item_table_name	Item Table Name	The name of the DBMS_Object_Name. It is a database object name, limited to 30 characters in length.	varchar(30)
item_defn_id	Item Definition Id	A Foreign Key to the Definition table for an instance of type Definition.	bigint

#### Attribute/Column(s) of Entity/Table "Target DBMS"

Physical Name	Name	Definition	Physical Data Type
target_dbms_id	Target DBMS Id	The unique identifier for an instance of Target DBMS. This identifier must be kept in sync with programming code that uses an enumeration or fixed constant to identify particular values.	int

target_dbms_name	Target DBMS Name	The name of the Target DBMS.	varchar(255)
------------------	------------------	------------------------------	--------------

#### ER Diagram(s) of Subject Area "110 - Id Generation"

Name	Physical Display Level
Logical	Column
Physical	Column

#### Entity/Table(s) of Subject Area "110 - Id Generation"

Physical Name	Name	Logical Only	Do Not Generate	Definition
id_generator	Id Generator		false	A table of incremental ids used as table primary keys and object identifiers.

#### Attribute/Column(s) of Entity/Table "Id Generator"

Physical Name	Name	Definition	Physical Data Type
db_table_name	Database Table Name	The value of a Id Generator, stored as a string.	varchar(255)
last_id_val	Last Id Value	The value that will be incremented	bigint

#### ER Diagram(s) of Subject Area "120 - Job Scheduler"

Name	Physical Display Level
Logical	Column
Physical	Column

#### Entity/Table(s) of Subject Area "120 - Job Scheduler"

Physical Name	Name	Logical Only	Do Not Generate	Definition
job_cfg_dtl	Job Configuration Detail		false	The job configuration details that will be used to create jobs on the job scheduler.
job_schedlr_cfg	Job Scheduler Configuration		false	An instance of job scheduler used for background job execution. Currently, the Quartz job scheduler is used. Usually one instance is sufficient.

#### Attribute/Column(s) of Entity/Table "Job Configuration Detail"

Physical Name	Name	Definition	Physical Data Type
job_cfg_dtl_id	Job Configuration Detail Id	The automatically generated primary key of the job_cfg_dtl table.	bigint
job_name	Job Name	The name of the job	varchar(100)
job_class	Job Class	The java class name of the job	VARCHAR(150)
grp_name	Group Name	Used for time based rules grouping.	varchar(100)
trigger_type_name	Trigger Type Name	Frequency type of trigger. eg Daily, Minutely, which can be qualified by the frequency. E.G. Minutely 15. Most jobs are controlled by Quartz. CRON jobs are controlled by the OS.	varchar(255)
trigger_start_date	Trigger Start Date	The start date of scheduled running. Not required	datetime
trigger_end_date	Trigger End Date	Trigger End Date.	datetime
trigger_hour	Trigger Hour	The Hours portion of a time of day	int
trigger_minute	Trigger Minute	The minutes portion of a time of day	int
trigger_freq	Trigger Frequency	Numeric frequency. e.g 15 in Minutely 15	int
is_job_enabled	Is Job Enabled	False if disabled.	bit
history_retain_days	History Retain Days	Days to retain history of jobs before cleanup	int
history_retain_ct	History Retain Count	Number of executions before cleanup.	int
job_parms	Job Parameters	Array of parameters used to pass to job.	varchar(max)
create_date	Create Date	The date that the Job Configuration Detail was created.	datetime
create_user	Create User	User that created job	varchar(255)
job_schedlr_cfg_id	Job Scheduler Configuration Id	A Foreign Key to the Job Scheduler Configuration table.	bigint
cron	Cron	For trigger type CRON JOB, the cron job arguments that will be passed on the command line.	varchar(255)
repeat_ct	Repeat Count	0 if not used. Can be set to a finite number.	int
time_zone	Time Zone	The value of a Job Configuration Detail, stored as a string.	varchar(255)
ver_id	Version ID	A Foreign Key to the Job Configuration Detail table.	bigint

#### Attribute/Column(s) of Entity/Table "Job Scheduler Configuration"

Physical Name	Name	Definition	Physical Data Type
---------------	------	------------	--------------------

job_schedlr_cfg_id	Job Scheduler Configuration Id	The automatically generated primary key of the job_schedlr_cfg table.	bigint
svc_host_name	Service Host Name	The name or IP address of the host machine.	varchar(255)
svc_port	Service Port	TCPIP port.	int
job_schedlr_instc_id	Job Scheduler Instance Id	An alternate automatically generated key of the job_schedlr_cfg table.	varchar(255)
job_schedlr_dflt_interrupt	Job Scheduler Default Interrupt	Default for boolean to interrupt long running jobs.	bit
job_schedlr_dflt_max_wait	Job Scheduler default Maximum Wait	Maximum wait time before interruption.	bigint

### ER Diagram(s) of Subject Area "140 - Workspace Calendar and Operation Schedule"

Name	Physical Display Level
Logical	Column
Physical	Column

### Entity/Table(s) of Subject Area "140 - Workspace Calendar and Operation Schedule"

Physical Name	Name	Logical Only	Do Not Generate	Definition
agent	Agent	true	true	a user authorized to use the FootPrints agent functions, which enable the agent to provide services to the customer. These include the ability to edit, escalate, and close tickets.
cal_item	Calendar Item		false	The main calendar item table for the calendar service. This table is populated and maintained by the Calendar Service API. It is not very suitable for database query outside of the API. The calendar item can be a personal or workspace calendar, or an appointment within the calendar. The calendar item can be synched using the iCal format.
cal_schd	Calendar Schedule	false	false	The association between the FootPrints calendar instance, and the implementation of the calendar schedule and appointments. This enables the reuse of a schedule between workspaces, but it does not require it.
cal_type	Calendar Type		false	The type of Calendar, Workspace or Personal
schd	Schedule	false	false	A personal or workspace calendar in Footprints. It is used to provide easy access to work hours and settings for appointment scheduling and metric reporting. The Calendar Service maintains the system of record tables for the schedule, but those tables are too difficult and time consuming to query for trend reporting.
schd_actual_period	Schedule Actual Period		false	Given the definitions of the standard work periods and holidays, these are the work periods. If overlapping shift work periods are defined, the time ranges here represent all shifts. This makes it easier to do math against working hours. The data in this table is time zone accurate for the schedule in the time zone of the workspace. Each minute within the begin and end date of the actual period must be a work minute. The master copy of the schedule is an iCal availability item stored in the Calendar service Calendar Item tables via the API. The data is redundantly and more conveniently stored so that the dashboard can use the information in the calculation of work hours elapsed.



time_zone_local_hour	Time Zone Local Hour		false	A bridge table that provides the UTC begin and End Hour datetimes for a local hour in an Olson timezone region. This table is used in Trend Reporting to give a high performance date calculation so that local hours and days can be used on graph axes and report heading with calculated totals. It is also used to load the summary tables using the local time dimension. High speed correct database-side time zone conversions are made by joining to this table.
time_zone_region	Time Zone Region		false	A region of the world defined in the Olson time zone database. Only the entries actually used by a FootPrints instance will be stored. The localized name of the region is provided by the Java.Util.Timezone getDisplayName() method. Only the java class can be used to determine the time zone offset from GMT, which varies historically and in the future. See the Time Zone Local Hour table. Take aspirin as needed.
wksp_cntr	Workspace Container	true	true	A workspace is the principle architectural structure in Numara FootPrints for storing and tracking information. A Workspace is a separate sub-database or work area within the system with its own fields, forms, options, workflow, settings and users. Most of the behaviour of the Workspace comes from the Container Abstract class, and only things that are specific to the Workspace are implemented here. For Example, Current FootPrints allows a single CMDB to be referenced from a Workspace, and a single Address book, yet these are listed as features for improvement. So convenience methods that get and set Container Group Relationship might be appropriate.

#### Attribute/Column(s) of Entity/Table "Agent"

Physical Name	Name	Definition	Physical Data Type
lic_type_id	License Type Id	A Foreign Key to the License Type table.	bigint
app_user_id	Application User Id	A Foreign Key to the Application User table.	bigint

#### Attribute/Column(s) of Entity/Table "Calendar Item"

Physical Name	Name	Definition	Physical Data Type
item_type	item type	item type.	varchar(16)
cal_item_id	Calendar Item Id	The automatically generated primary key of the cal_item table.	bigint
create_date	create date	create date, stored as a long integer to avoid using date time classes that do not handle timezone well.	int
etag	etag	etag.	varchar(255)
mdfy_date	modify date	modify date, stored as a long integer to avoid using date time classes that do not handle timezone well.	int
client_create_date	client create date	client create date, stored as a long integer to avoid using date time classes that do not handle timezone well.	int
client_mod_date	client modified date	client modified date, stored as a long integer to avoid using date time classes that do not handle timezone well.	int
display_name	display name	display name.	nvarchar(4000)
item_name	item name	item name.	nvarchar(4000)
intrnl_id	Internal Id	Internal Id.	varchar(255)
last_modf	last modification	The value of a Calendar Item, stored as a date.	int
last_mod_by	last modified by	last modified by.	nvarchar(4000)
needs_reply	needs reply	Indicates if needs reply.	bit
sent	sent	Indicates if sent.	bit
ical_uid	ical uid	ical uid.	varchar(255)
content_encoding	content encoding	content encoding.	varchar(32)
content_language	content language	content language.	varchar(32)
content_length	content length	The value of a Calendar Item stored as Long.	bigint

content_type	content type	content type.	varchar(64)
has_modifs	has modifications	Indicates if has modifications.	bit
cal_content_data_id	Calendar Content Data Id	A Foreign Key to the Calendar Content Data table.	bigint
app_user_id	Application User Id	A Foreign Key to the Application User table.	bigint
modifies_cal_item_id	Modifies Calendar Item Id	A Foreign Key to the Calendar Item table.	bigint
delete_date	Delete Date	The value of a Calendar Item, stored as a date.	datetime

#### Attribute/Column(s) of Entity/Table "Calendar Schedule"

Physical Name	Name	Definition	Physical Data Type
schd_id	Schedule Id	A Foreign Key to the Schedule table.	bigint
cal_schd_id	Calendar Schedule Id	The automatically generated primary key of the cal_schd table.	bigint
cal_item_id	Calendar Item Id	A Foreign Key to the Calendar Item table.	bigint

#### Attribute/Column(s) of Entity/Table "Calendar Type"

Physical Name	Name	Definition	Physical Data Type
cal_type_name	Calendar Type Name	The name of the Calendar Type.	varchar(255)
cal_type_id	Calendar Type Id	The automatically generated primary key of the cal_type table.	int

#### Attribute/Column(s) of Entity/Table "Schedule"

Physical Name	Name	Definition	Physical Data Type
schd_id	Schedule Id	The automatically generated primary key of the schd table.	bigint
schd_name	Schedule Name	The name of the Schedule.	varchar(255)
max_date_val	Maximum Date Value	The time spans are forecast out into the future as CalendarActualPeriod. This is the maximum date for which those periods are generated.	datetime
time_zone_region_name	Time Zone Region Name	The name of the Time Zone Region for the calendar, which derives from or overrides the value used for the system or workspace.	varchar(255)
cal_type_id	Calendar Type Id	A Foreign Key to the Calendar Type table.	int
app_user_id	Application User Id	A Foreign Key to the Application User table.	bigint
use_local_time	Use Local Time	Indicates that the workschedule is base on local time for the time zone region. If you want UTC time, use that as the region.	bit

#### Attribute/Column(s) of Entity/Table "Schedule Actual Period"

Physical Name	Name	Definition	Physical Data Type
schd_actual_period_id	Schedule Actual Period Id	The automatically generated primary key of the schd_actual_period table.	bigint
begin_date	Begin Date	Begin Date.	datetime
end_date	End Date	End Date.	datetime
work_durn_in_mins	Work Duration In Minutes	Work Time duration in minutes. This is the number of work minutes within the begin and end date of the period, which is equal to the end date minus the begin date, because the work periods can be sparsely populated. This value can be summed to get the total work minutes for a date range. If a date marker requested falls within the begin and end date of the actual period, then the minutes from the marker date and the begin/end point must be subtracted from this number.	int
schd_id	Schedule Id	A Foreign Key to the Schedule table.	bigint
cum_work_durn_in_mins	Cumulative Work Duration In Minutes	Cumulative Work Time duration in minutes. This is the number of work minutes from the actual first period to the end of the current actual period as calculated by a running total or cumulative sum of the Work Duration field. Because the work periods will be sparsely populated, the hard dates cannot be subtracted to yield Work Duration Difference. However, this Cumulative Work Duration can be subtracted to get the Work Duration difference. If a date marker requested falls within the begin and end date of the actual period, then the minutes from the marker date and the begin/end point must be subtracted from this	int

		number.	
begin_date_utc_s	Begin Date UTC S	A string version of the UTC date that is used in order to error correct JDBC/ Hibernate persistence of the date which produces errors on Daylight Savings Time boundaries.	varchar(23)
end_date_utc_s	End Date UTC S	A string version of the UTC date that is used in order to error correct JDBC/ Hibernate persistence of the date which produces errors on Daylight Savings Time boundaries.	varchar(23)

#### Attribute/Column(s) of Entity/Table "Time Zone Local Hour"

Physical Name	Name	Definition	Physical Data Type
begin_date_hour_utc	Begin Date Hour UTC	A date whole hour in the local timezone for the Begin Date Hour UTC. The Minutes portion of the date will always be 00. It is not stated whether daylight savings is in effect.	datetime
end_date_hour_utc	End Date Hour UTC	A date whole hour in the local timezone for the End Date Hour UTC. The Minutes portion of the date will always be 00. It is not stated whether daylight savings is in effect.	datetime
begin_date_hour_local	Begin Date Hour Local	A date time in UTC that cooresponds to a whole hour date in the local timezone for the Begin Date Hour Local. The Minutes portion of the date will usually be 00 but can be 30, 15, 45 in the case of half hour or quarter hour time zones. UTC never has daylight savings is in effect.	datetime
end_date_hour_local	End Date Hour Local	A date time in UTC that cooresponds to a whole hour date in the local timezone for the End Date Hour Local. The Minutes portion of the date will usually be 00 but can be 30, 15, 45 in the case of half hour or quarter hour time zones. UTC never has daylight savings is in effect.	datetime
begin_is_daylight_savings	Begin Is Daylight Savings	Flag indicating if daylight savings is in effect for the begin time. This helps identify the special hours that are missing or duplicated during daylight savings days.	bit
time_zone_region_id	Time Zone Region Id	An alternate Key to the Time Zone Region table, which is arbitrarily calculated and assigned by FootPrints. The reason for this is performance of the Foreign Key join to large numbers of rows. There is no suitable short data length key available in a published standard data source.	bigint
time_zone_local_hour_id	Time Zone Local Hour Id	The automatically generated primary key of the time_zone_local_hour table.	bigint
begin_date_hour_local_s	Begin Date Hour Local S	A string version of the local date that is used in order to error correct JDBC/ Hibernate persistence of the date which produces errors on Daylight Savings Time boundaries.	varchar(23)
end_date_hour_local_s	End Date Hour Local S	A string version of the local date that is used in order to error correct JDBC/ Hibernate persistence of the date which produces errors on Daylight Savings Time boundaries.	varchar(23)
begin_date_hour_utc_s	Begin Date Hour UTC S	A string version of the UTC date that is used in order to error correct JDBC/ Hibernate persistence of the date which produces errors on Daylight Savings Time boundaries.	varchar(23)
end_date_hour_utc_s	End Date Hour UTC S	A string version of the UTC date that is used in order to error correct JDBC/ Hibernate persistence of the date which produces errors on Daylight Savings Time boundaries.	varchar(23)
end_is_daylight_savings	End Is Daylight Savings	Flag indicating if daylight savings is in effect for the end time. This helps identify the special hours that are missing or duplicated during daylight savings days.	bit

#### Attribute/Column(s) of Entity/Table "Time Zone Region"

Physical Name	Name	Definition	Physical Data Type
time_zone_region_name	Time Zone Region Name	The name of the Time Zone Region. This is 2. The 'primary key' of the java timezone is the string region identifier in English, example 'America/Phoenix'. There is no numeric identifier. As the JRE is updated, or specific classes are updated, the list could change to include new entries, and the detail data behind the regions could change.	varchar(255)
time_zone_region_abbr	Time Zone Region Abbreviation	A well-known string identifier used as a code to identify the Time Zone Region. This could be provided by user entry, but it is not available from the Timezone class. It should probably not be used.	varchar(255)
time_zone_region_id	Time Zone Region Id	An alternate Key to the Time Zone Region table, which is arbitrarily calculated and assigned by FootPrints. The reason for this is performance of the Foreign Key join to large numbers of rows. There is no suitable short data length key available in a published standard data source.	bigint

#### Attribute/Column(s) of Entity/Table "Workspace Container"

Physical Name	Name	Definition	Physical Data Type
---------------	------	------------	--------------------

time_zone_region_name	Time Zone Region Name	The name of the Time Zone Region. This is 2. The 'primary key' of the java timezone is the string region identifier in English, example 'America/Phoenix'. There is no numeric identifier. As the JRE is updated, or specific classes are updated, the list could change to include new entries, and the detail data behind the regions could change.	varchar(255)
cntr_type_id	Container Type Id	A Foreign Key to the Container Type table.	int
cntr_id	Container Id	A Foreign Key to the Container table.	bigint

### ER Diagram(s) of Subject Area "141 - Calendar Appointments and Items"

Name	Physical Display Level
Logical	Column
Physical	Column

### Entity/Table(s) of Subject Area "141 - Calendar Appointments and Items"

Physical Name	Name	Logical Only	Do Not Generate	Definition
app_user	Application User		false	A user authorized to use the FootPrints system. The user is either a Customer User or an Agent User. The application user holds the minimum information needed to identify the user and associate the user with a user profile, where user settings are stored. A contact is not necessarily an application user. An application user entry is required in order for the user to access any online functions.
cal_coll_item	Calendar Collection Item		false	Holds calendar item collection relationships
cal_cntrnr_item	Calendar Container Item		false	The Container Item that is related to the Calendar Appointment Item. For Workspaces, the Workspace default calendar will have appointments between Issue Assignee agents and the Customer Contact in order to resolve the Issue/Workspace Item. It is called Container Item in case there is a capability to relate the appointment to a CMDB item or Knowledgebase Item.
cal_content_data	Calendar Content Data		false	Binary Large Objects (BLOBs) linked to an appointment item.
cal_event_stamp	Calendar Event Stamp		false	A log of calendar service events. A stamp associates a related set of properties/ attributes/methods to an Calendar Item. Event Stamp contains properties and apis relating to an iCal event.
cal_field	Calendar Field		false	A name/value data field value for a calendar item. Calendar Item has a map of Attribute objects, indexed by a QName . An Attribute can be associated to a single Item . QName is a composite key that represents a qualified name (namespace and local name). There are several Attribute implementations (String, Binary, Integer, etc) that are used to store different types of values. The polymorphic mapping allows any number of and any mix of Attribute implementations to be associated to an item.
cal_field_dict_val	Calendar Field Dictionary Value		false	Stores dictionary values for a calendar item. May not be used.
cal_field_multistring_val	Calendar Field Multistring Value		false	Stores multiple string values for a calendar item. May not be used. Has no primary key, its a bag of values.
cal_item	Calendar Item		false	The main calendar item table for the calendar service. This table is populated and maintained by the Calendar Service API. It is not very suitable for database query outside of the API. The calendar item can be a personal or workspace calendar, or an appointment within the calendar. The calendar item can be synched using the iCal format.

cal_schd	Calendar Schedule	false	false	The association between the FootPrints calendar instance, and the implementation of the calendar schedule and appointments. This enables the reuse of a schedule between workspaces, but it does not require it.
cal_stamp	Calendar Stamp		false	Calendar Item has a set of Calendar Stamp objects. A Calendar stamp can be associated to a single Calendar Item. A stamp associates a related set of properties/ attributes/methods to an Calendar Item. This is different from Calendar Attribute, which generally represents a single value. For example Event Stamp, contains properties and apis relating to an event and MessageStamp contains properties related to an email message. The stamp model allows items to be stamped and unstamped with these related sets of properties throughout its life-cycle.
cal_subscription	Calendar Subscription		false	A Calendar Subscription contains a ticket key and a collection uid. It is a way to associate a collection that is not owned by the user to the user. The ticket key provides the necessary authentication to access the collection. Not used.
cal_ticket	Calendar Ticket		false	This is unrelated and different than an ITSM ticket. See the Calendar Container Item table for ITSM tickets. A calendar ticket is an authorization object that grants some level of access to a calendar item and its descendants based on a unique ticket key. If a client has this key, then the client can gain access to the item and its descendants with the ticket key. May not be used in the FootPrints integration of calendaring code, because the calendar permissions are subordinate to the application security model.
cal_ticket_privilege	Calendar Ticket Privilege		false	(Not used, but referenced in Java code that must be removed before table deletion.) A permission granted via a calendar ticket. May not be used in the FootPrints integration of calendaring code.
cal_wksp	Calendar Workspace	false	false	The association of a workspace to a calendar. The workspace definition is associated with the Calendar Item that is the root entry for a recurring schedule item in the calendar service tables. The workspace can have one default work schedule, and it may have a second shift.
defn	Definition		false	A definition holds the structural and behavioral information needed for an application element that is configurable by the system designer and/or the end user at definition time. Current Footprints uses a combination of directories, files, tables, and programming code to represent this information. The new FootPrints will provide all of the same information within the definition. The definition can be persisted in a structure that is isolated at the Instance Level. The data for the definition will be stored as Configuration Settings associated with the Definition Version. See the database views v_defn_ver_active, v_cntrn_item_field.

item	Item	true	true	The Item is the abstract class for Issue, Configuration Item, Address Book entry, and other information items that are the main information entities in FootPrints. The Item structure and behavior is defined by its Definition. The Item is namespaced, identified by, and logically stored in the single parent container. It can be referenced in the same manner both inside and outside the container. The Item is the storage model and the implementation for the Item Definition. There are many important specializations of the Item class, and there even more Template definitions of Container/Item that provide an end-user application experience to the End-User. The Item implementation must implement flexible schema in an easy to manage, well engineered, and high performing way. The item can store user-defined and user-named fields defined in its definition. The Item could implement the nice-to-have features such as an inheritance model for Item subtypes, and have the ability to represent collections of subordinate items. The Item data can be imported, exported, or synchronized with external data sources. The Item data can be linked and synchronized with other items in the same Instance.
------	------	------	------	--

#### Attribute/Column(s) of Entity/Table "Application User"

Physical Name	Name	Definition	Physical Data Type
pwd	Password	This space intentionally left blank.	varchar(255)
user_name	User Name	The name of the Application User.	nvarchar(4000)
cust_login_type_id	Customer Login Type Id	A Foreign Key to the Customer Login Type table.	int
app_user_id	Application User Id	The automatically generated primary key of the app_user table.	bigint
lic_type_id	License Type Id	A Foreign Key to the License Type table.	bigint
user_login	User Login	The login name used to access the Application. This may not be filled in if external application security is used.	nvarchar(4000)
email_addr	Email Address	An email address for the Application User.	nvarchar(4000)
delete_date	Delete Date	The value of a Application User, stored as a date.	datetime

#### Attribute/Column(s) of Entity/Table "Calendar Collection Item"

Physical Name	Name	Definition	Physical Data Type
create_date	create date	create date, stored as a long integer to avoid using date time classes that do not handle timezone well.	int
reln_type	Relation Type	The value of a Calendar Collection Item, stored as a string.	varchar(16)
coll_cal_item	Collection Calendar item	A Foreign Key to the Calendar Item table.	bigint
cal_item_id	Calendar Item Id	A Foreign Key to the Calendar Item table.	bigint

#### Attribute/Column(s) of Entity/Table "Calendar Container Item"

Physical Name	Name	Definition	Physical Data Type
cal_cntrn_item_id	Calendar Container Item Id	The automatically generated primary key of the cal_cntrn_item table.	bigint
item_defn_id	Item Definition Id	A Foreign Key to the Definition table. Identifies the Container Definition which in turn identifies the dynamic object and table that contains the Item. This is the first part of a two-part identifier for the dynamic item.	bigint
item_id	Item Id	A logical Foreign Key to the Item table. It is an actual primary key in the dynamic object and table that contains the Item. This is the second part of a two-part identifier for the dynamic item.	bigint
appt_cal_item_id	Appointment Calendar Item Id	A Foreign Key to the Calendar Item table. This is an appointment type of item that has been set up related to the Workspace/Container Item.	bigint

field_defn_id	Field Definition Id	A Foreign Key to the Definition table.	bigint
is_original	Is Original	Indicates if this is the original container for the item.	bit
occurence_uid	Occurence UID	The value of a Calendar Container Item, stored as a string.	varchar(255)

#### Attribute/Column(s) of Entity/Table "Calendar Content Data"

Physical Name	Name	Definition	Physical Data Type
cal_content_data_id	Calendar Content Data Id	The automatically generated primary key of the cal_content_data table.	bigint
content_blob	content_blob		OID

#### Attribute/Column(s) of Entity/Table "Calendar Event Stamp"

Physical Name	Name	Definition	Physical Data Type
ical_data	ical data	The value of a Calendar Event Stamp, stored as a long text object.	varchar(max)
end_date	end date	end date.	varchar(255)
is_floating	is floating	Indicates if is floating.	bit
is_recurring	is recurring	Indicates if is recurring.	bit
start_date	start date	start date.	varchar(255)
cal_stamp_id	Calendar Stamp Id	A Foreign Key to the Calendar Stamp table.	bigint

#### Attribute/Column(s) of Entity/Table "Calendar Field"

Physical Name	Name	Definition	Physical Data Type
attr_type	attribute type	attribute type.	varchar(255)
cal_field_id	Calendar Field Id	The automatically generated primary key of the cal_field table.	bigint
create_date	create date	create date, stored as a long integer to avoid using date time classes that do not handle timezone well.	int
etag	etag	etag.	varchar(255)
mdfy_date	modify date	modify date, stored as a long integer to avoid using date time classes that do not handle timezone well.	int
local_name	local name	local name.	varchar(25)
nmsp	namespace	namespace.	varchar(220)
binary_val	binary value	Object Identifier binary value	OID
boolean_val	boolean value	generic boolean value.	bit
date_val	date value	generic date value	datetime
tz_val	tz value	time zone value.	varchar(255)
decimal_val	decimal value	generic decimal value	numeric(19,6)
text_val	text value	The value of a Calendar Field, stored as a long text object.	varchar(max)
integer_val	integer value	The value of a Calendar Field stored as Long.	bigint
string_val	string value	string value.	varchar(255)
cal_item_id	Calendar Item Id	A Foreign Key to the Calendar Item table.	bigint

#### Attribute/Column(s) of Entity/Table "Calendar Field Dictionary Value"

Physical Name	Name	Definition	Physical Data Type
cal_field_id	Calendar Field Id	A Foreign Key to the Calendar Field table.	bigint
string_val	string value	string value.	varchar(2048)
val_key	value_key	value_key.	varchar(255)

#### Attribute/Column(s) of Entity/Table "Calendar Field Multistring Value"

Physical Name	Name	Definition	Physical Data Type
cal_field_id	Calendar Field Id	A Foreign Key to the Calendar Field table.	bigint
string_val	string value	string value.	varchar(2048)

#### Attribute/Column(s) of Entity/Table "Calendar Item"

Physical Name	Name	Definition	Physical Data Type
item_type	item type	item type.	varchar(16)
cal_item_id	Calendar Item Id	The automatically generated primary key of the cal_item table.	bigint

create_date	create date	create date, stored as a long integer to avoid using date time classes that do not handle timezone well.	int
etag	etag	etag.	varchar(255)
mdfy_date	modify date	modify date, stored as a long integer to avoid using date time classes that do not handle timezone well.	int
client_create_date	client create date	client create date, stored as a long integer to avoid using date time classes that do not handle timezone well.	int
client_mod_date	client modified date	client modified date, stored as a long integer to avoid using date time classes that do not handle timezone well.	int
display_name	display name	display name.	nvarchar(4000)
item_name	item name	item name.	nvarchar(4000)
intrnl_id	Internal Id	Internal Id.	varchar(255)
last_modf	last modification	The value of a Calendar Item, stored as a date.	int
last_mod_by	last modified by	last modified by.	nvarchar(4000)
needs_reply	needs reply	Indicates if needs reply.	bit
sent	sent	Indicates if sent.	bit
ical_uid	ical uid	ical uid.	varchar(255)
content_encoding	content encoding	content encoding.	varchar(32)
content_language	content language	content language.	varchar(32)
content_length	content length	The value of a Calendar Item stored as Long.	bigint
content_type	content type	content type.	varchar(64)
has_modfs	has modifications	Indicates if has modifications.	bit
cal_content_data_id	Calendar Content Data Id	A Foreign Key to the Calendar Content Data table.	bigint
app_user_id	Application User Id	A Foreign Key to the Application User table.	bigint
modifies_cal_item_id	Modifies Calendar Item Id	A Foreign Key to the Calendar Item table.	bigint
delete_date	Delete Date	The value of a Calendar Item, stored as a date.	datetime

#### Attribute/Column(s) of Entity/Table "Calendar Schedule"

Physical Name	Name	Definition	Physical Data Type
schd_id	Schedule Id	A Foreign Key to the Schedule table.	bigint
cal_schd_id	Calendar Schedule Id	The automatically generated primary key of the cal_schd table.	bigint
cal_item_id	Calendar Item Id	A Foreign Key to the Calendar Item table.	bigint

#### Attribute/Column(s) of Entity/Table "Calendar Stamp"

Physical Name	Name	Definition	Physical Data Type
stamp_type	stamp type	stamp type.	varchar(16)
cal_stamp_id	Calendar Stamp Id	The automatically generated primary key of the cal_stamp table.	bigint
create_date	create date	create date, stored as a long integer to avoid using date time classes that do not handle timezone well.	int
etag	etag	etag.	varchar(255)
mdfy_date	modify date	modify date, stored as a long integer to avoid using date time classes that do not handle timezone well.	int
cal_item_id	Calendar Item Id	A Foreign Key to the Calendar Item table.	bigint

#### Attribute/Column(s) of Entity/Table "Calendar Subscription"

Physical Name	Name	Definition	Physical Data Type
cal_subscription_id	Calendar Subscription Id	The automatically generated primary key of the cal_subscription table.	bigint
create_date	create date	create date	int
etag	etag	etag	varchar(255)
mdfy_date	modify date	modify date	int
coll_uid	collection uid	id of the collection	varchar(255)
ticket_key	ticket key	permissions	varchar(255)
display_name	display name	Name of the calendar in the UI	varchar(255)
app_user_id	Application User Id	A Foreign Key to the Application User table.	bigint

#### Attribute/Column(s) of Entity/Table "Calendar Ticket"

Physical Name	Name	Definition	Physical Data Type
---------------	------	------------	--------------------



cal_ticket_id	Calendar Ticket Id	The automatically generated primary key of the cal_ticket table.	bigint
create_date	Create Date	Create date.	int
etag	etag	The value of a Calendar Ticket, stored as a string.	varchar(255)
mdfy_date	Modify Date	Change date.	int
creatr_date	Creation Date	Create date.	datetime
ticket_key	Ticket Key	Authorization id.	varchar(255)
app_user_id	Application User Id	A Foreign Key to the Application User table.	bigint
ticket_timeout	Ticket Timeout	Expiration of authority.	varchar(255)
cal_item_id	Calendar Item Id	A Foreign Key to the Calendar Item table.	bigint

#### Attribute/Column(s) of Entity/Table "Calendar Ticket Privilege"

Physical Name	Name	Definition	Physical Data Type
cal_ticket_id	Calendar Ticket Id	A Foreign Key to the Calendar Ticket table.	bigint
privilege	Privilege	The value of a Calendar Ticket Privilege, stored as a string.	varchar(255)

#### Attribute/Column(s) of Entity/Table "Calendar Workspace"

Physical Name	Name	Definition	Physical Data Type
cal_wksp_id	Calendar Workspace Id	The automatically generated primary key of the cal_wksp table.	bigint
wksp_defn_id	Workspace Definition Id	A Foreign Key to the Definition table. Identifies the Workspace that owns the Calendar.	bigint
cal_item_id	Calendar Item Id	A Foreign Key to the Calendar Item table. This is a Workspace Calendar type of Calendar Item that owns appointments.	bigint
is_dflt_flag	Is Default Flag	Indicates if Is Default Flag.	bit

#### Attribute/Column(s) of Entity/Table "Definition"

Physical Name	Name	Definition	Physical Data Type
defn_id	Definition Id	The automatically generated primary key of the defn table.	bigint
defn_name	Definition Name	The name of the Definition.	nvarchar(4000)
dsble_date	Disabled Date	The date that this instance of Definition is logically deleted and disabled.	datetime
defn_guid	Definition GUID	A Global Unique Identifier search key for the Definition, composed of letters and numbers. It is computed by the client and typically used in unstructured data.	varchar(50)
defn_type_name	Definition Type Name	The unique string identifier for an instance of Definition Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)
defn_subtype_id	Definition Subtype Id	A Foreign Key to the Definition Subtype table.	int
tmpl_export_guid	Template Export GUID	The value of a Definition, stored as a string.	VARCHAR(50)

#### Attribute/Column(s) of Entity/Table "Item"

Physical Name	Name	Definition	Physical Data Type
item_key	Item Key	A search key for the Item, composed of letters and numbers.	varchar(255)
item_id	Item Id	The automatically generated primary key of the item table.	bigint
cntrn_id	Container Id	A Foreign Key to the Container table.	bigint
item_defn_id	Item Definition Id	A Foreign Key to the Definition table for an instance of type Definition.	bigint

#### ER Diagram(s) of Subject Area "142 - Calendar Sync"

Name	Physical Display Level
Logical	Column
Physical	Column

#### Entity/Table(s) of Subject Area "142 - Calendar Sync"

Physical Name	Name	Logical Only	Do Not Generate	Definition
---------------	------	--------------	-----------------	------------

app_user	Application User		false	A user authorized to use the FootPrints system. The user is either a Customer User or an Agent User. The application user holds the minimum information needed to identify the user and associate the user with a user profile, where user settings are stored. A contact is not necessarily an application user. An application user entry is required in order for the user to access any online functions.
cal_item	Calendar Item		false	The main calendar item table for the calendar service. This table is populated and maintained by the Calendar Service API. It is not very suitable for database query outside of the API. The calendar item can be a personal or workspace calendar, or an appointment within the calendar. The calendar item can be synched using the iCal format.
defn	Definition		false	A definition holds the structural and behavioral information needed for an application element that is configurable by the system designer and/or the end user at definition time. Current Footprints uses a combination of directories, files, tables, and programming code to represent this information. The new FootPrints will provide all of the same information within the definition. The definition can be persisted in a structure that is isolated at the Instance Level. The data for the definition will be stored as Configuration Settings associated with the Definition Version. See the database views v_defn_ver_active, v_cntrn_item_field.
pim_appt_info	PIM Appointment Information		false	Information about calendar appointment synchronization with external calendars in a Personal Information Manager (PIM) such as a computer or phone calendar system.
pim_record_info	PIM Record Information		false	Information about calendar appointment data (such as the workspace and ticket number) synchronization with external calendars in a Personal Information Manager (PIM) such as a computer or phone calendar system.
syncing_job_dtl	Syncing Job Detail		false	A table used to control syncing of Calendar Items with a computer or phone calendar system.

#### Attribute/Column(s) of Entity/Table "Application User"

Physical Name	Name	Definition	Physical Data Type
pwd	Password	This space intentionally left blank.	varchar(255)
user_name	User Name	The name of the Application User.	nvarchar(4000)
cust_login_type_id	Customer Login Type Id	A Foreign Key to the Customer Login Type table.	int
app_user_id	Application User Id	The automatically generated primary key of the app_user table.	bigint
lic_type_id	License Type Id	A Foreign Key to the License Type table.	bigint
user_login	User Login	The login name used to access the Application. This may not be filled in if external application security is used.	nvarchar(4000)
email_addr	Email Address	An email address for the Application User.	nvarchar(4000)
delete_date	Delete Date	The value of a Application User, stored as a date.	datetime

#### Attribute/Column(s) of Entity/Table "Calendar Item"

Physical Name	Name	Definition	Physical Data Type
item_type	item type	item type.	varchar(16)
cal_item_id	Calendar Item Id	The automatically generated primary key of the cal_item table.	bigint
create_date	create date	create date, stored as a long integer to avoid using date time classes that do not handle timezone well.	int
etag	etag	etag.	varchar(255)
mdfy_date	modify date	modify date, stored as a long integer to avoid using date time classes that do not handle timezone well.	int

client_create_date	client create date	client create date, stored as a long integer to avoid using date time classes that do not handle timezone well.	int
client_mod_date	client modified date	client modified date, stored as a long integer to avoid using date time classes that do not handle timezone well.	int
display_name	display name	display name.	nvarchar(4000)
item_name	item name	item name.	nvarchar(4000)
intrnl_id	Internal Id	Internal Id.	varchar(255)
last_modf	last modification	The value of a Calendar Item, stored as a date.	int
last_mod_by	last modified by	last modified by.	nvarchar(4000)
needs_reply	needs reply	Indicates if needs reply.	bit
sent	sent	Indicates if sent.	bit
ical_uid	ical uid	ical uid.	varchar(255)
content_encoding	content encoding	content encoding.	varchar(32)
content_language	content language	content language.	varchar(32)
content_length	content length	The value of a Calendar Item stored as Long.	bigint
content_type	content type	content type.	varchar(64)
has_modfs	has modifications	Indicates if has modifications.	bit
cal_content_data_id	Calendar Content Data Id	A Foreign Key to the Calendar Content Data table.	bigint
app_user_id	Application User Id	A Foreign Key to the Application User table.	bigint
modifies_cal_item_id	Modifies Calendar Item Id	A Foreign Key to the Calendar Item table.	bigint
delete_date	Delete Date	The value of a Calendar Item, stored as a date.	datetime

#### Attribute/Column(s) of Entity/Table "Definition"

Physical Name	Name	Definition	Physical Data Type
defn_id	Definition Id	The automatically generated primary key of the defn table.	bigint
defn_name	Definition Name	The name of the Definition.	nvarchar(4000)
dsble_date	Disabled Date	The date that this instance of Definition is logically deleted and disabled.	datetime
defn_guid	Definition GUID	A Global Unique Identifier search key for the Definition, composed of letters and numbers. It is computed by the client and typically used in unstructured data.	varchar(50)
defn_type_name	Definition Type Name	The unique string identifier for an instance of Definition Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)
defn_subtype_id	Definition Subtype Id	A Foreign Key to the Definition Subtype table.	int
tmpl_export_guid	Template Export GUID	The value of a Definition, stored as a string.	VARCHAR(50)

#### Attribute/Column(s) of Entity/Table "PIM Appointment Information"

Physical Name	Name	Definition	Physical Data Type
pim_appt_info_id	PIM Appointment Information Id	The automatically generated primary key of the pim_appt_info table.	bigint
extl_id	External Id	The value of a PIM Appointment Information stored as Long.	varchar(255)
chg_key	Change Key	The value of a PIM Appointment Information, stored as a string.	varchar(255)
app_user_id	Application User Id	A Foreign Key to the Application User table.	bigint
cal_item_intrnl_id	Calendar Item Internal Id	Calendar Item Internal Id.	varchar(255)

#### Attribute/Column(s) of Entity/Table "PIM Record Information"

Physical Name	Name	Definition	Physical Data Type
pim_record_info_id	PIM Record Information Id	The automatically generated primary key of the pim_record_info table.	bigint
extl_id	External Id	The value of a PIM Record Information stored as Long.	varchar(255)
chg_key	Change Key	The value of a PIM Record Information, stored as a string.	varchar(255)

item_defn_id	Item Definition Id	A Foreign Key to the Definition table.	bigint
item_id	Item Id	The value of a PIM Record Information stored as Long.	bigint
app_user_id	Application User Id	A Foreign Key to the Application User table.	bigint

#### Attribute/Column(s) of Entity/Table "Syncing Job Detail"

Physical Name	Name	Definition	Physical Data Type
syncing_job_dtl_id	Syncing Job Detail Id	The automatically generated primary key of the syncing_job_dtl table.	bigint
app_user_id	Application User Id	A Foreign Key to the Application User table.	bigint
last_sync_date	Last Sync Date	The value of a Syncing Job Detail, stored as a date.	datetime
execn_date	Execution Date	The value of a Syncing Job Detail, stored as a date.	datetime

#### ER Diagram(s) of Subject Area "160 - Templating"

Name	Physical Display Level
Logical	Column
Physical	Column

#### Entity/Table(s) of Subject Area "160 - Templating"

Physical Name	Name	Logical Only	Do Not Generate	Definition
app_ele	Application Element		false	The identity and name of an element in the application implementation. An element can be one of many types, and can contain other elements. Example Application Elements include: a screen, a form, a dialog box, a text box, the Home Page, an Architecture Component, the Item Status field, the system values of the Item Status field. The Application Element is not the full implementation of the element, but is a key and nametag for the element. In many cases, a well-known algorithm can be used to locate the correct Application Element from the Application Element implementation. In other cases, an association to the Application Element Id must be recorded as part of the implementation.
defn	Definition		false	A definition holds the structural and behavioral information needed for an application element that is configurable by the system designer and/or the end user at definition time. Current Footprints uses a combination of directories, files, tables, and programming code to represent this information. The new FootPrints will provide all of the same information within the definition. The definition can be persisted in a structure that is isolated at the Instance Level. The data for the definition will be stored as Configuration Settings associated with the Definition Version. See the database views v_defn_ver_active, v_cntrn_item_field.
item_field_defn	Item Field Definition	true	true	Item Field Definition is an association between the Item Definition entity and the Field Definition entity. This association contains a distinct value for each of the intersection points between the two entities.
tmpl_attr_defn	Template Attribute Definition		false	A list of substitution attributes that can be used in an email template.
tmpl_ctgy	Template Category		false	A category of template, for example Email Template Definition.
tmpl_content_type	Template Content Type		false	A lookup table for the type of template content. Values include: PLAINTEXT, HTML.
tmpl_defn	Template Definition		false	A template used for notifications or description content with substitution attributes.
tmpl_defn_attr	Template Definition Attribute		false	The collection of template attributes associated with an instance of template definition.

tmpl_grp	Template Group		false	A group of templates.
----------	----------------	--	-------	-----------------------

#### Attribute/Column(s) of Entity/Table "Application Element"

Physical Name	Name	Definition	Physical Data Type
app_ele_id	Application Element Id	The automatically generated primary key of the app_ele table.	bigint
app_ele_name	Application Element Name	The name of the Application Element.	VARCHAR(2000)
intrnl_id	Internal Id	An alternate automatically generated key of the app_ele table.	int
is_intrnl_flag	Is Internal Flag	A boolean flag for indicating if the Application Element is defined internally by Engineering or by Partners or is defined by the customer.	bit
app_ele_type_id	Application Element Type Id	A Foreign Key to the Application Element Type table.	int
intrnl_id_type_id	Internal Identifier Type Id	A Foreign Key to the Internal Identifier Type table.	int
app_ele_nmsp_id	Application Element Namespace Id	A Foreign Key to the Application Element Namespace table.	bigint

#### Attribute/Column(s) of Entity/Table "Definition"

Physical Name	Name	Definition	Physical Data Type
defn_id	Definition Id	The automatically generated primary key of the defn table.	bigint
defn_name	Definition Name	The name of the Definition.	nvarchar(4000)
dsble_date	Disabled Date	The date that this instance of Definition is logically deleted and disabled.	datetime
defn_guid	Definition GUID	A Global Unique Identifier search key for the Definition, composed of letters and numbers. It is computed by the client and typically used in unstructured data.	varchar(50)
defn_type_name	Definition Type Name	The unique string identifier for an instance of Definition Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)
defn_subtype_id	Definition Subtype Id	A Foreign Key to the Definition Subtype table.	int
tmpl_export_guid	Template Export GUID	The value of a Definition, stored as a string.	VARCHAR(50)

#### Attribute/Column(s) of Entity/Table "Item Field Definition"

Physical Name	Name	Definition	Physical Data Type
item_field_defn_id	Item Field Definition Id	A Foreign Key to the Definition table for an instance of type Definition.	bigint
item_defn_id	Item Definition Id	A Foreign Key to the Definition table for an instance of type Definition.	bigint
field_defn_id	Field Definition Id	A Foreign Key to the Definition table.	bigint

#### Attribute/Column(s) of Entity/Table "Template Attribute Definition"

Physical Name	Name	Definition	Physical Data Type
tmpl_attr_defn_name	Template Attribute Definition Name	The value of a Template Attribute Definition, stored as a string.	varchar(255)
is_intrnl	Is Internal	Indicates if Is Internal.	bit
tmpl_ctgy_id	Template Category Id	A Foreign Key to the Template Category table.	bigint
tmpl_attr_defn_id	Template Attribute Definition Id	The automatically generated primary key of the tmpl_attr_defn table.	bigint
tmpl_attr_defn_keyword	Template Attribute Definition Keyword	The value of a Template Attribute Definition, stored as a string.	varchar(255)

#### Attribute/Column(s) of Entity/Table "Template Category"

Physical Name	Name	Definition	Physical Data Type
tmpl_ctgy_id	Template Category Id	The automatically generated primary key of the tmpl_ctgy table.	bigint
tmpl_ctgy_name	Template Category Name	The name of the Template Category.	varchar(255)
tmpl_ctgy_descr	Template Category Description	A descriptive name of the Template Category. This descriptive name may be localized.	varchar(512)

#### Attribute/Column(s) of Entity/Table "Template Content Type"

Physical Name	Name	Definition	Physical Data Type
---------------	------	------------	--------------------

tmpl_content_type_name	Template Content Type Name	The name of the Template Content Type.	varchar(255)
tmpl_content_type_id	Template Content Type Id	The automatically generated primary key of the tmpl_content_type table.	int
tmpl_content_type_descr	Template Content Type Description	A descriptive name of the Template Content Type. This descriptive name may be localized.	varchar(512)

#### Attribute/Column(s) of Entity/Table "Template Definition"

Physical Name	Name	Definition	Physical Data Type
tmpl_defn_id	Template Definition Id	The automatically generated primary key of the tmpl_defn table.	bigint
tmpl_defn_type	Template Definition Type	The Type of template.	varchar(255)
tmpl_content	Template Content	The value of a Template Definition, stored as a long text object.	varchar(max)
is_imported	Is_Imported	Indicates if Is_Imported.	bit
is_editable	Is Editable	Indicates if Is Editable.	bit
tmpl_grp_id	Template Group Id	A Foreign Key to the Template Group table.	bigint
tmpl_ctgy_id	Template Category Id	A Foreign Key to the Template Category table.	bigint
tmpl_content_type_id	Template Content Type Id	A Foreign Key to the Template Content Type table.	int
is_sys_tmpl	Is System Template	Indicates if Is System Template.	bit

#### Attribute/Column(s) of Entity/Table "Template Definition Attribute"

Physical Name	Name	Definition	Physical Data Type
tmpl_defn_id	Template Definition Id	A Foreign Key to the Template Definition table.	bigint
tmpl_attr_defn_id	Template Attribute Definition Id	The automatically generated primary key of the tmpl_attr_defn table.	bigint

#### Attribute/Column(s) of Entity/Table "Template Group"

Physical Name	Name	Definition	Physical Data Type
tmpl_grp_id	Template Group Id	The automatically generated primary key of the tmpl_grp table.	bigint
tmpl_grp_name	Template Group Name	The name of the Template Group.	varchar(255)
tmpl_grp_descr	Template Group Description	A descriptive name of the Template Group. This descriptive name may be localized.	varchar(512)

#### ER Diagram(s) of Subject Area "170 - Knowledge Base"

Name	Physical Display Level
Logical	Column
Physical	Column

#### Entity/Table(s) of Subject Area "170 - Knowledge Base"

Physical Name	Name	Logical Only	Do Not Generate	Definition
app_user	Application User		false	A user authorized to use the FootPrints system. The user is either a Customer User or an Agent User. The application user holds the minimum information needed to identify the user and associate the user with a user profile, where user settings are stored. A contact is not necessarily an application user. An application user entry is required in order for the user to access any online functions.
defn	Definition		false	A definition holds the structural and behavioral information needed for an application element that is configurable by the system designer and/or the end user at definition time. Current Footprints uses a combination of directories, files, tables, and programming code to represent this information. The new FootPrints will provide all of the same information within the definition. The definition can be persisted in a structure that is isolated at the Instance Level. The data for the definition will be stored as Configuration Settings associated with the Definition Version. See the database views v_defn_ver_active, v_cntrn_item_field.

item	Item	true	true	The Item is the abstract class for Issue, Configuration Item, Address Book entry, and other information items that are the main information entities in FootPrints. The Item structure and behavior is defined by its Definition. The Item is namespaced, identified by, and logically stored in the single parent container. It can be referenced in the same manner both inside and outside the container. The Item is the storage model and the implementation for the Item Definition. There are many important specializations of the Item class, and there even more Template definitions of Container/Item that provide an end-user application experience to the End-User. The Item implementation must implement flexible schema in an easy to manage, well engineered, and high performing way. The item can store user-defined and user-named fields defined in its definition. The Item could implement the nice-to-have features such as an inheritance model for Item subtypes, and have the ability to represent collections of subordinate items. The Item data can be imported, exported, or synchronized with external data sources. The Item data can be linked and synchronized with other items in the same Instance.
item_defn	Item Definition	true	true	The definition of an Item, which includes the Item Name and localizations and the collection of Item Fields. It also includes customizations and localizations of string resources for Item Names, and Item Field names.
item_useful_dtl	Item Useful Detail		false	An entry from an agent or customer classifying the quality and usefulness of an knowledgebase item or other item.
item_useful_sumry	Item Useful Summary		false	A standalone table holding a summary count of the usefulness of an Item, computed by summing the detail.
soln_item	Solution Item	true	true	An item that stores the solution information for a particular issue.
useful_cd	Useful Code		false	A lookup table for the useful code used in grading solutions. Examples include: I - Yes, but inaccurate, N - No, Y - Yes

#### Attribute/Column(s) of Entity/Table "Application User"

Physical Name	Name	Definition	Physical Data Type
pwd	Password	This space intentionally left blank.	varchar(255)
user_name	User Name	The name of the Application User.	nvarchar(4000)
cust_login_type_id	Customer Login Type Id	A Foreign Key to the Customer Login Type table.	int
app_user_id	Application User Id	The automatically generated primary key of the app_user table.	bigint
lic_type_id	License Type Id	A Foreign Key to the License Type table.	bigint
user_login	User Login	The login name used to access the Application. This may not be filled in if external application security is used.	nvarchar(4000)
email_addr	Email Address	An email address for the Application User.	nvarchar(4000)
delete_date	Delete Date	The value of a Application User, stored as a date.	datetime

#### Attribute/Column(s) of Entity/Table "Definition"

Physical Name	Name	Definition	Physical Data Type
defn_id	Definition Id	The automatically generated primary key of the defn table.	bigint
defn_name	Definition Name	The name of the Definition.	nvarchar(4000)
dsble_date	Disabled Date	The date that this instance of Definition is logically deleted and disabled.	datetime
defn_guid	Definition GUID	A Global Unique IDentifier search key for the Definition, composed of letters and numbers. It is computed by the client and typically used in unstructured data.	varchar(50)

defn_type_name	Definition Type Name	The unique string identifier for an instance of Definition Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)
defn_subtype_id	Definition Subtype Id	A Foreign Key to the Definition Subtype table.	int
tmpl_export_guid	Template Export GUID	The value of a Definition, stored as a string.	VARCHAR(50)

#### Attribute/Column(s) of Entity/Table "Item"

Physical Name	Name	Definition	Physical Data Type
item_key	Item Key	A search key for the Item, composed of letters and numbers.	varchar(255)
item_id	Item Id	The automatically generated primary key of the item table.	bigint
cntnr_id	Container Id	A Foreign Key to the Container table.	bigint
item_defn_id	Item Definition Id	A Foreign Key to the Definition table for an instance of type Definition.	bigint

#### Attribute/Column(s) of Entity/Table "Item Definition"

Physical Name	Name	Definition	Physical Data Type
item_table_name	Item Table Name	The name of the DBMS_Object_Name. It is a database object name, limited to 30 characters in length.	varchar(30)
item_defn_id	Item Definition Id	A Foreign Key to the Definition table for an instance of type Definition.	bigint

#### Attribute/Column(s) of Entity/Table "Item Useful Detail"

Physical Name	Name	Definition	Physical Data Type
item_useful_dtl_id	Item Useful Detail Id	The automatically generated primary key of the item_useful_dtl table.	bigint
app_user_id	Application User Id	A Foreign Key to the Application User table.	bigint
user_comment	User Comment	A free-form text comment describing the Item Useful Detail entry made by the application user.	nvarchar(4000)
create_date	Create Date	The date that the Item Useful Detail was created.	datetime
item_defn_id	Item Definition Id	A Foreign Key to the Definition table. Identifies the Container and Item Type.	bigint
item_id	Item Id	A Foreign Key to the generated DYNAMIC Item table key. Not database enforced.	bigint
useful_cd	Useful Code	Indicates the level of usefulness recorded by the user. A Foreign Key to the lookup table.	varchar(2)

#### Attribute/Column(s) of Entity/Table "Item Useful Summary"

Physical Name	Name	Definition	Physical Data Type
item_useful_sumry_id	Item Useful Summary Id	The automatically generated primary key of the item_useful_sumry table.	bigint
yes_ct	Yes Count	The summarized count of Yes responses, calculated from the detail table as stored here for convenience.	int
item_defn_id	Item Definition Id	A Foreign Key to the Definition table. Identifies the Container and Item Type.	bigint
inaccr_ct	Inaccurate Count	The summarized count of Inaccurate responses, calculated from the detail table as stored here for convenience. Perl FP does not show this column, it could be used to search for inaccurate items.	int
no_ct	No Count	The summarized count of No responses, calculated from the detail table as stored here for convenience. Perl FP does not show this column, it could be used to calculate a usefulness percentage.	int
item_id	Item Id	A Foreign Key to the generated DYNAMIC Item table key. Not database enforced.	bigint

#### Attribute/Column(s) of Entity/Table "Solution Item"

Physical Name	Name	Definition	Physical Data Type
item_id	Item Id	A Foreign Key to the Item table.	bigint

#### Attribute/Column(s) of Entity/Table "Useful Code"

Physical Name	Name	Definition	Physical Data Type
---------------	------	------------	--------------------



useful_cd_descr	Useful Code Description	A descriptive name of the Useful Code. This descriptive name may be localized.	varchar(512)
useful_cd	Useful Code	The name of the Useful Code.	varchar(2)

### ER Diagram(s) of Subject Area "190 - Password Reset"

Name	Physical Display Level
Logical	Column
Physical	Column

### Entity/Table(s) of Subject Area "190 - Password Reset"

Physical Name	Name	Logical Only	Do Not Generate	Definition
app_user	Application User		false	A user authorized to use the FootPrints system. The user is either a Customer User or an Agent User. The application user holds the minimum information needed to identify the user and associate the user with a user profile, where user settings are stored. A contact is not necessarily an application user. An application user entry is required in order for the user to access any online functions.
pwd_reset_question	Password Reset Question		false	Questions, once they've been saved, can only be disabled (deactivated). Users who have already used those questions as part of their password reset registration will still be able to use the disabled questions.
pwd_reset_question_answer	Password Reset Question Answer		false	This table holds the users' answers to the system questions set by the administrators and the Id to the actual question. Users can answer questions even after the administrator has disabled them, as long as they were available when the user first registered the answer.
pwd_reset_regtd_acct	Password Reset Registered Account		false	This holds LDAP information.

### Attribute/Column(s) of Entity/Table "Application User"

Physical Name	Name	Definition	Physical Data Type
pwd	Password	This space intentionally left blank.	varchar(255)
user_name	User Name	The name of the Application User.	nvarchar(4000)
cust_login_type_id	Customer Login Type Id	A Foreign Key to the Customer Login Type table.	int
app_user_id	Application User Id	The automatically generated primary key of the app_user table.	bigint
lic_type_id	License Type Id	A Foreign Key to the License Type table.	bigint
user_login	User Login	The login name used to access the Application. This may not be filled in if external application security is used.	nvarchar(4000)
email_addr	Email Address	An email address for the Application User.	nvarchar(4000)
delete_date	Delete Date	The value of a Application User, stored as a date.	datetime

### Attribute/Column(s) of Entity/Table "Password Reset Question"

Physical Name	Name	Definition	Physical Data Type
pwd_reset_question_id	Password Reset Question Id	The automatically generated primary key of the pwd_reset_question table.	bigint
question	Question	The value of a Password Reset Question, stored as a string.	varchar(255)

### Attribute/Column(s) of Entity/Table "Password Reset Question Answer"

Physical Name	Name	Definition	Physical Data Type
pwd_reset_question_answer_id	Password Reset Question Answer Id	The automatically generated primary key of the pwd_reset_question_answer table.	bigint
answer	Answer	The value of a Password Reset Question Answer, stored as a string.	nvarchar(max)
question	Question	The value of a Password Reset Question Answer, stored as a string.	varchar(255)
pwd_reset_regtd_acct_id	Password Reset Registered Account Id	The automatically generated primary key of the pwd_reset_regtd_acct table.	bigint

### Attribute/Column(s) of Entity/Table "Password Reset Registered Account"

Physical Name	Name	Definition	Physical Data Type
---------------	------	------------	--------------------

pwd_reset_regtd_acct_name	Password Reset Registered Account Name	This is the registered user name.	nvarchar(4000)
display_name	Display Name	This is the name pulled back from the active directory service.	varchar(255)
distinguished_name	Distinguished Name	The name of the Password Reset Registered Account.	varchar(255)
failed_reset_ct	Failed Reset Count	Count of failed resets.	int
failed_reset_timestamp	Failed Reset Timestamp	Last time of failed reset.	datetime
regtd_email	Registered Email	Email registered to handle reset verification.	varchar(255)
app_user_id	Application User Id	A Foreign Key to the Application User table.  The application user id will be used to tie AD accounts to existing Footprints customers.  If the user comes in anonymously through the browse feature, this field will be null.	bigint
pwd_reset_regtd_acct_id	Password Reset Registered Account Id	The automatically generated primary key of the pwd_reset_regtd_acct table.	bigint

### ER Diagram(s) of Subject Area "210 - Audit Trail"

Name	Physical Display Level
Logical	Comment
Physical	Column

### Entity/Table(s) of Subject Area "210 - Audit Trail"

Physical Name	Name	Logical Only	Do Not Generate	Definition
app_user	Application User		false	A user authorized to use the FootPrints system. The user is either a Customer User or an Agent User. The application user holds the minimum information needed to identify the user and associate the user with a user profile, where user settings are stored. A contact is not necessarily an application user. An application user entry is required in order for the user to access any online functions.
audit_event_type	Audit Event Type		false	A discriminator indicating the type of Audit Log Entry. Values include Audit Log Note, Audit Log Data Change, and Audit Log Data Version.
audit_field_cfg	Audit Field Configuration	false	false	Configuration values for the Audit that determine if a dynamic item should be audited. If only Dynamic Items are supported, then auditing may be something that cannot be turned off? Configuration Setting Values are used for the audit indicator. Within the Java code, automatic logging of classes can be turned on.
audit_log	Audit Log		false	The main Audit Log table which uses a serial key to establish the order of Audit Trail entries as committed to the system. It has subordinate notes and data values which are considered unordered within the Audit Log. It is possible and desirable that multiple Audit Log entries would have the same transaction id.
audit_msg_tmpl	Audit Message Template		false	A template message that is generated from the application with string substitution parameters. Only string variables are supported, because the eventual usage is just string presentation, and localization of the parameters is not supported.
audit_prop_chg	Audit Property Change		false	A data change record indicating the Inserted or Updated value. Deleted Values are not recorded here.
audit_prop_dim	Audit Property Dimension		false	A dimension table that holds every combination of Object Name, Property Name, Insert Update Delete Code, etc. that has been used in the Audit Trail. The purpose of this data design is to compress the repeated information in the Audit Trail, reducing data and speeding up indexed queries.

audit_rule_name	Audit Rule Name		false	This is a derived table of current and historical Rule Names that is used to make a permanent record of the rule name for the Audit trail. The replication will be done on publish and will avoid locking the definition tables during Audit History storage. The name of the Rule is replicated from the Rule Definition to avoid problems with query complexity, deletion and modification of Rule names, and locking. If a rule is deleted on publish, the delete date will be filled in. This table is cached in memory so the Audit Function does not have to repetitively retrieve it.
audit_txn	Audit Transaction		false	A record of the Principal and create time of the start of the commit time of a logical transaction. In many cases multiple Audit Log entries will use the same transaction.
item_field_defn	Item Field Definition	true	true	Item Field Definition is an association between the Item Definition entity and the Field Definition entity. This association contains a distinct value for each of the intersection points between the two entities.
sys_component	System Component		false	A lookup table for System Component. System Component is used to identify an area of the system. The primary use is when a component is updating system information with a transaction that is not initiated from a customer or agent.
val_type_cd	Value Type Code		false	A lookup table for the data value type discriminator. Many tables share this discriminator for name value pair datatypes. Values include: B Boolean D Timestamp - Date + Time I Integer L Long R Real Number S String T Text - long string

#### Attribute/Column(s) of Entity/Table "Application User"

Physical Name	Name	Definition	Physical Data Type
pwd	Password	This space intentionally left blank.	varchar(255)
user_name	User Name	The name of the Application User.	nvarchar(4000)
cust_login_type_id	Customer Login Type Id	A Foreign Key to the Customer Login Type table.	int
app_user_id	Application User Id	The automatically generated primary key of the app_user table.	bigint
lic_type_id	License Type Id	A Foreign Key to the License Type table.	bigint
user_login	User Login	The login name used to access the Application. This may not be filled in if external application security is used.	nvarchar(4000)
email_addr	Email Address	An email address for the Application User.	nvarchar(4000)
delete_date	Delete Date	The value of a Application User, stored as a date.	datetime

#### Attribute/Column(s) of Entity/Table "Audit Event Type"

Physical Name	Name	Definition	Physical Data Type
audit_event_type_id	Audit Event Type Id	The unique identifier for an instance of Audit Event Type. This identifier must be kept in sync with programming code that uses an enumeration or fixed constant to identify particular values.	BIGINT
audit_event_type_name	Audit Event Type Name	The name of the Audit Event Type.	varchar(255)

#### Attribute/Column(s) of Entity/Table "Audit Field Configuration"

Physical Name	Name	Definition	Physical Data Type
audit_field_cfg_id	Audit Field Configuration Id	The automatically generated primary key of the audit_field_cfg table.	bigint
is_audited	Is Audited	Indicates if changes to the field should be recorded in the Audit Trail.	bit
item_field_defn_id	Item Field Definition Id	A Foreign Key to the Definition table for an instance of type Definition.	bigint

#### Attribute/Column(s) of Entity/Table "Audit Log"

Physical Name	Name	Definition	Physical Data Type
---------------	------	------------	--------------------

audit_log_id	Audit Log Id	The automatically generated primary key of the audit_log table.	bigint
audit_event_type_id	Audit Event Type Id	The unique identifier for an instance of Audit Event Type. This identifier must be kept in sync with programming code that uses an enumeration or fixed constant to identify particular values.	BIGINT
audit_txn_id	Audit Transaction Id	The automatically generated primary key of the audit_txn table.	bigint
insert_update_delete_cd	Insert Update Delete Code	Indicates I-Insert, U-Update, D-Delete database action.	varchar(255)
string_parm_1	String Parameter 1	A substitution parameter for the referenced formatted string.	nvarchar(max)
string_parm_2	String Parameter 2	A substitution parameter for the referenced formatted string.	nvarchar(4000)
string_parm_3	String Parameter 3	A substitution parameter for the referenced formatted string.	nvarchar(4000)
string_parm_4	String Parameter 4	A substitution parameter for the referenced formatted string.	nvarchar(4000)
string_parm_5	String Parameter 5	A substitution parameter for the referenced formatted string.	nvarchar(4000)
string_parm_6	String Parameter 6	A substitution parameter for the referenced formatted string.	nvarchar(4000)
audit_prop_dim_id	Audit Property Dimension Id	A Foreign Key to the Audit Property Dimension table.	bigint
domain_obj_id	Domain Object Id	The numeric Primary Key of the Object that is autologged.	bigint
string_parm_7	String Parameter 7	A substitution parameter for the referenced formatted string.	nvarchar(4000)
string_parm_8	String Parameter 8	A substitution parameter for the referenced formatted string.	nvarchar(4000)
audit_msg_tmpl_id	Audit Message Template Id	A Foreign Key to the Audit Message Template table.	bigint
create_date	Create Date	The transaction date of the Audit Log entry. This date may not be in strict order with the Audit Log id in the event that features are used to continue a previous Audit Log transaction.	datetime
raw_msg	Raw Message	The value of a Audit Log, stored as a long text object.	nvarchar(max)
string_parm_7_int	String Parameter 7 Int	This is a computed column. The integer version of parameter 7. It is only indexed when parameter 8 is 'USER_SESSION'. It greatly increases the User browse screens ability to compute the last_login.	int

#### Attribute/Column(s) of Entity/Table "Audit Message Template"

Physical Name	Name	Definition	Physical Data Type
audit_msg_tmpl_id	Audit Message Template Id	The automatically generated primary key of the audit_msg_tmpl table.	bigint
audit_msg_tmpl_key	Audit Message Template Key	A search key for the Audit Message Template, composed of letters and numbers.	varchar(255)
audit_msg_tmpl	Audit Message Template	The template message consisting of text and substitution variables.	varchar(4000)

#### Attribute/Column(s) of Entity/Table "Audit Property Change"

Physical Name	Name	Definition	Physical Data Type
audit_prop_chg_id	Audit Property Change Id	The automatically generated primary key of the audit_prop_chg table.	bigint
audit_prop_dim_id	Audit Property Dimension Id	A Foreign Key to the Audit Property Dimension table.	bigint
new_string_val	New String Value	The new value of an inserted or updated property, for a property of type string.	nvarchar(4000)
new_text_val	New Text Value	The new value of an inserted or updated property, for a property of type text.	nvarchar(max)
new_date_val	New Date Value	The new value of an inserted or updated property, for a property of type date.	datetime
new_real_val	New Real Value	The new value of an inserted or updated property, for a property of type real.	numeric(38,20)
new_long_val	New Long Value	The new value of an inserted or updated property, for a property of type long integer. This may be a key, in which case the Convenience Value might also be stored.	bigint

audit_log_id	Audit Log Id	A Foreign Key to the Audit Log table.	bigint
old_string_val	Old String Value	The old value of an inserted or updated property, for a property of type string.	nvarchar(4000)
old_text_val	Old Text Value	The old value of an inserted or updated property, for a property of type text.	nvarchar(max)
old_date_val	Old Date Value	The old value of an inserted or updated property, for a property of type date.	datetime
old_real_val	Old Real Value	The old value of an inserted or updated property, for a property of type real.	numeric(38,20)
old_long_val	Old Long Value	The old value of an inserted or updated property, for a property of type long integer. This may be a key, in which case the Convenience Value might also be stored.	bigint
val_type_cd	Value Type Code	The data type for a Value Type Code. The choices are 'S' string, 'D' date, 'T' text, 'I' integer.	char(1)

#### Attribute/Column(s) of Entity/Table "Audit Property Dimension"

Physical Name	Name	Definition	Physical Data Type
audit_prop_dim_id	Audit Property Dimension Id	The automatically generated primary key of the audit_prop_dim table.	bigint
prop_mbr_name	Property Member Name	The name of the Object Property Name.	varchar(255)
val_type_cd	Value Type Code	The data type for a Value Type Code. The choices are 'S' string, 'D' date, 'T' text, 'I' integer.	char(1)
audit_class_name	Audit Class Name	The Java Class name of the object.	varchar(255)
prop_class_name	Property Class Name	The Java Class name of the object.	varchar(255)
coll_class_name	Collection Class Name	The Java Class name of the object.	varchar(255)
is_audited	Is Audited	Indicates if Is Audited.	bit
item_defn_id	Item Definition Id	A Foreign Key to the Definition table.	bigint
item_field_defn_id	Item Field Definition Id	A Foreign Key to the Definition table for an instance of type Definition.	bigint

#### Attribute/Column(s) of Entity/Table "Audit Rule Name"

Physical Name	Name	Definition	Physical Data Type
audit_rule_name_id	Audit Rule Name Id	The primary key is the original rule definition id. It will not change as the rule is renamed. If the rule is deleted, the deleted date will be filled in, but the rule name will still be known in this table.	bigint
rule_name	Rule Name	The name of the Rule is replicated from the Rule Definition to avoid problems with query complexity, deletion and modification of Rule names, and locking.	varchar(255)
deleted_date	Deleted Date	The date that the Audit Rule Name was published as deleted.	datetime

#### Attribute/Column(s) of Entity/Table "Audit Transaction"

Physical Name	Name	Definition	Physical Data Type
audit_txn_id	Audit Transaction Id	The automatically generated primary key of the audit_txn table.	bigint
txn_start_date	Transaction Start Date	The transaction date of the Audit Log entry. This date may not be in strict order with the Audit Log id in the event that features are used to continue a previous Audit Log transaction.	datetime
app_user_id	Application User Id	A Foreign Key to the Application User table. The audit trail transaction may or may not be associated with a customer or agent user. If this column is null, then a system component should be identified.	bigint
sys_component_id	System Component Id	This is a foreign key to the system component table. The audit trail transaction may or may not be associated with a customer or agent user. The system component will be set if there is no application user identified. It identifies the source of the transaction. Values include SYSTEM, WEB_SERVICES, ASSET_MANAGEMENT, CALENDAR, IMPORTING, etc. All identifiers are negative, so that they can be distinguished from userids in case they are both stored in the same dynamic table CreatedBy, and UpdatedBy fields. We are going to try first normal form normalization here and see if it works.	int

audit_rule_name_id	Audit Rule Name Id	The primary key is the original rule definition id. It will not change as the rule is renamed. If the rule is deleted, the deleted date will be filled in, but the rule name will still be known in this table.	bigint
--------------------	--------------------	---	--------

#### Attribute/Column(s) of Entity/Table "Item Field Definition"

Physical Name	Name	Definition	Physical Data Type
item_field_defn_id	Item Field Definition Id	A Foreign Key to the Definition table for an instance of type Definition.	bigint
item_defn_id	Item Definition Id	A Foreign Key to the Definition table for an instance of type Definition.	bigint
field_defn_id	Field Definition Id	A Foreign Key to the Definition table.	bigint

#### Attribute/Column(s) of Entity/Table "System Component"

Physical Name	Name	Definition	Physical Data Type
sys_component_id	System Component Id	The unique identifier for an instance of System Component. This identifier must be kept in sync with programming code that uses an enumeration or fixed constant to identify particular values.	int
sys_component_name	System Component Name	The unique string identifier for an instance of System Component. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)
sys_component_descr	System Component Description	A descriptive name of the System Component. This descriptive name may be localized.	varchar(512)

#### Attribute/Column(s) of Entity/Table "Value Type Code"

Physical Name	Name	Definition	Physical Data Type
val_type_cd	Value Type Code	The data type for a Value Type Code. The choices are 'S' string, 'D' date, 'I' text, 'I' integer.	char(1)
val_type_name	Value Type Name	The name of the Value Type Code.	varchar(255)

#### ER Diagram(s) of Subject Area "230 - Rules"

Name	Physical Display Level
Logical	Column
Physical	Column

#### Entity/Table(s) of Subject Area "230 - Rules"

Physical Name	Name	Logical Only	Do Not Generate	Definition
rule_usage_statistic	Rule Usage Statistic	false	false	A table that records statistics on rule usage.

#### Attribute/Column(s) of Entity/Table "Rule Usage Statistic"

Physical Name	Name	Definition	Physical Data Type
rule_usage_stat_id	Rule Usage Statistic Id	The automatically generated primary key of the rule_usage_statistic table.	bigint
defn_uuid	Definition UUID	The rule definition id.	varchar(255)
last_hit_timestamp	Last Hit Timestamp	The time of the last hit	datetime
hit_ct	Hit Count	Count of hits.	int
impact_ct	Impact Count	The count of the impact	int

#### ER Diagram(s) of Subject Area "240 - File Storage"

Name	Physical Display Level
Logical	Column
Physical	Column

#### Entity/Table(s) of Subject Area "240 - File Storage"

Physical Name	Name	Logical Only	Do Not Generate	Definition
defn	Definition		false	A definition holds the structural and behavioral information needed for an application element that is configurable by the system designer and/or the end user at definition time. Current Footprints uses a combination of directories, files, tables, and programming code to represent this information. The new FootPrints will provide all of the same information within the definition. The definition can be persisted in a structure that is isolated at the Instance Level. The data for the

				definition will be stored as Configuration Settings associated with the Definition Version. See the database views v_defn_ver_active, v_cntrn_item_field.
file_info	File Information		false	The URI in this table is a unique string identifier that points to the file stored in the jackrabbit tables.  This table is intended to be stand alone, and the application code will use the URI to actually do a lookup of the things that are attached to this file.
item_attch	Item Attachment		false	The link between a file stored in jackrabbit and a container item.

#### Attribute/Column(s) of Entity/Table "Definition"

Physical Name	Name	Definition	Physical Data Type
defn_id	Definition Id	The automatically generated primary key of the defn table.	bigint
defn_name	Definition Name	The name of the Definition.	nvarchar(4000)
dsble_date	Disabled Date	The date that this instance of Definition is logically deleted and disabled.	datetime
defn_guid	Definition GUID	A Global Unique Identifier search key for the Definition, composed of letters and numbers. It is computed by the client and typically used in unstructured data.	varchar(50)
defn_type_name	Definition Type Name	The unique string identifier for an instance of Definition Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)
defn_subtype_id	Definition Subtype Id	A Foreign Key to the Definition Subtype table.	int
tmpl_export_guid	Template Export GUID	The value of a Definition, stored as a string.	VARCHAR(50)

#### Attribute/Column(s) of Entity/Table "File Information"

Physical Name	Name	Definition	Physical Data Type
file_info_id	File Information Id	The automatically generated primary key of the file_info table.	bigint
file_info_uri	File Information URI	This will act as the string that points to the actual location of the file. The program will use this rather than relationship information stored in the database to find the file.	nvarchar(4000)
file_info_size	File Information Size	The size of the file.	bigint
create_date	Create Date	The create date of the file in the repository.	datetime
is_file_deleted	Is File Deleted	A boolean flag meant to indicate that the file has been deleted. This is used in a multi-phase deletion approach, where the deletion might not make it all the way through, and a cleanup process can recover where the deletion left off.	bit
expiration_date	Expiration Date	Expiry date	datetime

#### Attribute/Column(s) of Entity/Table "Item Attachment"

Physical Name	Name	Definition	Physical Data Type
item_attch_id	Item Attachment Id	The automatically generated primary key of the item_attch table.	bigint
file_info_id	File Information Id	A Foreign Key to the File Information table.	bigint
item_defn_id	Item Definition Id	A Foreign Key to the Definition table.	bigint
item_id	Item Id	A Foreign Key to the Item Attachment table.	bigint

#### ER Diagram(s) of Subject Area "250 - Apache Jack Rabbit JCR"

Name	Physical Display Level
Logical	Column
Physical	Column

#### Entity/Table(s) of Subject Area "250 - Apache Jack Rabbit JCR"

Physical Name	Name	Logical Only	Do Not Generate	Definition
dflt_binval	Default Binval		false	Apache JackRabbit Content Repository Table. Access via JCR API.

dfit_bundle	Default Bundle		false	Apache JackRabbit Content Repository Table. Access via JCR API.
dfit_names	Default Names		false	Apache JackRabbit Content Repository Table. Access via JCR API.
dfit_refs	Default Refs		false	Apache JackRabbit Content Repository Table. Access via JCR API.
secuty_binval	Security Binval		false	Apache JackRabbit Content Repository Table. Access via JCR API.
secuty_bundle	Security Bundle		false	Apache JackRabbit Content Repository Table. Access via JCR API.
secuty_names	Security Names		false	Apache JackRabbit Content Repository Table. Access via JCR API.
secuty_refs	Security Refs		false	Apache JackRabbit Content Repository Table. Access via JCR API.

#### Attribute/Column(s) of Entity/Table "Default Binval"

Physical Name	Name	Definition	Physical Data Type
binval_id	Binval Id	binval id	char(18)
binval_data	Binval Data	Binval data	char(18)

#### Attribute/Column(s) of Entity/Table "Default Bundle"

Physical Name	Name	Definition	Physical Data Type
node_id_hi	Node Id Hi	hi	char(18)
node_id_lo	Node Id Lo	low	char(18)
bundle_data	bundle Data	bundle	char(18)

#### Attribute/Column(s) of Entity/Table "Default Names"

Physical Name	Name	Definition	Physical Data Type
name	name	The name of the Default Names.	varchar(255)
id	id	id	char(18)

#### Attribute/Column(s) of Entity/Table "Default Refs"

Physical Name	Name	Definition	Physical Data Type
node_id_hi	Node Id Hi	hi	char(18)
node_id_lo	Node Id Lo	low	char(18)
refs_data	Refs Data	refs	char(18)

#### Attribute/Column(s) of Entity/Table "Security Binval"

Physical Name	Name	Definition	Physical Data Type
binval_id	Binval Id		char(18)
binval_data	Binval Data		char(18)

#### Attribute/Column(s) of Entity/Table "Security Bundle"

Physical Name	Name	Definition	Physical Data Type
node_id_hi	Node Id Hi		char(18)
node_id_lo	Node Id Lo		char(18)
bundle_data	bundle Data		char(18)

#### Attribute/Column(s) of Entity/Table "Security Names"

Physical Name	Name	Definition	Physical Data Type
name	name	The name of the Security Names.	varchar(255)
id	id		char(18)

#### Attribute/Column(s) of Entity/Table "Security Refs"

Physical Name	Name	Definition	Physical Data Type
node_id_hi	Node Id Hi		char(18)
node_id_lo	Node Id Lo		char(18)
refs_data	Refs Data		char(18)

#### ER Diagram(s) of Subject Area "262 - Import Items and Users"

Name	Physical Display Level
Logical	Column
Physical	Column

#### Entity/Table(s) of Subject Area "262 - Import Items and Users"

Physical Name	Name	Logical Only	Do Not Generate	Definition
---------------	------	--------------	-----------------	------------



app_role	Application Role		false	A User Security Role within Footprints. This includes Agent Users and Customer Users, and includes Role settings at the Instance and Container level. Workspace Role and CMDB Role may both be implemented as Container Role. The Role has associated Rights and Permissions, and is a way of granting Roles and Permissions to a User. There is some conceptual overlap between the User Profile and the Role. The User is assigned a single Role as part of the User Profile, and a User can only have one System Role and one Workspace Role. The User Profile can have property settings that override the Role Settings.
app_user	Application User		false	A user authorized to use the FootPrints system. The user is either a Customer User or an Agent User. The application user holds the minimum information needed to identify the user and associate the user with a user profile, where user settings are stored. A contact is not necessarily an application user. An application user entry is required in order for the user to access any online functions.
defn	Definition		false	A definition holds the structural and behavioral information needed for an application element that is configurable by the system designer and/or the end user at definition time. Current Footprints uses a combination of directories, files, tables, and programming code to represent this information. The new FootPrints will provide all of the same information within the definition. The definition can be persisted in a structure that is isolated at the Instance Level. The data for the definition will be stored as Configuration Settings associated with the Definition Version. See the database views v_defn_ver_active, v_cntrn_item_field.
imp	Import		false	
imp_asgnmt	Import Assignment		false	The default assignment details for an Import Job.
imp_defn_rule	Import Definition Rule		false	This table contains the information necessary for the application to do a conditional flow in the processing based on a text string (defn_value).
imp_error_result	Import Error Result		false	A list of errors that occurred during the execution of the import.
imp_execn_record	Import Execution Record		false	A log of the execution of an import.
imp_schdd_task	Import Scheduled Task		false	The import scheduled task is an association between the import and the scheduled task. Import specific information can be placed here. This was added when the scheduled task service was refactored to treat different schedule types with the same core code.
imp_user_wksp_role	Import User Workspace Role		false	An association between an Import and a role.
schdd_task	Scheduled Task		false	A scheduled task that may include recurrence. This is all of the information needed to provide the scheduler with data to execute a task. It is used for Reports, but is intended to be generic for other scheduled tasks. This is the root object. Report specific information will go in an association table.
schdd_task_notifn	Scheduled Task Notification		false	The collection of notification recipients for the Scheduled Task. Having true references here will make it easier to maintain changes to user's information such as LDAP lookup, name changes, and user deletion or replacement use cases.

team	Team		false	A team is a group of application users that can be used in Team level assignments.
------	------	--	-------	--

#### Attribute/Column(s) of Entity/Table "Application Role"

Physical Name	Name	Definition	Physical Data Type
app_role_id	Application Role Id	The automatically generated primary key of the app_role table.	bigint
tmpl_app_role_id	Template Application Role Id	A Foreign Key to the Application Role table. This indicates which Template Role was used to create a built-in role in case some features concerning new properties added to later release of the system will need to be added to existing templates.	bigint
app_role_name	Application Role Name	The name of the Application Role.	varchar(255)
defn_id	Definition Id	A Foreign Key to the Definition table.	bigint
admin_type_id	Administrator Type Id	A Foreign Key to the Administrator Type table.	int
is_built_in	Is Built In	Indicates that this instance of Application Role is a built-in instance that cannot be changed by the end user.	bit
app_role_descr	Application Role Description	A descriptive name of the Application Role. This descriptive name may be localized.	varchar(1024)
user_type_id	User Type Id	A Foreign Key to the User Type table.	int
defn_subtype_id	Definition Subtype Id	A Foreign Key to the Definition Subtype table.	int
tmpl_app_role_id	Template Application Role Id	A Foreign Key to the Application Role table.	bigint
tmpl_level_name	tmpl_level_name	The value of a Built_in_level, stored as a string.	varchar(255)

#### Attribute/Column(s) of Entity/Table "Application User"

Physical Name	Name	Definition	Physical Data Type
pwd	Password	This space intentionally left blank.	varchar(255)
user_name	User Name	The name of the Application User.	nvarchar(4000)
cust_login_type_id	Customer Login Type Id	A Foreign Key to the Customer Login Type table.	int
app_user_id	Application User Id	The automatically generated primary key of the app_user table.	bigint
lic_type_id	License Type Id	A Foreign Key to the License Type table.	bigint
user_login	User Login	The login name used to access the Application. This may not be filled in if external application security is used.	nvarchar(4000)
email_addr	Email Address	An email address for the Application User.	nvarchar(4000)
delete_date	Delete Date	The value of a Application User, stored as a date.	datetime

#### Attribute/Column(s) of Entity/Table "Definition"

Physical Name	Name	Definition	Physical Data Type
defn_id	Definition Id	The automatically generated primary key of the defn table.	bigint
defn_name	Definition Name	The name of the Definition.	nvarchar(4000)
dsble_date	Disabled Date	The date that this instance of Definition is logically deleted and disabled.	datetime
defn_guid	Definition GUID	A Global Unique Identifier search key for the Definition, composed of letters and numbers. It is computed by the client and typically used in unstructured data.	varchar(50)
defn_type_name	Definition Type Name	The unique string identifier for an instance of Definition Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)
defn_subtype_id	Definition Subtype Id	A Foreign Key to the Definition Subtype table.	int
tmpl_export_guid	Template Export GUID	The value of a Definition, stored as a string.	VARCHAR(50)

#### Attribute/Column(s) of Entity/Table "Import"

Physical Name	Name	Definition	Physical Data Type
---------------	------	------------	--------------------

imp_id	Import Id	The automatically generated primary key of the imp table.	bigint
imp_status	Import Status	Import Status	int
imp_type	Import Type	Import Type	varchar(255)
imp_user_sys_level_role	Import User System Level Role	A Foreign Key to the Application Role table.	bigint
imp_item_defn_id	Import Item Definition ID	A Foreign Key to the Definition table.	bigint
imp_defn_id	Import Definition ID	A Foreign Key to the Definition table.	bigint
imp_date	Import Date	Import Date	datetime
imp_create_date	Import Create Date	Import Create Date	datetime
imp_record_addr_book_id	Import Record Address Book Id	A Foreign Key to the Definition table.	bigint
imp_total_created	Import Total Created	Total Rows Created	int
imp_total_updated	Import Total Updated	Total Rows updated	int
imp_total_failures	Import Total Failures	Failure count.	int
imp_error_cd	Import Error Code	Import Error Code	varchar(255)
imp_create_user	Import Create User	Import Create User	int
imp_ext_db_defn_id	Import Ext DB Definition Id	A Foreign Key to the Definition table.	bigint
imp_ci_addr_book_id	Import CI Address Book Id	Import CI Address Book Id	bigint
imp_last_exec_date	Import Last Exec Date	Import Last Exec Date	datetime
imp_ci_rel_from_defn_id	Import CI Rel from Definition Id	A Foreign Key to the Definition table.	bigint
imp_ci_rel_to_defn_id	Import CI Rel to Definition Id	A Foreign Key to the Definition table.	bigint
imp_time_zone	Import Time Zone	Time zone of the import.	varchar(255)

#### Attribute/Column(s) of Entity/Table "Import Assignment"

Physical Name	Name	Definition	Physical Data Type
imp_asgmt_id	Import Assignment Id	The automatically generated primary key of the imp_asgmt table.	bigint
team_id	Team Id	A Foreign Key to the Team table.	bigint
app_user_id	Application User Id	A Foreign Key to the Application User table.	bigint
imp_id	Import Id	A Foreign Key to the Import table.	bigint

#### Attribute/Column(s) of Entity/Table "Import Definition Rule"

Physical Name	Name	Definition	Physical Data Type
imp_defn_rule_id	Import Definition Rule Id	The automatically generated primary key of the imp_defn_rule table.	bigint
imp_defn_rule_col_index	Import Definition Rule Column Index	stores the index of the column (i.e in a csv file) or the name of the column (i.e database table)	varchar(255)
imp_defn_rule_text_to_contain	Import Definition Rule Text To Contain	The text that will cause the rule to trigger a branch.	varchar(255)
imp_defn_rule_defn_id	Import Definition Rule Definition Id	The reference for the item definition (currently for the CI type but want to be more general).	bigint
imp_id	Import Id	FK to the import operation record in the existing import operation table.	bigint
imp_defn_rule_status	Import Definition Rule Status	Status can be different based on the CI type, but will be a fk to the allowed values inside the defn table.	bigint

#### Attribute/Column(s) of Entity/Table "Import Error Result"

Physical Name	Name	Definition	Physical Data Type
imp_error_result_id	Import Error Result Id	The automatically generated primary key of the imp_error_result table.	bigint
imp_error_result_string_inheri	Import Error Result String Inherited	Import Error Result String Inherited.	varchar(255)
imp_error_row	Import Error Row	The value of a Import Error Result, stored as a string.	varchar(max)
imp_error_line	Import Error Line	The value of a Import Error Result stored as boolean.	int
imp_error_parms	Import Error Parameters	The value of a Import Error Result, stored as a string.	varchar(255)
imp_execn_record_id	Import Execution Record Id	A Foreign Key to the Import Execution Record table.	bigint

#### Attribute/Column(s) of Entity/Table "Import Execution Record"

Physical Name	Name	Definition	Physical Data Type
imp_execn_record_id	Import Execution Record Id	The automatically generated primary key of the imp_execn_record table.	bigint
imp_id	Import Id	A Foreign Key to the Import table.	bigint
execn_date	Execution Date	Execution Date	datetime
created_ct	Created Count	Created Count	int

rejected_ct	Rejected Count	Rejected Count	int
updated_ct	Updated Count	Updated Count	int
unchgd_ct	Unchanged Count	Unchanged Count	int
imp_status	Import Status	Import Status	int
validated_ct	Validated Count	Validated Count	int

#### Attribute/Column(s) of Entity/Table "Import Scheduled Task"

Physical Name	Name	Definition	Physical Data Type
schdd_task_id	Scheduled Task Id	A Foreign Key to the Scheduled Task table. It also serves as the Primary Key here.	bigint
imp_id	Import Id	A Foreign Key to the Import table.	bigint
time_zone	Time Zone	Time zone of the import.	varchar(255)

#### Attribute/Column(s) of Entity/Table "Import User Workspace Role"

Physical Name	Name	Definition	Physical Data Type
imp_id	Import Id	A Foreign Key to the Import table.	bigint
app_role_id	Application Role Id	A Foreign Key to the Application Role table.	bigint
imp_user_wksp_role_id	Import User Workspace Role Id	The automatically generated primary key of the imp_user_wksp_role table.	bigint

#### Attribute/Column(s) of Entity/Table "Scheduled Task"

Physical Name	Name	Definition	Physical Data Type
schdd_task_name	Scheduled Task Name	The name of the Scheduled Task.	varchar(255)
schdd_task_id	Scheduled Task Id	The automatically generated primary key of the scheduled_task table.	bigint
schdd_task_descr	Scheduled Task Description	A descriptive name of the Scheduled Task. This descriptive name may be localized.	varchar(512)
recur_schd_type_id	Recur Schedule Type Id	The unique identifier for an instance of Recur Schedule Type. This identifier must be kept in sync with programming code that uses an enumeration or fixed constant to identify particular values.	integer
begin_date	Begin Date	Begin Date.	datetime
recur_period_type_id	Recur Period Type Id	The unique identifier for an instance of Recur Period Type. This identifier must be kept in sync with programming code that uses an enumeration or fixed constant to identify particular values.	integer
recur_hour	Recur Hour	The Hours portion of a time of day	integer
recur_minute	Recur Minute	The minutes portion of a time of day	integer
recur_days_of_week	Recur Days of Week	Recur Days of Week.	varchar(255)
recur_days_of_month	Recur Days of Month	Recur Days of Month.	varchar(255)
create_date	Create Date	The date that the Scheduled Task was created.	datetime
create_app_user_id	Create Application User Id	A Foreign Key to the Application User table.	bigint
schdd_task_type_id	Scheduled Task Type Id	The unique identifier for an instance of Scheduled Task Type. This identifier must be kept in sync with programming code that uses an enumeration or fixed constant to identify particular values.	integer

#### Attribute/Column(s) of Entity/Table "Scheduled Task Notification"

Physical Name	Name	Definition	Physical Data Type
schdd_task_id	Scheduled Task Id	A Foreign Key to the Scheduled Task table.	bigint
app_user_id	Application User Id	A Foreign Key to the Application User table.	bigint
team_id	Team Id	A Foreign Key to the Team table.	bigint
cntc_defn_id	Contact Definition Id	A Foreign Key to the Definition table.	bigint
email_addr	Email Address	An email address for the Scheduled Task Notification.	nvarchar(4000)
schdd_task_notifn_id	Scheduled Task Notification Id	The automatically generated primary key of the scheduled_task_notification table.	bigint
cntc_item_id	Contact Item Id	A Foreign Key to the Item table.	bigint
all_cntcs_flag	All_Contacts_Flag	A boolean flag for xxx	numeric(1)
cntc_email	Contact Email	An email address for the Scheduled Task Notification.	nvarchar(4000)

#### Attribute/Column(s) of Entity/Table "Team"

Physical Name	Name	Definition	Physical Data Type
team_id	Team Id	The automatically generated primary key of the team table.	bigint

team_name	Team Name	The name of the Team.	nvarchar(4000)
wksp_defn_id	Workspace Definition Id	A Foreign Key to the Definition table.	bigint
team_email_notifn_type_id	Team Email Notification Type Id	A Foreign Key to the Team Email Notification Type table.	int
team_lead_app_user_id	Team Lead Application User Id	A Foreign Key to the Application User table.	bigint

### ER Diagram(s) of Subject Area "280 - Session and Locking"

Name	Physical Display Level
Logical	Column
Physical	Column

### Entity/Table(s) of Subject Area "280 - Session and Locking"

Physical Name	Name	Logical Only	Do Not Generate	Definition
app_user	Application User		false	A user authorized to use the FootPrints system. The user is either a Customer User or an Agent User. The application user holds the minimum information needed to identify the user and associate the user with a user profile, where user settings are stored. A contact is not necessarily an application user. An application user entry is required in order for the user to access any online functions.
cntr_defn	Container Definition	true	true	The definition of a Container and the associated Configuration. Settings at the container level. This would include the collection of Item Definitions. The Container_Definition will have a Container_Class identifier (Workspace, CMDB, Address Book) and may have specializations such as Workspace_Container_Definition. As much as possible, the Container Definition will suffice to describe any Container Type. There will be an association to a default Container Template, as well as more application specific Container Templates that can be chosen to initially populate the Container Template. There will also be Container Group Definition, and Container Group Templates. See the database views v_cntr_defn, v_cntr_defn_active.
defn_lock	Definition Lock	true	true	A pessimistic lock of a definition. The lock helps implement single session editing. It is intended to be used for container definitions, but may be used by other definitions as well. This is not a table it is implemented with configuration settings.
item	Item	true	true	The Item is the abstract class for Issue, Configuration Item, Address Book entry, and other information items that are the main information entities in FootPrints. The Item structure and behavior is defined by its Definition. The Item is namespaced, identified by, and logically stored in the single parent container. It can be referenced in the same manner both inside and outside the container. The Item is the storage model and the implementation for the Item Definition. There are many important specializations of the Item class, and there even more Template definitions of Container/Item that provide an end-user application experience to the End-User. The Item implementation must implement flexible schema in an easy to manage, well engineered, and high performing way. The item can store user-defined and user-named fields defined in its definition. The Item could implement the nice-to-have features such as an inheritance model for Item subtypes, and have the ability to represent collections of subordinate items. The Item data can be imported, exported, or synchronized with external data sources. The Item data can be linked and synchronized with other items in the same Instance.

item_defn	Item Definition	true	true	The definition of an Item, which includes the Item Name and localizations and the collection of Item Fields. It also includes customizations and localizations of string resources for Item Names, and Item Field names.
item_lock	Item Lock		false	A pessimistic lock of a dynamic item instance. The lock helps implement single session editing of an ticket item , cmdb item, solution item, or any other item in the system.
sys_session	System Session	false	false	A table that holds records for currently logged in application users. It is not used for component sessions, and the last accessed time is not updated because of performance issues.

#### Attribute/Column(s) of Entity/Table "Application User"

Physical Name	Name	Definition	Physical Data Type
pwd	Password	This space intentionally left blank.	varchar(255)
user_name	User Name	The name of the Application User.	nvarchar(4000)
cust_login_type_id	Customer Login Type Id	A Foreign Key to the Customer Login Type table.	int
app_user_id	Application User Id	The automatically generated primary key of the app_user table.	bigint
lic_type_id	License Type Id	A Foreign Key to the License Type table.	bigint
user_login	User Login	The login name used to access the Application. This may not be filled in if external application security is used.	nvarchar(4000)
email_addr	Email Address	An email address for the Application User.	nvarchar(4000)
delete_date	Delete Date	The value of a Application User, stored as a date.	datetime

#### Attribute/Column(s) of Entity/Table "Container Definition"

Physical Name	Name	Definition	Physical Data Type
db_schema_name	Database Schema Name	The name of the database schema. It is a database object name, limited to 30 characters in length.	varchar(30)
defn_id	Definition Id	A Foreign Key to the Definition table.	bigint

#### Attribute/Column(s) of Entity/Table "Definition Lock"

Physical Name	Name	Definition	Physical Data Type
locked_by	Locked By	a string description suitable for display of a session user or component that owns a lock.	varchar(255)
create_date	Create Date	The date that the Definition Lock was created.	datetime
defn_id	Definition Id	A Foreign Key to the Definition table.	bigint
token	Token	A Global Unique Identifier search key for the System Session, composed of letters and numbers. The session table is not persisted to the database for performance and integrity reasons. Locks will be cleared by the application in normal use, and may be cleared if they are orphaned.	varchar(255)

#### Attribute/Column(s) of Entity/Table "Item"

Physical Name	Name	Definition	Physical Data Type
item_key	Item Key	A search key for the Item, composed of letters and numbers.	varchar(255)
item_id	Item Id	The automatically generated primary key of the item table.	bigint
cntnr_id	Container Id	A Foreign Key to the Container table.	bigint
item_defn_id	Item Definition Id	A Foreign Key to the Definition table for an instance of type Definition.	bigint

#### Attribute/Column(s) of Entity/Table "Item Definition"

Physical Name	Name	Definition	Physical Data Type
item_table_name	Item Table Name	The name of the DBMS_Object_Name. It is a database object name, limited to 30 characters in length.	varchar(30)
item_defn_id	Item Definition Id	A Foreign Key to the Definition table for an instance of type Definition.	bigint

#### Attribute/Column(s) of Entity/Table "Item Lock"

Physical Name	Name	Definition	Physical Data Type
---------------	------	------------	--------------------

item_id	Item Id	Identifies the Item being locked.	bigint
item_defn_id	Item Definition Id	Identifies the Item Definition for the item being locked.	bigint
session_token	Session Token	A Global Unique Identifier search key for the System Session, composed of letters and numbers. The session table is not persisted to the database for performance and integrity reasons. Locks will be cleared by the application in normal use, and may be cleared if they are orphaned.	varchar(255)
locked_by	Locked By	a string description suitable for display of a session user or component that owns a lock.	nvarchar(4000)
create_date	Create Date	The date that the Item Lock was created.	datetime
item_lock_id	Item Lock Id	The automatically generated primary key of the item_lock table.	bigint

#### Attribute/Column(s) of Entity/Table "System Session"

Physical Name	Name	Definition	Physical Data Type
sys_session_id	System Session Id	The automatically generated primary key of the sys_session table.	bigint
token	Token	A Global Unique Identifier search key for the System Session, composed of letters and numbers. It is computed by the client and typically used in unstructured data.	varchar(255)
app_user_id	Application User Id	A Foreign Key to the Application User table.	bigint
is_active	Is Active	Indicates if Is Active.	bit
create_time	Create Time	The value of a System Session, stored as a date.	datetime
last_accessed_time	Last Accessed Time	The last accessed time is not used due to performance problems. It is a transient column.	datetime
session_type	Session Type	The value of a System Session, stored as a string.	varchar(32)

#### ER Diagram(s) of Subject Area "290 - Security"

Name	Physical Display Level
Logical	Column
Physical	Column

#### Entity/Table(s) of Subject Area "290 - Security"

Physical Name	Name	Logical Only	Do Not Generate	Definition
secuty_policy	Security Policy		false	A XACML security policy.
target_match	Target Match		false	Matches the Secuty Policy to a Target using XACML rules.
target_match_type	Target Match Type		false	A lookup table for the XACML target match type. Values include ACTION, RESOURCE, SUBJECT

#### Attribute/Column(s) of Entity/Table "Security Policy"

Physical Name	Name	Definition	Physical Data Type
secuty_policy_id	Security Policy Id	The automatically generated primary key of the secuty_policy table.	bigint
policy_id	Policy Id	Policy Id	varchar(255)
policy_content	Policy Content	xml blob with the content of the XACML policy	varchar(max)
create_date	Create Date	Create Date	datetime
delete_date	Delete Date	Logical Delete date	datetime

#### Attribute/Column(s) of Entity/Table "Target Match"

Physical Name	Name	Definition	Physical Data Type
target_match_id	Target Match Id	The automatically generated primary key of the target_match table.	bigint
secuty_policy_id	Security Policy Id	A Foreign Key to the Security Policy table.	bigint
target_match_type_name	Target Match Type Name	The name of the Target Match Type. Values include Action, Resource, Subject.	varchar(70)
attr_id	Attribute Id	Concatenated identifier of the target attribute name in a name/value pair.	varchar(70)
attr_val	Attribute Value	The value of the name/value pair.	varchar(70)
create_date	Create Date	Create date.	datetime

delete_date	Delete Date	Logical Delete date.	datetime
-------------	-------------	----------------------	----------

#### Attribute/Column(s) of Entity/Table "Target Match Type"

Physical Name	Name	Definition	Physical Data Type
target_match_type_name	Target Match Type Name	The name of the Target Match Type. Values include Action, Resource, Subject.	VARCHAR(70)
target_match_type_descr	Target Match Type Description	A descriptive name of the Target Match Type. This descriptive name may be localized.	varchar(512)

#### ER Diagram(s) of Subject Area "300 - Ticket Data Mart"

Name	Physical Display Level
Logical	Column
Physical	Column

#### Entity/Table(s) of Subject Area "300 - Ticket Data Mart"

Physical Name	Name	Logical Only	Do Not Generate	Definition
:item_shortname_bal	:Item_Shortname Balance		false	A summary table to make calculation of hourly balances fast. The date range is a range of whole hours that identifies the period during which the ticket is unclosed at the end of the whole hour. The flags can be summed, and they indicated further whether the ticket was activated or resolved at the end of the hour. This table is only used for unclosed balance metrics, and is designed to be joined with the TIME DIMENSION. Summary higher than hour is possible just by joining the higher granularity time dimension with the date range. Interpolation below the hour can be done for the first or last hour of the ranges using the hard date time in the TICKET EVENT SUMMARY table.
:item_shortname_event_sum	:item_shortname Event Sum		false	The fact table for Ticket Event Summary. The date and time of lifecycle events is stored here, along with calculations for the elapsed work hours and clock hours. There are separate calculations that begin at ticket creation and ticket activation. Resolution Time and Closed Time are two different events. The data in this table is maintained by data mart stored procedures. The name of the table is qualified by the container schema and includes the item short name. In addition to the built-in reporting, this table can be used in any dimensional BI tool to produce useful reports.
:item_shortname_status	:item_shortname Status		false	The configuration settings for the item status and item lifecycle flags are maintained in this table for performance and history reasons. They are replicated here at the beginning of the data mart load process. Even a retired status will be in this table in order to support historical entries. : item_shortname will be substituted at publish.
:item_shortname_transn	:item_shortname Transition		false	A dimension table that records the details of a state transition, including the from and to state names, and the number of hops that are taken in the transition. This allows the precalculation of transitive states, so the transition from A to D via the path A>b>c>D would be recorded with a hopCt of 3.
:item_shortname_transn_sum	:item_shortname Transition Sum		false	The fact table for state transitions. It holds all transitions, whether they are transitive or not. Transitions with a hop count of 1 are traditional transitions. There can be multiple cycles between two states for a single ticket. The Is Longest Hop can be used to identify the distinct longest paths. Work Hours and Clock Hours are calculated for all transitions. In addition to the built-in reporting, this table can be used in any dimensional BI tool to produce useful reports.



life_cycle_event	Life Cycle Event		false	<p>This may come into play for the "First Contact" which is defined as:</p> <p>From Ursula Strahm (3/23/2012) First Call from the Executive Dashboard SRS:</p> <p>First Call.- The elapsed time between the From and To Event is 0 for the current Widget configuration. Examples of this are:</p> <p>i. The Issue is created in the "To Event" status. In this case, the Issue will always show in the First Call bucket.</p> <p>ii. Issue went from Created to "To Event" status outside of working hours and the Widget is configured to display "Work Hours"</p> <p>iii. Issue was Created in a "does not count for time" status and it stay in "does not count for time status" until it reached the to "To Event" and the Widget is configured to "only account for time spent in statuses included in the Issue Lifecycle Time Measurements"</p>
prev_timing	Previous Timing		false	A table with a generic timestamp. The timestamp is used to give the timestamps of previous and second previous timestamps according to a step as specified in the choices the user has for comparison reports.
time_dim	Time Dimension		false	A table with a generic timestamp. The timestamp is broken up into the specific pieces that the reports need to slice the information according to the user specifications.
time_zone_local_hour	Time Zone Local Hour		false	A bridge table that provides the UTC begin and End Hour datetimes for a local hour in an Olson timezone region. This table is used in Trend Reporting to give a high performance date calculation so that local hours and days can be used on graph axes and report heading with calculated totals. It is also used to load the summary tables using the local time dimension. High speed correct database-side time zone conversions are made by joining to this table.
time_zone_region	Time Zone Region		false	A region of the world defined in the Olson time zone database. Only the entries actually used by a FootPrints instance will be stored. The localized name of the region is provided by the Java.Util.Timezone.getDisplayName() method. Only the java class can be used to determine the time zone offset from GMT, which varies historically and in the future. See the Time Zone Local Hour table. Take aspirin as needed.

Attribute/Column(s) of Entity/Table ":Item_Shortname Balance"			
Physical Name	Name	Definition	Physical Data Type
item_id	Item Id	The ticket item id for this dynamic summary table that was generated for the item name.	bigint
begin_date_hour_local	Begin Date Hour Local	The begin whole hour of the time period for which this ticket is unclosed at the last minute of the first hour through the last minute of the end hour. Joins to TIME_DIM.	datetime
end_date_hour_local	End Date Hour Local	The end whole hour of the time period for which this ticket is unclosed at the last minute of the first hour through the last minute of the end hour. Joins to TIME_DIM.	datetime
activated_ct	Activated Count	Indicates if the Activation Event occurred prior to the end of the period. A one or zero integer count flag for the Item, designed to be summed in a query. It is important to identify the ONE ROW in this table which has the correct count flag by date joining, so that summing the count flag will not double count. That is the method of high performance aggregation.	numeric(13,2)

resolved_ct	Resolved Count	Indicates if the Resolved Event occurred prior to the end of the period. A one or zero integer count flag for the Item, designed to be summed in a query. It is important to identify the ONE ROW in this table which has the correct count flag by date joining, so that summing the count flag will not double count. That is the method of high performance aggregation.	numeric(13,2)
unresolved_ct	UnResolved Count	Indicates if the Resolved Event had not occurred prior to the end of the period. This is the inverse of the Resolved Ct, and makes it easier to count UnResolved. A one or zero integer count flag for the Item, designed to be summed in a query. It is important to identify the ONE ROW in this table which has the correct count flag by date joining, so that summing the count flag will not double count. That is the method of high performance aggregation.	numeric(13,2)

#### Attribute/Column(s) of Entity/Table ":item\_shortcode Event Sum"

Physical Name	Name	Definition	Physical Data Type
item_id	Item Id	The automatically generated primary key of the :item_shortcode_event_sum table.	bigint
event_datetime	Event DateTime	The exact event date and time in UTC. This is used to calculate exact times and durations. It is within the hour that is bound by the local hour.	datetime
creatn_clock_hours_ct	Creation Clock Hours Count	The elapsed clock hours from creation to the activity date. This does not include the time spent in DOES NOT COUNT FOR TIME states.	numeric(13,2)
creatn_work_hours_ct	Creation Work Hours Count	The elapsed work hours from creation to the activity date. This does not include the time spent in DOES NOT COUNT FOR TIME states. Work hours are defined by the schedule actual period table.	numeric(13,2)
all_creatn_clock_hours_ct	All Creation Clock Hours Count	The elapsed clock hours from creation to the activity date. The status flag DOES NOT COUNT FOR TIME is ignored here.	numeric(13,2)
all_creatn_work_hours_ct	All Creation Work Hours Count	The elapsed work hours from creation to the activity date. Work hours are defined by the schedule actual period table. The status flag DOES NOT COUNT FOR TIME is ignored here.	numeric(13,2)
actvn_clock_hours_ct	Activation Clock Hours Count	The elapsed clock hours from activation to the activity date. This does not include the time spent in DOES NOT COUNT FOR TIME states.	numeric(13,2)
actvn_work_hours_ct	Activation Work Hours Count	The elapsed work hours from activation to the activity date. This does not include the time spent in DOES NOT COUNT FOR TIME states. Work hours are defined by the schedule actual period table.	numeric(13,2)
all_actvn_clock_hours_ct	All Activation Clock Hours Count	The elapsed clock hours from activation to the activity date. The status flag DOES NOT COUNT FOR TIME is ignored here.	numeric(13,2)
all_actvn_work_hours_ct	All Activation Work Hours Count	The elapsed work hours from activation to the activity date. The status flag DOES NOT COUNT FOR TIME is ignored here. Work hours are defined by the schedule actual period table.	numeric(13,2)
life_cycle_event_id	Life Cycle Event Id	A Foreign Key to the Life Cycle Event table.	int
date_hour_local	Date Hour Local	A date time in UTC that corresponds to a whole hour date in the local timezone for the Date Hour Local. The Minutes portion of the date will usually be 00 but can be 30, 15, 45 in the case of half hour or quarter hour time zones. UTC never has daylight savings in effect.	datetime

#### Attribute/Column(s) of Entity/Table ":item\_shortcode Status"

Physical Name	Name	Definition	Physical Data Type
is_active	Is Active	Indicates if the state is an Activated state, used only to calculate Activated event date. Do not confuse with UnClosed, UnResolved.	bit
is_resolved	Is Resolved	Indicates if the state is an Resolved state, Used to calculate Resolved event date.	bit
is_closed	Is Closed	Indicates if the state is an Closed state, Used to calculate Closed event date. Currently, only one status is allow to be the Closed state.	bit
is_responded	Is Responded	Indicates if the state is an Responded state, Used to calculate Responded event date.	bit

does_ct_for_time	Does Count for Time	Indicates that this state does not count for time expended. This is used for the measures that do not include the ALL designation. It is anticipated that WAITING ON CUSTOMER RESPONSE and similar states would use this flag to show agent metrics without such states included in the time.	bit
item_status_name	Item Status Name	The name of the :item_shortcode Status.	varchar(255)
item_status_id	Item Status Id	The automatically generated primary key of the :item_shortcode_status table.	int

#### Attribute/Column(s) of Entity/Table ":item\_shortcode Transition"

Physical Name	Name	Definition	Physical Data Type
item_transn_id	Item Transition Id	The automatically generated primary key of the :item_shortcode_transn table.	bigint
hop_ct	Hop Count	The number of transitions. Open to Active to Closed will be stored additionally as Open to Closed with a hop count of 2.	int
is_longest_hop	Is Longest Hop	Indicates that this transition is a 'longest hop' transition. Many reports will want to find the longest transition between two states, and ignore cycles within the two.	bit
to_item_status_id	To Item Status Id	A Foreign Key to the :item_shortcode Status table.	int
from_item_status_id	From Item Status Id	A Foreign Key to the :item_shortcode Status table.	int
from_item_status_name	From Item Status Name	The name of the From Status.	varchar(255)
to_item_status_name	To Item Status Name	The name of the To Status.	varchar(255)

#### Attribute/Column(s) of Entity/Table ":item\_shortcode Transition Sum"

Physical Name	Name	Definition	Physical Data Type
item_id	Item Id	The automatically generated primary key of the :item_shortcode_transn_sum table.	bigint
transn_datetime	Transition DateTime	The exact event date and time in UTC. This is used to calculate exact times and durations. It is within the hour that is bound by the local hour.	datetime
clock_hours_ct	Clock Hours Count	The elapsed clock hours for the transition between the from and to states. This does not include the time spent in DOES NOT COUNT FOR TIME states.	numeric(13,2)
work_hours_ct	Work Hours Count	The elapsed work hours for the transition between the from and to states. This does not include the time spent in DOES NOT COUNT FOR TIME states. Work hours are defined by the schedule actual period table.	numeric(13,2)
all_clock_hours_ct	All Clock Hours Count	The elapsed clock hours for the transition between the from and to states. The status flag DOES NOT COUNT FOR TIME is ignored here.	numeric(13,2)
all_work_hours_ct	All Work Hours Count	The elapsed clock hours for the transition between the from and to states. The status flag DOES NOT COUNT FOR TIME is ignored here. Work hours are defined by the schedule actual period table.	numeric(13,2)
item_transn_id	Item Transition Id	A Foreign Key to the :item_shortcode Transition table.	bigint
date_hour_local	Date Hour Local	A date time in UTC that corresponds to a whole hour date in the local timezone for the Date Hour Local. The Minutes portion of the date will usually be 00 but can be 30, 15, 45 in the case of half hour or quarter hour time zones. UTC never has daylight savings in effect,	datetime

#### Attribute/Column(s) of Entity/Table "Life Cycle Event"

Physical Name	Name	Definition	Physical Data Type
life_cycle_event_id	Life Cycle Event Id	The automatically generated primary key of the life_cycle_event table.	int
life_cycle_event_name	Life Cycle Event Name	The name of the Life Cycle Event.	varchar(255)
life_cycle_event_descr	Life Cycle Event Description	A descriptive name of the Life Cycle Event. This descriptive name may be localized.	varchar(512)

#### Attribute/Column(s) of Entity/Table "Previous Timing"

Physical Name	Name	Definition	Physical Data Type
prev_day	Previous Day	The date for the start of the previous day for the start of the day that contains the current date hour local time.	datetime

prev_day_of_week	Previous Day of Week	The date for the start of the day one week back from the current date hour local time.	datetime
prev_day_of_month	Previous Day of Month	The date for the start of the day one month back from the current date hour local time.	datetime
prev_day_of_quarter	Previous Day of Quarter	The date for the start of the day one quarter back from the current date hour local time.	datetime
prev_day_of_year	Previous Day of Year	The date for the start of the day one year back from the current date hour local time.	datetime
prev_week	Previous Week	The date for the start of the week one week back from the start of the week that contains the current date hour local time.	datetime
prev_week_of_month	Previous Week of Month	The date for the start of the week whose number in the previous month corresponds to the week number in the current month of the current date hour local time.	datetime
prev_week_of_quarter	Previous Week of Quarter	The date for the start of the week whose number in the previous quarter corresponds to the week number in the current quarter of the current date hour local time.	datetime
prev_week_of_year	Previous Week of Year	The date for the start of the week whose number in the previous year corresponds to the week number in the current year of the current date hour local time.	datetime
prev_month	Previous Month	The start of the month one month before the month of the current date hour local time.	datetime
prev_month_of_quarter	Previous Month of Quarter	The start of the month three months before the month of the current date hour local time.	datetime
prev_month_of_year	Previous Month of Year	The start of the month twelve months before the month of the current date hour local time.	datetime
prev_quarter	Previous Quarter	The start of the quarter three months before the quarter of the current date hour local time.	datetime
prev_quarter_of_year	Previous Quarter of Year	The start of the quarter twelve months before the quarter of the current date hour local time.	datetime
prev_year	Previous Year	The start of the year one year before the year of the current date hour local time.	datetime
date_hour_local	Date Hour Local	A date time in UTC that cooresponds to a whole hour date in the local timezone for the Date Hour Local. The Minutes portion of the date will usually be 00 but can be 30, 15, 45 in the case of half hour or quarter hour time zones. UTC never has daylight savings is in effect,  This ties the previous and second previous dates to the date hour local of the time dimesnion table.	datetime
second_prev_day	Second Previous Day	The date for the start of the second previous day for the start of the day that contains the current date hour local time.	datetime
second_prev_day_of_week	Second Previous Day of Week	The date for the start of the day two weeks back from the current date hour local time.	datetime
second_prev_day_of_month	Second Previous Day of Month	The date for the start of the day two months back from the current date hour local time.	datetime
second_prev_day_of_quarter	Second Previous Day of Quarter	The date for the start of the day two quarters back from the current date hour local time.	datetime
second_prev_day_of_year	Second Previous Day of Year	The date for the start of the day two years back from the current date hour local time.	datetime
second_prev_week	Second Previous Week	The date for the start of the week two weeks back from the start of the week that contains the current date hour local time.	datetime
second_prev_week_of_month	Second Previous Week of Month	The date for the start of the week whose number in the second previous month corresponds to the week number in the current month of the current date hour local time.	datetime
second_prev_week_of_quarter	Second Previous Week of Quarter	The date for the start of the week whose number in the second previous quarter corresponds to the week number in the current quarter of the current date hour local time.	datetime
second_prev_week_of_year	Second Previous Week of Year	The date for the start of the week whose number in the second previous year corresponds to the week number in the current year of the current date hour local time.	datetime
second_prev_month	Second Previous Month	The start of the month two months before the month of the current date hour local time.	datetime
second_prev_month_of_quarter	Second Previous Month of Quarter	The start of the month six months before the month of the current date hour local time.	datetime
second_prev_month_of_year	Second Previous Month of Year	The start of the month 24 months before the month of the current date hour local time.	datetime
second_prev_quarter	Second Previous Quarter	The start of the quarter six months before the quarter of the current date hour local time.	datetime

second_prev_quarter_of_year	Second Previous Quarter of Year	The start of the quarter 24 months before the quarter of the current date hour local time.	datetime
second_prev_year	Second Previous Year	The start of the year two years before the year of the current date hour local time.	datetime

#### Attribute/Column(s) of Entity/Table "Time Dimension"

Physical Name	Name	Definition	Physical Data Type
day_of_week	Day of Week	Integer Day of Week.	int
td_day	TD Day	Integer value for the day portion of the date hour local.	int
td_week	TD Week	Integer value for the week of the year based on the ISO 8601 system used for calculating the week of the year.  The first week of the year corresponds to the first week in the year of the Gregorian calendar that contains 4 days starting on a Monday. Weeks are then incremented each Monday, until the next years count starts.	int
td_month	TD Month	Integer value for the month portion of the date hour local.	int
td_quarter	TD Quarter	Integer value for the quarter of the year. This is based on dividing 12 months into 4 quarters of three consecutive months.	int
td_year	TD Year	Integer value for the year portion of the date hour local.	int
day_date	Day Date	Today's date with time truncated.	datetime
start_of_week_date	Start of Week Date	The start of the week that contains the current date hour local time.	datetime
start_of_month_date	Start of Month Date	The start of the month that contains the current date hour local time.	datetime
start_of_quarter_date	Start of Quarter Date	The start of the quarter that contains the current date hour local time.	datetime
start_of_year_date	Start of Year Date	The start of the year that contains the current date hour local time.	datetime
td_hour	TD Hour	Integer value for the hour portion of the date hour local.	int
day_of_month	Day of Month	Integer Day of Month.	int
day_of_quarter	Day of Quarter	Integer Day of Quarter.	int
day_of_year	Day of Year	Integer Day of Year	int
week_of_month	Week of Month	Integer Week of month based on the ISO 8601 system used for calculating the week of the year.  The first week of the month corresponds to the first week in the month of the Gregorian calendar that contains 4 days starting on a Monday.	int
week_of_quarter	Week of Quarter	Integer Week of quarter based on the ISO 8601 system used for calculating the week of the year.  The first week of the quarter corresponds to the first week in the first quarter month of the Gregorian calendar that contains 4 days starting on a Monday.	int
date_hour_local	Date Hour Local	A date time in UTC that cooresponds to a whole hour date in the local timezone for the Date Hour Local. The Minutes portion of the date will usually be 00 but can be 30, 15, 45 in the case of half hour or quarter hour time zones. UTC never has daylight savings in effect,	datetime
end_of_hour_date	End of Hour Date	The next hour increment of the current date hour local time.	datetime
end_of_day_date	End of Day Date	The next day increment of the current date hour local time.	datetime
end_of_week_date	End of Week Date	The next week increment of the current date hour local time.	datetime
end_of_month_date	End of Month Date	The next month increment of the current date hour local time.	datetime
end_of_quarter_date	End of Quarter Date	The next quarter increment of the current date hour local time.	datetime
end_of_year_date	End of Year Date	The next year increment of the current date hour local time.	datetime

#### Attribute/Column(s) of Entity/Table "Time Zone Local Hour"

Physical Name	Name	Definition	Physical Data Type
begin_date_hour_utc	Begin Date Hour UTC	A date whole hour in the local timezone for the Begin Date Hour UTC. The Minutes portion of the date will always be 00. It is not stated whether daylight	datetime

		savings is in effect.	
end_date_hour_utc	End Date Hour UTC	A date whole hour in the local timezone for the End Date Hour UTC. The Minutes portion of the date will always be 00. It is not stated whether daylight savings is in effect.	datetime
begin_date_hour_local	Begin Date Hour Local	A date time in UTC that cooresponds to a whole hour date in the local timezone for the Begin Date Hour Local. The Minutes portion of the date will usually be 00 but can be 30, 15, 45 in the case of half hour or quarter hour time zones. UTC never has daylight savings is in effect,	datetime
end_date_hour_local	End Date Hour Local	A date time in UTC that cooresponds to a whole hour date in the local timezone for the End Date Hour Local. The Minutes portion of the date will usually be 00 but can be 30, 15, 45 in the case of half hour or quarter hour time zones. UTC never has daylight savings is in effect,	datetime
begin_is_daylight_savings	Begin Is Daylight Savings	Flag indicating if daylight savings is in effect for the begin time. This helps identify the special hours that are missing or duplicated during daylight savings days.	bit
time_zone_region_id	Time Zone Region Id	An alternate Key to the Time Zone Region table, which is arbitrarily calculated and assigned by FootPrints. The reason for this is performance of the Foreign Key join to large numbers of rows. There is no suitable short data length key available in a published standard data source.	bigint
time_zone_local_hour_id	Time Zone Local Hour Id	The automatically generated primary key of the time_zone_local_hour table.	bigint
begin_date_hour_local_s	Begin Date Hour Local S	A string version of the local date that is used in order to error correct JDBC/ Hibernate persistence of the date which produces errors on Daylight Savings Time boundaries.	varchar(23)
end_date_hour_local_s	End Date Hour Local S	A string version of the local date that is used in order to error correct JDBC/ Hibernate persistence of the date which produces errors on Daylight Savings Time boundaries.	varchar(23)
begin_date_hour_utc_s	Begin Date Hour UTC S	A string version of the UTC date that is used in order to error correct JDBC/ Hibernate persistence of the date which produces errors on Daylight Savings Time boundaries.	varchar(23)
end_date_hour_utc_s	End Date Hour UTC S	A string version of the UTC date that is used in order to error correct JDBC/ Hibernate persistence of the date which produces errors on Daylight Savings Time boundaries.	varchar(23)
end_is_daylight_savings	End Is Daylight Savings	Flag indicating if daylight savings is in effect for the end time. This helps identify the special hours that are missing or duplicated during daylight savings days.	bit

#### Attribute/Column(s) of Entity/Table "Time Zone Region"

Physical Name	Name	Definition	Physical Data Type
time_zone_region_name	Time Zone Region Name	The name of the Time Zone Region. This is 2. The 'primary key' of the java timezone is the string region identifier in English, example 'America/Phoenix'. There is no numeric identifier. As the JRE is updated, or specific classes are updated, the list could change to include new entries, and the detail data behind the regions could change.	varchar(255)
time_zone_region_abbr	Time Zone Region Abbreviation	A well-known string identifier used as a code to identify the Time Zone Region. This could be provided by user entry, but it is not available from the Timezone class. It should probably not be used.	varchar(255)
time_zone_region_id	Time Zone Region Id	An alternate Key to the Time Zone Region table, which is arbitrarily calculated and assigned by FootPrints. The reason for this is performance of the Foreign Key join to large numbers of rows. There is no suitable short data length key available in a published standard data source.	bigint

#### ER Diagram(s) of Subject Area "305 - Data Mart Load"

Name	Physical Display Level
Logical	Column
Physical	Column

#### Entity/Table(s) of Subject Area "305 - Data Mart Load"

Physical Name	Name	Logical Only	Do Not Generate	Definition
audit_txn	Audit Transaction		false	A record of the Principal and create time of the start of the commit time of a logical transaction. In many cases multiple Audit Log entries will use the same transaction.

data_load_type	Data Load Type		false	A lookup table for the load type. Values include: FULL_LOAD Full deletion and re-adding of item data in the data mart INCR_LOAD Incremental deletion and re-adding of item data changed in the data mart.
data_mart_load_history	Data Mart Load History		false	The history of data mart full and incremental loads. A full load is chunked into 10K row batches and is restartable and re-runnable. An incremental load processes all the activity since the previous run, which is assumed to be a short duration. This table holds an Audit Transaction Id reference to the Audit tables, which identify the last transaction processed. This table is designed to handle many types of data marts in multiple different containers. Currently, only two Ticket data marts are defined and run. If a new data mart fact table were defined, and stored procedures written for the full and incremental load, and all entries defined in the request tables, the data mart java process should manage the loading.
data_mart_load_lock	Data Mart Load Lock		false	A pessimistic lock table designed to prevent multiple loads from running against the same item. It is also used to count and throttle the number of simultaneous loads. Both the full and incremental loads drop and create tables so that faster non-logged database operations can be used (Select * INTO..., CREATE TABLE AS SELECT *,...) The tables have item specific names, so they must single thread.
data_mart_rqst_history	Data Mart Request History	false	false	A history table detailing the results of Data Mart request execution. It is also used to report the last run of the data mart on data mart reports. It may need to be purged from time to time as each request is logged even if no updates occur, and the frequency defaults to 10 minutes.
data_mart_rqst_queue	Data Mart Request Queue	false	false	The queue of active data mart load requests. An entry in this table will be picked up by the Java Data Mart manager and used to initiate and then maintain state of the request. The FootPrints UI controls data mart execution by creating and updating rows in this table. During upgrade, a request to update all data marts is added, and once the system comes up, Data Marts are updated in round robin fashion.
data_mart_rqst_state	Data Mart Request State	false	false	A lookup table for the state of data mart request processing. Values include: PENDING, CANCEL_REQUEST, IN_PROCESS, DONE, FAILED, CANCELED.
data_mart_rqst_type	Data Mart Request Type	false	false	A lookup table for the data mart load request type. Values include: FULL_LOAD, SCHEDULED_INC_LOAD, REQUESTED_INC_LOAD, FULL_LOAD_WITH_SCHD_UPDATE, REBUILD_ALL_DATAMARTS.
defn	Definition		false	A definition holds the structural and behavioral information needed for an application element that is configurable by the system designer and/or the end user at definition time. Current Footprints uses a combination of directories, files, tables, and programming code to represent this information. The new FootPrints will provide all of the same information within the definition. The definition can be persisted in a structure that is isolated at the Instance Level. The data for the definition will be stored as Configuration Settings associated with the Definition Version. See the database views v_defn_ver_active, v_cntrn_item_field.

#### Attribute/Column(s) of Entity/Table "Audit Transaction"

Physical Name	Name	Definition	Physical Data Type
---------------	------	------------	--------------------

audit_txn_id	Audit Transaction Id	The automatically generated primary key of the audit_txn table.	bigint
txn_start_date	Transaction Start Date	The transaction date of the Audit Log entry. This date may not be in strict order with the Audit Log id in the event that features are used to continue a previous Audit Log transaction.	datetime
app_user_id	Application User Id	A Foreign Key to the Application User table. The audit trail transaction may or may not be associated with a customer or agent user. If this column is null, then a system component should be identified.	bigint
sys_component_id	System Component Id	This is a foreign key to the system component table. The audit trail transaction may or may not be associated with a customer or agent user. The system component will be set if there is no application user identified. It identifies the source of the transaction. Values include SYSTEM, WEB_SERVICES, ASSET_MANAGEMENT, CALENDAR, IMPORTING, etc. All identifiers are negative, so that they can be distinguished from userids in case they are both stored in the same dynamic table CreatedBy, and UpdatedBy fields. We are going to try first normal form normalization here and see if it works.	int
audit_rule_name_id	Audit Rule Name Id	The primary key is the original rule definition id. It will not change as the rule is renamed. If the rule is deleted, the deleted date will be filled in, but the rule name will still be known in this table.	bigint

#### Attribute/Column(s) of Entity/Table "Data Load Type"

Physical Name	Name	Definition	Physical Data Type
data_load_type_name	Data Load Type Name	The name of the Data Load Type.	varchar(20)
data_load_type_descr	Data Load Type Description	A descriptive name of the Data Load Type. This descriptive name may be localized.	varchar(512)
load_proc_suffix	Load Process Suffix	The name of the DBMS_Object_Name. It is a database object name, limited to 30 characters in length.	varchar(30)

#### Attribute/Column(s) of Entity/Table "Data Mart Load History"

Physical Name	Name	Definition	Physical Data Type
data_mart_load_history_id	Data Mart Load History Id	The automatically generated primary key of the data_mart_load_history table.	bigint
data_mart_name	Data Mart Name	The name of the Data Mart Load History.	varchar(255)
item_defn_id	Item Definition Id	A Foreign Key to the Definition table.	bigint
load_start_date	Load Start Date	Begin Date.	datetime
load_end_date	Load End Date	Load end date in UTC	datetime
last_audit_txn_id	Last Audit Transaction Id	The automatically generated primary key of the audit_txn table.	bigint
is_waiting	Is Waiting	Indicates if Is Waiting.	bit
load_proc_name	Load Process Name	The name of the DBMS_Object_Name. It is a database object name, limited to 30 characters in length.	varchar(30)
data_load_type_name	Data Load Type Name	The name of the Data Load Type.	varchar(20)

#### Attribute/Column(s) of Entity/Table "Data Mart Load Lock"

Physical Name	Name	Definition	Physical Data Type
data_mart_name	Data Mart Name	The name of the Data Mart Load Lock.	varchar(255)
item_defn_id	Item Definition Id	A Foreign Key to the Definition table.	bigint
is_locked	Is Locked	Indicates if Is Locked.	bit
owning_proc_id	Owning Process Id	The name of the Data Mart Load Lock.	varchar(255)
data_mart_load_lock_id	Data Mart Load Lock Id	A Foreign Key to the Data Mart Load Lock table.	bigint

#### Attribute/Column(s) of Entity/Table "Data Mart Request History"

Physical Name	Name	Definition	Physical Data Type
data_mart_rqst_history_id	Data Mart Request History Id	The automatically generated primary key of the data_mart_rqst_history table.	bigint
item_defn_id	Item Definition Id	A Foreign Key to the Definition table.	bigint
rqst_begin_date	Request Begin Date	Request Begin Date	datetime
rqst_begin_date_utc_string	Request Begin Date UTC String	Begin Date as a UTC string to avoid JDBC/Hibernate errors.	varchar(23)



rqst_end_date	Request End Date	End Date	datetime
rqst_end_date_utc_string	Request End Date UTC String	End Date as a UTC string to avoid JDBC/Hibernate errors.	varchar(23)
data_mart_rqst_type_id	Data Mart Request Type Id	A Foreign Key to the Data Mart Request Type table.	int
data_mart_rqst_state_id	Data Mart Request State Id	A Foreign Key to the Data Mart Request State table.	int

#### Attribute/Column(s) of Entity/Table "Data Mart Request Queue"

Physical Name	Name	Definition	Physical Data Type
data_mart_rqst_queue_id	Data Mart Request Queue Id	The automatically generated primary key of the data_mart_rqst_queue table.	bigint
item_defn_id	Item Definition Id	A Foreign Key to the Definition table.	bigint
rqst_date	Request Date	Request date	datetime
rqst_date_utc_string	Request Date UTC String	Request Date as a UTC string to avoid JDBC/Hibernate errors.	varchar(23)
data_mart_rqst_type_id	Data Mart Request Type Id	A Foreign Key to the Data Mart Request Type table.	int
data_mart_rqst_state_id	Data Mart Request State Id	A Foreign Key to the Data Mart Request State table.	int

#### Attribute/Column(s) of Entity/Table "Data Mart Request State"

Physical Name	Name	Definition	Physical Data Type
data_mart_rqst_state_id	Data Mart Request State Id	The automatically generated primary key of the data_mart_rqst_state table.	int
data_mart_rqst_state_name	Data Mart Request State Name	The name of the Data Mart Request State.	varchar(255)
data_mart_rqst_state_descr	Data Mart Request State Description	A descriptive name of the Data Mart Request State. This descriptive name may be localized.	varchar(512)

#### Attribute/Column(s) of Entity/Table "Data Mart Request Type"

Physical Name	Name	Definition	Physical Data Type
data_mart_rqst_type_name	Data Mart Request Type Name	The name of the Data Mart Request Type.	varchar(255)
data_mart_rqst_type_descr	Data Mart Request Type Description	A descriptive name of the Data Mart Request Type. This descriptive name may be localized.	varchar(512)
data_mart_rqst_type_id	Data Mart Request Type Id	The automatically generated primary key of the data_mart_rqst_type table.	int

#### Attribute/Column(s) of Entity/Table "Definition"

Physical Name	Name	Definition	Physical Data Type
defn_id	Definition Id	The automatically generated primary key of the defn table.	bigint
defn_name	Definition Name	The name of the Definition.	nvarchar(4000)
dsble_date	Disabled Date	The date that this instance of Definition is logically deleted and disabled.	datetime
defn_guid	Definition GUID	A Global Unique Identifier search key for the Definition, composed of letters and numbers. It is computed by the client and typically used in unstructured data.	varchar(50)
defn_type_name	Definition Type Name	The unique string identifier for an instance of Definition Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)
defn_subtype_id	Definition Subtype Id	A Foreign Key to the Definition Subtype table.	int
tmpl_export_guid	Template Export GUID	The value of a Definition, stored as a string.	VARCHAR(50)

#### ER Diagram(s) of Subject Area "310 - Auxiliary Tables"

Name	Physical Display Level
Logical	Column
Physical	Column

#### Entity/Table(s) of Subject Area "310 - Auxiliary Tables"

Physical Name	Name	Logical Only	Do Not Generate	Definition
cfg_file_repository	Configuration File Repository		false	Now during publishing, new dynamic HBM files will be stored in DB and fetched during session restart.

				<p>Configuration files are also stored here.</p> <p>Design Notes:</p> <p>Identify all configuration files that application (Renoir) need for startup, and running. - Very important that we identify all the files that application is using today. This will help us with #5.</p> <p>a. Examples; Hibernate Configuration Files, and . Properties files. This does not include Spring Specific files.</p> <p>/application/src/main/resources/footprints-email-incoming.properties  /business/src/main/resources/ical4j.properties  /business/src/main/resources/assetcore-searchdevice-one-result-response.xml  /business/src/main/resources/cosmo.version.txt  /business/src/main/resources/jecllicense  /business/src/main/resources/privkeystore  /business/src/main/resources/timezone.alias  /core/src/main/resources/mappings/*.  /core/src/main/resources/footprints-external-data.properties  /core/src/main/conf/mysql/*.  /core/src/main/conf/oracle/*.  /core/src/main/conf/postgres/*.  /core/src/main/conf/sqlserver/*.  /infrastructure/src/main/resources/dbschema/*.  /infrastructure/src/main/conf/*.properties  /web/src/main/conf/footprints-application-key.properties  /build/src/main/deployment/log4j.properties</p> <p>2. Where to store the configuration files (file data store) - This is</p> <p>a. Database of File Storage  We plan doing this as more general file Repository interface that will allow us to select any specific implementation. But in our case we decided to use DB.</p> <p>3. How to retrieve the files from file data store -  Store them in blob format of DB and on demand transform into ByteArrayInputStream.</p>
--	--	--	--	---

extl_sys_certificate	External System Certificate		false	Holds information about external system certificates.
----------------------	-----------------------------	--	-------	---

#### Attribute/Column(s) of Entity/Table "Configuration File Repository"

Physical Name	Name	Definition	Physical Data Type
cfg_file_related_path	Configuration File Related Path	File name and extension, although some names have non-standard extensions.	varchar(255)
cfg_file_content	Configuration File Content	The file contents as a binary array.	varbinary(max)

#### Attribute/Column(s) of Entity/Table "External System Certificate"

Physical Name	Name	Definition	Physical Data Type
extl_sys_certificate_id	External System Certificate Id	The automatically generated primary key of the extl_sys_certificate table.	bigint
extl_sys_certificate_name	External System Certificate Name	The name of the External System Certificate.	varchar(255)
certificate	Certificate	Certificate binary array.	varbinary(max)

#### ER Diagram(s) of Subject Area "320 - Report"

Name	Physical Display Level
Logical	Column
Physical	Column

#### Entity/Table(s) of Subject Area "320 - Report"

Physical Name	Name	Logical Only	Do Not Generate	Definition
app_user	Application User		false	A user authorized to use the FootPrints system. The user is either a Customer User or an Agent User. The application user holds the minimum information needed to identify the user and associate the user with a user profile, where user settings are stored. A contact is not necessarily an application user. An application user entry is required in order for the user to access any online functions.
defn	Definition		false	A definition holds the structural and behavioral information needed for an application element that is configurable by the system designer and/or the end user at definition time. Current Footprints uses a combination of directories, files, tables, and programming code to represent this information. The new FootPrints will provide all of the same information within the definition. The definition can be persisted in a structure that is isolated at the Instance Level. The data for the definition will be stored as Configuration Settings associated with the Definition Version. See the database views v_defn_ver_active, v_cntrn_item_field.
defn_ver	Definition Version		false	A version of a Definition with a version identifier that changes when data changes are made to the template. The definition version will suffice to store the current state of the FootPrints Configuration. A full copy of the definition tree is versioned when a container is published. See the database views v_defn_ver_active, v_cntrn_item_field.
recur_period_type	Recur Period Type		false	The recurrence type. Choices include No Recurrence, Daily, Weekly, Monthly. This is an enum table that must be kept in sync with the application code. It explains the values for database queries only, and is not mapped for application access.
recur_schd_type	Recur Schedule Type		false	The Schedule type. Choices include IMMEDIATELY/ SCHEDULED_ONCE/ MANUAL_ONLY. This is an enum table that must be kept in sync with the application code. It explains the values for database queries only, and is not mapped for application access.

rpt	Report		false	A dashboard report definition. The report information is used with the Logi Analytics embedded reporting engine to render reports in the UI.
rpt_cfg	Report Configuration		false	Configuration data values associated with a dashboard report.
rpt_format	Report Format		false	Report Format such as PDF, EXCEL.
rpt_schdd_task	Report Scheduled Task		false	The report schedule is association between a report and the scheduled task. Report schedule specific information can also be placed here.
rpt_visibility_type	Report Visibility Type		false	A lookup table for report visibility type. Values include: Private, Shared.
schd_task_run_stat	Schedule Task Run Statistic		false	An individual statistic item for a run. This has a compound key so it can be mapped as a simple key value map.
schdd_task	Scheduled Task		false	A scheduled task that may include recurrence. This is all of the information needed to provide the scheduler with data to execute a task. It is used for Reports, but is intended to be generic for other scheduled tasks. This is the root object. Report specific information will go in an association table.
schdd_task_notifn	Scheduled Task Notification		false	The collection of notification recipients for the Scheduled Task. Having true references here will make it easier to maintain changes to user's information such as LDAP lookup, name changes, and user deletion or replacement use cases.
schdd_task_run	Scheduled Task Run		false	An instance of a run for a schedule task used for last run and previous run information. The run date and the status of the run are kept here. There is only one row per run, so the time of each status change is not known.
schdd_task_stat	Scheduled Task Statistic		false	A named statistic that can be associated with a run. This allows free form name value statistics to be dynamically associated with the run.
schdd_task_status	Scheduled Task Status		false	The run status enumeration. Choices include Scheduled, In Progress, Completed, Completed with Errors.
schdd_task_type	Scheduled Task Type		false	A lookup table for the Scheduled Task Type that must be kept in sync with an enumeration in the Java. Values include: IMPORT_USER, MASSS_EMAIL, REPORT_SCHEDULE etc.
val_type_cd	Value Type Code		false	A lookup table for the data value type discriminator. Many tables share this discriminator for name value pair datatypes. Values include: B Boolean D Timestamp - Date + Time I Integer L Long R Real Number S String T Text - long string

#### Attribute/Column(s) of Entity/Table "Application User"

Physical Name	Name	Definition	Physical Data Type
pwd	Password	This space intentionally left blank.	varchar(255)
user_name	User Name	The name of the Application User.	nvarchar(4000)
cust_login_type_id	Customer Login Type Id	A Foreign Key to the Customer Login Type table.	int
app_user_id	Application User Id	The automatically generated primary key of the app_user table.	bigint
lic_type_id	License Type Id	A Foreign Key to the License Type table.	bigint
user_login	User Login	The login name used to access the Application. This may not be filled in if external application security is used.	nvarchar(4000)
email_addr	Email Address	An email address for the Application User.	nvarchar(4000)
delete_date	Delete Date	The value of a Application User, stored as a date.	datetime

#### Attribute/Column(s) of Entity/Table "Definition"

Physical Name	Name	Definition	Physical Data Type
---------------	------	------------	--------------------

defn_id	Definition Id	The automatically generated primary key of the defn table.	bigint
defn_name	Definition Name	The name of the Definition.	nvarchar(4000)
dsble_date	Disabled Date	The date that this instance of Definition is logically deleted and disabled.	datetime
defn_guid	Definition GUID	A Global Unique Identifier search key for the Definition, composed of letters and numbers. It is computed by the client and typically used in unstructured data.	varchar(50)
defn_type_name	Definition Type Name	The unique string identifier for an instance of Definition Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)
defn_subtype_id	Definition Subtype Id	A Foreign Key to the Definition Subtype table.	int
tmpl_export_guid	Template Export GUID	The value of a Definition, stored as a string.	VARCHAR(50)

#### Attribute/Column(s) of Entity/Table "Definition Version"

Physical Name	Name	Definition	Physical Data Type
defn_id	Definition Id	A Foreign Key to the Definition table.	bigint
defn_ver_id	Definition Version Id	The automatically generated primary key of the defn_ver table.	bigint
defn_ver_name	Definition Version Name	The name of the Definition Version.	nvarchar(4000)
defn_ver_status_id	Definition Version Status Id	A Foreign Key to the Definition Version Status table.	int
defn_ver_guid	Definition Version GUID	A Global Unique Identifier search key for the Definition Version, composed of letters and numbers. It is computed by the client and typically used in unstructured data.	varchar(50)

#### Attribute/Column(s) of Entity/Table "Recur Period Type"

Physical Name	Name	Definition	Physical Data Type
recur_period_type_id	Recur Period Type Id	The unique identifier for an instance of Recur Period Type. This identifier must be kept in sync with programming code that uses an enumeration or fixed constant to identify particular values.	integer
recur_period_type_name	Recur Period Type Name	The unique string identifier for an instance of Recur Period Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)

#### Attribute/Column(s) of Entity/Table "Recur Schedule Type"

Physical Name	Name	Definition	Physical Data Type
recur_schd_type_id	Recur Schedule Type Id	The unique identifier for an instance of Recur Schedule Type. This identifier must be kept in sync with programming code that uses an enumeration or fixed constant to identify particular values.	integer
recur_schd_type_name	Recur Schedule Type Name	The unique string identifier for an instance of Recur Schedule Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)

#### Attribute/Column(s) of Entity/Table "Report"

Physical Name	Name	Definition	Physical Data Type
rpt_id	Report Id	The automatically generated primary key of the rpt table.	bigint
rpt_name	Report Name	The name of the Report.	varchar(255)
created_by	Created By	A Foreign Key to the Application User table.	bigint
created_on	Created On	The value of a Report, stored as a date.	datetime
mod_by	Modified By	A Foreign Key to the Application User table.	bigint
mod_on	Modified On	The value of a Report, stored as a date.	datetime
rpt_visibility_type_id	Report Visibility Type Id	A Foreign Key to the Report Visibility Type table.	int
next_schdd_run_date	Next Scheduled Run Date	Next run date	datetime
rpt_comment	Report Comment	The value of a Report, stored as a string.	varchar(255)

item_defn_id	Item Definition Id	A Foreign Key to the Definition Version table.	bigint
rpt_descr	Report Description	A descriptive name of the Report. This descriptive name may be localized.	varchar(512)
rpt_title	Report Title	The name of the Report.	varchar(255)
rpt_grp_id	Report Group Id	A Foreign Key to the Definition Version table.	bigint
deleted_on	Deleted On	The value of a Report, stored as a date.	datetime
rpt_type_id	Report Type Id	A Foreign Key to the Definition table.	bigint

#### Attribute/Column(s) of Entity/Table "Report Configuration"

Physical Name	Name	Definition	Physical Data Type
rpt_cfg_id	Report Configuration Id	The automatically generated primary key of the rpt_cfg table.	bigint
rpt_id	Report Id	A Foreign Key to the Report table.	bigint
rpt_cfg_name	Report Configuration Name	The name of the Report Configuration.	varchar(255)
string_val	String Value	The value of a Report Configuration, stored as a string.	varchar(255)
text_val	Text Value	The value of a Report Configuration, stored as a long text object.	varchar(max)
date_val	Date Value	The value of a Report Configuration, stored as a date.	datetime
real_val	Real Value	The value of a Report Configuration stored as real.	numeric(38,20)
int_val	Int Value	The value of a Report Configuration stored as boolean.	int
long_val	Long Value	The value of a Report Configuration stored as Long.	bigint
boolean_val	Boolean Value	The Boolean Value.	bit
val_type_cd	Value Type Code	The data type for a Value Type Code. The choices are 'S' string, 'D' date, 'T' text, 'I' integer.	char(1)

#### Attribute/Column(s) of Entity/Table "Report Format"

Physical Name	Name	Definition	Physical Data Type
rpt_format_name	Report Format Name	The unique string identifier for an instance of Report Format. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)
rpt_format_descr	Report Format Description	A descriptive name of the Report Format. This descriptive name may be localized.	varchar(512)

#### Attribute/Column(s) of Entity/Table "Report Scheduled Task"

Physical Name	Name	Definition	Physical Data Type
rpt_id	Report Id	A Foreign Key to the Report table.	bigint
schdd_task_id	Scheduled Task Id	A Foreign Key to the Scheduled Task table.	bigint
mail_empty_rpt_flag	Mail Empty Report Flag	A boolean flag for xxx	numeric(1)
rpt_folder_name	Report Folder Name	A search key for the Report Scheduled Task, composed of letters and numbers.	varchar(255)
rpt_format_name	Report Format Name	The unique string identifier for an instance of Report Format. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)

#### Attribute/Column(s) of Entity/Table "Report Visibility Type"

Physical Name	Name	Definition	Physical Data Type
rpt_visibility_type_id	Report Visibility Type Id	The automatically generated primary key of the rpt_visibility_type table.	int
rpt_visibility_type_name	Report Visibility Type Name	The name of the Report Visibility Type.	varchar(255)
rpt_visibility_type_descr	Report Visibility Type Description	A descriptive name of the Report Visibility Type. This descriptive name may be localized.	varchar(512)

#### Attribute/Column(s) of Entity/Table "Schedule Task Run Statistic"

Physical Name	Name	Definition	Physical Data Type
schdd_task_run_id	Scheduled Task Run Id	A Foreign Key to the Scheduled Task Run table.	bigint

schdd_task_stat_id	Scheduled Task Statistic Id	The unique identifier for an instance of Scheduled Task Statistic. This identifier must be kept in sync with programming code that uses an enumeration or fixed constant to identify particular values.	integer
int_val	Int Value	The value of a Schedule Task Run Statistic stored as boolean.	integer

#### Attribute/Column(s) of Entity/Table "Scheduled Task"

Physical Name	Name	Definition	Physical Data Type
schdd_task_name	Scheduled Task Name	The name of the Scheduled Task.	varchar(255)
schdd_task_id	Scheduled Task Id	The automatically generated primary key of the scheduled_task table.	bigint
schdd_task_descr	Scheduled Task Description	A descriptive name of the Scheduled Task. This descriptive name may be localized.	varchar(512)
recur_schd_type_id	Recur Schedule Type Id	The unique identifier for an instance of Recur Schedule Type. This identifier must be kept in sync with programming code that uses an enumeration or fixed constant to identify particular values.	integer
begin_date	Begin Date	Begin Date.	datetime
recur_period_type_id	Recur Period Type Id	The unique identifier for an instance of Recur Period Type. This identifier must be kept in sync with programming code that uses an enumeration or fixed constant to identify particular values.	integer
recur_hour	Recur Hour	The Hours portion of a time of day	integer
recur_minute	Recur Minute	The minutes portion of a time of day	integer
recur_days_of_week	Recur Days of Week	Recur Days of Week.	varchar(255)
recur_days_of_month	Recur Days of Month	Recur Days of Month.	varchar(255)
create_date	Create Date	The date that the Scheduled Task was created.	datetime
create_app_user_id	Create Application User Id	A Foreign Key to the Application User table.	bigint
schdd_task_type_id	Scheduled Task Type Id	The unique identifier for an instance of Scheduled Task Type. This identifier must be kept in sync with programming code that uses an enumeration or fixed constant to identify particular values.	integer

#### Attribute/Column(s) of Entity/Table "Scheduled Task Notification"

Physical Name	Name	Definition	Physical Data Type
schdd_task_id	Scheduled Task Id	A Foreign Key to the Scheduled Task table.	bigint
app_user_id	Application User Id	A Foreign Key to the Application User table.	bigint
team_id	Team Id	A Foreign Key to the Team table.	bigint
cntc_defn_id	Contact Definition Id	A Foreign Key to the Definition table.	bigint
email_addr	Email Address	An email address for the Scheduled Task Notification.	nvarchar(4000)
schdd_task_notifn_id	Scheduled Task Notification Id	The automatically generated primary key of the scheduled_task_notification table.	bigint
cntc_item_id	Contact Item Id	A Foreign Key to the Item table.	bigint
all_cntcs_flag	All_Contacts_Flag	A boolean flag for xxx	numeric(1)
cntc_email	Contact Email	An email address for the Scheduled Task Notification.	nvarchar(4000)

#### Attribute/Column(s) of Entity/Table "Scheduled Task Run"

Physical Name	Name	Definition	Physical Data Type
schdd_task_run_id	Scheduled Task Run Id	The automatically generated primary key of the scheduled_task_run table.	bigint
schdd_task_id	Scheduled Task Id	A Foreign Key to the Scheduled Task table.	bigint
schdd_task_status_id	Scheduled Task Status Id	The unique identifier for an instance of Scheduled Task Status. This identifier must be kept in sync with programming code that uses an enumeration or fixed constant to identify particular values.	integer
status_chg_date	Status Change Date	The date that the Scheduled Task Run was created.	datetime
dtl_status	Detail Status	The name of the Scheduled Task Run.	varchar(255)
schdd_run_date	Scheduled Run Date		datetime

#### Attribute/Column(s) of Entity/Table "Scheduled Task Statistic"

Physical Name	Name	Definition	Physical Data Type
schdd_task_stat_id	Scheduled Task Statistic Id	The unique identifier for an instance of Scheduled Task Statistic. This identifier must be kept in sync with programming code that uses an enumeration or fixed constant to identify particular values.	integer

schdd_task_stat_name	Scheduled Task Statistic Name	The unique string identifier for an instance of Scheduled Task Statistic. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)
----------------------	-------------------------------	--	--------------

#### Attribute/Column(s) of Entity/Table "Scheduled Task Status"

Physical Name	Name	Definition	Physical Data Type
schdd_task_status_id	Scheduled Task Status Id	The unique identifier for an instance of Scheduled Task Status. This identifier must be kept in sync with programming code that uses an enumeration or fixed constant to identify particular values.	integer
schdd_task_status_name	Scheduled Task Status Name	The unique string identifier for an instance of Scheduled Task Status. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)

#### Attribute/Column(s) of Entity/Table "Scheduled Task Type"

Physical Name	Name	Definition	Physical Data Type
schdd_task_type_id	Scheduled Task Type Id	The unique identifier for an instance of Scheduled Task Type. This identifier must be kept in sync with programming code that uses an enumeration or fixed constant to identify particular values.	integer
schdd_task_type_name	Scheduled Task Type Name	The unique string identifier for an instance of Scheduled Task Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)

#### Attribute/Column(s) of Entity/Table "Value Type Code"

Physical Name	Name	Definition	Physical Data Type
val_type_cd	Value Type Code	The data type for a Value Type Code. The choices are 'S' string, 'D' date, 'T' text, 'I' integer.	char(1)
val_type_name	Value Type Name	The name of the Value Type Code.	varchar(255)

#### ER Diagram(s) of Subject Area "332 - Change Management Voting"

Name	Physical Display Level
Logical	Column
Physical	Column

#### Entity/Table(s) of Subject Area "332 - Change Management Voting"

Physical Name	Name	Logical Only	Do Not Generate	Definition
app_user	Application User		false	A user authorized to use the FootPrints system. The user is either a Customer User or an Agent User. The application user holds the minimum information needed to identify the user and associate the user with a user profile, where user settings are stored. A contact is not necessarily an application user. An application user entry is required in order for the user to access any online functions.
defn	Definition		false	A definition holds the structural and behavioral information needed for an application element that is configurable by the system designer and/or the end user at definition time. Current Footprints uses a combination of directories, files, tables, and programming code to represent this information. The new FootPrints will provide all of the same information within the definition. The definition can be persisted in a structure that is isolated at the Instance Level. The data for the definition will be stored as Configuration Settings associated with the Definition Version. See the database views v_defn_ver_active, v_cntrn_item_field.



item	Item	true	true	The Item is the abstract class for Issue, Configuration Item, Address Book entry, and other information items that are the main information entities in FootPrints. The Item structure and behavior is defined by its Definition. The Item is namespaced, identified by, and logically stored in the single parent container. It can be referenced in the same manner both inside and outside the container. The Item is the storage model and the implementation for the Item Definition. There are many important specializations of the Item class, and there even more Template definitions of Container/Item that provide an end-user application experience to the End-User. The Item implementation must implement flexible schema in an easy to manage, well engineered, and high performing way. The item can store user-defined and user-named fields defined in its definition. The Item could implement the nice-to-have features such as an inheritance model for Item subtypes, and have the ability to represent collections of subordinate items. The Item data can be imported, exported, or synchronized with external data sources. The Item data can be linked and synchronized with other items in the same Instance.
vote	Vote		false	The vote of a single approver on a single item.
vote_history	Vote History		false	The history of votes in the case that voting changes.
voting_result	Voting Result		false	The root voting result entry for a ticket item that requires voting approval.

#### Attribute/Column(s) of Entity/Table "Application User"

Physical Name	Name	Definition	Physical Data Type
pwd	Password	This space intentionally left blank.	varchar(255)
user_name	User Name	The name of the Application User.	nvarchar(4000)
cust_login_type_id	Customer Login Type Id	A Foreign Key to the Customer Login Type table.	int
app_user_id	Application User Id	The automatically generated primary key of the app_user table.	bigint
lic_type_id	License Type Id	A Foreign Key to the License Type table.	bigint
user_login	User Login	The login name used to access the Application. This may not be filled in if external application security is used.	nvarchar(4000)
email_addr	Email Address	An email address for the Application User.	nvarchar(4000)
delete_date	Delete Date	The value of a Application User, stored as a date.	datetime

#### Attribute/Column(s) of Entity/Table "Definition"

Physical Name	Name	Definition	Physical Data Type
defn_id	Definition Id	The automatically generated primary key of the defn table.	bigint
defn_name	Definition Name	The name of the Definition.	nvarchar(4000)
dsble_date	Disabled Date	The date that this instance of Definition is logically deleted and disabled.	datetime
defn_guid	Definition GUID	A Global Unique Identifier search key for the Definition, composed of letters and numbers. It is computed by the client and typically used in unstructured data.	varchar(50)
defn_type_name	Definition Type Name	The unique string identifier for an instance of Definition Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)
defn_subtype_id	Definition Subtype Id	A Foreign Key to the Definition Subtype table.	int
tmpl_export_guid	Template Export GUID	The value of a Definition, stored as a string.	VARCHAR(50)

#### Attribute/Column(s) of Entity/Table "Item"

Physical Name	Name	Definition	Physical Data Type
---------------	------	------------	--------------------

item_key	Item Key	A search key for the Item, composed of letters and numbers.	varchar(255)
item_id	Item Id	The automatically generated primary key of the item table.	bigint
cntrn_id	Container Id	A Foreign Key to the Container table.	bigint
item_defn_id	Item Definition Id	A Foreign Key to the Definition table for an instance of type Definition.	bigint

#### Attribute/Column(s) of Entity/Table "Vote"

Physical Name	Name	Definition	Physical Data Type
vote_id	Vote Id	The automatically generated primary key of the vote table.	bigint
voting_timestamp	Voting Timestamp	Voting Timestamp	datetime
approver_id	Approver Id	A Foreign Key to the Application User table.	bigint
comments	Comments	Vote comments	nvarchar(max)
voting_state	Voting State	State of the vote	varchar(64)
voting_result_id	Voting Result Id	A Foreign Key to the Voting Result table.	bigint

#### Attribute/Column(s) of Entity/Table "Vote History"

Physical Name	Name	Definition	Physical Data Type
vote_history_id	Vote History Id	The automatically generated primary key of the vote_history table.	bigint
voting_timestamp	Voting Timestamp	History timestamp	datetime
comments	Comments	History comments	nvarchar(max)
voting_state	Voting State	History state	varchar(64)
vote_id	Vote Id	A Foreign Key to the Vote table. The source of the history entry.	bigint

#### Attribute/Column(s) of Entity/Table "Voting Result"

Physical Name	Name	Definition	Physical Data Type
voting_result_id	Voting Result Id	The automatically generated primary key of the voting_result table.	bigint
voting_state	Voting State	Overall Vote State.	varchar(64)
item_defn_id	Item Definition Id	A Foreign Key to the Definition table.	bigint
item_id	item_id	The value of a Voting Result stored as Long.	bigint
voting_metadata_id	Voting Metadata Id	A Foreign Key to the Definition table. Indicates type of voting.	bigint
voting_timestamp	Voting Timestamp	Time of the overall vote	datetime
item_id	Item Id	A Foreign Key to the Item table.	bigint

#### ER Diagram(s) of Subject Area "340 - Search"

Name	Physical Display Level
Logical	Column
Physical	Column

#### Entity/Table(s) of Subject Area "340 - Search"

Physical Name	Name	Logical Only	Do Not Generate	Definition
app_user	Application User		false	A user authorized to use the FootPrints system. The user is either a Customer User or an Agent User. The application user holds the minimum information needed to identify the user and associate the user with a user profile, where user settings are stored. A contact is not necessarily an application user. An application user entry is required in order for the user to access any online functions.
defn	Definition		false	A definition holds the structural and behavioral information needed for an application element that is configurable by the system designer and/or the end user at definition time. Current Footprints uses a combination of directories, files, tables, and programming code to represent this information. The new FootPrints will provide all of the same information within the definition. The definition can be persisted in a structure that is isolated at the Instance Level. The data for the definition will be stored as Configuration Settings

				associated with the Definition Version. See the database views v_defn_ver_active, v_cntrn_item_field.
defn_ver	Definition Version		false	A version of a Definition with a version identifier that changes when data changes are made to the template. The definition version will suffice to store the current state of the FootPrints Configuration. A full copy of the definition tree is versioned when a container is published. See the database views v_defn_ver_active, v_cntrn_item_field.
logical_data_type	Logical Data Type		false	A reference table for logical data type. The logical type maps to the database type in DDL generation.
search_col	Search Column		false	The number and order of search result columns returned when the search is executed.
search_cond	Search Condition		false	The set of logical search conditions for the query. This is rendered into the actual SQL or filtering code.
search_cstr_type	Search Constraint Type		false	A lookup table for the various operators that can be used in a search condition. These are logical operators.
search_item_query	Search Item Query		false	Identifies the specific item definitions associated with the search query to aid in visibility and permissions.
search_query	Search Query		false	The root table for saved searches. It names the search, and indicates the visibility in various parts of the application. The search is saved in an implementation neutral format so that it can be used in many different layers.
search_view	Search View		false	The UI details of the default view of the saved search.

#### Attribute/Column(s) of Entity/Table "Application User"

Physical Name	Name	Definition	Physical Data Type
pwd	Password	This space intentionally left blank.	varchar(255)
user_name	User Name	The name of the Application User.	nvarchar(4000)
cust_login_type_id	Customer Login Type Id	A Foreign Key to the Customer Login Type table.	int
app_user_id	Application User Id	The automatically generated primary key of the app_user table.	bigint
lic_type_id	License Type Id	A Foreign Key to the License Type table.	bigint
user_login	User Login	The login name used to access the Application. This may not be filled in if external application security is used.	nvarchar(4000)
email_addr	Email Address	An email address for the Application User.	nvarchar(4000)
delete_date	Delete Date	The value of a Application User, stored as a date.	datetime

#### Attribute/Column(s) of Entity/Table "Definition"

Physical Name	Name	Definition	Physical Data Type
defn_id	Definition Id	The automatically generated primary key of the defn table.	bigint
defn_name	Definition Name	The name of the Definition.	nvarchar(4000)
dsble_date	Disabled Date	The date that this instance of Definition is logically deleted and disabled.	datetime
defn_guid	Definition GUID	A Global Unique Identifier search key for the Definition, composed of letters and numbers. It is computed by the client and typically used in unstructured data.	varchar(50)
defn_type_name	Definition Type Name	The unique string identifier for an instance of Definition Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)

defn_subtype_id	Definition Subtype Id	A Foreign Key to the Definition Subtype table.	int
tmpl_export_guid	Template Export GUID	The value of a Definition, stored as a string.	VARCHAR(50)

#### Attribute/Column(s) of Entity/Table "Definition Version"

Physical Name	Name	Definition	Physical Data Type
defn_id	Definition Id	A Foreign Key to the Definition table.	bigint
defn_ver_id	Definition Version Id	The automatically generated primary key of the defn_ver table.	bigint
defn_ver_name	Definition Version Name	The name of the Definition Version.	nvarchar(4000)
defn_ver_status_id	Definition Version Status Id	A Foreign Key to the Definition Version Status table.	int
defn_ver_guid	Definition Version GUID	A Global Unique Identifier search key for the Definition Version, composed of letters and numbers. It is computed by the client and typically used in unstructured data	varchar(50)

#### Attribute/Column(s) of Entity/Table "Logical Data Type"

Physical Name	Name	Definition	Physical Data Type
logical_data_type_name	Logical Data Type Name	The name of the Logical Data Type.	varchar(255)
logical_data_type_id	Logical Data Type Id	The automatically generated primary key of the logical_data_type table.	int

#### Attribute/Column(s) of Entity/Table "Search Column"

Physical Name	Name	Definition	Physical Data Type
search_col_id	Search Column Id	The automatically generated primary key of the search_col table.	bigint
search_col_name	Search Column Name	The name of the Search Column.	varchar(255)
is_ascending_order	Is Ascending Order	Indicates if Is Ascending Order.	bit
search_view_id	Search View Id	A Foreign Key to the Search View table.	bigint
sequence	Sequence	The value of a Search Column stored as Long.	bigint
is_automatic	Is Automatic	Indicates if Is Automatic.	bit

#### Attribute/Column(s) of Entity/Table "Search Condition"

Physical Name	Name	Definition	Physical Data Type
search_cond_id	Search Condition Id	The automatically generated primary key of the search_cond table.	bigint
parent	Parent	A Foreign Key to the Search Condition table. Parent search.	bigint
search_cond_cd	Search Condition Code	A well-known string identifier used as a code to identify the Search Condition.	varchar(255)
field_name	Field Name	Field name.	varchar(255)
cstr_val	Constraint Value	Constraint value	nvarchar(4000)
search_item_query_id	Search Item Query Id	A Foreign Key to the Search Item Query table.	bigint
search_cstr_type_id	Search Constraint Type Id	A Foreign Key to the Search Constraint Type table.	int
sequence	Sequence	sequence integer	bigint
search_cstr_type	Search Constraint Type	Constraint type	varchar(255)
depth	Depth	tree depth	bigint
defn_id	Definition Id	A Foreign Key to the Definition table.	bigint
field_type	Field Type	A Foreign Key to the Logical Data Type table.	int
is_journal	Is Journal	Indicates if Is Journal.	bit
search_query_id	Search Query Id	A Foreign Key to the Search Query table.	bigint
predicate	Predicate	Search Predicate	varchar(255)
cond_name	Condition Name	Name of condition	varchar(255)

#### Attribute/Column(s) of Entity/Table "Search Constraint Type"

Physical Name	Name	Definition	Physical Data Type
search_cstr_type_id	Search Constraint Type Id	The automatically generated primary key of the search_cstr_type table.	int
search_cstr_type_name	Search Constraint Type Name	The name of the Search Constraint Type.	varchar(255)
search_cstr_type_descr	Search Constraint Type Description	A descriptive name of the Search Constraint Type. This descriptive name may be localized.	varchar(512)

#### Attribute/Column(s) of Entity/Table "Search Item Query"

Physical Name	Name	Definition	Physical Data Type
---------------	------	------------	--------------------

search_item_query_id	Search Item Query Id	The automatically generated primary key of the search_item_query table.	bigint
search_item_query_name	Search Item Query Name	The name of the Search Item Query.	varchar(255)
search_query_id	Search Query Id	A Foreign Key to the Search Query table.	bigint
extl_mapping_defn_id	External Mapping Definition Id	A Foreign Key to the Definition table. External data source.	bigint
meta_id	Meta Id	A Foreign Key to the Definition Version table.	bigint
defn_id	Definition Id	A Foreign Key to the Definition table.	bigint
schema_name	Schema Name	Database schema name	varchar(255)

#### Attribute/Column(s) of Entity/Table "Search Query"

Physical Name	Name	Definition	Physical Data Type
search_query_id	Search Query Id	The automatically generated primary key of the search_query table.	bigint
search_query_name	Search Query Name	The name of the Search Query.	varchar(255)
start_from_pos	Start From Position	Start From Position	bigint
limit_ct	Limit Count	query limit.	bigint
parent_name	Parent Name	Parent Name	varchar(255)
search_source	Search Source	Source	int
notes	Notes	User notes	varchar()
search_query_descr	Search Query Description	A descriptive name of the Search Query. This descriptive name may be localized.	varchar(1024)
tab	Tab	Tab name in UI	varchar(255)
query_type	Query Type	Advanced or Global or other type	varchar(255)
in_quick_searches	In Quick Searches	Indicates if In Quick Searches.	bit
in_search_analytics	In Search Analytics	Indicates if In Search Analytics.	bit
search_in_all	Search In All	Indicates if Search In All containers	bit
in_mobile	In Mobile	Indicates if In Mobile.	bit

#### Attribute/Column(s) of Entity/Table "Search View"

Physical Name	Name	Definition	Physical Data Type
search_view_id	Search View Id	The automatically generated primary key of the search_view table.	bigint
search_view_name	Search View Name	The name of the Search View.	varchar(255)
create_date	Create Date	Create Date	datetime
is_shared	Is Shared	Indicates if Is Shared.	bit
search_query_id	Search Query Id	A Foreign Key to the Search Query table.	bigint
create_user	Create User	A Foreign Key to the Application User table.	bigint
is_dflt	Is Default	Indicates if Is Default.	bit
show_in_menu	Show In Menu	Indicates if Show In Menu.	bit
search_view_descr	Search View Description	A descriptive name of the Search View. This descriptive name may be localized.	varchar(512)
cxt	Context	Search Context	varchar(255)
update_user	Update User	A Foreign Key to the Application User table.	bigint
update_date	Update Date	Update Date	datetime

#### ER Diagram(s) of Subject Area "350 - Chat"

Name	Physical Display Level
Logical	Column
Physical	Column

#### Entity/Table(s) of Subject Area "350 - Chat"

Physical Name	Name	Logical Only	Do Not Generate	Definition
alert_role_type	Alert Role Type		false	
app_user	Application User		false	A user authorized to use the FootPrints system. The user is either a Customer User or an Agent User. The application user holds the minimum information needed to identify the user and associate the user with a user profile, where user settings are stored. A contact is not necessarily an application user. An application user entry is required in order for the user to access any online functions.

broadcast_alert	Broadcast Alert		false	A broadcast alert message.
broadcast_msg	Broadcast Message		false	
broadcast_msg_type	Broadcast Message Type		false	
chat_channel	Chat Channel		false	A chat channel.
chat_channel_subscriber	Chat Channel Subscriber		false	A subscriber to a chat channel.
chat_msg	Chat Message		false	A message broadcast on a chat channel.
chat_status	Chat Status		false	A lookup table for chat status. Values include: AVAILABLE AWAY INVISIBLE_TO_CUSTOMERS INVISIBLE BUSY OFFLINE
chat_user_status	Chat User Status		false	The current chat status of an agent or customer application user.
defn	Definition		false	A definition holds the structural and behavioral information needed for an application element that is configurable by the system designer and/or the end user at definition time. Current Footprints uses a combination of directories, files, tables, and programming code to represent this information. The new FootPrints will provide all of the same information within the definition. The definition can be persisted in a structure that is isolated at the Instance Level. The data for the definition will be stored as Configuration Settings associated with the Definition Version. See the database views v_defn_ver_active, v_cntrn_item_field.
defn_ver	Definition Version		false	A version of a Definition with a version identifier that changes when data changes are made to the template. The definition version will suffice to store the current state of the FootPrints Configuration. A full copy of the definition tree is versioned when a container is published. See the database views v_defn_ver_active, v_cntrn_item_field.
item	Item	true	true	The Item is the abstract class for Issue, Configuration Item, Address Book entry, and other information items that are the main information entities in FootPrints. The Item structure and behavior is defined by its Definition. The Item is namespaced, identified by, and logically stored in the single parent container. It can be referenced in the same manner both inside and outside the container. The Item is the storage model and the implementation for the Item Definition. There are many important specializations of the Item class, and there even more Template definitions of Container/Item that provide an end-user application experience to the End-User. The Item implementation must implement flexible schema in an easy to manage, well engineered, and high performing way. The item can store user-defined and user-named fields defined in its definition. The Item could implement the nice-to-have features such as an inheritance model for Item subtypes, and have the ability to represent collections of subordinate items. The Item data can be imported, exported, or synchronized with external data sources. The Item data can be linked and synchronized with other items in the same Instance.

item_defn	Item Definition	true	true	The definition of an Item, which includes the Item Name and localizations and the collection of Item Fields. It also includes customizations and localizations of string resources for Item Names, and Item Field names.
ticket_alert	Ticket Alert		false	
wksp_cntnr	Workspace Container	true	true	A workspace is the principle architectural structure in Numara FootPrints for storing and tracking information. A Workspace is a separate sub-database or work area within the system with its own fields, forms, options, workflow, settings and users. Most of the behaviour of the Workspace comes from the Container Abstract class, and only things that are specific to the Workspace are implemented here. For Example, Current FootPrints allows a single CMDB to be referenced from a Workspace, and a single Address book, yet these are listed as features for improvement. So convenience methods that get and set Container Group Relationship might be appropriate.

#### Attribute/Column(s) of Entity/Table "Alert Role Type"

Physical Name	Name	Definition	Physical Data Type
alert_role_type_id	Alert Role Type Id	The unique identifier for an instance of Alert Role Type. This identifier must be kept in sync with programming code that uses an enumeration or fixed constant to identify particular values.	integer
alert_role_type_name	Alert Role Type Name	The unique string identifier for an instance of Alert Role Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)

#### Attribute/Column(s) of Entity/Table "Application User"

Physical Name	Name	Definition	Physical Data Type
pwd	Password	This space intentionally left blank.	varchar(255)
user_name	User Name	The name of the Application User.	varchar(4000)
cust_login_type_id	Customer Login Type Id	A Foreign Key to the Customer Login Type table.	int
app_user_id	Application User Id	The automatically generated primary key of the app_user table.	bigint
lic_type_id	License Type Id	A Foreign Key to the License Type table.	bigint
user_login	User Login	The login name used to access the Application. This may not be filled in if external application security is used.	nvarchar(4000)
email_addr	Email Address	An email address for the Application User.	nvarchar(4000)
delete_date	Delete Date	The value of a Application User, stored as a date.	datetime

#### Attribute/Column(s) of Entity/Table "Broadcast Alert"

Physical Name	Name	Definition	Physical Data Type
broadcast_alert_id	Broadcast Alert Id	The automatically generated primary key of the broadcast_alert table.	bigint
recipient_id	Recipient Id	A Foreign Key to the Application User table.	bigint
ticket_alert_id	Ticket Alert Id	A Foreign Key to the Ticket Alert table.	bigint

#### Attribute/Column(s) of Entity/Table "Broadcast Message"

Physical Name	Name	Definition	Physical Data Type
broadcast_msg_id	Broadcast Message Id	The automatically generated primary key of the broadcast_message table.	bigint
broadcast_msg_type_id	Broadcast Message Type Id	The unique identifier for an instance of Broadcast Message Type. This identifier must be kept in sync with programming code that uses an enumeration or fixed constant to identify particular values.	integer

msg_tmpl_defn_id	Message Template Definition Id	A Foreign Key to the Definition table.	bigint
msg_text	Message Text	The value of a Broadcast Message, stored as a long text object.	nvarchar(max)

#### Attribute/Column(s) of Entity/Table "Broadcast Message Type"

Physical Name	Name	Definition	Physical Data Type
broadcast_msg_type_id	Broadcast Message Type Id	The unique identifier for an instance of Broadcast Message Type. This identifier must be kept in sync with programming code that uses an enumeration or fixed constant to identify particular values.	integer
broadcast_msg_type_name	Broadcast Message Type Name	The unique string identifier for an instance of Broadcast Message Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)

#### Attribute/Column(s) of Entity/Table "Chat Channel"

Physical Name	Name	Definition	Physical Data Type
chat_channel_id	Chat Channel Id	The automatically generated primary key of the chat_channel table.	bigint
chat_channel_ctgy	Chat Channel Category	Category of chat.	varchar(50)
chat_channel_subctgy	Chat Channel Subcategory	Subcategory of chat	varchar(50)

#### Attribute/Column(s) of Entity/Table "Chat Channel Subscriber"

Physical Name	Name	Definition	Physical Data Type
chat_channel_subscriber_id	Chat Channel Subscriber Id	The automatically generated primary key of the chat_channel_subscriber table.	bigint
chat_channel_id	Chat Channel Id	A Foreign Key to the Chat Channel table.	bigint
user_id	User Id	A Foreign Key to the Application User table.	bigint
last_time_queried	Last Time Queried	The value of a Chat Channel Subscriber, stored as a date.	datetime

#### Attribute/Column(s) of Entity/Table "Chat Message"

Physical Name	Name	Definition	Physical Data Type
chat_msg_id	Chat Message Id	The automatically generated primary key of the chat_msg table.	bigint
arrive_date	Arrive Date	The value of a Chat Message, stored as a date.	datetime
msg	Message	The value of a Chat Message, stored as a long text object.	nvarchar(max)
chat_channel_id	Chat Channel Id	A Foreign Key to the Chat Channel table.	bigint
user_id_from	User Id From	A Foreign Key to the Application User table.	bigint

#### Attribute/Column(s) of Entity/Table "Chat Status"

Physical Name	Name	Definition	Physical Data Type
chat_status_id	Chat Status Id	The automatically generated primary key of the chat_status table.	int
chat_status_name	Chat Status Name	The name of the Chat Status.	varchar(255)
chat_status_descr	Chat Status Description	A descriptive name of the Chat Status. This descriptive name may be localized.	varchar(512)

#### Attribute/Column(s) of Entity/Table "Chat User Status"

Physical Name	Name	Definition	Physical Data Type
chat_user_status_id	Chat User Status Id	The automatically generated primary key of the chat_user_status table.	bigint
app_user_id	Application User Id	A Foreign Key to the Application User table.	bigint
chat_status_id	Chat Status Id	A Foreign Key to the Chat Status table.	int

#### Attribute/Column(s) of Entity/Table "Definition"

Physical Name	Name	Definition	Physical Data Type
defn_id	Definition Id	The automatically generated primary key of the defn table.	bigint



defn_name	Definition Name	The name of the Definition.	nvarchar(4000)
dsble_date	Disabled Date	The date that this instance of Definition is logically deleted and disabled.	datetime
defn_guid	Definition GUID	A Global Unique Identifier search key for the Definition, composed of letters and numbers. It is computed by the client and typically used in unstructured data.	varchar(50)
defn_type_name	Definition Type Name	The unique string identifier for an instance of Definition Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)
defn_subtype_id	Definition Subtype Id	A Foreign Key to the Definition Subtype table.	int
tmpl_export_guid	Template Export GUID	The value of a Definition, stored as a string.	VARCHAR(50)

#### Attribute/Column(s) of Entity/Table "Definition Version"

Physical Name	Name	Definition	Physical Data Type
defn_id	Definition Id	A Foreign Key to the Definition table.	bigint
defn_ver_id	Definition Version Id	The automatically generated primary key of the defn_ver table.	bigint
defn_ver_name	Definition Version Name	The name of the Definition Version.	nvarchar(4000)
defn_ver_status_id	Definition Version Status Id	A Foreign Key to the Definition Version Status table.	int
defn_ver_guid	Definition Version GUID	A Global Unique Identifier search key for the Definition Version, composed of letters and numbers. It is computed by the client and typically used in unstructured data.	varchar(50)

#### Attribute/Column(s) of Entity/Table "Item"

Physical Name	Name	Definition	Physical Data Type
item_key	Item Key	A search key for the Item, composed of letters and numbers.	varchar(255)
item_id	Item Id	The automatically generated primary key of the item table.	bigint
cntrn_id	Container Id	A Foreign Key to the Container table.	bigint
item_defn_id	Item Definition Id	A Foreign Key to the Definition table for an instance of type Definition.	bigint

#### Attribute/Column(s) of Entity/Table "Item Definition"

Physical Name	Name	Definition	Physical Data Type
item_table_name	Item Table Name	The name of the DBMS_Object_Name. It is a database object name, limited to 30 characters in length.	varchar(30)
item_defn_id	Item Definition Id	A Foreign Key to the Definition table for an instance of type Definition.	bigint

#### Attribute/Column(s) of Entity/Table "Ticket Alert"

Physical Name	Name	Definition	Physical Data Type
ticket_alert_id	Ticket Alert Id	The automatically generated primary key of the ticket_alert table.	bigint
alert_role_type_id	Alert Role Type Id	The unique identifier for an instance of Alert Role Type. This identifier must be kept in sync with programming code that uses an enumeration or fixed constant to identify particular values.	integer
defn_ver_id	Definition Version Id	A Foreign Key to the Definition Version table.	bigint
item_id	Item Id	A Foreign Key to the Item table.	bigint
item_defn_id	Item Definition Id	A Foreign Key to the Definition table for an instance of type Definition.	bigint
cntrn_id	Container Id	A Foreign Key to the Container table.	bigint

#### Attribute/Column(s) of Entity/Table "Workspace Container"

Physical Name	Name	Definition	Physical Data Type
time_zone_region_name	Time Zone Region Name	The name of the Time Zone Region. This is 2. The 'primary key' of the java timezone is the string region identifier in English, example 'America/Phoenix'. There is no numeric identifier. As the JRE is updated, or specific classes are updated, the list could change to include new entries, and the detail data behind the regions could change.	varchar(255)

cntrnr_type_id	Container Type Id	A Foreign Key to the Container Type table.	int
cntrnr_id	Container Id	A Foreign Key to the Container table.	bigint

#### ER Diagram(s) of Subject Area "360 - Survey"

Name	Physical Display Level
Logical	Column
Physical	Column

#### Entity/Table(s) of Subject Area "360 - Survey"

Physical Name	Name	Logical Only	Do Not Generate	Definition
defn_ver	Definition Version		false	A version of a Definition with a version identifier that changes when data changes are made to the template. The definition version will suffice to store the current state of the FootPrints Configuration. A full copy of the definition tree is versioned when a container is published. See the database views v_defn_ver_active, v_cntrnr_item_field.
survey_sent	Survey Sent		false	A record of a survey sent to an application user.

#### Attribute/Column(s) of Entity/Table "Definition Version"

Physical Name	Name	Definition	Physical Data Type
defn_id	Definition Id	A Foreign Key to the Definition table.	bigint
defn_ver_id	Definition Version Id	The automatically generated primary key of the defn_ver table.	bigint
defn_ver_name	Definition Version Name	The name of the Definition Version.	nvarchar(4000)
defn_ver_status_id	Definition Version Status Id	A Foreign Key to the Definition Version Status table.	int
defn_ver_guid	Definition Version GUID	A Global Unique Identifier search key for the Definition Version, composed of letters and numbers. It is computed by the client and typically used in unstructured data.	varchar(50)

#### Attribute/Column(s) of Entity/Table "Survey Sent"

Physical Name	Name	Definition	Physical Data Type
survey_sent_id	Survey Sent Id	The automatically generated primary key of the survey_sent table.	bigint
survey_sent_email	Survey Sent Email	Email address of recipient.	varchar(50)
date_sent	Date Sent	Date sent.	datetime
item_defn_id	Item Definition Id	A Foreign Key to the Definition Version table.	bigint

#### ER Diagram(s) of Subject Area "370 - Deployment"

Name	Physical Display Level
Logical	Column
Physical	Column

#### Entity/Table(s) of Subject Area "370 - Deployment"

Physical Name	Name	Logical Only	Do Not Generate	Definition
deployment_update	Deployment Update		false	A log of updates to the database structure performed by the installation deployment. This log ensures that the correct scripts are run in order, and prevents the re-running of scripts.

#### Attribute/Column(s) of Entity/Table "Deployment Update"

Physical Name	Name	Definition	Physical Data Type
start_datetime	Start Datetime	Start of deployment	datetime
finish_datetime	Finish Datetime	End of deployment	datetime
update_pack_id	Update Pack Id	The value of a Deployment Update, stored as a string.	varchar(64)
from_ver	From Version	From FP version	varchar(64)
to_ver	To Version	To FP version	varchar(64)
success	Success	Indicates if Success.	bit
error	Error	Error Messages	varchar(4000)
build_number	Build Number	Build number	varchar(64)

prod_ver	Product Version	Product Version	varchar(64)
----------	-----------------	-----------------	-------------

### ER Diagram(s) of Subject Area "380 - Portal"

Name	Physical Display Level
Logical	Column
Physical	Column

### Entity/Table(s) of Subject Area "380 - Portal"

Physical Name	Name	Logical Only	Do Not Generate	Definition
app_user	Application User		false	A user authorized to use the FootPrints system. The user is either a Customer User or an Agent User. The application user holds the minimum information needed to identify the user and associate the user with a user profile, where user settings are stored. A contact is not necessarily an application user. An application user entry is required in order for the user to access any online functions.
console	Console		false	A console in the FootPrints 12 User Interface. A console can contain other consoles, for example the Home Page.
portal	Portal		false	A portal in the FootPrints 12 User Interface.
portal_console	Portal Console		false	The association between a Console and a Portal.
widget	Widget		false	A widget. A root widget contains subordinate widgets. Widget Types include: ACTION_PANE ADMINISTRATION ALL_ITEMS CALENDAR CHART NEW_ITEM ROOT SAVED_SEARCH_WIDGET TAB_PANE URL WIDGET_PANE

### Attribute/Column(s) of Entity/Table "Application User"

Physical Name	Name	Definition	Physical Data Type
pwd	Password	This space intentionally left blank.	varchar(255)
user_name	User Name	The name of the Application User.	nvarchar(4000)
cust_login_type_id	Customer Login Type Id	A Foreign Key to the Customer Login Type table.	int
app_user_id	Application User Id	The automatically generated primary key of the app_user table.	bigint
lic_type_id	License Type Id	A Foreign Key to the License Type table.	bigint
user_login	User Login	The login name used to access the Application. This may not be filled in if external application security is used.	nvarchar(4000)
email_addr	Email Address	An email address for the Application User.	nvarchar(4000)
delete_date	Delete Date	The value of a Application User, stored as a date.	datetime

### Attribute/Column(s) of Entity/Table "Console"

Physical Name	Name	Definition	Physical Data Type
console_id	Console Id	The automatically generated primary key of the console table.	bigint
source_console_id	Source Console Id	A Foreign Key to the Console table.	bigint
console_guid	Console GUID	Identifier used from Javascript.	varchar(255)
console_name	Console Name	Console name.	nvarchar(4000)
widget_id	Widget Id	A Foreign Key to the Widget table.	bigint
cntr_type	Container Type	The name of the container type, such as workspace, cmdb etc.	varchar(50)
sys_name	System Name	internal identifier of the console	varchar(255)
console_descr	Console Description	Console Description.	nvarchar(4000)
user_id	User Id	A Foreign Key to the Application User table.	bigint

### Attribute/Column(s) of Entity/Table "Portal"

Physical Name	Name	Definition	Physical Data Type
---------------	------	------------	--------------------

portal_id	Portal Id	The automatically generated primary key of the portal table.	bigint
portal_guid	Portal GUID	The value of a Portal, stored as a string.	varchar(255)
portal_name	Portal Name	Name of the portal.	nvarchar(4000)
portal_theme	Portal Theme	Theme name to be used in Portal rendering.	varchar(255)
portal_logo_guid	Portal Logo GUID	Logo to be displayed	varchar(255)
portal_active_console_id	Portal Active Console Id	A Foreign Key to the Console table.	bigint
is_sys	Is System	Indicates if Is System or user custom portal.	bit
portal_descr	Portal Description	Portal Description.	nvarchar(4000)
allow_anonymous_access	Allow Anonymous Access	Indicates if Allow Anonymous Access.	bit

#### Attribute/Column(s) of Entity/Table "Portal Console"

Physical Name	Name	Definition	Physical Data Type
console_id	Console Id	A Foreign Key to the Console table.	bigint
portal_id	Portal Id	A Foreign Key to the Portal table.	bigint
is_vsbble	Is Visible	Indicates if Is Visible.	bit
console_index	Console Index	The order of the console in the portal.	int
is_closable	Is Closable	Indicates if Is Closable.	bit

#### Attribute/Column(s) of Entity/Table "Widget"

Physical Name	Name	Definition	Physical Data Type
widget_id	Widget Id	The automatically generated primary key of the widget table.	bigint
parent_widget_id	Parent Widget Id	A Foreign Key to the Widget table.	bigint
widget_guid	Widget GUID	The value of a Widget, stored as a string.	varchar(255)
widget_descr	Widget Description	The value of a Widget, stored as a long text object.	varchar(max)
widget_layout	Widget Layout	The value of a Widget, stored as a long text object.	varchar(max)
widget_type	Widget Type	The value of a Widget, stored as a string.	varchar(255)
widget_cfg	Widget Configuration	The value of a Widget, stored as a long text object.	varchar(max)
widget_view_cfg	Widget View Configuration	The value of a Widget, stored as a long text object.	varchar(max)
widget_index	Widget Index	The value of a Widget stored as boolean.	int

#### ER Diagram(s) of Subject Area "390 - Business Process Template"

Name	Physical Display Level
Logical	Column
Physical	Column

#### Entity/Table(s) of Subject Area "390 - Business Process Template"

Physical Name	Name	Logical Only	Do Not Generate	Definition
bus_proc	Business Process		false	A business process is a set of containers, items, business rules, etc. created from a business process template.
bus_proc_defn	Business Process Definition		false	The collection of definitions associated with a business process that was created as a result of using a business process template to create a business process.
defn	Definition		false	A definition holds the structural and behavioral information needed for an application element that is configurable by the system designer and/or the end user at definition time. Current Footprints uses a combination of directories, files, tables, and programming code to represent this information. The new FootPrints will provide all of the same information within the definition. The definition can be persisted in a structure that is isolated at the Instance Level. The data for the definition will be stored as Configuration Settings associated with the Definition Version. See the database views v_defn_ver_active, v_cntrn_item_field.

defn_subtype	Definition Subtype		false	The subtype of the definition used as a discriminator for the class of definition. Each subtype will have different configuration settings and associated definitions. These include associated built-in field definitions for item subtypes. See the database view <code>v_defn_subtype</code> .
tmpl_package	Template Package		false	A template specification that can be used to create an Item, Container, or Group of Containers.
tmpl_package_type	Template Package Type		false	A lookup table for the Template Package Type. Values include: BUSINESS_PROCESS, CONTAINER, ITEM

#### Attribute/Column(s) of Entity/Table "Business Process"

Physical Name	Name	Definition	Physical Data Type
bus_proc_id	Business Process Id	The automatically generated primary key of the bus_proc table.	bigint
bus_proc_name	Business Process Name	The name of the Business Process.	nvarchar(4000)
bus_proc_descr	Business Process Description	A descriptive name of the Business Process. This descriptive name may be localized.	nvarchar(4000)
bus_proc_tmpl_id	Business Process Template Id	A Foreign Key to the Template Package table.	bigint

#### Attribute/Column(s) of Entity/Table "Business Process Definition"

Physical Name	Name	Definition	Physical Data Type
bus_proc_defn_id	Business Process Definition Id	The automatically generated primary key of the bus_proc_defn table.	bigint
bus_proc_id	Business Process Id	A Foreign Key to the Business Process table.	bigint
defn_id	Definition Id	A Foreign Key to the Definition table.	bigint

#### Attribute/Column(s) of Entity/Table "Definition"

Physical Name	Name	Definition	Physical Data Type
defn_id	Definition Id	The automatically generated primary key of the defn table.	bigint
defn_name	Definition Name	The name of the Definition.	nvarchar(4000)
dsble_date	Disabled Date	The date that this instance of Definition is logically deleted and disabled.	datetime
defn_guid	Definition GUID	A Global Unique IDentifier search key for the Definition, composed of letters and numbers. It is computed by the client and typically used in unstructured data.	varchar(50)
defn_type_name	Definition Type Name	The unique string identifier for an instance of Definition Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)
defn_subtype_id	Definition Subtype Id	A Foreign Key to the Definition Subtype table.	int
tmpl_export_guid	Template Export GUID	The value of a Definition, stored as a string.	VARCHAR(50)

#### Attribute/Column(s) of Entity/Table "Definition Subtype"

Physical Name	Name	Definition	Physical Data Type
defn_subtype_name	Definition Subtype Name	The name of the Definition Subtype.	varchar(255)
defn_subtype_descr	Definition Subtype Description	A descriptive name of the Definition Subtype. This descriptive name may be localized.	varchar(512)
defn_subtype_id	Definition Subtype Id	The automatically generated primary key of the defn_subtype table.	int
defn_parent_subtype_id	definition parent subtype id	A Foreign Key to the Definition Subtype table.	int
defn_type_name	Definition Type Name	The unique string identifier for an instance of Definition Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)

max_in_parent	Maximum In Parent	A Foreign Key to the Definition Subtype table.	int
---------------	-------------------	--	-----

#### Attribute/Column(s) of Entity/Table "Template Package"

Physical Name	Name	Definition	Physical Data Type
tmpl_package_id	Template Package Id	The automatically generated primary key of the tmpl_package table.	bigint
tmpl_package_content	Template Package Content	The value of a Template Package, stored as a long text object.	varchar(max)
tmpl_package_name	Template Package Name	The name of the Template Package.	varchar(255)
tmpl_package_type_id	Template Package Type Id	A Foreign Key to the Template Package Type table.	int
defn_subtype_id	Definition Subtype Id	A Foreign Key to the Definition Subtype table.	int
is_sys_tmpl	Is System Template	Indicates if Is System Template.	bit
tmpl_package_intrnl_name	Template Package Internal Name	The value of a Template Package, stored as a string.	varchar(255)

#### Attribute/Column(s) of Entity/Table "Template Package Type"

Physical Name	Name	Definition	Physical Data Type
tmpl_package_type_id	Template Package Type Id	The automatically generated primary key of the tmpl_package_type table.	int
tmpl_package_type_name	Template Package Type Name	The name of the Template Package Type.	varchar(255)

#### ER Diagram(s) of Subject Area "400 - Asset Management"

Name	Physical Display Level
Logical	Column
Physical	Column

#### Entity/Table(s) of Subject Area "400 - Asset Management"

Physical Name	Name	Logical Only	Do Not Generate	Definition
defn	Definition		false	A definition holds the structural and behavioral information needed for an application element that is configurable by the system designer and/or the end user at definition time. Current Footprints uses a combination of directories, files, tables, and programming code to represent this information. The new FootPrints will provide all of the same information within the definition. The definition can be persisted in a structure that is isolated at the Instance Level. The data for the definition will be stored as Configuration Settings associated with the Definition Version. See the database views v_defn_ver_active, v_cntrn_item_field.
item	Item	true	true	The Item is the abstract class for Issue, Configuration Item, Address Book entry, and other information items that are the main information entities in FootPrints. The Item structure and behavior is defined by its Definition. The Item is namespaced, identified by, and logically stored in the single parent container. It can be referenced in the same manner both inside and outside the container. The Item is the storage model and the implementation for the Item Definition. There are many important specializations of the Item class, and there even more Template definitions of Container/Item that provide an end-user application experience to the End-User. The Item implementation must implement flexible schema in an easy to manage, well engineered, and high performing way. The item can store user-defined and user-named fields defined in its definition. The Item could implement the nice-to-have features such as an inheritance model for Item subtypes, and have the ability to represent collections of subordinate items. The Item data can be imported, exported, or synchronized with external data sources. The Item data can be linked and

				synchronized with other items in the same Instance.
pend_advsn_actn	Pending Advanced Action		false	An advanced action to be taken by integration with Asset Core.

#### Attribute/Column(s) of Entity/Table "Definition"

Physical Name	Name	Definition	Physical Data Type
defn_id	Definition Id	The automatically generated primary key of the defn table.	bigint
defn_name	Definition Name	The name of the Definition.	nvarchar(4000)
dsble_date	Disabled Date	The date that this instance of Definition is logically deleted and disabled.	datetime
defn_guid	Definition GUID	A Global Unique Identifier search key for the Definition, composed of letters and numbers. It is computed by the client and typically used in unstructured data.	varchar(50)
defn_type_name	Definition Type Name	The unique string identifier for an instance of Definition Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)
defn_subtype_id	Definition Subtype Id	A Foreign Key to the Definition Subtype table.	int
tmpl_export_guid	Template Export GUID	The value of a Definition, stored as a string.	VARCHAR(50)

#### Attribute/Column(s) of Entity/Table "Item"

Physical Name	Name	Definition	Physical Data Type
item_key	Item Key	A search key for the Item, composed of letters and numbers.	varchar(255)
item_id	Item Id	The automatically generated primary key of the item table.	bigint
cntrn_id	Container Id	A Foreign Key to the Container table.	bigint
item_defn_id	Item Definition Id	A Foreign Key to the Definition table for an instance of type Definition.	bigint

#### Attribute/Column(s) of Entity/Table "Pending Advanced Action"

Physical Name	Name	Definition	Physical Data Type
pend_advsn_actn_id	Pending Advanced Action Id	The automatically generated primary key of the pend_advsn_actn table.	bigint
pend_advsn_actn_advsn_actn_id	Pending Advanced Action Advanced Action Id	A Foreign Key to the Pending Advanced Action table.	bigint
pend_advsn_actn_asset_id	Pending Advanced Action Asset Id	A Foreign Key to the Pending Advanced Action table.	bigint
pend_advsn_actn_dyn_item_id	Pending Advanced Action Dynamic Item Id	A Foreign Key to the Item table.	bigint
pend_advsn_actn_defn_id	Pending Advanced Action Definition Id	A Foreign Key to the Definition table.	bigint

#### ER Diagram(s) of Subject Area "410 - Publishing"

Name	Physical Display Level
Logical	Column
Physical	Column

#### Entity/Table(s) of Subject Area "410 - Publishing"

Physical Name	Name	Logical Only	Do Not Generate	Definition
app_user	Application User		false	A user authorized to use the FootPrints system. The user is either a Customer User or an Agent User. The application user holds the minimum information needed to identify the user and associate the user with a user profile, where user settings are stored. A contact is not necessarily an application user. An application user entry is required in order for the user to access any online functions.

defn	Definition		false	A definition holds the structural and behavioral information needed for an application element that is configurable by the system designer and/or the end user at definition time. Current Footprints uses a combination of directories, files, tables, and programming code to represent this information. The new FootPrints will provide all of the same information within the definition. The definition can be persisted in a structure that is isolated at the Instance Level. The data for the definition will be stored as Configuration Settings associated with the Definition Version. See the database views v_defn_ver_active, v_cntr_item_field.
item	Item	true	true	The Item is the abstract class for Issue, Configuration Item, Address Book entry, and other information items that are the main information entities in FootPrints. The Item structure and behavior is defined by its Definition. The Item is namespaced, identified by, and logically stored in the single parent container. It can be referenced in the same manner both inside and outside the container. The Item is the storage model and the implementation for the Item Definition. There are many important specializations of the Item class, and there even more Template definitions of Container/Item that provide an end-user application experience to the End-User. The Item implementation must implement flexible schema in an easy to manage, well engineered, and high performing way. The item can store user-defined and user-named fields defined in its definition. The Item could implement the nice-to-have features such as an inheritance model for Item subtypes, and have the ability to represent collections of subordinate items. The Item data can be imported, exported, or synchronized with external data sources. The Item data can be linked and synchronized with other items in the same Instance.
publ_error	Publish Error		false	An error that occurred during publishing.
publ_log	Publish Log		false	A log of container and item publish actions. The definitions are used to create schema, table, view, stored procedure, and index objects in the relational database in order to store dynamically defined information.

#### Attribute/Column(s) of Entity/Table "Application User"

Physical Name	Name	Definition	Physical Data Type
pwd	Password	This space intentionally left blank.	varchar(255)
user_name	User Name	The name of the Application User.	nvarchar(4000)
cust_login_type_id	Customer Login Type Id	A Foreign Key to the Customer Login Type table.	int
app_user_id	Application User Id	The automatically generated primary key of the app_user table.	bigint
lic_type_id	License Type Id	A Foreign Key to the License Type table.	bigint
user_login	User Login	The login name used to access the Application. This may not be filled in if external application security is used.	nvarchar(4000)
email_addr	Email Address	An email address for the Application User.	nvarchar(4000)
delete_date	Delete Date	The value of a Application User, stored as a date.	datetime

#### Attribute/Column(s) of Entity/Table "Definition"

Physical Name	Name	Definition	Physical Data Type
defn_id	Definition Id	The automatically generated primary key of the defn table.	bigint



defn_name	Definition Name	The name of the Definition.	nvarchar(4000)
dsble_date	Disabled Date	The date that this instance of Definition is logically deleted and disabled.	datetime
defn_guid	Definition GUID	A Global Unique Identifier search key for the Definition, composed of letters and numbers. It is computed by the client and typically used in unstructured data.	varchar(50)
defn_type_name	Definition Type Name	The unique string identifier for an instance of Definition Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)
defn_subtype_id	Definition Subtype Id	A Foreign Key to the Definition Subtype table.	int
tmpl_export_guid	Template Export GUID	The value of a Definition, stored as a string.	VARCHAR(50)

#### Attribute/Column(s) of Entity/Table "Item"

Physical Name	Name	Definition	Physical Data Type
item_key	Item Key	A search key for the Item, composed of letters and numbers.	varchar(255)
item_id	Item Id	The automatically generated primary key of the item table.	bigint
cntnr_id	Container Id	A Foreign Key to the Container table.	bigint
item_defn_id	Item Definition Id	A Foreign Key to the Definition table for an instance of type Definition.	bigint

#### Attribute/Column(s) of Entity/Table "Publish Error"

Physical Name	Name	Definition	Physical Data Type
publ_error_id	Publish Error Id	The automatically generated primary key of the publ_error table.	bigint
publ_log_id	Publish Log Id	A Foreign Key to the Publish Log table.	bigint
error_type_name	Error Type Name	Error Type Name.	varchar(10)
location_name	Location Name	Location Name.	nvarchar(max)
location_type_name	Location Type Name	Location Type Name.	varchar(255)
location_id	Location Identifier	Location Identifier	varchar(36)
item_name	Item Name	The name of the item being published.	nvarchar(max)
msg	Message	Error message.	nvarchar(max)
item_id	item_id	The value of a Publish Error stored as Long.	bigint
item_id	Item Id	A Foreign Key to the Item table.	bigint

#### Attribute/Column(s) of Entity/Table "Publish Log"

Physical Name	Name	Definition	Physical Data Type
publ_log_id	Publish Log Id	The automatically generated primary key of the publ_log table.	bigint
cntnr_defn_id	Container Definition Id	A Foreign Key to the Definition table. Container being published.	bigint
publ_date	Publish Date	Log date.	datetime
author_name	Author Name	Publish admin user.	nvarchar(4000)
status	Status	Published status.	bit
error_ct	Error Count	Count of errors.	smallint
warn_ct	Warning Count	Count of warnings.	smallint
user_id	user_id	A Foreign Key to the Application User table. Identifies the user that performed the publish.	bigint

#### ER Diagram(s) of Subject Area "430 - UI Component State"

Name	Physical Display Level
Logical	Column
Physical	Column

#### Entity/Table(s) of Subject Area "430 - UI Component State"

Physical Name	Name	Logical Only	Do Not Generate	Definition
app_user	Application User		false	A user authorized to use the FootPrints system. The user is either a Customer User or an Agent User. The application user holds the minimum information needed to identify the user and associate the user with a user profile, where user settings are stored. A contact is not necessarily an application user. An application user entry is required in order for the user

				to access any online functions.
ui_component_state	UI Component State		false	The EXTJS grid has the ability to store it's state allowing the preservation of grid layout properties. This table supports that. It might also be used for other UI control objects.

#### Attribute/Column(s) of Entity/Table "Application User"

Physical Name	Name	Definition	Physical Data Type
pwd	Password	This space intentionally left blank.	varchar(255)
user_name	User Name	The name of the Application User.	nvarchar(4000)
cust_login_type_id	Customer Login Type Id	A Foreign Key to the Customer Login Type table.	int
app_user_id	Application User Id	The automatically generated primary key of the app_user table.	bigint
lic_type_id	License Type Id	A Foreign Key to the License Type table.	bigint
user_login	User Login	The login name used to access the Application. This may not be filled in if external application security is used.	nvarchar(4000)
email_addr	Email Address	An email address for the Application User.	nvarchar(4000)
delete_date	Delete Date	The value of a Application User, stored as a date.	datetime

#### Attribute/Column(s) of Entity/Table "UI Component State"

Physical Name	Name	Definition	Physical Data Type
ui_component_state_id	UI Component State Id	The automatically generated primary key of the ui_component_state table.	bigint
state_key	State Key	A search key for the UI Component State, composed of letters and numbers. It must be unique when combined with the app user id. It is an intelligent key, but the only use is to store and retrieve values.	varchar(255)
state_val	State Value	The value of a UI Component State, stored as a long text object.	varchar(max)
app_user_id	Application User Id	A Foreign Key to the Application User table. Identifies the user that owns the layout.	bigint

#### ER Diagram(s) of Subject Area "900 To Be Deleted"

Name	Physical Display Level
Physical	Column

#### Entity/Table(s) of Subject Area "900 To Be Deleted"

Physical Name	Name	Logical Only	Do Not Generate	Definition
cal_ticket_privilege	Calendar Ticket Privilege		false	(Not used, but referenced in Java code that must be removed before table deletion.) A permission granted via a calendar ticket. May not be used in the FootPrints integration of calendaring code.
local_col	Localized Column		false	(Not used, but referenced in Java code that must be removed before table deletion.) (Not currently used. Data localization is implemented in Configuration Setting Value. May be retired in the future). A Database Column name used in the Transparent Database Localization pattern. The data in the column is the original data for which a translated value can exist. In the case of English, this is a little strange because the English translation can be different than the original data.

local_col_val	Localized Column Value		false	(Not used, but referenced in Java code that must be removed before table deletion.) (Not currently used. Data localization is implemented in Configuration Setting Value. May be retired in the future). A data value from a column and row in a Localized table. The initial english value is obtained from the table, but may be modified for visual display purposes. The Primary Key identifier of the source table is used along with the table name to establish the row identity.
role_grp	Role Group		false	(Not used, but referenced in Java code that must be removed before table deletion.) The role group will contain the roles that the user has for each of the workspaces that the user has access. The user will always have a role group even if the administrator chooses the Perl Footprints functionality.
role_grp_role	Role Group Role		false	(Not used, but referenced in Java code that must be removed before table deletion.) an association table that resolves the many to many relationship between the Role Group and the Role entities.
vt\$item_fid_defn_networkadapte	vt\$item_fid_defn_NetworkAdapter		false	
vt\$item_fid_networkadapter\$chg	vt\$item_fid_NetworkAdapter\$-changes		false	

#### Attribute/Column(s) of Entity/Table "Calendar Ticket Privilege"

Physical Name	Name	Definition	Physical Data Type
cal_ticket_id	Calendar Ticket Id	A Foreign Key to the Calendar Ticket table.	bigint
privilege	Privilege	The value of a Calendar Ticket Privilege, stored as a string.	varchar(255)

#### Attribute/Column(s) of Entity/Table "Localized Column"

Physical Name	Name	Definition	Physical Data Type
app_ele_id	Application Element Id	A Foreign Key to the Application Element table.	bigint
schema_name	Schema Name	The name of the Schema containing the Localized table. In Internal systems and in Definition level systems, this will be the logical name "FP", "FP Workspace", but in the Runtime system, this will be "FP 01", "FP 01 Workspace 001" for example.	varchar(30)
table_name	Table Name	The name of the Localized Table.	varchar(30)
col_name	Column Name	The name of the DBMS_Object_Name. It is a database object name, limited to 30 characters in length.	varchar(30)

#### Attribute/Column(s) of Entity/Table "Localized Column Value"

Physical Name	Name	Definition	Physical Data Type
app_ele_id	Application Element Id	A Foreign Key to the Application Element table.	bigint
local_col_id	Localized Column Id	A Foreign Key to the Application Element table.	bigint
init_col_string_val	Initial Column String Value	Initial Column String Value.	varchar(255)
prmry_key_id	Primary Key Id	The integer primary key value of the row in the named Table that is being localized.	bigint

#### Attribute/Column(s) of Entity/Table "Role Group"

Physical Name	Name	Definition	Physical Data Type
role_grp_id	Role Group Id	The automatically generated primary key of the role_grp table.	bigint
role_grp_name	Role Group Name	The name of the Role Group.	varchar(255)
role_grp_descr	Role Group Description	A descriptive name of the Role Group. This descriptive name may be localized.	varchar(512)

#### Attribute/Column(s) of Entity/Table "Role Group Role"

Physical Name	Name	Definition	Physical Data Type
app_role_id	Application Role Id	A Foreign Key to the Application Role table.	bigint

role_grp_role_id	Role Group Role Id	The automatically generated primary key of the role_grp_role table.	bigint
role_grp_id	Role Group Id	A Foreign Key to the Role Group table.	bigint

#### Attribute/Column(s) of Entity/Table "vt\$item fld\_defn NetworkAdapter"

Physical Name	Name	Definition	Physical Data Type
field_extlname_en_us	field_externalName_en_US	field_externalName_en_US.	varchar(255)
fieldrole	fieldRole	fieldRole.	varchar(255)
fieldlogicaldtype	fieldLogicalDataType	fieldLogicalDataType.	varchar(255)
fld_defn_id	fld_defn_id		bigint
fld_defn_ver_id	fld_defn_ver_id		bigint

#### Attribute/Column(s) of Entity/Table "vt\$item fld NetworkAdapter\$changes"

Physical Name	Name	Definition	Physical Data Type
field_extlname_en_us	field_externalName_en_US	field_externalName_en_US.	varchar(255)
fieldrole	fieldRole	fieldRole.	varchar(255)
fieldlogicaldtype	fieldLogicalDataType	fieldLogicalDataType.	varchar(255)
fld_defn_id	fld_defn_id		bigint
fld_defn_ver_id	fld_defn_ver_id		bigint
draft_fld_defn_ver_id	draft_fld_defn_ver_id		bigint
active_cfg_setting_val_id	active_cfg_setting_val_id		bigint
active_string_val	active_string_val	active_string_val.	varchar(255)
draft_cfg_setting_val_id	draft_cfg_setting_val_id		bigint
draft_string_val	draft_string_val	draft_string_val.	varchar(255)
dyn_col_spec_id	dyn_col_spec_id		bigint
logical_data_type_id	logical_data_type_id		int
logical_data_type_name	logical_data_type_name	logical_data_type_name.	varchar(255)
text_logical_data_type_id	TEXT_logical_data_type_id		int
schema_name	schema_name	schema_name.	varchar(30)
table_name	table_name	table_name.	varchar(30)
col_name	col_name	col_name.	varchar(255)

#### ER Diagram(s) of Subject Area "901 - Tables Added"

Name	Physical Display Level
Logical	Column
Physical	Column

#### Entity/Table(s) of Subject Area "901 - Tables Added"

Physical Name	Name	Logical Only	Do Not Generate	Definition
cntnr_snapshot	Container Snapshot		false	
snapshot_cfg_setting_val	Snapshot Configuration Setting Value		false	Configuration Setting Value is an association between the Configuration Setting entity and a container entity (i.e. User Profile, Role, and Definition). This association contains a distinct value for each of the intersection points between the two entities.
snapshot_defn	Snapshot Definition		false	
snapshot_defn_reln	Snapshot Definition Relationship		false	All relationships between definitions are stored in this table in order to make the model able to handle new relationships without changes to tables.
snapshot_defn_ver	Snapshot Definition Version		false	A version of a Definition with a version identifier that changes when data changes are made to the template. The definition version will suffice to store the current state of the FootPrints Configuration. A full copy of the definition tree is versioned when a container is published. See the database views v_defn_ver_active, v_cntnr_item_field.
snapshot_defn_ver_assoc	Snapshot Definition Version Assoc		false	
snapshot_status	Snapshot Status		false	
snapshot_type	Snapshot Type		false	

#### Attribute/Column(s) of Entity/Table "Container Snapshot"

Physical Name	Name	Definition	Physical Data Type
cntnr_snapshot_id	Container Snapshot Id	The automatically generated primary key of the cntnr_snapshot table.	bigint

cntr_defn_id	Container Definition Id	A Foreign Key to the Definition table.	bigint
create_date	Create Date	The date that the Container Snapshot was created.	datetime
snapshot_status_name	Snapshot Status Name	The unique string identifier for an instance of Snapshot Status. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)
last_good_publ_date	Last Good Publish Date		datetime

#### Attribute/Column(s) of Entity/Table "Snapshot Configuration Setting Value"

Physical Name	Name	Definition	Physical Data Type
cfg_setting_val_id	Configuration Setting Value Id	The automatically generated primary key of the snapshot_cfg_setting_val table.	bigint
cfg_setting_id	Configuration Setting Id	A Foreign Key to the Snapshot Configuration Setting Value table.	bigint
val_type_cd	Value Type Code	The data type for a Snapshot Configuration Setting Value. The choices are 'S' string, 'D' date, 'T' text, 'I' integer.	char(1)
string_val	String Value	The value of a Snapshot Configuration Setting Value, stored as a string.	varchar(255)
text_val	Text Value	The value of a Snapshot Configuration Setting Value, stored as a long text object.	varchar(max)
real_val	Real Value	The value of a Configuration Setting Value stored as a real.	numeric(38,20)
long_val	Long Value	The value of a Configuration Setting Value stored as Long.	bigint
date_val	Date Value	The value of a Snapshot Configuration Setting Value, stored as a date.	datetime
boolean_val	Boolean Value	The value of a Configuration Setting Value stored as Boolean.	bit
int_val	Int Value	The value of a Configuration Setting Value stored as Int.	int
owner_id	Owner Id	A Foreign Key to the Owning instance in either the Definition, Definition Version, Role, or User Profile tables.	bigint
val_map_key	Value Map Key	The value of a Snapshot Configuration Setting Value, stored as a string.	varchar(255)
cfg_setting_owner_type_name	Configuration Setting Owner Type Name	The unique string identifier for an instance of Snapshot Configuration Setting Value. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)
cntr_snapshot_id	Container Snapshot Id	A Foreign Key to the Container Snapshot table.	bigint

#### Attribute/Column(s) of Entity/Table "Snapshot Definition"

Physical Name	Name	Definition	Physical Data Type
cntr_snapshot_id	Container Snapshot Id	A Foreign Key to the Container Snapshot table.	bigint
defn_id	Definition Id	A Foreign Key to the Definition table.	bigint
defn_name	Definition Name	The name of the Snapshot Definition.	nvarchar(4000)
dsble_date	Disabled Date	The date that this instance of Snapshot Definition is logically deleted and disabled.	datetime
defn_guid	Definition GUID	A Global Unique Identifier search key for the Snapshot Definition, composed of letters and numbers. It is computed by the client and typically used in unstructured data.	varchar(50)
defn_type_name	Definition Type Name	The unique string identifier for an instance of Snapshot Definition. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)
defn_subtype_id	Definition Subtype Id	A Foreign Key to the Snapshot Definition table.	int
tmpl_export_guid	Template Export GUID	The value of a Snapshot Definition, stored as a string.	VARCHAR(50)

#### Attribute/Column(s) of Entity/Table "Snapshot Definition Relationship"

Physical Name	Name	Definition	Physical Data Type
defn_reln_id	Definition Relationship Id	The automatically generated primary key of the snapshot_definition_relationsh table.	bigint

from_defn_ver_id	From Definition Version Id	A Foreign Key to the Snapshot Definition Relationship table.	bigint
to_defn_ver_id	To Definition Version Id	A Foreign Key to the Snapshot Definition Relationship table.	bigint
defn_reln_type_id	Definition Relationship Type Id	A Foreign Key to the Snapshot Definition Relationship table.	int
cntr_snapshot_id	Container Snapshot Id	A Foreign Key to the Container Snapshot table.	bigint

#### Attribute/Column(s) of Entity/Table "Snapshot Definition Version"

Physical Name	Name	Definition	Physical Data Type
defn_id	Definition Id	A Foreign Key to the Snapshot Definition Version table.	bigint
defn_ver_id	Definition Version Id	The automatically generated primary key of the snapshot_definition_version table.	bigint
defn_ver_name	Definition Version Name	The name of the Snapshot Definition Version.	varchar(255)
defn_ver_status_id	Definition Version Status Id	A Foreign Key to the Snapshot Definition Version table.	int
defn_ver_guid	Definition Version GUID	A Global Unique Identifier search key for the Snapshot Definition Version, composed of letters and numbers. It is computed by the client and typically used in unstructured data.	varchar(50)
cntr_snapshot_id	Container Snapshot Id	A Foreign Key to the Container Snapshot table.	bigint

#### Attribute/Column(s) of Entity/Table "Snapshot Definition Version Assoc"

Physical Name	Name	Definition	Physical Data Type
defn_ver_association_id	Definition Version Association Id	The automatically generated primary key of the snapshot_definition_version_as table.	bigint
from_defn_ver_id	From Definition Version ID	A Foreign Key to the Snapshot Definition Version Assoc table.	bigint
to_defn_ver_id	To Definition Version ID	A Foreign Key to the Snapshot Definition Version Assoc table.	bigint
association_type_id	Association Type Id	A Foreign Key to the Snapshot Definition Version Assoc table.	bigint
cntr_snapshot_id	Container Snapshot Id	A Foreign Key to the Container Snapshot table.	bigint

#### Attribute/Column(s) of Entity/Table "Snapshot Status"

Physical Name	Name	Definition	Physical Data Type
snapshot_status_id	Snapshot Status Id	The unique identifier for an instance of Snapshot Status. This identifier must be kept in sync with programming code that uses an enumeration or fixed constant to identify particular values.	integer
snapshot_status_name	Snapshot Status Name	The unique string identifier for an instance of Snapshot Status. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)

#### Attribute/Column(s) of Entity/Table "Snapshot Type"

Physical Name	Name	Definition	Physical Data Type
snapshot_type_id	Snapshot Type Id	The unique identifier for an instance of Snapshot Type. This identifier must be kept in sync with programming code that uses an enumeration or fixed constant to identify particular values.	integer
snapshot_type_name	Snapshot Type Name	The unique string identifier for an instance of Snapshot Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)

#### ER Diagram(s) of Subject Area "902 - Columns Added"

Name	Physical Display Level
Logical	Column
Physical	Column

#### Entity/Table(s) of Subject Area "902 - Columns Added"

Physical Name	Name	Logical Only	Do Not Generate	Definition
---------------	------	--------------	-----------------	------------

audit_txn	Audit Transaction		false	A record of the Principal and create time of the start of the commit time of a logical transaction. In many cases multiple Audit Log entries will use the same transaction.
-----------	-------------------	--	-------	---

#### Attribute/Column(s) of Entity/Table "Audit Transaction"

Physical Name	Name	Definition	Physical Data Type
audit_txn_id	Audit Transaction Id	The automatically generated primary key of the audit_txn table.	bigint
txn_start_date	Transaction Start Date	The transaction date of the Audit Log entry. This date may not be in strict order with the Audit Log id in the event that features are used to continue a previous Audit Log transaction.	datetime
app_user_id	Application User Id	A Foreign Key to the Application User table. The audit trail transaction may or may not be associated with a customer or agent user. If this column is null, then a system component should be identified.	bigint
sys_component_id	System Component Id	This is a foreign key to the system component table. The audit trail transaction may or may not be associated with a customer or agent user. The system component will be set if there is no application user identified. It identifies the source of the transaction. Values include SYSTEM, WEB_SERVICES, ASSET_MANAGEMENT, CALENDAR, IMPORTING, etc. All identifiers are negative, so that they can be distinguished from userids in case they are both stored in the same dynamic table CreatedBy, and UpdatedBy fields. We are going to try first normal form normalization here and see if it works.	int
audit_rule_name_id	Audit Rule Name Id	The primary key is the original rule definition id. It will not change as the rule is renamed. If the rule is deleted, the deleted date will be filled in, but the rule name will still be known in this table.	bigint

#### ER Diagram(s) of Subject Area "903 - Tables Deleted"

Name	Physical Display Level
Logical	Column
Physical	Column

#### ER Diagram(s) of Subject Area "904 - Columns Deleted"

Name	Physical Display Level
Logical	Column
Physical	Column

#### ER Diagram(s) of Subject Area "Legend"

Name	Physical Display Level
Legend	Column

#### Entity/Table(s) of Subject Area "Legend"

Physical Name	Name	Logical Only	Do Not Generate	Definition
added_col	Added Column		false	
added_table	Added Table		false	
cfg_setting	Configuration Setting		false	A configuration variable that can be identified in programming code and which has an atomic value that affects the behavior of the application. Configuration values are either scalars, or are supported by value lists that also can be programmatically identified. They have localized labels and value choices, and are categorized to support organization and separation in programming namespaces and configuration screens. They support the concept of Built-In Values, Default Values, and Customized Values. Configuration should be supported by an Audit Trail, and should be able to determine if configuration has changed by level, category, and for the individual value in order to support cacheing. A configuration value has a definition to identify its usage, and an enum that identifies if a change will require a restart and what level of restart is required

				if the value changes. If a restart is required, both the currently running value and the value after restart should be accessible. This means that historical values of the Configuration Settings should be accessible.
cfg_setting_owner_type	Configuration Setting Owner Type		false	A lookup table for Owner Types of Configuration Setting Values. Values include: APPLICATION_USER, DEFINITION_IMPORTS_ROLE, SIMPLE_CONFIGURABLE_OBJECT, USER_PROFILE.
delete_entry	Delete Entry		false	
deleted_col	Deleted Column		false	
deleted_table	Deleted Table		false	
item_defn	Item Definition	true	true	The definition of an Item, which includes the Item Name and localizations and the collection of Item Fields. It also includes customizations and localizations of string resources for Item Names, and Item Field names.
subtype	Subtype		false	
subtype	Subtype		false	
subtype	Subtype		false	
supertype	Supertype		false	
to	To		false	
to	To		false	
to	To		false	
to	To		false	
to_one_entity	To One Entity		false	
to_one_or_more_entities	To One or More Entities		false	
to_zero_or_more_entities	To Zero or More Entities		false	
to_zero_or_one_entity	To Zero or One Entity		false	

#### Attribute/Column(s) of Entity/Table "Added Column"

Physical Name	Name	Definition	Physical Data Type
added_col_name	Added Column Name	The name of the Added Column.	varchar(255)
added_col_descr	Added Column Description	A descriptive name of the Added Column. This descriptive name may be localized.	varchar(512)
added_col_id	Added Column Id	The automatically generated primary key of the added_col table.	bigint

#### Attribute/Column(s) of Entity/Table "Added Table"

Physical Name	Name	Definition	Physical Data Type
added_col_name	Added Column Name	The name of the Added Table.	varchar(255)
added_col_descr	Added Column Description	A descriptive name of the Added Table. This descriptive name may be localized.	varchar(512)
added_table_id	Added Table Id	The automatically generated primary key of the added_table table.	bigint

#### Attribute/Column(s) of Entity/Table "Configuration Setting"

Physical Name	Name	Definition	Physical Data Type
cfg_ctgy_id	Configuration Category Id	A Foreign Key to the Configuration Category table.	bigint
cfg_setting_const	Configuration Setting Constant	The constant name of the Configuration Setting Constant, which can be aligned with an enumeration or other compile time constant value for access in Java.	varchar(255)
dflt_setting_label	Default Setting Label	The name of the Configuration Setting.	varchar(255)
dflt_setting_descr	Default Setting Description	The description of the purpose and usage of the configuration setting. This description serves for documentation, and will be used to help develop UI strings for the Label, ToolTip, HelpText etc. of the UI for the configuration setting.	varchar(255)
cfg_setting_id	Configuration Setting Id	The automatically generated primary key of the cfg_setting table.	bigint
agent_only	Agent Only	Indicates if Agent Only.	bit
unique_user_only	Unique User Only	Indicates if Unique User Only.	bit
cfg_val_list_id	Configuration Value List Id	A Foreign Key to the Configuration Value List table.	bigint
val_type_cd	Value Type Code	The data type for a Configuration Setting. The choices are 'S' string, 'D' date, 'T' text, 'L' Long, 'R' real, and 'A' for any type.	char(1)



cfg_list_type_name	Configuration List Type Name	The unique string identifier for an instance of Configuration List Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)
spec_class_name	Specification Class Name	The value of a Configuration Setting, stored as a string.	varchar(255)
cfg_is_val_mapped	Configuration Is Value Mapped	Indicates if Configuration Is Value Mapped.	bit

#### Attribute/Column(s) of Entity/Table "Configuration Setting Owner Type"

Physical Name	Name	Definition	Physical Data Type
cfg_setting_owner_type_descr	Configuration Setting Owner Type Description	A descriptive name of the Configuration Setting Owner Type. This descriptive name may be localized.	varchar(512)
cfg_setting_owner_type_name	Configuration Setting Owner Type Name	The unique string identifier for an instance of Configuration Setting Owner Type. This identifier must be kept in sync with programming code that uses an enumeration constant to identify particular values. A scan program is generally used to populate this table.	varchar(255)

#### Attribute/Column(s) of Entity/Table "Delete Entry"

Physical Name	Name	Definition	Physical Data Type
delete_entry_number_inherited	Delete Entry Number Inherited	The automatically generated primary key of the delete_entry table.	int
delete_entry_name	Delete Entry Name	The name of the Delete Entry.	varchar(255)
delete_entry_comment	Delete Entry Comment	A free-form text comment describing the Delete Entry.	varchar(255)

#### Attribute/Column(s) of Entity/Table "Deleted Column"

Physical Name	Name	Definition	Physical Data Type
deleted_col_id	Deleted Column Id	The automatically generated primary key of the deleted_col table.	bigint
deleted_col_name	Deleted Column Name	The name of the Deleted Column.	varchar(255)
deleted_col_descr	Deleted Column Description	A descriptive name of the Deleted Column. This descriptive name may be localized.	varchar(512)

#### Attribute/Column(s) of Entity/Table "Deleted Table"

Physical Name	Name	Definition	Physical Data Type
deleted_table_id	Deleted Table Id	The automatically generated primary key of the deleted_table table.	bigint
deleted_table_name	Deleted Table Name	The name of the Deleted Table.	varchar(255)
deleted_table_descr	Deleted Table Description	A descriptive name of the Deleted Table. This descriptive name may be localized.	varchar(512)

#### Attribute/Column(s) of Entity/Table "Item Definition"

Physical Name	Name	Definition	Physical Data Type
item_table_name	Item Table Name	The name of the DBMS_Object_Name. It is a database object name, limited to 30 characters in length.	varchar(30)
item_defn_id	Item Definition Id	A Foreign Key to the Definition table for an instance of type Definition.	bigint

#### Attribute/Column(s) of Entity/Table "To"

Physical Name	Name	Definition	Physical Data Type
to_id	To Id	The automatically generated primary key of the to table.	bigint

#### Attribute/Column(s) of Entity/Table "To"

Physical Name	Name	Definition	Physical Data Type
to_id	To Id	The automatically generated primary key of the to table.	bigint

#### Attribute/Column(s) of Entity/Table "To"

Physical Name	Name	Definition	Physical Data Type
to_id	To Id	The automatically generated primary key of the to table.	bigint

#### Attribute/Column(s) of Entity/Table "To"

Physical Name	Name	Definition	Physical Data Type
---------------	------	------------	--------------------

to_id	To Id	The automatically generated primary key of the to table.	bigint
-------	-------	--	--------

#### Attribute/Column(s) of Entity/Table "To One Entity"

Physical Name	Name	Definition	Physical Data Type
to_id	To Id	A Foreign Key to the To table.	bigint

#### Attribute/Column(s) of Entity/Table "To One or More Entities"

Physical Name	Name	Definition	Physical Data Type
to_id	To Id	A Foreign Key to the To table.	bigint

#### Attribute/Column(s) of Entity/Table "To Zero or More Entities"

Physical Name	Name	Definition	Physical Data Type
to_id	To Id	A Foreign Key to the To table.	bigint

#### Attribute/Column(s) of Entity/Table "To Zero or One Entity"

Physical Name	Name	Definition	Physical Data Type
to_id	To Id	A Foreign Key to the To table.	bigint

# Index

.

:Item\_Shortname Balance [236, 237](#)  
:item\_shortname Event Sum [236, 238](#)  
:item\_shortname Status [236, 238](#)  
:item\_shortname Transition [236, 239](#)  
:item\_shortname Transition Sum [236, 239](#)  
:item\_shortname\_bal [236](#)  
:item\_shortname\_event\_sum [236](#)  
:item\_shortname\_status [236](#)  
:item\_shortname\_transn [236](#)  
:item\_shortname\_transn\_sum [236](#)

## 0

000 - Very High Level Domain Model [118, 119](#)  
001 - High Level Domain Model [118, 130](#)  
010 - Translation [118, 147](#)  
040 - Container [118, 149](#)  
041 - Definition Detail [118, 152](#)  
046 - Workspace Item Assignment [118, 155](#)  
047 - Workspace Time Tracking [118, 159](#)  
048 - Item Field Journaling [118, 161](#)  
050 - Item Relationship [118, 163](#)  
051 - Item Field Collection [118, 165](#)  
060 - Role [118, 166](#)  
065 - Configuration Setting [118, 172](#)  
066 - Container Snapshot [118, 179](#)  
070 - User [118, 186](#)  
071 - Team [118, 189](#)  
080 - User Profile [118, 192](#)  
096 - Email Notification [118, 197](#)

## 1

101 - Data Access System [118, 198](#)  
102 - Data Access Scripts [118, 200](#)  
110 - Id Generation [118, 203](#)  
120 - Job Scheduler [118, 203](#)  
140 - Workspace Calendar and Operation Schedule [118, 204](#)  
141 - Calendar Appointments and Items [118, 208](#)  
142 - Calendar Sync [118, 213](#)  
160 - Templating [118, 216](#)  
170 - Knowledge Base [118, 218](#)  
190 - Password Reset [118, 221](#)

## 2

210 - Audit Trail [119, 222](#)  
230 - Rules [119, 226](#)  
240 - File Storage [119, 226](#)  
250 - Apache Jack Rabbit JCR [119, 227](#)  
262 - Import Items and Users [119, 228](#)  
280 - Session and Locking [119, 233](#)  
290 - Security [119, 235](#)

## 3

300 - Ticket Data Mart [119, 236](#)  
305 - Data Mart Load [119, 242](#)  
310 - Auxiliary Tables [119, 245](#)  
320 - Report [119, 247](#)  
332 - Change Management Voting [119, 252](#)  
340 - Search [119, 254](#)  
350 - Chat [119, 257](#)  
360 - Survey [119, 262](#)  
370 - Deployment [119, 262](#)  
380 - Portal [119, 263](#)  
390 - Business Process Template [119, 264](#)

## 4

400 - Asset Management [119, 266](#)  
410 - Publishing [119, 267](#)  
430 - UI Component State [119, 269](#)

## 9

900 To Be Deleted [119, 270](#)  
901 - Tables Added [119, 272](#)  
902 - Columns Added [119, 274](#)  
903 - Tables Deleted [119, 275](#)  
904 - Columns Deleted [119, 275](#)

## A

Activated Count [237](#)

## a

activated\_ct [237](#)

## A

Activation Clock Hours Count [238](#)  
Activation Work Hours Count [238](#)

## a

active\_cfg\_setting\_val\_id [272](#)  
active\_string\_val [272](#)

## A

Activity Definition [130, 138](#)

## a

actvn\_clock\_hours\_ct [238](#)  
actvn\_work\_hours\_ct [238](#)  
actvy\_defn [130](#)

## A

Added Column [275, 276](#)  
Added Column Description [276](#)  
Added Column Id [276](#)  
Added Column Name [276](#)  
Added Table [275, 276](#)  
Added Table Id [276](#)

## a

added\_col [275](#)  
added\_col\_descr [276](#)  
added\_col\_id [276](#)  
added\_col\_name [276](#)  
added\_table [275](#)  
added\_table\_id [276](#)  
addr\_book [119, 130, 149, 198](#)  
addr\_book\_cntnr\_name [124, 138, 151](#)  
addr\_book\_entry [121, 132, 135, 156](#)  
addr\_book\_type\_id [124, 138, 151](#)

## A

Address Book [198](#)  
Address Book Container [119, 124, 130, 138, 149, 150](#)  
Address Book Container Name [124, 138, 151](#)  
Address Book Type Id [124, 138, 151](#)

## a

admin [167](#)  
admin\_type [167](#)  
admin\_type\_id [125, 139, 168, 169, 174, 194, 230](#)  
admin\_type\_name [168](#)

## A

Administrator CMDB Role [167, 168](#)  
Administrator Type [167, 168](#)  
Administrator Type Id [125, 139, 168, 169, 174, 194, 230](#)  
Administrator Type Name [168](#)

## a

agent [130, 186, 204](#)

## A

Agent [130, 138, 186, 187, 204, 205](#)  
Agent Authentication Type [192, 194](#)  
Agent Authentication Type Id [129, 146, 171, 178, 188, 194, 197](#)  
Agent Authentication Type Name [194](#)

Agent Only [126, 140, 176, 182, 276](#)  
Agent Profile [192, 194](#)  
Agent Role [167, 168](#)  
Agent Team [189, 190](#)  
Agent Team Id [190](#)

## a

agent\_authcn\_type [192](#)  
agent\_authcn\_type\_id [129, 146, 171, 178, 188, 194, 197](#)  
agent\_authcn\_type\_name [194](#)  
agent\_only [126, 140, 176, 182, 276](#)  
agent\_prof [192](#)  
agent\_role [167](#)  
agent\_team [189](#)  
agent\_team\_id [190](#)

## A

Alert Role Type [257, 259](#)  
Alert Role Type Id [259, 261](#)  
Alert Role Type Name [259](#)

## a

alert\_role\_type [257](#)  
alert\_role\_type\_id [259, 261](#)  
alert\_role\_type\_name [259](#)

## A

All Activation Clock Hours Count [238](#)  
All Activation Work Hours Count [238](#)  
All Clock Hours Count [239](#)  
All Creation Clock Hours Count [238](#)  
All Creation Work Hours Count [238](#)  
All Work Hours Count [239](#)

## a

all\_actvn\_clock\_hours\_ct [238](#)  
all\_actvn\_work\_hours\_ct [238](#)  
all\_clock\_hours\_ct [239](#)  
all\_cntcs\_flag [232, 251](#)

## A

All\_Contacts\_Flag [232, 251](#)

## a

all\_creatn\_clock\_hours\_ct [238](#)  
all\_creatn\_work\_hours\_ct [238](#)  
all\_work\_hours\_ct [239](#)

## A

Allow Anonymous Access [264](#)

## a

allow\_anonymous\_access [264](#)

## A

Anonymous Customer [187](#)  
Anonymous Customer Profile [192, 194](#)

## a

anonymous\_cust [187](#)  
anonymous\_cust\_prof [192](#)  
answer [221](#)

## A

Answer [221](#)

## a

app\_ele [119, 130, 147, 216](#)  
app\_ele\_id [124, 125, 138, 141, 143, 144, 145, 146, 148, 217, 271](#)  
app\_ele\_name [124, 138, 148, 217](#)  
app\_ele\_nmsp [147](#)  
app\_ele\_nmsp\_id [124, 138, 148, 217](#)  
app\_ele\_nmsp\_name [148](#)  
app\_ele\_string [120, 130, 147](#)  
app\_ele\_string\_id [124, 128, 138, 145, 148, 149](#)

[app\\_ele\\_string\\_usage](#) 147  
[app\\_ele\\_type](#) 147  
[app\\_ele\\_type\\_id](#) 124, 138, 148, 149, 217  
[app\\_ele\\_type\\_name](#) 149  
[app\\_role](#) 120, 130, 167, 172, 192, 229  
[app\\_role\\_descr](#) 125, 139, 169, 174, 194, 230  
[app\\_role\\_id](#) 125, 129, 138, 144, 146, 168, 169, 170, 171, 172, 174, 178, 188, 194, 196, 230, 232, 271  
[app\\_role\\_name](#) 125, 139, 169, 174, 194, 230  
[app\\_user](#) 120, 131, 155, 159, 161, 187, 189, 193, 208, 214, 218, 221, 222, 229, 233, 247, 252, 254, 257, 263, 267, 269  
[app\\_user\\_id](#) 125, 128, 138, 139, 141, 144, 157, 160, 162, 163, 187, 188, 190, 191, 195, 205, 206, 210, 212, 213, 214, 215, 216, 219, 220, 221, 222, 223, 225, 230, 231, 232, 234, 235, 244, 248, 251, 253, 255, 259, 260, 263, 268, 270, 275  
[app\\_user\\_prof\\_link](#) 120, 187, 193  
[app\\_user\\_prof\\_link\\_id](#) 125, 188, 195

## A

[Application Element](#) 119, 124, 130, 138, 147, 148, 216, 217  
[Application Element Id](#) 124, 125, 138, 141, 143, 144, 145, 146, 148, 217, 271  
[Application Element Name](#) 124, 138, 148, 217  
[Application Element Namespace](#) 147, 148  
[Application Element Namespace Id](#) 124, 138, 148, 217  
[Application Element Namespace Name](#) 148  
[Application Element String](#) 120, 124, 130, 138, 147, 148  
[Application Element String Id](#) 124, 128, 138, 145, 148, 149  
[Application Element String Usage](#) 147, 148  
[Application Element Type](#) 147, 149  
[Application Element Type Id](#) 124, 138, 148, 149, 217  
[Application Element Type Name](#) 149  
[Application Role](#) 120, 125, 130, 138, 167, 169, 172, 174, 192, 194, 229, 230  
[Application Role Description](#) 125, 139, 169, 174, 194, 230  
[Application Role Id](#) 125, 129, 138, 144, 146, 168, 169, 170, 171, 172, 174, 178, 188, 194, 196, 230, 232, 271  
[Application Role Name](#) 125, 139, 169, 174, 194, 230  
[Application User](#) 120, 125, 131, 139, 155, 157, 159, 160, 161, 162, 187, 189, 190, 193, 195, 208, 210, 214, 218, 219, 221, 222, 223, 229, 230, 233, 234, 247, 248, 252, 253, 254, 255, 257, 259, 263, 267, 268, 269, 270  
[Application User Id](#) 125, 128, 138, 139, 141, 144, 157, 160, 162, 163, 187, 188, 190, 191, 195, 205, 206, 210, 212, 213, 214, 215, 216, 219, 220, 221, 222, 223, 225, 230, 231, 232, 234, 235, 244, 248, 251, 253, 255, 259, 260, 263, 268, 270, 275  
[Application User Profile Link](#) 120, 125, 187, 193, 195  
[Application User Profile Link Id](#) 125, 188, 195  
[Appointment](#) 120, 131  
[Appointment Calendar Item Id](#) 210  
[Approver Id](#) 254

## a

[approver\\_id](#) 254  
[appt](#) 120, 131  
[appt\\_cal\\_item\\_id](#) 210

## A

[Arrive Date](#) 260

## a

[arrive\\_date](#) 260  
[asgnmt](#) 156, 189  
[asgnmt\\_id](#) 157, 190  
[asgnmt\\_type](#) 156, 189  
[asgnmt\\_type\\_descr](#) 158, 191  
[asgnmt\\_type\\_id](#) 157, 158, 191  
[asgnmt\\_type\\_name](#) 158, 191

## A

[Asset Item](#) 131, 139

## a

[asset\\_item](#) 131

## A

[Assignment](#) 156, 157, 189, 190  
[Assignment Id](#) 157, 190  
[Assignment Type](#) 156, 158, 189, 191  
[Assignment Type Description](#) 158, 191  
[Assignment Type Id](#) 157, 158, 191  
[Assignment Type Name](#) 158, 191  
[Association Type](#) 152, 153  
[Association Type Description](#) 153  
[Association Type Id](#) 153, 155, 184, 186, 274

Association Type Name 153

## a

association\_type 152

association\_type\_descr 153

association\_type\_id 153, 155, 184, 186, 274

association\_type\_name 153

## A

Attachment 131

Attachment Inbound Email 131

Attachment Outbound Email 131

## a

attch 131

attch\_inbd\_email 131

attch\_outbd\_email 131

attr\_id 235

attr\_type 211

attr\_val 235

## A

Attribute Id 235

## a

attribute type 211

## A

Attribute Value 235

Audit Class Name 225

Audit Event Type 222, 223

Audit Event Type Id 223, 224

Audit Event Type Name 223

Audit Field Configuration 222, 223

Audit Field Configuration Id 223

Audit Log 222, 223

Audit Log Id 224, 225

Audit Message Template 222, 224

Audit Message Template Id 224

Audit Message Template Key 224

Audit Property Change 222, 224

Audit Property Change Id 224

Audit Property Dimension 222, 225

Audit Property Dimension Id 224, 225

Audit Rule Name 223, 225

Audit Rule Name Id 225, 226, 244, 275

Audit Trail 120, 125, 131, 139

Audit Transaction 131, 223, 225, 242, 243, 275

Audit Transaction Audit Trail 120, 125, 131, 139

Audit Transaction Id 224, 225, 244, 275

## a

audit\_class\_name 225

audit\_event\_type 222

audit\_event\_type\_id 223, 224

audit\_event\_type\_name 223

audit\_field\_cfg 222

audit\_field\_cfg\_id 223

audit\_log 222

audit\_log\_id 224, 225

audit\_msg\_tmpl 222, 224

audit\_msg\_tmpl\_id 224

audit\_msg\_tmpl\_key 224

audit\_prop\_chg 222

audit\_prop\_chg\_id 224

audit\_prop\_dim 222

audit\_prop\_dim\_id 224, 225

audit\_rule\_name 223

audit\_rule\_name\_id 225, 226, 244, 275

audit\_trail 120, 131

audit\_txn 131, 223, 242, 275

audit\_txn\_audit\_trail 120, 131

audit\_txn\_id 224, 225, 244, 275

## A

Author Name [269](#)

## a

author\_name [269](#)

## A

Auto Add Customer Profile [193, 195](#)

## a

auto\_add\_cust\_prof [193](#)

## B

Batch Id [200](#)

## b

batch\_id [200](#)

## B

Begin Date [157, 191, 206, 232, 251](#)

Begin Date Hour Local [207, 237, 242](#)

Begin Date Hour Local S [207, 242](#)

Begin Date Hour UTC [207, 241](#)

Begin Date Hour UTC S [207, 242](#)

Begin Date UTC S [207](#)

Begin Is Daylight Savings [207, 242](#)

## b

begin\_date [157, 191, 206, 232, 251](#)

begin\_date\_hour\_local [207, 237, 242](#)

begin\_date\_hour\_local\_s [207, 242](#)

begin\_date\_hour\_utc [207, 241](#)

begin\_date\_hour\_utc\_s [207, 242](#)

begin\_date\_utc\_s [207](#)

begin\_is\_daylight\_savings [207, 242](#)

binary value [211](#)

binary\_val [211](#)

## B

Binval Data [228](#)

Binval Id [228](#)

## b

binval\_data [228](#)

binval\_id [228](#)

boolean [175](#)

## B

Boolean [175](#)

## b

boolean value [211](#)

## B

Boolean Value [140, 169, 177, 182, 185, 250, 273](#)

## b

boolean\_val [140, 169, 177, 178, 182, 185, 211, 250, 273](#)

## B

Broadcast Alert [258, 259](#)

Broadcast Alert Id [259](#)

Broadcast Message [258, 259](#)

Broadcast Message Id [259](#)

Broadcast Message Type [258, 260](#)

Broadcast Message Type Id [259, 260](#)

Broadcast Message Type Name [260](#)

## b

broadcast\_alert [258](#)

broadcast\_alert\_id [259](#)

broadcast\_msg [258](#)

broadcast\_msg\_id [259](#)

broadcast\_msg\_type [258](#)

broadcast\_msg\_type\_id [259, 260](#)



broadcast\_msg\_type\_name [260](#)

## B

Build Number [262](#)

## b

build\_number [262](#)

bundle Data [228](#)

bundle\_data [228](#)

bus\_proc [264](#)

bus\_proc\_defn [264](#)

bus\_proc\_defn\_id [265](#)

bus\_proc\_descr [265](#)

bus\_proc\_id [265](#)

bus\_proc\_name [265](#)

bus\_proc\_tmpl\_id [265](#)

bus\_rule\_defn [131](#)

## B

Business Process [264](#), [265](#)

Business Process Definition [264](#), [265](#)

Business Process Definition Id [265](#)

Business Process Description [265](#)

Business Process Id [265](#)

Business Process Name [265](#)

Business Process Template Id [265](#)

Business Rule Definition [131](#), [139](#)

## C

cal\_cntrn\_item [208](#)

cal\_cntrn\_item\_id [210](#)

cal\_coll\_item [208](#)

cal\_content\_data [208](#)

cal\_content\_data\_id [206](#), [211](#), [212](#), [215](#)

cal\_event\_stamp [208](#)

cal\_field [208](#)

cal\_field\_dict\_val [208](#)

cal\_field\_id [211](#)

cal\_field\_multistring\_val [208](#)

cal\_item [204](#), [208](#), [214](#)

cal\_item\_id [205](#), [206](#), [210](#), [211](#), [212](#), [213](#), [214](#)

cal\_item\_intrnl\_id [215](#)

cal\_schd [204](#), [209](#)

cal\_schd\_id [206](#), [212](#)

cal\_stamp [209](#)

cal\_stamp\_id [211](#), [212](#)

cal\_subscription [209](#)

cal\_subscription\_id [212](#)

cal\_ticket [209](#)

cal\_ticket\_id [213](#), [271](#)

cal\_ticket\_privilege [209](#), [270](#)

cal\_type [204](#)

cal\_type\_id [128](#), [144](#), [206](#)

cal\_type\_name [206](#)

cal\_wksp [209](#)

cal\_wksp\_id [213](#)

## C

Calendar Collection Item [208](#), [210](#)

Calendar Container Item [208](#), [210](#)

Calendar Container Item Id [210](#)

Calendar Content Data [208](#), [211](#)

Calendar Content Data Id [206](#), [211](#), [212](#), [215](#)

Calendar Event Stamp [208](#), [211](#)

Calendar Field [208](#), [211](#)

Calendar Field Dictionary Value [208](#), [211](#)

Calendar Field Id [211](#)

Calendar Field Multistring Value [208](#), [211](#)

Calendar Item [204](#), [205](#), [208](#), [211](#), [214](#)

Calendar Item Id [205](#), [206](#), [210](#), [211](#), [212](#), [213](#), [214](#)

Calendar Item Internal Id [215](#)

Calendar Schedule [204](#), [206](#), [209](#), [212](#)

Calendar Schedule Id [206](#), [212](#)

Calendar Stamp [209, 212](#)  
Calendar Stamp Id [211, 212](#)  
Calendar Subscription [209, 212](#)  
Calendar Subscription Id [212](#)  
Calendar Ticket [209, 212](#)  
Calendar Ticket Id [213, 271](#)  
Calendar Ticket Privilege [209, 213, 270, 271](#)  
Calendar Type [204, 206](#)  
Calendar Type Id [128, 144, 206](#)  
Calendar Type Name [206](#)  
Calendar Workspace [209, 213](#)  
Calendar Workspace Id [213](#)  
Cardinality Node 1 [165](#)  
Cardinality Node 2 [165](#)

## C

cardinality\_node\_1 [165](#)  
cardinality\_node\_2 [165](#)

## C

CC Email Address [157, 191](#)

## C

cc\_email\_addr [157, 191](#)  
certificate [247](#)

## C

Certificate [247](#)

## C

cfg\_allow\_val [172](#)  
cfg\_allow\_val\_descr [175](#)  
cfg\_allow\_val\_id [175](#)  
cfg\_area\_type [172](#)  
cfg\_area\_type\_descr [175](#)  
cfg\_area\_type\_name [175](#)  
cfg\_ctgy [172](#)  
cfg\_ctgy\_id [126, 139, 175, 176, 182, 276](#)  
cfg\_ctgy\_type [173](#)  
cfg\_ctgy\_type\_descr [175](#)  
cfg\_ctgy\_type\_name [175](#)  
cfg\_file\_content [247](#)  
cfg\_file\_related\_path [247](#)  
cfg\_file\_repository [245](#)  
cfg\_is\_val\_mapped [126, 140, 176, 182, 277](#)  
cfg\_item [120, 131, 156](#)  
cfg\_list\_type [173](#)  
cfg\_list\_type\_descr [176](#)  
cfg\_list\_type\_name [126, 140, 175, 176, 177, 182, 277](#)  
cfg\_obj [173](#)  
cfg\_obj\_id [176](#)  
cfg\_obj\_subtype [176](#)  
cfg\_obj\_type [176](#)  
cfg\_setting [121, 131, 173, 179, 275](#)  
cfg\_setting\_const [126, 140, 176, 182, 276](#)  
cfg\_setting\_id [126, 140, 169, 176, 177, 178, 182, 184, 273, 276](#)  
cfg\_setting\_owner\_type [173, 276](#)  
cfg\_setting\_owner\_type\_descr [176, 277](#)  
cfg\_setting\_owner\_type\_name [140, 169, 176, 177, 178, 182, 185, 273, 277](#)  
cfg\_setting\_val [132, 167, 173, 180](#)  
cfg\_setting\_val\_id [140, 169, 177, 178, 182, 184, 273](#)  
cfg\_val\_list [173](#)  
cfg\_val\_list\_id [126, 140, 175, 176, 177, 182, 276](#)

## C

Change Key [215](#)  
Chat Channel [258, 260](#)  
Chat Channel Category [260](#)  
Chat Channel Id [260](#)  
Chat Channel Subcategory [260](#)  
Chat Channel Subscriber [258, 260](#)  
Chat Channel Subscriber Id [260](#)  
Chat Message [258, 260](#)  
Chat Message Id [260](#)

Chat Status [258, 260](#)  
Chat Status Description [260](#)  
Chat Status Id [260](#)  
Chat Status Name [260](#)  
Chat User Status [258, 260](#)  
Chat User Status Id [260](#)

## C

chat\_channel [258](#)  
chat\_channel\_ctgy [260](#)  
chat\_channel\_id [260](#)  
chat\_channel\_subctgy [260](#)  
chat\_channel\_subscriber [258](#)  
chat\_channel\_subscriber\_id [260](#)  
chat\_msg [258](#)  
chat\_msg\_id [260](#)  
chat\_status [258](#)  
chat\_status\_descr [260](#)  
chat\_status\_id [260](#)  
chat\_status\_name [260](#)  
chat\_user\_status [258](#)  
chat\_user\_status\_id [260](#)  
chg\_key [215](#)  
client create date [205, 212, 215](#)  
client modified date [205, 212, 215](#)  
client\_create\_date [205, 212, 215](#)  
client\_mod\_date [205, 212, 215](#)

## C

Clock Hours Count [239](#)

## C

clock\_hours\_ct [239](#)

## C

CMDB Container [120, 125, 131, 139, 150, 151](#)  
CMDB Role [167, 169](#)

## C

cmdb\_cntnr [120, 131, 150](#)  
cmdb\_role [167](#)  
cntc\_defn\_id [232, 251](#)  
cntc\_email [232, 251](#)  
cntc\_item\_id [232, 251](#)  
cntnr [121, 132, 150, 180](#)  
cntnr\_defn [121, 132, 150, 152, 180, 200, 233](#)  
cntnr\_defn\_id [183, 184, 202, 269, 273](#)  
cntnr\_grp\_defn [132](#)  
cntnr\_grp\_reln [132](#)  
cntnr\_grp\_reln\_id [141](#)  
cntnr\_home\_page [132](#)  
cntnr\_id [124, 126, 127, 130, 138, 139, 141, 143, 144, 147, 151, 152, 158, 160, 162, 164, 166, 182, 186, 191, 208, 213, 220, 234, 254, 261, 262, 267, 269](#)  
cntnr\_name [126, 140, 151, 182](#)  
cntnr\_snapshot [180, 272](#)  
cntnr\_snapshot\_id [183, 185, 186, 272, 273, 274](#)  
cntnr\_type [263](#)  
cntnr\_type\_id [124, 126, 130, 138, 139, 143, 144, 147, 151, 152, 186, 208, 262](#)  
col\_name [144, 199, 271, 272](#)  
coll\_cal\_item [210](#)  
coll\_class\_name [225](#)  
coll\_uid [212](#)

## C

Collection Calendar item [210](#)  
Collection Class Name [225](#)

## C

collection uid [212](#)

## C

Color Value Setting [165](#)

## C

color\_val\_setting [165](#)

## C

Column Name [144, 199, 271](#)

## C

comments [254](#)

## C

Comments [254](#)

Completed Date [198](#)

## C

completed\_date [198](#)

cond\_name [256](#)

## C

Condition Name [256](#)

Configuration Allowed Value [172, 174](#)

Configuration Allowed Value Description [175](#)

Configuration Allowed Value Id [175](#)

Configuration Area Type [172, 175](#)

Configuration Area Type Description [175](#)

Configuration Area Type Name [175](#)

Configuration Category [172, 175](#)

Configuration Category Id [126, 139, 175, 176, 182, 276](#)

Configuration Category Type [173, 175](#)

Configuration Category Type Description [175](#)

Configuration Category Type Name [175](#)

Configuration File Content [247](#)

Configuration File Related Path [247](#)

Configuration File Repository [245, 247](#)

Configuration Is Value Mapped [126, 140, 176, 182, 277](#)

Configuration Item [120, 126, 131, 139, 156, 158](#)

Configuration List Type [173, 175](#)

Configuration List Type Description [176](#)

Configuration List Type Name [126, 140, 175, 176, 177, 182, 277](#)

Configuration Object [173, 176](#)

Configuration Object Id [176](#)

Configuration Object Subtype [176](#)

Configuration Object Type [176](#)

Configuration Setting [121, 126, 131, 139, 173, 176, 179, 181, 275, 276](#)

Configuration Setting Constant [126, 140, 176, 182, 276](#)

Configuration Setting Id [126, 140, 169, 176, 177, 182, 184, 273, 276](#)

Configuration Setting Owner Type [173, 176, 276, 277](#)

Configuration Setting Owner Type Description [176, 277](#)

Configuration Setting Owner Type Name [140, 169, 176, 177, 182, 185, 273, 277](#)

Configuration Setting Value [132, 140, 167, 169, 173, 176, 180, 182](#)

Configuration Setting Value Id [140, 169, 177, 182, 184, 273](#)

Configuration Value List [173, 177](#)

Configuration Value List Id [126, 140, 175, 176, 177, 182, 276](#)

## C

console [263](#)

## C

Console [263](#)

Console Description [263](#)

Console GUID [263](#)

Console Id [263, 264](#)

Console Index [264](#)

Console Name [263](#)

## C

console\_descr [263](#)

console\_guid [263](#)

console\_id [263, 264](#)

console\_index [264](#)

console\_name [263](#)

## C

Constraint Value [199, 256](#)

Contact Definition Id [232, 251](#)

Contact Email [232, 251](#)

Contact Item [121, 126, 132, 140, 156, 158](#)  
 Contact Item Id [232, 251](#)  
 Container [121, 126, 132, 140, 150, 151, 180, 182](#)  
 Container Definition [121, 126, 132, 141, 150, 151, 152, 153, 180, 183, 200, 201, 233, 234](#)  
 Container Definition Id [183, 184, 202, 269, 273](#)  
 Container Group Definition [132, 141](#)  
 Container Group Relationship [132, 141](#)  
 Container Group Relationship Id [141](#)  
 Container Home Page [132, 141](#)  
 Container Id [124, 126, 127, 130, 138, 139, 141, 143, 144, 147, 151, 152, 158, 160, 162, 164, 166, 182, 186, 191, 208, 213, 220, 234, 254, 261, 262, 267, 269](#)  
 Container Name [126, 140, 151, 182](#)  
 Container Snapshot [180, 183, 272](#)  
 Container Snapshot Id [183, 185, 186, 272, 273, 274](#)  
 Container Type [263](#)  
 Container Type Id [124, 126, 130, 138, 139, 143, 144, 147, 151, 152, 186, 208, 262](#)

## C

content encoding [205, 212, 215](#)  
 content language [205, 212, 215](#)  
 content length [205, 212, 215](#)  
 content type [206, 212, 215](#)  
 content\_blob [211](#)  
 content\_encoding [205, 212, 215](#)  
 content\_language [205, 212, 215](#)  
 content\_length [205, 212, 215](#)  
 content\_type [206, 212, 215](#)

## C

Context [257](#)  
 Context Phrase [128, 145, 149](#)  
 Create Application User Id [232, 251](#)

## C

create date [205, 210, 211, 212, 214](#)

## C

Create Date [163, 183, 184, 197, 202, 203, 213, 220, 224, 227, 232, 234, 235, 251, 257, 273](#)  
 Create Edit CMDB Role [167, 169](#)  
 Create Time [235](#)  
 Create User [197, 203, 257](#)

## C

create\_app\_user\_id [232, 251](#)  
 create\_date [163, 183, 184, 197, 202, 203, 205, 210, 211, 212, 213, 214, 220, 224, 227, 232, 234, 235, 251, 257, 273](#)  
 create\_edit [167](#)  
 create\_time [235](#)  
 create\_user [197, 203, 257](#)

## C

Created By [249](#)  
 Created Count [231](#)  
 Created On [249](#)

## C

created\_by [249](#)  
 created\_ct [231](#)  
 created\_on [249](#)

## C

Creation Clock Hours Count [238](#)  
 Creation Date [213](#)  
 Creation Work Hours Count [238](#)

## C

creatn\_clock\_hours\_ct [238](#)  
 creatn\_date [213](#)  
 creatn\_work\_hours\_ct [238](#)  
 cron [203](#)

## C

Cron [203](#)

## C

cstr\_val [199, 256](#)

cum\_work\_durn\_in\_mins 206

## C

Cumulative Work Duration In Minutes 206

## C

cust 132, 187

cust\_authcn\_type 193

cust\_authcn\_type\_id 129, 146, 172, 179, 188, 195, 197

cust\_authcn\_type\_name 195

cust\_login\_type 187

cust\_login\_type\_id 125, 139, 141, 157, 160, 162, 187, 188, 190, 195, 210, 214, 219, 221, 223, 230, 234, 248, 253, 255, 259, 263, 268, 270

cust\_login\_type\_name 188

cust\_prof 193

cust\_read\_knowledge\_base\_and\_r 167

cust\_read\_knowledge\_base\_role 167

cust\_read\_submit\_edit\_role 167

cust\_read\_submit\_role 167

cust\_role 167

custom 167

## C

Custom CMDB Role 167, 170

Customer 132, 141, 187, 188

Customer Authentication Type 193, 195

Customer Authentication Type Id 129, 146, 172, 179, 188, 195, 197

Customer Authentication Type Name 195

Customer Login Type 187, 188

Customer Login Type Id 125, 139, 141, 157, 160, 162, 187, 188, 190, 195, 210, 214, 219, 221, 223, 230, 234, 248, 253, 255, 259, 263, 268, 270

Customer Login Type Name 188

Customer Profile 193, 195

Customer Read Knowledge Base and Requests Role 167, 170

Customer Read Knowledge Base Role 167, 170

Customer Read/Submit Role 167, 170

Customer Read/Submit/Edit Role 167, 170

Customer Role 167, 170

## C

cxt 257

cxt\_phrase 128, 145, 149

## D

Data Load Type 243, 244

Data Load Type Description 244

Data Load Type Name 244

Data Mart Load History 243, 244

Data Mart Load History Id 244

Data Mart Load Lock 243, 244

Data Mart Load Lock Id 244

Data Mart Name 244

Data Mart Request History 243, 244

Data Mart Request History Id 244

Data Mart Request Queue 243, 245

Data Mart Request Queue Id 245

Data Mart Request State 243, 245

Data Mart Request State Description 245

Data Mart Request State Id 245

Data Mart Request State Name 245

Data Mart Request Type 243, 245

Data Mart Request Type Description 245

Data Mart Request Type Id 245

Data Mart Request Type Name 245

## d

data\_load\_type 243

data\_load\_type\_descr 244

data\_load\_type\_name 244

data\_mart\_load\_history 243

data\_mart\_load\_history\_id 244

data\_mart\_load\_lock 243

data\_mart\_load\_lock\_id 244

data\_mart\_name 244

[data\\_mart\\_rqst\\_history](#) 243  
[data\\_mart\\_rqst\\_history\\_id](#) 244  
[data\\_mart\\_rqst\\_queue](#) 243  
[data\\_mart\\_rqst\\_queue\\_id](#) 245  
[data\\_mart\\_rqst\\_state](#) 243  
[data\\_mart\\_rqst\\_state\\_descr](#) 245  
[data\\_mart\\_rqst\\_state\\_id](#) 245  
[data\\_mart\\_rqst\\_state\\_name](#) 245  
[data\\_mart\\_rqst\\_type](#) 243  
[data\\_mart\\_rqst\\_type\\_descr](#) 245  
[data\\_mart\\_rqst\\_type\\_id](#) 245  
[data\\_mart\\_rqst\\_type\\_name](#) 245

## D

[Database Data Type](#) 198, 199  
[Database Data Type Id](#) 199  
[Database Data Type Name](#) 199  
[Database Object](#) 132  
[Database Object Definition](#) 132, 141  
[Database Schema Name](#) 126, 141, 145, 151, 153, 155, 183, 201, 234  
[Database Table Name](#) 203  
[Date Hour Local](#) 238, 239, 240, 241  
[Date Sent](#) 262

## d

[date value](#) 211

## D

[Date Value](#) 140, 169, 175, 177, 182, 185, 250, 273

## d

[date\\_hour\\_local](#) 238, 239, 240, 241  
[date\\_sent](#) 262  
[date\\_val](#) 140, 169, 175, 177, 178, 182, 185, 211, 250, 273

## D

[Day Date](#) 241  
[Day of Month](#) 241  
[Day of Quarter](#) 241  
[Day of Week](#) 241  
[Day of Year](#) 241

## d

[day\\_date](#) 241  
[day\\_of\\_month](#) 241  
[day\\_of\\_quarter](#) 241  
[day\\_of\\_week](#) 241  
[day\\_of\\_year](#) 241  
[db\\_data\\_type](#) 198  
[db\\_data\\_type\\_id](#) 199  
[db\\_data\\_type\\_name](#) 199  
[db\\_obj](#) 132  
[db\\_obj\\_defn](#) 132  
[db\\_schema\\_name](#) 126, 141, 145, 151, 153, 155, 183, 201, 234  
[db\\_table\\_name](#) 203  
[decimal value](#) 211  
[decimal\\_val](#) 211

## D

[Default Binval](#) 227, 228  
[Default Bundle](#) 228  
[Default Category Description](#) 175  
[Default Configuration Setting Value](#) 174, 177  
[Default Configuration Setting Value Id](#) 177  
[Default Date Value](#) 199  
[Default Int Value](#) 199  
[Default Long Value](#) 199  
[Default Names](#) 228  
[Default Refs](#) 228  
[Default Rule](#) 133, 141  
[Default Setting Description](#) 126, 140, 176, 182, 276  
[Default Setting Label](#) 126, 140, 176, 182, 276  
[Default String Value](#) 199  
[Definition](#) 121, 126, 133, 141, 150, 151, 152, 153, 156, 158, 159, 160, 167, 170, 174, 177, 180, 183, 189, 191, 193, 195, 209,

213, 214, 215, 216, 217, 218, 219, 226, 227, 229, 230, 243, 245, 247, 248, 252, 253, 254, 255, 258, 260, 264, 265, 266, 267, 268

Definition GUID 127, 141, 151, 153, 158, 160, 170, 178, 183, 185, 191, 195, 213, 215, 217, 219, 227, 230, 245, 249, 253, 255, 261, 265, 267, 269, 273

Definition Id 125, 126, 127, 129, 138, 139, 141, 142, 145, 146, 147, 151, 153, 154, 155, 158, 160, 169, 170, 171, 174, 178, 183, 184, 185, 188, 191, 192, 194, 195, 196, 201, 213, 215, 217, 219, 227, 230, 234, 245, 249, 253, 255, 256, 257, 260, 261, 262, 265, 267, 268, 273, 274

Definition Lock 180, 183, 233, 234

Definition Name 126, 141, 151, 153, 158, 160, 170, 178, 183, 185, 191, 195, 213, 215, 217, 219, 227, 230, 245, 249, 253, 255, 261, 265, 267, 269, 273

## d

definition parent subtype id 142, 154, 170, 196, 201, 265

## D

Definition Relationship 122, 127, 152, 154, 180, 183

Definition Relationship Id 127, 154, 183, 185, 273

Definition Relationship Type 152, 154, 180, 183

Definition Relationship Type Description 154, 184

Definition Relationship Type Id 127, 154, 183, 184, 185, 274

Definition Relationship Type Name 154, 184

Definition Subtype 133, 141, 152, 154, 168, 170, 193, 195, 201, 265

Definition Subtype Description 141, 154, 170, 196, 201, 265

Definition Subtype Id 125, 127, 129, 139, 141, 142, 146, 151, 154, 158, 160, 169, 170, 172, 174, 178, 179, 183, 185, 188, 191, 194, 195, 196, 197, 201, 202, 213, 215, 217, 220, 227, 230, 245, 249, 253, 256, 261, 265, 266, 267, 269, 273

Definition Subtype Name 141, 154, 170, 196, 201, 265

Definition Type 133, 142, 152, 154, 193, 196

Definition Type Description 142, 154, 196

Definition Type Name 127, 141, 142, 151, 154, 158, 160, 170, 178, 183, 185, 191, 195, 196, 201, 213, 215, 217, 220, 227, 230, 245, 249, 253, 255, 261, 265, 267, 269, 273

Definition UUID 226

Definition Version 122, 127, 133, 142, 152, 154, 174, 178, 180, 184, 247, 249, 255, 256, 258, 261, 262

Definition Version Association 153, 154, 181, 184

Definition Version Association Id 155, 184, 186, 274

Definition Version GUID 127, 142, 154, 178, 184, 186, 249, 256, 261, 262, 274

Definition Version Id 127, 142, 154, 178, 184, 185, 249, 256, 261, 262, 274

Definition Version Name 127, 142, 154, 178, 184, 185, 249, 256, 261, 262, 274

Definition Version Status 153, 155, 181, 184

Definition Version Status Id 127, 142, 154, 155, 178, 184, 186, 249, 256, 261, 262, 274

Definition Version Status Name 155, 184

## d

defn 121, 133, 150, 152, 156, 159, 167, 174, 180, 189, 193, 209, 214, 216, 218, 226, 229, 243, 247, 252, 254, 258, 264, 266, 268

defn\_guid 127, 141, 151, 153, 158, 160, 170, 178, 183, 185, 191, 195, 213, 215, 217, 219, 227, 230, 245, 249, 253, 255, 261, 265, 267, 269, 273

defn\_id 125, 126, 127, 129, 138, 139, 141, 142, 145, 146, 147, 151, 153, 154, 155, 158, 160, 169, 170, 171, 174, 178, 183, 184, 185, 188, 191, 192, 194, 195, 196, 201, 213, 215, 217, 219, 227, 230, 234, 245, 249, 253, 255, 256, 257, 260, 261, 262, 265, 267, 268, 273, 274

defn\_lock 180, 233

defn\_name 126, 141, 151, 153, 158, 160, 170, 178, 183, 185, 191, 195, 213, 215, 217, 219, 227, 230, 245, 249, 253, 255, 261, 265, 267, 269, 273

defn\_parent\_subtype\_id 142, 154, 170, 196, 201, 265

defn\_reln 122, 152, 180

defn\_reln\_id 127, 154, 183, 185, 273

defn\_reln\_type 152, 180

defn\_reln\_type\_descr 154, 184

defn\_reln\_type\_id 127, 154, 183, 184, 185, 274

defn\_reln\_type\_name 154, 184

defn\_subtype 133, 152, 168, 193, 201, 265

defn\_subtype\_descr 141, 154, 170, 196, 201, 265

defn\_subtype\_id 125, 127, 129, 139, 141, 142, 146, 151, 154, 158, 160, 169, 170, 172, 174, 178, 179, 183, 185, 188, 191, 194, 195, 196, 197, 201, 202, 213, 215, 217, 220, 227, 230, 245, 249, 253, 256, 261, 265, 266, 267, 269, 273

defn\_subtype\_name 141, 154, 170, 196, 201, 265

defn\_type 133, 152, 193

defn\_type\_descr 142, 154, 196

defn\_type\_name 127, 141, 142, 151, 154, 158, 160, 170, 178, 183, 185, 191, 195, 196, 201, 213, 215, 217, 220, 227, 230, 245, 249, 253, 255, 261, 265, 267, 269, 273

defn\_uuid 226

defn\_ver 122, 133, 152, 174, 180, 247, 255, 258, 262

defn\_ver\_association 153, 181

defn\_ver\_association\_id 155, 184, 186, 274

defn\_ver\_guid 127, 142, 154, 178, 184, 186, 249, 256, 261, 262, 274



defn\_ver\_id [127, 142, 154, 178, 184, 185, 249, 256, 261, 262, 274](#)  
defn\_ver\_name [127, 142, 154, 178, 184, 185, 249, 256, 261, 262, 274](#)  
defn\_ver\_status [153, 181](#)  
defn\_ver\_status\_id [127, 142, 154, 155, 178, 184, 186, 249, 256, 261, 262, 274](#)  
defn\_ver\_status\_name [155, 184](#)

## D

Delay Between Batches [197](#)

### d

delay\_between\_batches [197](#)

## D

Delete Date [125, 129, 139, 146, 157, 160, 162, 172, 179, 187, 188, 190, 195, 197, 206, 210, 212, 214, 215, 219, 221, 223, 230, 234, 235, 236, 248, 253, 255, 259, 263, 268, 270](#)

Delete Entry [276, 277](#)

Delete Entry Comment [277](#)

Delete Entry Name [277](#)

Delete Entry Number Inherited [277](#)

Delete Transaction Id [157, 191](#)

### d

delete\_date [125, 129, 139, 146, 157, 160, 162, 172, 179, 187, 188, 190, 195, 197, 206, 210, 212, 214, 215, 219, 221, 223, 230, 234, 235, 236, 248, 253, 255, 259, 263, 268, 270](#)

delete\_entry [276](#)

delete\_entry\_comment [277](#)

delete\_entry\_name [277](#)

delete\_entry\_number\_inherited [277](#)

delete\_txn\_id [157, 191](#)

## D

Deleted Column [276, 277](#)

Deleted Column Description [277](#)

Deleted Column Id [277](#)

Deleted Column Name [277](#)

Deleted Date [225](#)

Deleted On [250](#)

Deleted Table [276, 277](#)

Deleted Table Description [277](#)

Deleted Table Id [277](#)

Deleted Table Name [277](#)

### d

deleted\_col [276](#)

deleted\_col\_descr [277](#)

deleted\_col\_id [277](#)

deleted\_col\_name [277](#)

deleted\_date [225](#)

deleted\_on [250](#)

deleted\_table [276](#)

deleted\_table\_descr [277](#)

deleted\_table\_id [277](#)

deleted\_table\_name [277](#)

## D

Deployment Update [262](#)

### d

deployment\_update [262](#)

depth [256](#)

## D

Depth [256](#)

Detail Status [251](#)

### d

dflt\_binval [227](#)

dflt\_bundle [228](#)

dflt\_cfg\_setting\_val [174](#)

dflt\_cfg\_setting\_val\_id [177](#)

dflt\_ctgy\_descr [175](#)

dflt\_date\_val [199](#)

dflt\_int\_val [199](#)

dflt\_long\_val [199](#)

dflt\_names [228](#)

dflt\_refs [228](#)  
dflt\_rule [133](#)  
dflt\_setting\_descr [126, 140, 176, 182, 276](#)  
dflt\_setting\_label [126, 140, 176, 182, 276](#)  
dflt\_string\_val [199](#)

## D

Disabled Date [126, 141, 151, 153, 158, 160, 170, 178, 183, 185, 191, 195, 213, 215, 217, 219, 227, 230, 245, 249, 253, 255, 261, 265, 267, 269, 273](#)

## d

display name [205, 212, 215](#)

## D

Display Name [222](#)

## d

display\_name [205, 212, 215, 222](#)

## D

Distinguished Name [222](#)

## d

distinguished\_name [222](#)

## D

Does Count for Time [239](#)

## d

does\_ct\_for\_time [239](#)

## D

Domain Object Id [224](#)

## d

domain\_obj\_id [224](#)  
draft\_cfg\_setting\_val\_id [272](#)  
draft\_fld\_defn\_ver\_id [272](#)  
draft\_string\_val [272](#)  
dsble\_date [126, 141, 151, 153, 158, 160, 170, 178, 183, 185, 191, 195, 213, 215, 217, 219, 227, 230, 245, 249, 253, 255, 261, 265, 267, 269, 273](#)  
dtl\_status [251](#)

## D

Duration Minutes [161](#)

## d

durn\_mins [161](#)  
dyn\_col\_cstr\_spec [198](#)  
dyn\_col\_cstr\_spec\_id [199](#)  
dyn\_col\_pos [200](#)  
dyn\_col\_spec [198](#)  
dyn\_col\_spec\_id [199, 200, 272](#)  
dyn\_table\_index [199](#)  
dyn\_table\_index\_col [199](#)  
dyn\_table\_index\_id [200](#)  
dyn\_table\_script [201](#)  
dyn\_table\_script\_history [181, 201](#)  
dyn\_table\_script\_history\_id [184, 202](#)  
dyn\_table\_script\_id [184, 202](#)  
dyn\_table\_script\_name [202](#)  
dyn\_table\_script\_type [201](#)  
dyn\_table\_script\_type\_descr [202](#)  
dyn\_table\_script\_type\_id [202](#)  
dyn\_table\_script\_type\_name [202](#)  
dyn\_table\_spec [199](#)  
dyn\_table\_spec\_id [199, 200](#)

## D

Dynamic Column Constraint Specification [198, 199](#)  
Dynamic Column Constraint Specification Id [199](#)  
Dynamic Column Position [200](#)  
Dynamic Column Specification [198, 199](#)  
Dynamic Column Specification Id [199, 200](#)  
Dynamic Table Index [199](#)

Dynamic Table Index Column [199, 200](#)  
Dynamic Table Index Id [200](#)  
Dynamic Table Script [201, 202](#)  
Dynamic Table Script History [181, 184, 201, 202](#)  
Dynamic Table Script History Id [184, 202](#)  
Dynamic Table Script Id [184, 202](#)  
Dynamic Table Script Name [202](#)  
Dynamic Table Script Type [201, 202](#)  
Dynamic Table Script Type Description [202](#)  
Dynamic Table Script Type Id [202](#)  
Dynamic Table Script Type Name [202](#)  
Dynamic Table Specification [199, 200](#)  
Dynamic Table Specification Id [199, 200](#)

## E

Email Address [125, 139, 157, 160, 162, 187, 190, 195, 210, 214, 219, 221, 223, 230, 232, 234, 248, 251, 253, 255, 259, 263, 268, 270](#)  
Email Attachment URI [198](#)  
Email Notification [197](#)  
Email Notification Attachment [197, 198](#)  
Email Notification Attachment Id [198](#)  
Email Notification Description [197](#)  
Email Notification Id [197, 198](#)  
Email Notification Name [197](#)  
Email Notification Type [197, 198](#)  
Email Notification Type Id [198](#)  
Email Notification Type Name [198](#)

## e

email\_addr [125, 139, 157, 160, 162, 187, 190, 195, 210, 214, 219, 221, 223, 230, 232, 234, 248, 251, 253, 255, 259, 263, 268, 270](#)  
email\_attch\_uri [198](#)  
email\_notifn [197](#)  
email\_notifn\_attch [197](#)  
email\_notifn\_attch\_id [198](#)  
email\_notifn\_descr [197](#)  
email\_notifn\_id [197, 198](#)  
email\_notifn\_name [197](#)  
email\_notifn\_type [197](#)  
email\_notifn\_type\_id [198](#)  
email\_notifn\_type\_name [198](#)

## E

Enable Master Contact Record [124, 138, 151](#)

## e

enable\_master\_cntc\_record [124, 138, 151](#)  
end date [211](#)

## E

End Date [157, 191, 206](#)  
End Date Hour Local [207, 237, 242](#)  
End Date Hour Local S [207, 242](#)  
End Date Hour UTC [207, 242](#)  
End Date Hour UTC S [207, 242](#)  
End Date Time [160](#)  
End Date UTC S [207](#)  
End Is Daylight Savings [207, 242](#)  
End of Day Date [241](#)  
End of Hour Date [241](#)  
End of Month Date [241](#)  
End of Quarter Date [241](#)  
End of Week Date [241](#)  
End of Year Date [241](#)

## e

end\_date [157, 191, 206, 211](#)  
end\_date\_hour\_local [207, 237, 242](#)  
end\_date\_hour\_local\_s [207, 242](#)  
end\_date\_hour\_utc [207, 242](#)  
end\_date\_hour\_utc\_s [207, 242](#)  
end\_date\_time [160](#)  
end\_date\_utc\_s [207](#)  
end\_is\_daylight\_savings [207, 242](#)

end\_of\_day\_date [241](#)  
end\_of\_hour\_date [241](#)  
end\_of\_month\_date [241](#)  
end\_of\_quarter\_date [241](#)  
end\_of\_week\_date [241](#)  
end\_of\_year\_date [241](#)

## E

Entity Level [13](#), [15](#), [119](#), [130](#)

## e

error [262](#)

## E

Error [262](#)

Error Count [269](#)

Error Type Name [269](#)

## e

error\_ct [269](#)

error\_type\_name [269](#)

etag [205](#), [211](#), [212](#), [213](#), [214](#)

event [133](#)

## E

Event [133](#)

Event DateTime [238](#)

Event Workflow Process Definition [133](#), [142](#)

## e

event\_datetime [238](#)

event\_wkflw\_proc\_defn [133](#)

execn\_date [216](#), [231](#)

## E

Execution Date [216](#), [231](#)

Expiration Date [227](#)

## e

expiration\_date [227](#)

## E

External Data Source Definition [133](#)

External Data Source Definition Database Object Definition [133](#), [142](#)

External Id [215](#)

External Mapping Definition Id [257](#)

External System Certificate [247](#)

External System Certificate Id [247](#)

External System Certificate Name [247](#)

## e

extl\_data\_source\_defn [133](#)

extl\_data\_source\_defn\_db\_obj\_d [133](#)

extl\_id [215](#)

extl\_mapping\_defn\_id [257](#)

extl\_sys\_certificate [247](#)

extl\_sys\_certificate\_id [247](#)

extl\_sys\_certificate\_name [247](#)

## F

Failed Reset Count [222](#)

Failed Reset Timestamp [222](#)

## f

failed\_reset\_ct [222](#)

failed\_reset\_timestamp [222](#)

## F

Field Column Name [127](#), [142](#), [151](#), [155](#), [184](#)

Field Constraint Type [199](#)

Field Datatype [133](#), [142](#), [153](#), [155](#)

Field Datatype Id [127](#), [142](#), [151](#), [155](#), [184](#)

Field Datatype Name [142](#), [155](#)

Field Definition [122](#), [127](#), [133](#), [142](#), [150](#), [151](#), [153](#), [155](#), [181](#), [184](#)

Field Definition Id [127](#), [128](#), [142](#), [143](#), [151](#), [155](#), [163](#), [166](#), [184](#), [211](#), [217](#), [226](#)

Field Dependency Rule [133, 142](#)  
Field Description [127, 142, 151, 155, 184](#)  
Field Id [158, 191](#)  
Field Name [256](#)  
Field Type [256](#)

## f

field\_col\_name [127, 142, 151, 155, 184](#)  
field\_cstr\_type [199](#)  
field\_defn [122, 133, 150, 153, 181](#)  
field\_defn\_id [127, 128, 142, 143, 151, 155, 163, 166, 184, 211, 217, 226](#)  
field\_descr [127, 142, 151, 155, 184](#)  
field\_dpncy\_rule [133](#)  
field\_dtype [133, 153](#)  
field\_dtype\_id [127, 142, 151, 155, 184](#)  
field\_dtype\_name [142, 155](#)  
field\_externalName\_en\_US [272](#)  
field\_extlName\_en\_us [272](#)  
field\_id [158, 191](#)  
field\_name [256](#)  
field\_type [256](#)  
fieldLogicalDataType [272](#)  
fieldlogicaldtype [272](#)  
fieldrole [272](#)  
fieldRole [272](#)

## F

File Information [227](#)  
File Information Id [227](#)  
File Information Size [227](#)  
File Information URI [227](#)

## f

file\_info [227](#)  
file\_info\_id [227](#)  
file\_info\_size [227](#)  
file\_info\_uri [227](#)

## F

Finish Datetime [262](#)

## f

finish\_datetime [262](#)  
fld\_defn\_id [272](#)  
fld\_defn\_ver\_id [272](#)

## F

From Definition Version Id [127, 154, 183, 185, 274](#)  
From Definition Version ID [155, 184, 186, 274](#)  
From Item Definition Id [128, 143, 164](#)  
From Item Id [128, 143, 164](#)  
From Item Status Id [239](#)  
From Item Status Name [239](#)  
From Member Role Type Id [128, 165](#)  
From Name [197](#)  
From Version [262](#)

## f

from\_defn\_ver\_id [127, 154, 155, 183, 184, 185, 186, 274](#)  
from\_item\_defn\_id [128, 143, 164](#)  
from\_item\_id [128, 143, 164](#)  
from\_item\_status\_id [239](#)  
from\_item\_status\_name [239](#)  
from\_mbr\_role\_type\_id [128, 165](#)  
from\_name [197](#)  
from\_ver [262](#)

## G

Gets Individual Assignment for Team [190](#)

## g

gets\_individual\_asgnmt\_for\_tea [190](#)

## G

Group Name [203](#)

## g

grp\_name 203

## h

has modifications 206, 212, 215

has\_modfs 206, 212, 215

## H

History Retain Count 203

History Retain Days 203

## h

history\_retain\_ct 203

history\_retain\_days 203

## H

Hit Count 226

## h

hit\_ct 226

## H

Hop Count 239

## h

hop\_ct 239

## H

HTML Body 197

## h

html\_body 197

## i

ical data 211

ical uid 205, 212, 215

ical\_data 211

ical\_uid 205, 212, 215

id 163, 228

## I

Id Generator 203

## i

id\_generator 203

imp 229

imp\_asgnmt 229

imp\_asgnmt\_id 231

imp\_ci\_addr\_book\_id 231

imp\_ci\_rel\_from\_defn\_id 231

imp\_ci\_rel\_to\_defn\_id 231

imp\_create\_date 231

imp\_create\_user 231

imp\_date 231

imp\_defn\_id 231

imp\_defn\_rule 229

imp\_defn\_rule\_col\_index 231

imp\_defn\_rule\_defn\_id 231

imp\_defn\_rule\_id 231

imp\_defn\_rule\_status 231

imp\_defn\_rule\_text\_to\_contain 231

imp\_error\_cd 231

imp\_error\_line 231

imp\_error\_parms 231

imp\_error\_result 229

imp\_error\_result\_id 231

imp\_error\_result\_string\_inheri 231

imp\_error\_row 231

imp\_execn\_record 229

imp\_execn\_record\_id 231

imp\_ext\_db\_defn\_id 231

imp\_id 231, 232

imp\_item\_defn\_id 231

imp\_last\_exec\_date 231

imp\_record\_addr\_book\_id 231  
imp\_schdd\_task 229  
imp\_status 231, 232  
imp\_time\_zone 231  
imp\_total\_created 231  
imp\_total\_failures 231  
imp\_total\_updated 231  
imp\_type 231  
imp\_user\_sys\_level\_role 231  
imp\_user\_wksp\_role 229  
imp\_user\_wksp\_role\_id 232

|  
Impact Count 226

i  
impact\_ct 226

|  
Import 229, 230  
Import Assignment 229, 231  
Import Assignment Id 231  
Import CI Address Book Id 231  
Import CI Rel from Definition Id 231  
Import CI Rel to Definition Id 231  
Import Create Date 231  
Import Create User 231  
Import Date 231  
Import Definition ID 231  
Import Definition Rule 229, 231  
Import Definition Rule Column Index 231  
Import Definition Rule Definition Id 231  
Import Definition Rule Id 231  
Import Definition Rule Status 231  
Import Definition Rule Text To Contain 231  
Import Error Code 231  
Import Error Line 231  
Import Error Parameters 231  
Import Error Result 229, 231  
Import Error Result Id 231  
Import Error Result String Inherited 231  
Import Error Row 231  
Import Execution Record 229, 231  
Import Execution Record Id 231  
Import Ext DB Definition Id 231  
Import Id 231, 232  
Import Item Definition ID 231  
Import Last Exec Date 231  
Import Record Address Book Id 231  
Import Scheduled Task 229, 232  
Import Status 231, 232  
Import Time Zone 231  
Import Total Created 231  
Import Total Failures 231  
Import Total Updated 231  
Import Type 231  
Import User System Level Role 231  
Import User Workspace Role 229, 232  
Import User Workspace Role Id 232  
In Mobile 257  
In Quick Searches 257  
In Search Analytics 257

i  
in\_mobile 257  
in\_quick\_searches 257  
in\_search\_analytics 257  
inaccr\_ct 220

|  
Inaccurate Count 220

i  
inbd\_email 133

|  
 Inbound Email [133](#)  
 Index Name [200](#)  
 Index Type [199, 200](#)  
 Index Type Id [200](#)  
 Index Type Name [200](#)  
 |  
 |  
 index\_name [200](#)  
 index\_type [199](#)  
 index\_type\_id [200](#)  
 index\_type\_name [200](#)  
 |  
 |  
 Initial Column String Value [144, 271](#)  
 |  
 |  
 initl\_col\_string\_val [144, 271](#)  
 |  
 |  
 Insert Update Delete Code [224](#)  
 |  
 |  
 insert\_update\_delete\_cd [224](#)  
 |  
 |  
 Int Value [140, 169, 175, 177, 182, 185, 250, 251, 273](#)  
 |  
 |  
 int\_val [140, 169, 175, 177, 178, 182, 185, 250, 251, 273](#)  
 integer value [211](#)  
 integer\_val [211](#)  
 |  
 |  
 Internal Id [124, 138, 148, 205, 212, 215, 217](#)  
 Internal Identifier Type [148, 149](#)  
 Internal Identifier Type Description [149](#)  
 Internal Identifier Type Id [124, 138, 148, 149, 217](#)  
 |  
 |  
 intrnl\_id [124, 138, 148, 205, 212, 215, 217](#)  
 intrnl\_id\_type [148](#)  
 intrnl\_id\_type\_descr [149](#)  
 intrnl\_id\_type\_id [124, 138, 148, 149, 217](#)  
 |  
 |  
 Is Active [202, 235, 238](#)  
 Is Ascending Order [256](#)  
 Is Audited [223, 225](#)  
 Is Automatic [256](#)  
 Is Built In [125, 129, 139, 146, 169, 171, 174, 178, 188, 194, 197, 230](#)  
 Is Closable [264](#)  
 Is Closed [238](#)  
 Is Default [257](#)  
 Is Default Flag [213](#)  
 Is Editable [218](#)  
 Is File Deleted [227](#)  
 |  
 |  
 is floating [211](#)  
 |  
 |  
 Is Internal [217](#)  
 Is Internal Flag [124, 138, 148, 217](#)  
 Is Job Enabled [203](#)  
 Is Journal [199, 256](#)  
 Is Locked [244](#)  
 Is Longest Hop [239](#)  
 Is Original [211](#)  
 |  
 |  
 is recurring [211](#)  
 |  
 |  
 Is Resolved [238](#)



Is Responded 238  
Is Shared 257  
Is System 129, 146, 172, 179, 188, 197, 264  
Is System Administrator 196  
Is System Flag 128, 143, 149  
Is System Template 218, 266  
Is Team 157, 191  
Is Visible 264  
Is Waiting 244

## i

is\_active 202, 235, 238  
is\_ascending\_order 256  
is\_audited 223, 225  
is\_automatic 256  
is\_built\_in 125, 129, 139, 146, 169, 171, 174, 178, 188, 194, 197, 230  
is\_closable 264  
is\_closed 238  
is\_dflt 257  
is\_dflt\_flag 213  
is\_editable 218  
is\_file\_deleted 227  
is\_floating 211  
is\_imported 218

## I

Is\_Imported 218

## i

is\_intrnl 217  
is\_intrnl\_flag 124, 138, 148, 217  
is\_job\_enabled 203  
is\_journal 199, 256  
is\_locked 244  
is\_longest\_hop 239  
is\_original 211  
is\_recurring 211  
is\_resolved 238  
is\_responded 238  
is\_shared 257  
is\_sys 129, 146, 172, 179, 188, 197, 264  
is\_sys\_admin 196  
is\_sys\_flag 128, 143, 149  
is\_sys\_tmpl 218, 266  
is\_team 157, 191  
is\_vsble 264  
is\_waiting 244  
item 122, 133, 156, 159, 161, 163, 165, 189, 210, 219, 233, 253, 258, 266, 268

## I

Item 122, 127, 133, 142, 156, 158, 159, 160, 161, 162, 163, 164, 165, 166, 189, 191, 210, 213, 219, 220, 233, 234, 253, 258, 261, 266, 267, 268, 269  
Item Attachment 227  
Item Attachment Id 227  
Item Definition 123, 127, 134, 143, 153, 155, 162, 164, 166, 201, 202, 219, 220, 234, 259, 261, 276, 277  
Item Definition Id 127, 128, 143, 155, 157, 158, 160, 161, 162, 164, 166, 184, 191, 202, 210, 213, 216, 217, 220, 225, 226, 227, 234, 235, 244, 245, 250, 254, 261, 262, 267, 269, 277  
Item Field 134, 143, 157, 158, 162, 166  
Item Field Collection 166  
Item Field Definition 123, 127, 134, 143, 153, 155, 162, 166, 216, 217, 223, 226  
Item Field Definition Id 127, 143, 155, 162, 163, 166, 217, 223, 225, 226  
Item Field Journal 162, 163  
Item Id 125, 126, 127, 129, 139, 140, 142, 143, 144, 145, 147, 157, 158, 159, 160, 161, 162, 163, 164, 166, 191, 210, 213, 216, 220, 227, 234, 235, 237, 238, 239, 254, 261, 267, 269  
Item Key 127, 142, 158, 160, 162, 164, 166, 191, 213, 220, 234, 254, 261, 267, 269  
Item Lock 234  
Item Lock Id 235

## i

item name 205, 212, 215

## I

Item Name 269  
Item Relationship 123, 128, 134, 143, 164

Item Relationship Definition [123, 128, 164](#)  
Item Relationship Definition Id [128, 143, 164](#)  
Item Relationship Id [128, 143, 164](#)  
Item Status Id [239](#)  
Item Status Name [239](#)  
Item Table Name [127, 143, 155, 162, 164, 166, 202, 220, 234, 261, 277](#)  
Item Transition Id [239](#)

## I

item type [205, 211, 214](#)

## I

Item Type [198](#)  
Item type Node 1 [165](#)  
Item type Node 2 [165](#)  
Item Useful Detail [219, 220](#)  
Item Useful Detail Id [220](#)  
Item Useful Summary [219, 220](#)  
Item Useful Summary Id [220](#)  
Item Vote [134, 143](#)

## I

item\_attch [227](#)  
item\_attch\_id [227](#)  
item\_defn [123, 134, 153, 162, 164, 166, 201, 219, 234, 259, 276](#)  
item\_defn\_id [127, 128, 143, 155, 157, 158, 160, 161, 162, 164, 166, 184, 191, 202, 210, 213, 216, 217, 220, 225, 226, 227, 234, 235, 244, 245, 250, 254, 261, 262, 267, 269, 277](#)  
item\_field [134, 157, 162, 166](#)  
item\_field\_coll [166](#)  
item\_field\_defn [123, 134, 153, 162, 166, 216, 223](#)  
item\_field\_defn\_id [127, 143, 155, 162, 163, 166, 217, 223, 225, 226](#)  
item\_field\_journal [162](#)  
item\_id [125, 126, 127, 129, 139, 140, 142, 143, 144, 145, 147, 157, 158, 159, 160, 161, 162, 163, 164, 166, 191, 210, 213, 216, 220, 227, 234, 235, 237, 238, 239, 254, 261, 267, 269](#)  
item\_key [127, 142, 158, 160, 162, 164, 166, 191, 213, 220, 234, 254, 261, 267, 269](#)  
item\_lock [234](#)  
item\_lock\_id [235](#)  
item\_name [205, 212, 215, 269](#)  
item\_reln [123, 134, 164](#)  
item\_reln\_defn [123, 164](#)  
item\_reln\_defn\_id [128, 143, 164](#)  
item\_reln\_id [128, 143, 164](#)  
item\_status\_id [239](#)  
item\_status\_name [239](#)  
item\_table\_name [127, 143, 155, 162, 164, 166, 202, 220, 234, 261, 277](#)  
item\_transn\_id [239](#)  
item\_type [198, 205, 211, 214](#)  
item\_type\_node\_1 [165](#)  
item\_type\_node\_2 [165](#)  
item\_useful\_dtl [219](#)  
item\_useful\_dtl\_id [220](#)  
item\_useful\_sumry [219](#)  
item\_useful\_sumry\_id [220](#)  
item\_vote [134](#)

## J

Job Class [203](#)  
Job Configuration Detail [203](#)  
Job Configuration Detail Id [203](#)  
Job Name [203](#)  
Job Parameters [203](#)  
Job Scheduler Configuration [203](#)  
Job Scheduler Configuration Id [203, 204](#)  
Job Scheduler Default Interrupt [204](#)  
Job Scheduler default Maximum Wait [204](#)  
Job Scheduler Instance Id [204](#)

## j

job\_cfg\_dtl [203](#)  
job\_cfg\_dtl\_id [203](#)  
job\_class [203](#)  
job\_name [203](#)  
job\_parms [203](#)

job\_schedlr\_cfg [203](#)  
job\_schedlr\_cfg\_id [203](#), [204](#)  
job\_schedlr\_dflt\_interrupt [204](#)  
job\_schedlr\_dflt\_max\_wait [204](#)  
job\_schedlr\_instc\_id [204](#)

## K

Knowledge Base Container [134](#), [143](#)

## k

knowledge\_base\_cntnr [134](#)

## L

Last Accessed Time [235](#)  
Last Audit Transaction Id [244](#)  
Last Good Publish Date [183](#), [273](#)  
Last Hit Timestamp [226](#)  
Last Id Value [203](#)  
Last Login [129](#), [146](#), [172](#), [179](#), [189](#), [197](#)

## l

last modification [205](#), [212](#), [215](#)  
last modified by [205](#), [212](#), [215](#)

## L

Last Sync Date [216](#)  
Last Time Queried [260](#)

## l

last\_accessed\_time [235](#)  
last\_audit\_txn\_id [244](#)  
last\_good\_publ\_date [183](#), [273](#)  
last\_hit\_timestamp [226](#)  
last\_id\_val [203](#)  
last\_login [129](#), [146](#), [172](#), [179](#), [189](#), [197](#)  
last\_mod\_by [205](#), [212](#), [215](#)  
last\_modf [205](#), [212](#), [215](#)  
last\_sync\_date [216](#)  
last\_time\_queried [260](#)

## L

Legend [117](#), [119](#), [275](#)

## l

lic\_type [134](#), [187](#)  
lic\_type\_id [125](#), [138](#), [139](#), [143](#), [157](#), [160](#), [162](#), [187](#), [188](#), [190](#), [195](#), [205](#), [210](#), [214](#), [219](#), [221](#), [223](#), [230](#), [234](#), [248](#), [253](#), [255](#), [259](#), [263](#), [268](#), [270](#)

## L

License Type [134](#), [143](#), [187](#), [188](#)  
License Type Id [125](#), [138](#), [139](#), [143](#), [157](#), [160](#), [162](#), [187](#), [188](#), [190](#), [195](#), [205](#), [210](#), [214](#), [219](#), [221](#), [223](#), [230](#), [234](#), [248](#), [253](#), [255](#), [259](#), [263](#), [268](#), [270](#)  
Life Cycle Event [237](#), [239](#)  
Life Cycle Event Description [239](#)  
Life Cycle Event Id [238](#), [239](#)  
Life Cycle Event Name [239](#)

## l

life\_cycle\_event [237](#)  
life\_cycle\_event\_descr [239](#)  
life\_cycle\_event\_id [238](#), [239](#)  
life\_cycle\_event\_name [239](#)

## L

Limit Count [257](#)

## l

limit\_ct [257](#)

## L

Link Type Definition Version [164](#), [165](#)  
Link Type Definition Version Id [128](#), [143](#), [164](#), [165](#)

## l

link\_type\_defn\_ver [164](#)  
link\_type\_defn\_ver\_id [128](#), [143](#), [164](#), [165](#)

## L

Load End Date [244](#)  
Load Process Name [244](#)  
Load Process Suffix [244](#)  
Load Start Date [244](#)

## I

load\_end\_date [244](#)  
load\_proc\_name [244](#)  
load\_proc\_suffix [244](#)  
load\_start\_date [244](#)  
local name [211](#)  
local\_col [134, 270](#)  
local\_col\_id [144, 271](#)  
local\_col\_val [135, 271](#)  
local\_name [211](#)  
locale [123, 134, 148](#)

## L

Locale [123, 128, 134, 143, 148, 149](#)  
Locale Code [128, 143, 149](#)  
Locale Id [128, 129, 143, 145, 149](#)  
Locale Name [128, 143, 149](#)

## I

locale\_cd [128, 143, 149](#)  
locale\_id [128, 129, 143, 145, 149](#)  
locale\_name [128, 143, 149](#)

## L

Localized Column [134, 143, 270, 271](#)  
Localized Column Id [144, 271](#)  
Localized Column Value [135, 144, 271](#)  
Location Identifier [269](#)  
Location Name [269](#)  
Location Type Name [269](#)

## I

location\_id [269](#)  
location\_name [269](#)  
location\_type\_name [269](#)

## L

Locked By [183, 234, 235](#)

## I

locked\_by [183, 234, 235](#)

## L

Logical [17, 19, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92, 94, 96, 98, 100, 102, 104, 106, 109, 111, 113, 115, 147, 149, 152, 155, 159, 161, 163, 165, 166, 172, 179, 186, 189, 192, 197, 198, 200, 203, 204, 208, 213, 216, 218, 221, 222, 226, 227, 228, 233, 235, 236, 242, 245, 247, 252, 254, 257, 262, 263, 264, 266, 267, 269, 272, 274, 275](#)  
Logical Data Type [199, 200, 255, 256](#)  
Logical Data Type Id [199, 200, 256](#)  
Logical Data Type Name [200, 256](#)

## I

logical\_data\_type [199, 255](#)  
logical\_data\_type\_id [199, 200, 256, 272](#)  
logical\_data\_type\_name [200, 256, 272](#)

## L

Long Value [140, 169, 175, 177, 182, 185, 250, 273](#)

## I

long\_val [140, 169, 175, 177, 178, 182, 185, 250, 273](#)

## M

Mail Empty Report Flag [250](#)

## m

mail\_empty\_rpt\_flag [250](#)

## M

Master Contact Item [135, 144](#)  
Master Contact Item Id [126, 140, 144, 158](#)

## m

master\_cntc\_item\_id [126, 140, 144, 158](#)  
max\_date\_val [128, 144, 206](#)  
max\_in\_parent [142, 154, 171, 196, 202, 266](#)

## M

Maximum Date Value [128, 144, 206](#)  
Maximum In Parent [142, 154, 171, 196, 202, 266](#)

## m

mdfy\_date [205, 211, 212, 213, 214](#)

## M

Message [260, 269](#)  
Message Template Definition Id [260](#)  
Message Text [260](#)  
Messages Per Batch [197](#)  
Meta Id [257](#)

## m

meta\_id [257](#)

## M

Metric Definition [135](#)  
Metric Threshold [135](#)

## m

metric\_defn [135](#)  
metric\_threshold [135](#)  
mod\_by [249](#)  
mod\_on [249](#)

## M

Modified By [249](#)  
Modified On [249](#)  
Modifies Calendar Item Id [206, 212, 215](#)

## m

modifies\_cal\_item\_id [206, 212, 215](#)  
modify date [205, 211, 212, 214](#)

## M

Modify Date [213](#)

## m

msg [260, 269](#)  
msg\_text [260](#)  
msg\_tmpl\_defn\_id [260](#)  
msgs\_per\_batch [197](#)

## n

name [228](#)  
namespace [211](#)

## N

Needs Multiple Plural Forms Flag [125, 138, 148](#)

## n

needs reply [205, 212, 215](#)  
needs\_mpl\_plural\_forms\_flag [125, 138, 148](#)  
needs\_reply [205, 212, 215](#)

## N

New Date Value [224](#)  
New Long Value [224](#)  
New Real Value [224](#)  
New String Value [224](#)  
New Text Value [224](#)

## n

new\_date\_val [224](#)  
new\_long\_val [224](#)  
new\_real\_val [224](#)

new\_string\_val 224  
new\_text\_val 224

## N

Next Scheduled Run Date 249

## n

next\_schdd\_run\_date 249  
nmsp 211

## N

No Count 220

## n

no\_ct 220

## N

Node 1 Name 165  
Node 2 Name 165  
Node Id Hi 228  
Node Id Lo 228

## n

node\_1\_name 165  
node\_2\_name 165  
node\_id\_hi 228  
node\_id\_lo 228  
notes 257

## N

Notes 257  
Notification 135  
Notification Recipient Status 197, 198  
Notification Status Code 198

## n

notifn 135  
notifn\_recipient\_status 197  
notifn\_status\_code 198

## O

Occurence UID 211

## O

occurence\_uid 211

## O

Old Date Value 225  
Old Long Value 225  
Old Real Value 225  
Old String Value 225  
Old Text Value 225

## O

old\_date\_val 225  
old\_long\_val 225  
old\_real\_val 225  
old\_string\_val 225  
old\_text\_val 225  
op 135  
op\_visibility 135

## O

Operation 135  
Operation Visibility 135  
Order Index 176

## O

order\_index 176  
orgal\_field 124, 138, 151

## O

Organizational Field 124, 138, 151

## O

outbd\_email 135

## O

Outbound Email [135](#)

Owner Id [140](#), [169](#), [176](#), [177](#), [182](#), [185](#), [273](#)

## O

owner\_id [140](#), [169](#), [176](#), [177](#), [178](#), [182](#), [185](#), [273](#)

## O

Owning Process Id [244](#)

## O

owning\_proc\_id [244](#)

## p

parent [256](#)

## P

Parent [256](#)

Parent Name [257](#)

Parent Widget Id [264](#)

## p

parent\_name [257](#)

parent\_widget\_id [264](#)

## P

Part Of Speech Id [125](#), [138](#), [148](#)

## p

part\_of\_speech\_id [125](#), [138](#), [148](#)

## P

Password [125](#), [139](#), [157](#), [160](#), [162](#), [187](#), [190](#), [195](#), [210](#), [214](#), [219](#), [221](#), [223](#), [230](#), [234](#), [248](#), [253](#), [255](#), [259](#), [263](#), [268](#), [270](#)

Password Reset Question [221](#)

Password Reset Question Answer [221](#)

Password Reset Question Answer Id [221](#)

Password Reset Question Id [221](#)

Password Reset Registered Account [221](#)

Password Reset Registered Account Id [221](#), [222](#)

Password Reset Registered Account Name [222](#)

## p

pend\_adv\_n\_actn [267](#)

pend\_adv\_n\_actn\_adv\_n\_actn\_id [267](#)

pend\_adv\_n\_actn\_asset\_id [267](#)

pend\_adv\_n\_actn\_defn\_id [267](#)

pend\_adv\_n\_actn\_dyn\_item\_id [267](#)

pend\_adv\_n\_actn\_id [267](#)

## P

Pending Advanced Action [267](#)

Pending Advanced Action Advanced Action Id [267](#)

Pending Advanced Action Asset Id [267](#)

Pending Advanced Action Definition Id [267](#)

Pending Advanced Action Dynamic Item Id [267](#)

Pending Advanced Action Id [267](#)

Physical [18](#), [21](#), [23](#), [25](#), [27](#), [29](#), [31](#), [33](#), [35](#), [37](#), [39](#), [41](#), [43](#), [45](#), [47](#), [49](#), [51](#), [53](#), [55](#), [57](#), [59](#), [61](#), [63](#), [65](#), [67](#), [69](#), [71](#), [73](#), [75](#), [77](#), [79](#), [81](#), [83](#), [85](#), [87](#), [89](#), [91](#), [93](#), [95](#), [97](#), [99](#), [101](#), [103](#), [105](#), [107](#), [108](#), [110](#), [112](#), [114](#), [116](#), [147](#), [152](#), [155](#), [159](#), [161](#), [163](#), [165](#), [166](#), [172](#), [179](#), [186](#), [189](#), [192](#), [197](#), [198](#), [200](#), [203](#), [204](#), [208](#), [213](#), [216](#), [218](#), [221](#), [222](#), [226](#), [227](#), [228](#), [233](#), [235](#), [236](#), [242](#), [245](#), [247](#), [252](#), [254](#), [257](#), [262](#), [263](#), [264](#), [266](#), [267](#), [269](#), [270](#), [272](#), [274](#), [275](#)

PIM Appointment Information [214](#), [215](#)

PIM Appointment Information Id [215](#)

PIM Record Information [214](#), [215](#)

PIM Record Information Id [215](#)

## p

pim\_appt\_info [214](#)

pim\_appt\_info\_id [215](#)

pim\_record\_info [214](#)

pim\_record\_info\_id [215](#)

## P

Plain Text Auto [197](#)

Plain Text Body [197](#)

## p

plain\_text\_auto [197](#)  
plain\_text\_body [197](#)

## P

Policy Content [235](#)  
Policy Id [235](#)

## p

policy\_content [235](#)  
policy\_id [235](#)  
portal [263](#)

## P

Portal [263](#)  
Portal Active Console Id [264](#)  
Portal Console [263](#), [264](#)  
Portal Description [264](#)  
Portal GUID [264](#)  
Portal Id [264](#)  
Portal Logo GUID [264](#)  
Portal Name [264](#)  
Portal Theme [264](#)

## p

portal\_active\_console\_id [264](#)  
portal\_console [263](#)  
portal\_descr [264](#)  
portal\_guid [264](#)  
portal\_id [264](#)  
portal\_logo\_guid [264](#)  
portal\_name [264](#)  
portal\_theme [264](#)  
predicate [256](#)

## P

Predicate [256](#)  
Presentation Mode [135](#)

## p

prev\_day [239](#)  
prev\_day\_of\_month [240](#)  
prev\_day\_of\_quarter [240](#)  
prev\_day\_of\_week [240](#)  
prev\_day\_of\_year [240](#)  
prev\_month [240](#)  
prev\_month\_of\_quarter [240](#)  
prev\_month\_of\_year [240](#)  
prev\_quarter [240](#)  
prev\_quarter\_of\_year [240](#)  
prev\_timing [237](#)  
prev\_week [240](#)  
prev\_week\_of\_month [240](#)  
prev\_week\_of\_quarter [240](#)  
prev\_week\_of\_year [240](#)  
prev\_year [240](#)

## P

Previous Day [239](#)  
Previous Day of Month [240](#)  
Previous Day of Quarter [240](#)  
Previous Day of Week [240](#)  
Previous Day of Year [240](#)  
Previous Month [240](#)  
Previous Month of Quarter [240](#)  
Previous Month of Year [240](#)  
Previous Quarter [240](#)  
Previous Quarter of Year [240](#)  
Previous Timing [237](#), [239](#)  
Previous Week [240](#)  
Previous Week of Month [240](#)  
Previous Week of Quarter [240](#)  
Previous Week of Year [240](#)



Previous Year [240](#)  
Primary Key [124, 138, 151](#)  
Primary Key Id [144, 271](#)

**p**  
priority [197](#)

**P**  
Priority [197](#)

**p**  
privilege [213, 271](#)

**P**  
Privilege [213, 271](#)

**p**  
prmry\_key [124, 138, 151](#)  
prmry\_key\_id [144, 271](#)  
prob\_item [135, 138](#)  
prod\_rlse\_id [129, 145, 149](#)  
prod\_ver [135, 263](#)

**P**  
Product Release Id [129, 145, 149](#)  
Product Version [135, 263](#)

**p**  
prop\_class\_name [225](#)  
prop\_mbr\_name [225](#)

**P**  
Property Class Name [225](#)  
Property Member Name [225](#)  
Protocol Id [198](#)

**p**  
protocol\_id [198](#)  
prsntn\_mode [135](#)  
publ\_date [269](#)  
publ\_error [268](#)  
publ\_error\_id [269](#)  
publ\_log [268](#)  
publ\_log\_id [269](#)

**P**  
Publish Date [269](#)  
Publish Error [268, 269](#)  
Publish Error Id [269](#)  
Publish Log [268, 269](#)  
Publish Log Id [269](#)

**p**  
pwd [125, 139, 157, 160, 162, 187, 190, 195, 210, 214, 219, 221, 223, 230, 234, 248, 253, 255, 259, 263, 268, 270](#)  
pwd\_reset\_question [221](#)  
pwd\_reset\_question\_answer [221](#)  
pwd\_reset\_question\_answer\_id [221](#)  
pwd\_reset\_question\_id [221](#)  
pwd\_reset\_regtd\_acct [221](#)  
pwd\_reset\_regtd\_acct\_id [221, 222](#)  
pwd\_reset\_regtd\_acct\_name [222](#)

**Q**  
Query Type [257](#)

**q**  
query\_type [257](#)  
question [221](#)

**Q**  
Question [221](#)

**R**  
Rate Description [161](#)  
Rate Per Hour [161](#)

## **r**

rate\_descr [161](#)  
rate\_per\_hour [161](#)

## **R**

Raw Message [224](#)

## **r**

raw\_msg [224](#)

## **R**

Read Only CMDB Role [168](#), [171](#)

## **r**

read\_only [168](#)

## **R**

Real Value [140](#), [169](#), [175](#), [177](#), [182](#), [185](#), [250](#), [273](#)

## **r**

real\_val [140](#), [169](#), [175](#), [177](#), [178](#), [182](#), [185](#), [250](#), [273](#)

## **R**

Recipient Contact [198](#)  
Recipient Id [259](#)  
Recipient Status Id [198](#)

## **r**

recipient\_contact [198](#)  
recipient\_id [259](#)  
recipient\_status\_id [198](#)

## **R**

Recur Days of Month [232](#), [251](#)  
Recur Days of Week [232](#), [251](#)  
Recur Hour [232](#), [251](#)  
Recur Minute [232](#), [251](#)  
Recur Period Type [247](#), [249](#)  
Recur Period Type Id [232](#), [249](#), [251](#)  
Recur Period Type Name [249](#)  
Recur Schedule Type [247](#), [249](#)  
Recur Schedule Type Id [232](#), [249](#), [251](#)  
Recur Schedule Type Name [249](#)

## **r**

recur\_days\_of\_month [232](#), [251](#)  
recur\_days\_of\_week [232](#), [251](#)  
recur\_hour [232](#), [251](#)  
recur\_minute [232](#), [251](#)  
recur\_period\_type [247](#)  
recur\_period\_type\_id [232](#), [249](#), [251](#)  
recur\_period\_type\_name [249](#)  
recur\_schd\_type [247](#)  
recur\_schd\_type\_id [232](#), [249](#), [251](#)  
recur\_schd\_type\_name [249](#)

## **R**

Reference Id [199](#), [200](#)

## **r**

reference\_id [199](#), [200](#)

## **R**

Refs Data [228](#)

## **r**

refs\_data [228](#)

## **R**

Registered Email [222](#)

## **r**

regtd\_email [222](#)

## **R**

Rejected Count [232](#)

## r

rejected\_ct [232](#)

## R

Relation Type [210](#)

Relationship Type Id [128](#), [165](#)

## r

reln\_type [210](#)

reln\_type\_id [128](#), [165](#)

## R

Renoir Domain Model Detail [118](#)

Repeat Count [203](#)

## r

repeat\_ct [203](#)

## R

Reply Address [197](#)

## r

reply\_addr [197](#)

## R

Report [248](#), [249](#)

Report Comment [249](#)

Report Configuration [248](#), [250](#)

Report Configuration Id [250](#)

Report Configuration Name [250](#)

Report Description [250](#)

Report Folder Name [250](#)

Report Format [248](#), [250](#)

Report Format Description [250](#)

Report Format Name [250](#)

Report Group Id [250](#)

Report Id [249](#), [250](#)

Report Name [249](#)

Report Scheduled Task [248](#), [250](#)

Report Title [250](#)

Report Type Id [250](#)

Report Visibility Type [248](#), [250](#)

Report Visibility Type Description [250](#)

Report Visibility Type Id [249](#), [250](#)

Report Visibility Type Name [250](#)

Request Begin Date [244](#)

Request Begin Date UTC String [244](#)

Request Date [245](#)

Request Date UTC String [245](#)

Request End Date [245](#)

Request End Date UTC String [245](#)

Request For Change Item [135](#), [144](#)

Resolved Count [238](#)

## r

resolved\_ct [238](#)

## R

Retries Ct [198](#)

## r

retries\_ct [198](#)

## R

Role Group [168](#), [171](#), [193](#), [196](#), [271](#)

Role Group Description [171](#), [196](#), [271](#)

Role Group Id [129](#), [146](#), [171](#), [178](#), [188](#), [196](#), [271](#), [272](#)

Role Group Name [171](#), [196](#), [271](#)

Role Group Role [168](#), [171](#), [271](#)

Role Group Role Id [171](#), [272](#)

## r

role\_grp [168](#), [193](#), [271](#)

role\_grp\_descr [171](#), [196](#), [271](#)

role\_grp\_id 129, 146, 171, 178, 188, 196, 271, 272  
role\_grp\_name 171, 196, 271  
role\_grp\_role 168, 271  
role\_grp\_role\_id 171, 272  
rpt 248  
rpt\_cfg 248  
rpt\_cfg\_id 250  
rpt\_cfg\_name 250  
rpt\_comment 249  
rpt\_descr 250  
rpt\_folder\_name 250  
rpt\_format 248  
rpt\_format\_descr 250  
rpt\_format\_name 250  
rpt\_grp\_id 250  
rpt\_id 249, 250  
rpt\_name 249  
rpt\_schdd\_task 248  
rpt\_title 250  
rpt\_type\_id 250  
rpt\_visibility\_type 248  
rpt\_visibility\_type\_descr 250  
rpt\_visibility\_type\_id 249, 250  
rpt\_visibility\_type\_name 250  
rqst\_begin\_date 244  
rqst\_begin\_date\_utc\_string 244  
rqst\_date 245  
rqst\_date\_utc\_string 245  
rqst\_end\_date 245  
rqst\_end\_date\_utc\_string 245  
rqst\_for\_chg\_item 135

## R

Rule Name 225  
Rule Usage Statistic 226  
Rule Usage Statistic Id 226

## r

rule\_name 225  
rule\_usage\_stat\_id 226  
rule\_usage\_statistic 226

## R

Run As Proxy 202

## r

run\_as\_proxy 202

## S

schd 123, 135, 204  
schd\_actual\_period 204  
schd\_actual\_period\_id 206  
schd\_id 128, 144, 206, 212  
schd\_name 128, 144, 206  
schd\_task\_run\_stat 248  
schdd\_date 197, 198  
schdd\_run\_date 251  
schdd\_task 229, 248  
schdd\_task\_descr 232, 251  
schdd\_task\_id 232, 250, 251  
schdd\_task\_name 232, 251  
schdd\_task\_notifn 229, 248  
schdd\_task\_notifn\_id 232, 251  
schdd\_task\_run 248  
schdd\_task\_run\_id 250, 251  
schdd\_task\_stat 248  
schdd\_task\_stat\_id 251  
schdd\_task\_stat\_name 252  
schdd\_task\_status 248  
schdd\_task\_status\_id 251, 252  
schdd\_task\_status\_name 252  
schdd\_task\_type 248  
schdd\_task\_type\_id 232, 251, 252

schdd\_task\_type\_name 252

## S

Schedule 123, 128, 135, 144, 204, 206

Schedule Actual Period 204, 206

Schedule Actual Period Id 206

Schedule Id 128, 144, 206, 212

Schedule Name 128, 144, 206

Schedule Task Run Statistic 248, 250

Scheduled Date 197, 198

Scheduled Run Date 251

Scheduled Task 229, 232, 248, 251

Scheduled Task Description 232, 251

Scheduled Task Id 232, 250, 251

Scheduled Task Name 232, 251

Scheduled Task Notification 229, 232, 248, 251

Scheduled Task Notification Id 232, 251

Scheduled Task Run 248, 251

Scheduled Task Run Id 250, 251

Scheduled Task Statistic 248, 251

Scheduled Task Statistic Id 251

Scheduled Task Statistic Name 252

Scheduled Task Status 248, 252

Scheduled Task Status Id 251, 252

Scheduled Task Status Name 252

Scheduled Task Type 248, 252

Scheduled Task Type Id 232, 251, 252

Scheduled Task Type Name 252

Schema Name 144, 200, 257, 271

## S

schema\_name 144, 200, 257, 271, 272

## S

Script Sort Order 202

Script Type Id 202

## S

script\_sort\_order 202

script\_type\_id 202

## S

Search Column 255, 256

Search Column Id 256

Search Column Name 256

Search Condition 255, 256

Search Condition Code 256

Search Condition Id 256

Search Constraint Type 255, 256

Search Constraint Type Description 256

Search Constraint Type Id 256

Search Constraint Type Name 256

Search In All 257

Search Item Query 255, 256

Search Item Query Id 256, 257

Search Item Query Name 257

Search Query 255, 257

Search Query Description 257

Search Query Id 256, 257

Search Query Name 257

Search Source 257

Search View 255, 257

Search View Description 257

Search View Id 256, 257

Search View Name 257

## S

search\_col 255

search\_col\_id 256

search\_col\_name 256

search\_cond 255

search\_cond\_cd 256

search\_cond\_id 256

search\_cstr\_type 255, 256

search\_cstr\_type\_descr [256](#)  
search\_cstr\_type\_id [256](#)  
search\_cstr\_type\_name [256](#)  
search\_in\_all [257](#)  
search\_item\_query [255](#)  
search\_item\_query\_id [256](#), [257](#)  
search\_item\_query\_name [257](#)  
search\_query [255](#)  
search\_query\_descr [257](#)  
search\_query\_id [256](#), [257](#)  
search\_query\_name [257](#)  
search\_source [257](#)  
search\_view [255](#)  
search\_view\_descr [257](#)  
search\_view\_id [256](#), [257](#)  
search\_view\_name [257](#)

## S

Second Previous Day [240](#)  
Second Previous Day of Month [240](#)  
Second Previous Day of Quarter [240](#)  
Second Previous Day of Week [240](#)  
Second Previous Day of Year [240](#)  
Second Previous Month [240](#)  
Second Previous Month of Quarter [240](#)  
Second Previous Month of Year [240](#)  
Second Previous Quarter [240](#)  
Second Previous Quarter of Year [241](#)  
Second Previous Week [240](#)  
Second Previous Week of Month [240](#)  
Second Previous Week of Quarter [240](#)  
Second Previous Week of Year [240](#)  
Second Previous Year [241](#)

## S

second\_prev\_day [240](#)  
second\_prev\_day\_of\_month [240](#)  
second\_prev\_day\_of\_quarter [240](#)  
second\_prev\_day\_of\_week [240](#)  
second\_prev\_day\_of\_year [240](#)  
second\_prev\_month [240](#)  
second\_prev\_month\_of\_quarter [240](#)  
second\_prev\_month\_of\_year [240](#)  
second\_prev\_quarter [240](#)  
second\_prev\_quarter\_of\_year [241](#)  
second\_prev\_week [240](#)  
second\_prev\_week\_of\_month [240](#)  
second\_prev\_week\_of\_quarter [240](#)  
second\_prev\_week\_of\_year [240](#)  
second\_prev\_year [241](#)

## S

Security Binval [228](#)  
Security Bundle [228](#)  
Security Names [228](#)  
Security Permission [135](#)  
Security Policy [235](#)  
Security Policy Id [235](#)  
Security Refs [228](#)  
Security Right [135](#), [144](#)

## S

secuty\_binval [228](#)  
secuty\_bundle [228](#)  
secuty\_names [228](#)  
secuty\_permnsn [135](#)  
secuty\_policy [235](#)  
secuty\_policy\_id [235](#)  
secuty\_refs [228](#)  
secuty\_right [135](#)

## S

Selected Field [198](#)

## S

selected\_field 198  
sent 205, 212, 215  
sequence 256

## S

Sequence 256  
Service Host Name 204  
Service Item 135, 144  
Service Level Agreement 135  
Service Level Target Item 135, 144  
Service Port 204  
Service Portfolio Container 135, 144  
Session Token 235  
Session Type 235

## S

session\_token 235  
session\_type 235

## S

Shared Customer 187, 188  
Shared Customer Profile 193, 196

## S

shared\_cust 187  
shared\_cust\_prof 193

## S

Show In Menu 257

## S

show\_in\_menu 257

## S

Snapshot Configuration Setting Value 181, 184, 272, 273  
Snapshot Definition 181, 185, 272, 273  
Snapshot Definition Relationship 181, 185, 272, 273  
Snapshot Definition Version 181, 185, 272, 274  
Snapshot Definition Version Assoc 181, 186, 272, 274  
Snapshot Status 181, 186, 272, 274  
Snapshot Status Id 186, 274  
Snapshot Status Name 183, 186, 273, 274  
Snapshot Type 181, 186, 272, 274  
Snapshot Type Id 186, 274  
Snapshot Type Name 186, 274

## S

snapshot\_cfg\_setting\_val 181, 272  
snapshot\_defn 181, 272  
snapshot\_defn\_reln 181, 272  
snapshot\_defn\_ver 181, 272  
snapshot\_defn\_ver\_assoc 181, 272  
snapshot\_status 181, 272  
snapshot\_status\_id 186, 274  
snapshot\_status\_name 183, 186, 273, 274  
snapshot\_type 181, 272  
snapshot\_type\_id 186, 274  
snapshot\_type\_name 186, 274

## S

Soft Delete Id 163

## S

soft\_delete\_id 163  
soln\_item 135, 219

## S

Solution Item 135, 144, 219, 220  
Sort Order 157, 191

## S

sort\_order 157, 191

## S

Source Console Id [263](#)

## S

source\_console\_id [263](#)

spec\_class\_name [126, 140, 176, 182, 277](#)

## S

Specification Class Name [126, 140, 176, 182, 277](#)

SQL Script [202](#)

## S

sql\_script [202](#)

stamp type [212](#)

stamp\_type [212](#)

## S

Standard Agent Profile [193, 196](#)

## S

standard\_agent\_prof [193](#)

start date [211](#)

## S

Start Date Time [160](#)

Start Datetime [262](#)

Start From Position [257](#)

Start Mail Sent [198](#)

Start of Month Date [241](#)

Start of Quarter Date [241](#)

Start of Week Date [241](#)

Start of Year Date [241](#)

## S

start\_date [211](#)

start\_date\_time [160](#)

start\_datetime [262](#)

start\_from\_pos [257](#)

start\_mail\_sent [198](#)

start\_of\_month\_date [241](#)

start\_of\_quarter\_date [241](#)

start\_of\_week\_date [241](#)

start\_of\_year\_date [241](#)

## S

State Key [270](#)

State Value [270](#)

## S

state\_key [270](#)

state\_val [270](#)

status [200, 269](#)

## S

Status [200, 269](#)

Status Change Date [251](#)

## S

status\_chg\_date [251](#)

## S

String Parameter 1 [224](#)

String Parameter 2 [224](#)

String Parameter 3 [224](#)

String Parameter 4 [224](#)

String Parameter 5 [224](#)

String Parameter 6 [224](#)

String Parameter 7 [224](#)

String Parameter 7 Int [224](#)

String Parameter 8 [224](#)

String Resource Value [123, 128, 135, 145, 148, 149](#)

String Resource Value Id [128, 145, 149](#)

String Usage Type [148, 149](#)

String Usage Type Id [125, 138, 148, 149](#)

String Usage Type Name [149](#)



## S

string value [211](#)

## S

String Value [128](#), [140](#), [145](#), [149](#), [163](#), [166](#), [169](#), [175](#), [177](#), [182](#), [185](#), [250](#), [273](#)

## S

string\_parm\_1 [224](#)

string\_parm\_2 [224](#)

string\_parm\_3 [224](#)

string\_parm\_4 [224](#)

string\_parm\_5 [224](#)

string\_parm\_6 [224](#)

string\_parm\_7 [224](#)

string\_parm\_7\_int [224](#)

string\_parm\_8 [224](#)

string\_rsrc\_val [123](#), [135](#), [148](#)

string\_rsrc\_val\_id [128](#), [145](#), [149](#)

string\_usage\_type [148](#)

string\_usage\_type\_id [125](#), [138](#), [148](#), [149](#)

string\_usage\_type\_name [149](#)

string\_val [128](#), [140](#), [145](#), [149](#), [163](#), [166](#), [169](#), [175](#), [177](#), [178](#), [182](#), [185](#), [211](#), [250](#), [273](#)

subject [197](#)

## S

Subject [197](#)

## S

subtype [276](#)

## S

Subtype [276](#)

## S

success [262](#)

## S

Success [262](#)

## S

supertype [276](#)

## S

Supertype [276](#)

Survey Item [135](#), [145](#)

Survey Sent [262](#)

Survey Sent Email [262](#)

Survey Sent Id [262](#)

## S

survey\_item [135](#)

survey\_sent [262](#)

survey\_sent\_email [262](#)

survey\_sent\_id [262](#)

svc\_ctlg\_cntrnr [135](#)

svc\_host\_name [204](#)

svc\_level\_agrmt [135](#)

svc\_port [204](#)

svc\_rqst [135](#)

## S

Symmetric Link [165](#)

## S

symmetric\_link [165](#)

## S

Syncing Job Detail [214](#), [216](#)

Syncing Job Detail Id [216](#)

## S

syncing\_job\_dtl [214](#)

syncing\_job\_dtl\_id [216](#)

sys\_admin\_prof [194](#)

sys\_admin\_role [168](#)

sys\_component 223  
sys\_component\_descr 226  
sys\_component\_id 225, 226, 244, 275  
sys\_component\_name 226  
sys\_env 135  
sys\_env\_id 129, 145  
sys\_env\_name 145  
sys\_log 136  
sys\_mfgr 136  
sys\_name 263  
sys\_prvdr 136  
sys\_session 234  
sys\_session\_id 235

## S

System Administration Profile 194, 196  
System Administrator Role 168, 171  
System Component 223, 226  
System Component Description 226  
System Component Id 225, 226, 244, 275  
System Component Name 226  
System Environment 135, 145  
System Environment Id 129, 145  
System Environment Name 145  
System Log 136, 145  
System Manufacturer 136  
System Name 263  
System Provider 136  
System Session 234, 235  
System Session Id 235

## t

tab 257

## T

Tab 257  
Table Level 14, 16, 119, 130  
Table Name 144, 200, 271

## t

table\_name 144, 200, 271, 272

## T

Target DBMS 199, 200, 201, 202  
Target DBMS Id 199, 200, 202  
Target DBMS Name 200, 203  
Target Match 235  
Target Match Id 235  
Target Match Type 235, 236  
Target Match Type Description 236  
Target Match Type Name 235, 236

## t

target\_dbms 199, 201  
target\_dbms\_id 199, 200, 202  
target\_dbms\_name 200, 203  
target\_match 235  
target\_match\_id 235  
target\_match\_type 235  
target\_match\_type\_descr 236  
target\_match\_type\_name 235, 236

## T

TD Day 241  
TD Hour 241  
TD Month 241  
TD Quarter 241  
TD Week 241  
TD Year 241

## t

td\_day 241  
td\_hour 241  
td\_month 241

td\_quarter [241](#)  
td\_week [241](#)  
td\_year [241](#)  
team [157](#), [190](#), [230](#)

## T

Team [157](#), [158](#), [190](#), [191](#), [230](#), [232](#)  
Team Email Notification Type [190](#), [192](#)  
Team Email Notification Type Description [192](#)  
Team Email Notification Type Id [158](#), [192](#), [233](#)  
Team Email Notification Type Name [192](#)  
Team Id [157](#), [158](#), [190](#), [191](#), [192](#), [231](#), [232](#), [251](#)  
Team Item Association [190](#), [192](#)  
Team Lead Application User Id [158](#), [192](#), [233](#)  
Team Name [158](#), [192](#), [233](#)

## t

team\_email\_notifn\_type [190](#)  
team\_email\_notifn\_type\_descr [192](#)  
team\_email\_notifn\_type\_id [158](#), [192](#), [233](#)  
team\_email\_notifn\_type\_name [192](#)  
team\_id [157](#), [158](#), [190](#), [191](#), [192](#), [231](#), [232](#), [251](#)  
team\_item\_association [190](#)  
team\_lead\_app\_user\_id [158](#), [192](#), [233](#)  
team\_name [158](#), [192](#), [233](#)

## T

Template [123](#), [129](#), [136](#), [145](#)  
Template Application Role [168](#), [171](#)  
Template Application Role Id [125](#), [138](#), [139](#), [169](#), [174](#), [194](#), [230](#)  
Template Attribute Definition [216](#), [217](#)  
Template Attribute Definition Id [217](#), [218](#)  
Template Attribute Definition Keyword [217](#)  
Template Attribute Definition Name [217](#)  
Template Category [216](#), [217](#)  
Template Category Description [217](#)  
Template Category Id [217](#), [218](#)  
Template Category Name [217](#)  
Template Content [218](#)  
Template Content Type [216](#), [217](#)  
Template Content Type Description [218](#)  
Template Content Type Id [218](#)  
Template Content Type Name [218](#)  
Template Definition [216](#), [218](#)  
Template Definition Attribute [216](#), [218](#)  
Template Definition Id [218](#)  
Template Definition Type [218](#)  
Template Export GUID [127](#), [141](#), [151](#), [154](#), [158](#), [160](#), [170](#), [178](#), [183](#), [185](#), [191](#), [195](#), [213](#), [215](#), [217](#), [220](#), [227](#), [230](#), [245](#), [249](#), [253](#), [256](#), [261](#), [265](#), [267](#), [269](#), [273](#)  
Template Group [217](#), [218](#)  
Template Group Description [218](#)  
Template Group Id [218](#)  
Template Group Name [218](#)  
Template Package [265](#), [266](#)  
Template Package Content [266](#)  
Template Package Id [266](#)  
Template Package Internal Name [266](#)  
Template Package Name [266](#)  
Template Package Type [265](#), [266](#)  
Template Package Type Id [266](#)  
Template Package Type Name [266](#)  
Template User Profile Id [129](#), [146](#), [171](#), [178](#), [188](#), [196](#)

## t

tenant [136](#)

## T

Tenant [136](#)  
Tenant Partition [123](#), [129](#), [136](#), [145](#)  
Tenant Partition Definition [136](#), [145](#), [153](#), [155](#)  
Tenant Partition Id [126](#), [129](#), [141](#), [145](#), [151](#), [182](#)

## t

tenant\_parttn [123](#), [136](#)

tenant\_parttn\_defn 136, 153  
tenant\_parttn\_id 126, 129, 141, 145, 151, 182  
text value 211

## T

Text Value 140, 169, 175, 177, 182, 185, 250, 273

## t

text\_logical\_data\_type\_id 272

## T

TEXT\_logical\_data\_type\_id 272

## t

text\_val 140, 169, 175, 177, 178, 182, 185, 211, 250, 273

## T

Ticket Alert 259, 261

Ticket Alert Id 259, 261

Ticket Item 123, 129, 136, 145, 157, 158, 162, 163

## t

ticket key 212

## T

Ticket Key 213

Ticket Timeout 213

## t

ticket\_alert 259

ticket\_alert\_id 259, 261

ticket\_key 212, 213

ticket\_timeout 213

## T

Time Before Break 197

Time Dimension 237, 241

Time Tracking 160

Time Tracking Description 160

Time Tracking Id 160

Time Tracking Type 160, 161

Time Tracking Type Description 161

Time Tracking Type Id 160, 161

Time Tracking Type Name 161

Time Zone 203, 232

Time Zone Local Hour 205, 207, 237, 241

Time Zone Local Hour Id 207, 242

Time Zone Region 205, 207, 237, 242

Time Zone Region Abbreviation 207, 242

Time Zone Region Id 207, 242

Time Zone Region Name 128, 129, 144, 147, 151, 186, 206, 207, 208, 242, 261

## t

time\_before\_break 197

time\_dim 237

time\_tracking 160

time\_tracking\_descr 160

time\_tracking\_id 160

time\_tracking\_type 160

time\_tracking\_type\_descr 161

time\_tracking\_type\_id 160, 161

time\_tracking\_type\_name 161

time\_zone 203, 232

time\_zone\_local\_hour 205, 237

time\_zone\_local\_hour\_id 207, 242

time\_zone\_region 205, 237

time\_zone\_region\_abbr 207, 242

time\_zone\_region\_id 207, 242

time\_zone\_region\_name 128, 129, 144, 147, 151, 186, 206, 207, 208, 242, 261

tmp\_cfg\_setting\_val 174, 178

tmpl 123, 136

tmpl\_app\_role 168

tmpl\_app\_role\_id 125, 138, 139, 169, 174, 194, 230

tmpl\_attr\_defn 216

tmpl\_attr\_defn\_id 217, 218

[tmpl\\_attr\\_defn\\_keyword](#) 217  
[tmpl\\_attr\\_defn\\_name](#) 217  
[tmpl\\_content](#) 218  
[tmpl\\_content\\_type](#) 216  
[tmpl\\_content\\_type\\_descr](#) 218  
[tmpl\\_content\\_type\\_id](#) 218  
[tmpl\\_content\\_type\\_name](#) 218  
[tmpl\\_ctgy](#) 216  
[tmpl\\_ctgy\\_descr](#) 217  
[tmpl\\_ctgy\\_id](#) 217, 218  
[tmpl\\_ctgy\\_name](#) 217  
[tmpl\\_defn](#) 216  
[tmpl\\_defn\\_attr](#) 216  
[tmpl\\_defn\\_id](#) 218  
[tmpl\\_defn\\_type](#) 218  
[tmpl\\_export\\_guid](#) 127, 141, 151, 154, 158, 160, 170, 178, 183, 185, 191, 195, 213, 215, 217, 220, 227, 230, 245, 249, 253, 256, 261, 265, 267, 269, 273  
[tmpl\\_grp](#) 217  
[tmpl\\_grp\\_descr](#) 218  
[tmpl\\_grp\\_id](#) 218  
[tmpl\\_grp\\_name](#) 218  
[tmpl\\_level\\_name](#) 125, 129, 139, 146, 169, 172, 174, 179, 189, 195, 197, 230  
[tmpl\\_package](#) 265  
[tmpl\\_package\\_content](#) 266  
[tmpl\\_package\\_id](#) 266  
[tmpl\\_package\\_intrnl\\_name](#) 266  
[tmpl\\_package\\_name](#) 266  
[tmpl\\_package\\_type](#) 265  
[tmpl\\_package\\_type\\_id](#) 266  
[tmpl\\_package\\_type\\_name](#) 266  
[tmpl\\_user\\_prof\\_id](#) 129, 146, 171, 178, 188, 196  
[to](#) 276

## T

[To](#) 276, 277  
[To Definition Version Id](#) 127, 154, 183, 185, 274  
[To Definition Version ID](#) 155, 184, 186, 274  
[To Id](#) 277, 278  
[To Item Definition Id](#) 128, 143, 164  
[To Item Id](#) 128, 143, 164  
[To Item Status Id](#) 239  
[To Item Status Name](#) 239  
[To Member Role Type Id](#) 128, 165  
[To One Entity](#) 276, 278  
[To One or More Entities](#) 276, 278  
[To Version](#) 262  
[To Zero or More Entities](#) 276, 278  
[To Zero or One Entity](#) 276, 278

## t

[to\\_defn\\_ver\\_id](#) 127, 154, 155, 183, 184, 185, 186, 274  
[to\\_id](#) 277, 278  
[to\\_item\\_defn\\_id](#) 128, 143, 164  
[to\\_item\\_id](#) 128, 143, 164  
[to\\_item\\_status\\_id](#) 239  
[to\\_item\\_status\\_name](#) 239  
[to\\_mbr\\_role\\_type\\_id](#) 128, 165  
[to\\_one\\_entity](#) 276  
[to\\_one\\_or\\_more\\_entities](#) 276  
[to\\_ver](#) 262  
[to\\_zero\\_or\\_more\\_entities](#) 276  
[to\\_zero\\_or\\_one\\_entity](#) 276  
[token](#) 183, 234, 235

## T

[Token](#) 183, 234, 235  
[Transaction Start Date](#) 225, 244, 275  
[Transition DateTime](#) 239

## t

[transn\\_datetime](#) 239

## T

Trigger End Date [203](#)  
Trigger Frequency [203](#)  
Trigger Hour [203](#)  
Trigger Minute [203](#)  
Trigger Start Date [203](#)  
Trigger Type Name [203](#)

## t

trigger\_end\_date [203](#)  
trigger\_freq [203](#)  
trigger\_hour [203](#)  
trigger\_minute [203](#)  
trigger\_start\_date [203](#)  
trigger\_type\_name [203](#)  
txn\_start\_date [225](#), [244](#), [275](#)

## T

Type Sort Order [202](#)

## t

type\_sort\_order [202](#)  
tz value [211](#)  
tz\_val [211](#)

## U

UI Component State [270](#)  
UI Component State Id [270](#)  
UI Element [136](#), [145](#)  
UI Element Definition [136](#), [145](#)  
UI Field [136](#), [145](#)  
UI Field Definition [136](#), [145](#)  
UI Form [136](#), [146](#)  
UI Form Definition [136](#), [146](#)  
UI Tab [137](#), [146](#)  
UI Tab Definition [137](#), [146](#)

## u

ui\_component\_state [270](#)  
ui\_component\_state\_id [270](#)  
ui\_ele [136](#)  
ui\_ele\_defn [136](#)  
ui\_field [136](#)  
ui\_field\_defn [136](#)  
ui\_form [136](#)  
ui\_form\_defn [136](#)  
ui\_tab [137](#)  
ui\_tab\_defn [137](#)

## U

Unchanged Count [232](#)

## u

unchgd\_ct [232](#)

## U

Unique Customer [187](#), [188](#)  
Unique Customer Profile [194](#), [196](#)  
Unique User Only [126](#), [140](#), [176](#), [182](#), [276](#)

## u

unique\_cust [187](#)  
unique\_cust\_prof [194](#)  
unique\_user\_only [126](#), [140](#), [176](#), [182](#), [276](#)

## U

UnResolved Count [238](#)

## u

unresolved\_ct [238](#)

## U

Update Date [257](#)  
Update Pack Id [262](#)  
Update User [257](#)

## U

update\_date [257](#)  
update\_pack\_id [262](#)  
update\_user [257](#)

## U

Updated Count [232](#)

## U

updated\_ct [232](#)

## U

Use Local Time [128](#), [144](#), [206](#)

## U

use\_local\_time [128](#), [144](#), [206](#)

## U

Useful Code [219](#), [220](#), [221](#)  
Useful Code Description [221](#)

## U

useful\_cd [219](#), [220](#), [221](#)  
useful\_cd\_descr [221](#)

## U

User Comment [220](#)  
User Defined Agent Role [168](#), [171](#)  
User Defined Customer Role [168](#), [171](#)  
User Id [260](#), [263](#)  
User Id From [260](#)  
User Login [125](#), [139](#), [157](#), [160](#), [162](#), [187](#), [190](#), [195](#), [210](#), [214](#), [219](#), [221](#), [223](#), [230](#), [234](#), [248](#), [253](#), [255](#), [259](#), [263](#), [268](#), [270](#)  
User Name [125](#), [139](#), [157](#), [160](#), [162](#), [187](#), [190](#), [195](#), [210](#), [214](#), [219](#), [221](#), [223](#), [230](#), [234](#), [248](#), [253](#), [255](#), [259](#), [263](#), [268](#), [270](#)  
User Preference Definition [137](#), [146](#)  
User Profile [123](#), [129](#), [137](#), [146](#), [168](#), [171](#), [174](#), [178](#), [187](#), [188](#), [194](#), [196](#)  
User Profile Id [125](#), [129](#), [146](#), [171](#), [178](#), [188](#), [194](#), [195](#), [196](#), [197](#)  
User Profile Name [129](#), [146](#), [171](#), [178](#), [188](#), [197](#)  
User Type [168](#), [172](#), [194](#), [197](#)  
User Type Description [172](#), [197](#)  
User Type Id [125](#), [129](#), [139](#), [146](#), [169](#), [172](#), [174](#), [179](#), [188](#), [194](#), [197](#), [230](#)  
User Type Name [172](#), [197](#)

## U

user\_comment [220](#)  
user\_defined\_agent\_role [168](#)  
user\_defined\_cust\_role [168](#)  
user\_id [260](#), [263](#), [269](#)  
user\_id\_from [260](#)  
user\_login [125](#), [139](#), [157](#), [160](#), [162](#), [187](#), [190](#), [195](#), [210](#), [214](#), [219](#), [221](#), [223](#), [230](#), [234](#), [248](#), [253](#), [255](#), [259](#), [263](#), [268](#), [270](#)  
user\_name [125](#), [139](#), [157](#), [160](#), [162](#), [187](#), [190](#), [195](#), [210](#), [214](#), [219](#), [221](#), [223](#), [230](#), [234](#), [248](#), [253](#), [255](#), [259](#), [263](#), [268](#), [270](#)  
user\_prefc\_defn [137](#)  
user\_prof [123](#), [137](#), [168](#), [174](#), [187](#), [194](#)  
user\_prof\_id [125](#), [129](#), [146](#), [171](#), [178](#), [188](#), [194](#), [195](#), [196](#), [197](#)  
user\_prof\_name [129](#), [146](#), [171](#), [178](#), [188](#), [197](#)  
user\_type [168](#), [194](#)  
user\_type\_descr [172](#), [197](#)  
user\_type\_id [125](#), [129](#), [139](#), [146](#), [169](#), [172](#), [174](#), [179](#), [188](#), [194](#), [197](#), [230](#)  
user\_type\_name [172](#), [197](#)

## V

val\_key [211](#)  
val\_map\_key [140](#), [169](#), [177](#), [178](#), [182](#), [185](#), [273](#)  
val\_type\_cd [126](#), [140](#), [169](#), [174](#), [175](#), [176](#), [177](#), [178](#), [179](#), [182](#), [184](#), [223](#), [225](#), [226](#), [248](#), [250](#), [252](#), [273](#), [276](#)  
val\_type\_name [179](#), [226](#), [252](#)  
valdn\_rule [137](#)

## V

Valid Values Rule [137](#), [146](#)

## V

valid\_vals\_rule [137](#)

## V

Validated Count [232](#)

## V

validated\_ct [232](#)

## V

Validation Rule [137](#), [146](#)

Value Map Key [140](#), [169](#), [177](#), [182](#), [185](#), [273](#)

Value Type Code [126](#), [140](#), [169](#), [174](#), [175](#), [176](#), [177](#), [179](#), [182](#), [184](#), [223](#), [225](#), [226](#), [248](#), [250](#), [252](#), [273](#), [276](#)

Value Type Name [179](#), [226](#), [252](#)

## V

value\_key [211](#)

ver\_id [203](#)

## V

Version ID [203](#)

## V

vote [253](#)

## V

Vote [253](#), [254](#)

Vote History [253](#), [254](#)

Vote History Id [254](#)

Vote Id [254](#)

## V

vote\_history [253](#)

vote\_history\_id [254](#)

vote\_id [254](#)

## V

Voting Metadata Id [254](#)

Voting Result [253](#), [254](#)

Voting Result Id [254](#)

Voting State [254](#)

Voting Timestamp [254](#)

## V

voting\_metadata\_id [254](#)

voting\_result [253](#)

voting\_result\_id [254](#)

voting\_state [254](#)

voting\_timestamp [254](#)

vt\$item\_fld\_defn\_networkadapte [271](#)

vt\$item\_fld\_defn\_NetworkAdapter [271](#), [272](#)

vt\$item\_fld\_NetworkAdapter\$changes [271](#), [272](#)

vt\$item\_fld\_networkadapter\$chg [271](#)

## W

warn\_ct [269](#)

## W

Warning Count [269](#)

Week of Month [241](#)

Week of Quarter [241](#)

## W

week\_of\_month [241](#)

week\_of\_quarter [241](#)

widget [263](#)

## W

Widget [263](#), [264](#)

Widget Configuration [264](#)

Widget Description [264](#)

Widget GUID [264](#)

Widget Id [263](#), [264](#)

Widget Index [264](#)

Widget Layout [264](#)

Widget Type [264](#)

Widget View Configuration [264](#)

## W

widget\_cfg [264](#)

widget\_descr [264](#)



widget\_guid 264  
widget\_id 263, 264  
widget\_index 264  
widget\_layout 264  
widget\_type 264  
widget\_view\_cfg 264  
wkflw\_actn 137  
wkflw\_proc\_defn 137  
wkflw\_proc\_defn\_wkflw\_actn 137  
wksp\_actvy\_data\_mart 124, 137  
wksp\_admin\_prof 194  
wksp\_admin\_role 168  
wksp\_cntnr 124, 137, 150, 181, 205, 259  
wksp\_defn\_id 158, 192, 213, 233  
wksp\_item 123, 136, 157, 162

## W

Work Duration In Minutes 206  
Work Hours Count 239

## W

work\_durn\_in\_mins 206  
work\_hours\_ct 239

## W

Workflow Action 137  
Workflow Process Definition 137, 146  
Workflow Process Definition Workflow Action 137, 147  
Workspace Activity Data Mart 124, 137  
Workspace Administration Profile 194, 197  
Workspace Administrator Role 168, 172  
Workspace Container 124, 129, 137, 147, 150, 151, 181, 186, 205, 207, 259, 261  
Workspace Definition Id 158, 192, 213, 233  
WorkTarget Item 138, 147  
WorkTarget Item Id 147

## W

worktarget\_item\_id 147

## Y

Yes Count 220

## y

yes\_ct 220