SQL Performance for DB2®
User Guide

Supporting

Version 10.1 of SQL Performance for DB2

April 2011
Contacting BMC Software

You can access the BMC Software website at http://www.bmc.com. From this website, you can obtain information about the company, its products, corporate offices, special events, and career opportunities.

United States and Canada

Address  
BMC SOFTWARE INC  
2101 CITYWEST BLVD  
HOUSTON TX 77042-2827  
USA

Telephone  
713 918 8800  or  
800 841 2031

Fax  
713 918 8000

Outside United States and Canada

Telephone  
(01) 713 918 8800

Fax  
(01) 713 918 8000

© Copyright 1996 - 2011 BMC Software, Inc.

BMC, BMC Software, and the BMC Software logo are the exclusive properties of BMC Software, Inc., are registered with the U.S. Patent and Trademark Office, and may be registered or pending registration in other countries. All other BMC trademarks, service marks, and logos may be registered or pending registration in the U.S. or in other countries. All other trademarks or registered trademarks are the property of their respective owners.

CICS, DB2, IBM, IMS, MVS, OS/390, RACF, z/OS, and zSeries are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

UNIX is the registered trademark of The Open Group in the US and other countries.

The information included in this documentation is the proprietary and confidential information of BMC Software, Inc., its affiliates, or licensors. Your use of this information is subject to the terms and conditions of the applicable End User License agreement for the product and to the proprietary and restricted rights notices included in the product documentation.

Restricted rights legend

U.S. Government Restricted Rights to Computer Software. UNPUBLISHED -- RIGHTS RESERVED UNDER THE COPYRIGHT LAWS OF THE UNITED STATES. Use, duplication, or disclosure of any data and computer software by the U.S. Government is subject to restrictions, as applicable, set forth in FAR Section 52.227-14, DFARS 252.227-7013, DFARS 252.227-7014, DFARS 252.227-7015, and DFARS 252.227-7025, as amended from time to time. Contractor/Manufacturer is BMC SOFTWARE INC, 2101 CITYWEST BLVD, HOUSTON TX 77042-2827, USA. Any contract notices should be sent to this address.
Customer support

You can obtain technical support by using the BMC Software Customer Support website or by contacting Customer Support by telephone or e-mail. To expedite your inquiry, see “Before contacting BMC.”

Support website

You can obtain technical support from BMC 24 hours a day, 7 days a week at http://www.bmc.com/support. From this website, you can

- read overviews about support services and programs that BMC offers
- find the most current information about BMC products
- search a database for issues similar to yours and possible solutions
- order or download product documentation
- download products and maintenance
- report an issue or ask a question
- subscribe to receive proactive e-mail alerts when new product notices are released
- find worldwide BMC support center locations and contact information, including e-mail addresses, fax numbers, and telephone numbers

Support by telephone or e-mail

In the United States and Canada, if you need technical support and do not have access to the web, call 800 537 1813 or send an e-mail message to customer_support@bmc.com. (In the subject line, enter SupID:<yourSupportContractID>, such as SupID:12345). Outside the United States and Canada, contact your local support center for assistance.

Before contacting BMC

Have the following information available so that Customer Support can begin working on your issue immediately:

- product information
  - product name
  - product version (release number)
  - license number and password (trial or permanent)
- operating system and environment information
  - machine type
  - operating system type, version, and service pack or other maintenance level such as PUT or PTF
  - system hardware configuration
  - serial numbers
  - related software (database, application, and communication) including type, version, and service pack or maintenance level
- sequence of events leading to the issue
- commands and options that you used
- messages received (and the time and date that you received them)
  - product error messages
  - messages from the operating system, such as file system full
  - messages from related software
License key and password information

If you have questions about your license key or password, use one of the following methods to get assistance:

- Send an e-mail message to customer_support@bmc.com.
## Contents

### About this book

15

- Related publications
  
- Conventions
  
- Summary of changes

### Chapter 1 Overview of SQL Performance

21

- Overview of SQL Performance
  
- SQL Performance solution components
  
- SQL Performance features
  
- SQL Performance for DB2 architecture
  
- Data Collector
  
- Report Manager
  
- Sysplex support
  
- Shared infrastructure components

### Chapter 2 Getting started with SQL Performance

39

- Logging on
  
- Navigation
  
- Panel layout
  
- Online report layout

### Chapter 3 Managing performance with Performance Advisor

49

- Overview of Performance Advisor
  
- Why you need Performance Advisor
  
- Reorganizing objects
  
- Eliminating obsolete indexes
  
- Identifying degraded objects
  
- Task summary
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Working with object sets</th>
<th>61</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Overview of object sets</td>
<td>61</td>
</tr>
<tr>
<td></td>
<td>Description of object set patterns</td>
<td>63</td>
</tr>
<tr>
<td></td>
<td>Inclusion versus exclusion of objects from the workload</td>
<td>64</td>
</tr>
<tr>
<td></td>
<td>Defining an object set</td>
<td>65</td>
</tr>
<tr>
<td></td>
<td>Managing object set definitions</td>
<td>68</td>
</tr>
<tr>
<td></td>
<td>Selecting an object set to use</td>
<td>68</td>
</tr>
<tr>
<td></td>
<td>Editing an object set</td>
<td>68</td>
</tr>
<tr>
<td></td>
<td>Renaming an object set</td>
<td>70</td>
</tr>
<tr>
<td></td>
<td>Copying an object set</td>
<td>71</td>
</tr>
<tr>
<td></td>
<td>Viewing an object set</td>
<td>71</td>
</tr>
<tr>
<td></td>
<td>Deleting an object set</td>
<td>72</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Comparing access paths for workloads</th>
<th>73</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Overview of workload comparisons</td>
<td>73</td>
</tr>
<tr>
<td></td>
<td>Benefits</td>
<td>74</td>
</tr>
<tr>
<td></td>
<td>Workload Access Path Compare process</td>
<td>75</td>
</tr>
<tr>
<td></td>
<td>Comparing results</td>
<td>77</td>
</tr>
<tr>
<td></td>
<td>Extracting and explaining a workload</td>
<td>78</td>
</tr>
<tr>
<td></td>
<td>Explaining a workload a second time</td>
<td>84</td>
</tr>
<tr>
<td></td>
<td>Comparing two or more workloads</td>
<td>87</td>
</tr>
<tr>
<td></td>
<td>Working with workload comparison reports</td>
<td>90</td>
</tr>
<tr>
<td></td>
<td>Sample batch report for Workload Access Path Compare</td>
<td>91</td>
</tr>
<tr>
<td></td>
<td>Specifying report options for workload comparisons</td>
<td>96</td>
</tr>
<tr>
<td></td>
<td>Customizing the comparison report</td>
<td>98</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Recommending indexes for workloads</th>
<th>99</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Overview of the Workload Index Advisor</td>
<td>99</td>
</tr>
<tr>
<td></td>
<td>Benefits</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>Workload Index Advisor process</td>
<td>101</td>
</tr>
<tr>
<td></td>
<td>Extracting SQL and recommending an index</td>
<td>103</td>
</tr>
<tr>
<td></td>
<td>Recommending an index for a workload a second time</td>
<td>108</td>
</tr>
<tr>
<td></td>
<td>Reviewing index recommendations</td>
<td>110</td>
</tr>
<tr>
<td></td>
<td>Working with index recommendation reports</td>
<td>114</td>
</tr>
<tr>
<td></td>
<td>Sample batch report for Index Advisor</td>
<td>114</td>
</tr>
<tr>
<td></td>
<td>Specifying report options for Index Advisor reports</td>
<td>117</td>
</tr>
<tr>
<td></td>
<td>Customizing index recommendation reports</td>
<td>119</td>
</tr>
</tbody>
</table>
# Chapter 7 Exploring advice for exceptions

Overview of the Exception Advisor .................................................. 122
Benefits ......................................................................................... 122
Exception Advisor input ................................................................. 123
Exception Advisor process ............................................................... 123
Running the Exception Advisor batch job ........................................ 124
Rules ............................................................................................. 124
Exception Advisor example ............................................................. 125
Exception Advisor parameters ......................................................... 130
Viewing the Exception Advisor reports online ................................... 135
Exception Advisor output ............................................................... 138
Exception information ................................................................. 138
History of statement performance .................................................. 139
Comparison graphs ....................................................................... 140
SQL text and host variables ......................................................... 141
Advice ......................................................................................... 142

# Chapter 8 Index Component reports

Index Component report set ........................................................... 145
Report hierarchies ....................................................................... 146
Action codes ................................................................................. 150
Reporting intervals ..................................................................... 150
Using panels ............................................................................... 150
Using online reports ................................................................. 151
Viewing reports ......................................................................... 152
Application groups and application profiles .................................. 153
IN-SQL elapsed time measurement .............................................. 153
Explain processing ..................................................................... 155
Additional reporting features ..................................................... 156
Batch reporting .......................................................................... 157
Report format .............................................................................. 157
Batch utilities .............................................................................. 158
Batch report qualifiers ............................................................... 159

# Chapter 9 Using the Index Component

Overview of the Index Component .................................................. 161
Preparing to use What-If Index ....................................................... 162
Identifying What-If Index candidates ............................................. 165
Using the What-If Index function .................................................. 170
Adding an index by using What-If Index ........................................ 171
Dropping an index by using What-If Index ..................................... 177
Updating statistics on an index by using What-If Index .................. 180
Saving What-If Index DDL changes ............................................. 183
Importing What-If Index changes to another statement for a table ... 186
Adding an expression to a What-If Index ....................................... 191
Figures

SQL Performance for DB2 (AFD) architecture ............................................. 30
Sysplex communication .......................................................... 32
Shared infrastructure component architecture ...................................... 33
System and SQL Performance for DB2 logo panel ................................. 40
SQL Performance panels .............................................................. 41
Common online report elements ...................................................... 43
System and SQL Performance for DB2 main menu (DOMESELT) .............. 46
Example index query .............................................................. 51
Example object query .............................................................. 52
Performance Advisor task overview .............................................. 52
Example DELETE SQL statements ................................................. 58
Object sets and source parameters ........................................... 62
Object set definition on Extract SQL for Workload panel (PSSWC105) ...... 65
Object Set Specification panel (PSSWC180) .................................... 66
Enter Name Specification panel (PSSWC185) .................................... 67
Object Set List panel (PSSWC175) ................................................ 69
Workload Access Path Compare process (two different subsystems) ......... 76
Workload Access Path Compare Main Menu (PSSWC100) ..................... 79
Extract SQL for Workload (PSSWC105) .......................................... 79
SQL Workload Explain (PSSWC120) ............................................. 81
Batch Job panel (PSSPB000) .................................................... 83
SQL Workload Compare List for Explain (PSSWC115) ............................ 85
SQL Workload Compare (PSSWC130) ............................................ 87
SQL Workload Compare Report (PSSWC140) .................................... 89
SQL Workload Compare Detail Report (PSSWC142) ............................... 89
Batch Workload Access Path Compare report .................................. 91
Report options panel (PSSWC137) ............................................... 96
Index Advisor process ........................................................... 102
Index Advisor Menu (PSSWC100) .............................................. 104
Recommindex panel (PSSWC120) ............................................... 106
SQL Workload List for Recommindex (PSSWC115) ............................... 108
Index Recommendation Report ............................................... 111
Index Advisor Workloads (PSSWC245) .......................................... 112
Recommindex: Index Recommendation Report panel (PSSWC140) ............ 112
Sample of Index Advisor report ............................................ 114
Index Advisor Report Options (PSSWC275) ..................................... 117
Layout Data Set panel (PSSWCLDS) .............................................. 118
Exception Advisor panel (IODEWCMB/I) ........................................ 135
XCEPTION ADVISOR-EXCEPTIONS panel ...................................... 137
Index Component reports ......................................................... 146
<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Browse DDL data set member</td>
<td>188</td>
</tr>
<tr>
<td>List of members in DDL data set</td>
<td>188</td>
</tr>
<tr>
<td>Import DDL panel (PSSPWDI0)</td>
<td>187</td>
</tr>
<tr>
<td>Explain Results panel showing cloned objects</td>
<td>187</td>
</tr>
<tr>
<td>Browse saved DDL data set</td>
<td>188</td>
</tr>
<tr>
<td>Explain Results panel (PSSPW200)</td>
<td>186</td>
</tr>
<tr>
<td>Results of dynamic Explain on cloned objects after statistical update</td>
<td>186</td>
</tr>
<tr>
<td>Update Index Access Path Statistics panel (PSSPWIXS)</td>
<td>185</td>
</tr>
<tr>
<td>Display of cloned objects</td>
<td>185</td>
</tr>
<tr>
<td>Specifying statistics to be updated for an index using What-If Index</td>
<td>185</td>
</tr>
<tr>
<td>Save DDL panel (PSSPWD00)</td>
<td>184</td>
</tr>
<tr>
<td>Explain Results panel (PSSPW200)</td>
<td>184</td>
</tr>
<tr>
<td>Save DDL data set</td>
<td>184</td>
</tr>
<tr>
<td>Explain Results panel showing cloned objects</td>
<td>187</td>
</tr>
<tr>
<td>Import DDL panel (PSSPWDI0)</td>
<td>187</td>
</tr>
<tr>
<td>List of members in DDL data set</td>
<td>188</td>
</tr>
<tr>
<td>Browse DDL data set member</td>
<td>188</td>
</tr>
</tbody>
</table>

10  SQL Performance for DB2 User Guide
Figures 11

Import DDL Member Selection List panel (PSSPMEML) .......................... 189
Message displayed when import process is complete .......................... 189
Explain Results panel (PSSPW200) .................................................. 190
Results of dynamic Explain on imported DDL ................................. 190
Explain Edit panel ................................................................. 191
Explain or Execute Parameters panel (PSSPA117) .............................. 191
Explain Results panel showing an ad hoc query ............................ 192
Explain Results panel showing the clone ................................. 192
Explain Edit panel showing an expression ........................................ 193
Select Key Columns showing an expression ...................................... 193
Explain Results panel showing an index on expression .................. 194
Example of a PSSEXTRJ error report ............................................. 197
Example of a PSSEXTRJ error report ............................................. 200
DMDBMERG JCL ................................................................. 204
DMDBRTSU JCL ................................................................. 216
Default layout data set member (PSSREPB) ................................... 236
System and SQL Performance for DB2 main menu (DOMESELT) ........ 246
Tables

Shared infrastructure components ............................................................ 25
Features and benefits of SQL Performance ............................................... 26
Collection options ..................................................................................... 35
Panel elements .......................................................................................... 41
Report elements ......................................................................................... 44
Performance Advisor sample members for creating tables ..................... 53
Object types for object definitions ........................................................... 63
Source Workload Extract panels by source .............................................. 81
Batch Job options ..................................................................................... 83
Reviewing an online report ....................................................................... 89
Source Workload Extract panels by source .............................................. 105
Exception Advisor parameters ................................................................. 130
Report qualifiers used by the Index Component batch reports ............. 159
PSSEXTRJ input parameters for DB2 Version 8 subsystem .................... 196
PSSEXTRJ input parameters for DB2 Version 9 subsystem .................... 197
PSSCOMPJ input parameters for DB2 Version 9 subsystem ................... 198
PSSEXTRJ input parameters .................................................................... 199
PSSCOMPJ input parameters ................................................................... 200
DMDBMERG data sets ............................................................................. 208
DMDBMERG SYSIN control statements .................................................. 210
INTERVAL statement and parameter keywords ...................................... 211
QUALIFIER statement and parameter keywords .................................... 213
DMDBRTSU data sets ............................................................................. 217
Performance Advisor tables ..................................................................... 218
Performance Advisor samples ................................................................. 220
Information for DB2 catalog sources ....................................................... 224
Information for APPTUNE trace data sources ......................................... 224
Information for DBRM libraries ............................................................... 226
Information for Performance Advisor Database sources ...................... 226
Information for SQL text data set sources ............................................. 229
Information for DB2 statement cache sources ....................................... 230
Workload Access Path Compare and Index Advisor report field labels
  (from PSSREPB layout data set) ................................................................. 231
Return codes for the Workload Access Path Compare function ............ 240
Return codes for the Index Advisor function .......................................... 240
DSN_SORT_TABLE ................................................................................... 241
DSN_SORTKEY_TABLE .......................................................................... 241
DSN_VIRTUAL_INDEXES .......................................................................... 242
IFCID used by SQL Performance reports ................................................. 243
About this book

This book contains detailed information about the BMC Software SQL Performance for DB2® solution and is intended for IBM® DB2 Universal Database (DB2) system administrators, database administrators (DBAs), DB2 application programmers, and others.

To use this book, you should be familiar with the following items:

- IBM DATABASE 2 Universal Database Server for OS/390® and z/OS®
- Multiple Virtual Storage (MVS®/XA or MVS/ESA) systems
- Job Control Language (JCL)
- Interactive System Productivity Facility (ISPF)
- your host operating system

Like most BMC Software documentation, this book is available in printed and online formats. Visit the BMC Software Customer Support page at http://www.bmc.com/support_home to request additional printed books or to view online books and notices (such as release notes and technical bulletins). Some product shipments also include the online books on a documentation CD.

**NOTE**

Online books are formatted as Portable Document Format (PDF) or HTML files. To view, print, or copy PDF books, use the free Adobe Reader from Adobe Systems. If your product installation does not install the reader, you can obtain the reader at http://www.adobe.com.

The software also offers online Help. To access Help, press F1 within any product. For more information, see “Online assistance” on page 47.
Related publications

The following related publications supplement this book and the online Help:

<table>
<thead>
<tr>
<th>Category</th>
<th>Document</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>installation</td>
<td>System and SQL Performance for DB2 Installation Guide</td>
<td>contains installation, customization, and configuration information for the SQL Performance products</td>
</tr>
<tr>
<td>documents</td>
<td>System and SQL Performance for DB2 Administrator Guide</td>
<td>provides detailed information about the administrative functions for the common SQL Performance components</td>
</tr>
<tr>
<td></td>
<td>APPTUNE for DB2 User Guide</td>
<td>describes the functions of the APPTUNE for DB2 product and provides guidelines for using the product</td>
</tr>
<tr>
<td></td>
<td>SQL Explorer for DB2 User Guide</td>
<td>describes the functions of the SQL Explorer for DB2 product and provides guidelines for using the product</td>
</tr>
<tr>
<td>core documents</td>
<td>BMC Runtime Component System Configuration and Administration Guide</td>
<td>provides information about configuring and administering the Runtime Component System (RTCS)</td>
</tr>
<tr>
<td></td>
<td>BMC Common Components Messages Manual</td>
<td>provides message information for infrastructure components used by SQL Performance</td>
</tr>
<tr>
<td>notices</td>
<td>SQL Performance for DB2 Release Notes</td>
<td>provide updates to the installation instructions and last-minute product information</td>
</tr>
</tbody>
</table>

Conventions

This book uses the following types of special text:

---

**NOTE**

Notes contain important information that you should consider.

---

**WARNING**

Warnings alert you to situations that could cause problems, such as loss of data, if you do not follow instructions carefully.
Summary of changes

This section summarizes changes to the functionality of the solution, listing the changes by product version and release date. The summary includes enhancements to the solution and any major changes to the documentation.

Version 10.1.00  April 2011

Major enhancements to version 10.1.00 of the SQL Performance for DB2 solution are listed in the following sections.

DB2 support

SQL Performance for DB2 and the associated product components now support DB2 Version 10.

Workload Index Advisor component

This new component helps create the right indexes to minimize the cost of running a set of SQL statements. Workload Index Advisor uses the Explain capability to extract and explain the SQL from a user-defined workload; the component then analyzes the indexes to provide estimated costs and recommendations for additional indexes. Workload Index Advisor validates the results by testing the recommendations with virtual indexes.
Summary of changes

Workload Access Path Compare component

This component now uses object sets to define the workloads to be used. This release also adds additional sources for SQL that you can use within workloads. The sources now include:

- APPTUNE archived trace data sets
- BMC Performance Database (new)
- DB2 catalog
- DB2 statement cache (new)
- a DBRM library
- SQL text data set (new)

The Workload Access Path Compare component also now uses timerons instead of service units as the basis of cost comparisons.

Exception Advisor component

This new component uses the Performance Advisor Database (PADB) to identify the root causes of exceptional executions and to make recommendations. Exception Advisor compares the accompanying data to past execution statistics for the same statement in baseline or aggregated tables.

Infrastructure changes

This release changes the infrastructure of the SQL Performance solution to include the following new technology components:

- In previous versions, the Data Collector ran as its own started task. Now, the Data Collector runs within the DB2 Component Services (DBC) subsystem. DBC provides a persistent z/OS subsystem address space into which BMC products can dynamically initialize their own product services.

- This release changes the method that APPTUNE uses to log data. APPTUNE now uses the Next Generation Logger (NGL), a logging facility that logs and retrieves data based on application-defined keys and a time span. NGL runs as a service within the DBC subsystem and relies on the Runtime Component System (RTCS) for registry services. APPTUNE uses NGL for archiving trace data sets.

- This release changes the method in which you specify DOMPLEX and APPTUNE filter parameters. To configure the DOMPLEX and APPTUNE filter parameters, you now use the DB2 Product Configuration technology and create option sets that SQL Performance uses to manage data collection.
The DB2 Product Configuration technology separates product (or solution) installation from configuration. DB2 Product Configuration simplifies configuration and deployment by setting default option values for you. (You can change the values, if needed.) The online interface of DB2 Product Configuration simplifies navigation by allowing you to expand or contract sections as needed.

DB2 Product Configuration is integrated within the Installation System so that you can configure the DOMPLEX during installation. DB2 Product Configuration is also available through the Administration option on the System and SQL Performance for DB2 main menu. DB2 Product Configuration runs within the DBC subsystem.

**Version 6.2.00 November 2008**

SQL Performance for DB2 version 6.2.00 includes the following product components

- APPTUNE version 6.2
- SQL Explorer version 6.2

Major enhancements to the solution are listed in the following sections.

**Performance Advisor component**

This new component helps you set up and maintain a Performance Management Database (PMDB) that improves performance by identifying high-impact tuning operations, such as reorganizing objects, eliminating obsolete indexes, and identifying degraded objects.

**Workload Access Path Compare component**

This new component helps you identify changed access paths in a user-defined workload. Workload Access Path Compare analyzes SQL statements and provides reports that indicate specific differences between statements, including such things as differences in statistics, SQL text, and indexes which might have caused changes to the access paths.

**Support for views and aliases**

The Index component has been enhanced to support views and aliases used in the What-If Index function.

**DB2 version support**

SQL Performance supports DB2 Version 8 and Version 9. Support for DB2 Version 7 has been eliminated.
Summary of changes
Overview of SQL Performance

The BMC SQL Performance for DB2 solution enables DBAs, application developers, and system programmers to identify and correct performance problems in DB2 applications that run in IBM CICS®, IMS®, and OS/390 environments. This solution is the answer to the ongoing quest for optimal application efficiency. SQL Performance provides a unique set of tools that allow an application to be fine-tuned through its conception and growth, to its retirement.

The SQL Performance for DB2 solution provides the following benefits:

- solves application performance problems by quickly identifying the most expensive SQL statements and making tuning recommendations
- avoids application performance problems by proactively correcting problems before an application reaches production
- increases programmer productivity and efficiency by providing both analysis of access paths and easy-to-use tuning tools

The BMC SQL Performance for DB2 solution enables DBAs, application developers, and system programmers to identify and correct performance problems in DB2 applications that run in IBM CICS®, IMS®, and OS/390 environments. This solution is the answer to the ongoing quest for optimal application efficiency. SQL Performance provides a unique set of tools that allow an application to be fine-tuned through its conception and growth, to its retirement.

The SQL Performance for DB2 solution provides the following benefits:

- solves application performance problems by quickly identifying the most expensive SQL statements and making tuning recommendations
- avoids application performance problems by proactively correcting problems before an application reaches production
- increases programmer productivity and efficiency by providing both analysis of access paths and easy-to-use tuning tools
SQL Performance solution components

- quickly and easily pinpoints resource-consuming SQL statements without executing a DB2 SQL trace

- provides in-depth index optimization recommendations, including identification of unused indexes and “What-If Index” analysis

For more information, see “SQL Performance features” on page 26.

The SQL Performance solution includes the following components:

- APPTUNE for DB2
- SQL Explorer for DB2
- Performance Advisor
- Workload Access Path Compare component
- Workload Index Advisor component
- Exception Advisor component
- REORG Advisor component
- Index component

**NOTE**
The Performance Advisor, Workload Access Path Compare, Workload Index Advisor, Exception Advisor, REORG Advisor, and Index components are available only as part of the SQL Performance solution (not as stand-alone components).

The solution comes with a comprehensive set of flexible reports that lets you obtain the best picture of application information that is meaningful to your needs.

You can access all SQL Performance components through a common interface, without having to know which component to use to solve a problem. The solution guides you to the information you need to solve application tuning problems, using the appropriate components to gather the information and make the needed changes.

**APPTUNE for DB2 component**

The APPTUNE for DB2 component is an application performance and resource analysis facility that is used to gather and display data from a single SQL statement or a set of SQL statements. The gathered data provides valuable information about the performance of and resource use by DB2 applications. APPTUNE collects all relevant performance measures in real time for every SQL statement executed in one or more DB2 subsystems. The collected data is then summarized and stored for analysis.
SQL Explorer for DB2 component

The SQL Explorer for DB2 component is an SQL analysis tool that you can use to proactively manage performance problems. With SQL Explorer, DBAs and application developers can quickly and easily analyze SQL statements and database structures to optimize the performance of applications before the applications are put in production. SQL Explorer can also be used to identify and correct problems in production applications.

Performance Advisor component

The Performance Advisor component helps you set up and maintain a Performance Advisor Database (PADB). The PADB improves performance by identifying high-impact tuning operations, such as reorganizing objects, eliminating obsolete indexes, and identifying degraded objects.

NOTE

The Performance Advisor Component is available only as part of the SQL Performance solution (not as a stand-alone component).

For more information about the Performance Advisor component, see Chapter 3, “Managing performance with Performance Advisor.”

Workload Access Path Compare component

The Workload Access Path Compare component is an automated tool that identifies and compares changed access paths and provides analysis and drill-down capability. The tool enables you to

- compare workloads when running an application on different DB2 subsystems
- predict changes in the access path within an application before migration to a new release of DB2
- analyze and predict access path changes for dynamic and static SQL

NOTE

The Workload Access Path Compare Component is available only as part of the SQL Performance solution (not as a stand-alone component).

For more information about the Workload Access Path Compare Component, see Chapter 5, “Comparing access paths for workloads.”
Workload Index Advisor component

The Workload Index Advisor component helps you create the right indexes to minimize the cost of running a set of SQL statements. Workload Index Advisor uses the Explain capability to extract and explain the SQL from a user-defined workload; the component then analyzes the indexes to provide estimated costs and recommendations for additional indexes. Workload Index Advisor then validates the results by testing the recommendations with virtual indexes.

NOTE
The Workload Index Advisor component is available only as part of the SQL Performance solution (not as a stand-alone component).

For more information about the Workload Index Advisor component, see Chapter 6, “Recommending indexes for workloads.”

Exception Advisor component

The Performance Advisor component of SQL Performance for DB2 maintains a DB2 Performance Advisor Database (PADB). Using the PADB, Exception Advisor identifies the root causes of previously triggered exception executions and makes recommendations.

Exception Advisor compares the accompanying data to past execution statistics for the same statement in baseline or aggregated tables. You can adjust the analysis without changing code. Exception Advisor bases its recommendations on observed conditions.

NOTE
The Exception Advisor component is available only as part of the SQL Performance solution (not as a stand-alone component).

For more information about the Exception Advisor component, see Chapter 7, “Exploring advice for exceptions.”

REORG Advisor

The REORG Advisor is a REXX® EXEC program, IODADREO, that identifies potential candidate objects for REORG, based on two criteria:

- the level of disorganization of the object
- the level of performance degradation for the object since the last REORG occurred.
The advisor helps you avoid performing unnecessary reorganizations for tables and indexes that are disorganized, yet do not show degraded performance since the time of the last REORG.

For more information, see “Using REORG Advisor” on page 60.

**Index component**

The Index component automatically collects and displays actual access counts for each unique SQL statement (table and index, and predicate usage frequencies). A *What-If Index* analysis lets you model changes to indexes. The Index Component provides on-demand, dynamic data collection of index dependencies and catalog statistics. Table and index reports provide quick access to listings of the most-used object based on getpage volume or ratio.

*NOTE*

The Index Component is available only as part of the SQL Performance solution (not as a stand-alone component).

For more information about the Index Component, see Chapter 9, “Using the Index Component.”

**Shared infrastructure components**

SQL Performance also uses the infrastructure components described in Table 1.

<table>
<thead>
<tr>
<th>Table 1</th>
<th>Shared infrastructure components (part 1 of 2)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Component</strong></td>
<td><strong>Purpose</strong></td>
</tr>
<tr>
<td>DB2 Component Services (DBC)</td>
<td>DBC provides a persistent z/OS subsystem address space into which BMC products can dynamically initialize their own product services. Other shared infrastructure components (such as the Data Collector, NGL, DB2 Product Configuration, and RTCS) require the DBC subsystem.</td>
</tr>
<tr>
<td>Next Generation Logger (NGL)</td>
<td>NGL is a logging facility that logs and retrieves data based on application-defined keys and a time span. NGL runs as a service within the DBC subsystem and relies on the Runtime Component System (RTCS) for registry services. APPTUNE requires NGL for archiving trace data sets.</td>
</tr>
</tbody>
</table>
SQL Performance features

Table 1  Shared infrastructure components (part 2 of 2)

<table>
<thead>
<tr>
<th>Component</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Runtime Component System (RTCS)</td>
<td>RTCS runs as a started task and provides programming services to various BMC mainframe products. RTCS is designed for continuous operation and seldom, if ever, needs to be stopped.</td>
</tr>
<tr>
<td>DB2 Product Configuration</td>
<td>The DB2 Product Configuration technology separates product (or solution) installation from configuration. Through its online interface, DB2 Product Configuration simplifies configuration and deployment by setting default option values for you. (You can change the values, if needed.) DB2 Product Configuration panels simplify navigation by allowing you to expand or contract sections as needed. Also, you can link to DB2 Product Configuration from within your product or solution, thus maintaining a consistent look and feel, and retaining your changes from version to version.</td>
</tr>
</tbody>
</table>

SQL Performance features

Table 2 summarizes the major features and benefits of the SQL Performance for DB2 solution.

Table 2  Features and benefits of SQL Performance (part 1 of 4)

<table>
<thead>
<tr>
<th>Feature</th>
<th>Benefit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analysis criteria</td>
<td>Using SQL Performance, you can specify the time period, DB2 subsystems, and data source (archived data or online data) used to generate online or batch reports. You can also specify the format—numeric or graphical—of the reports generated. This feature allows you to exclude unnecessary information from generated reports, include real-time data as needed, analyze historical data, and choose a report format that best suits your needs.</td>
</tr>
<tr>
<td>Analysis of dynamic SQL from trace data sets</td>
<td>This feature of SQL Performance is useful when you plan to migrate to a different version of DB2 and want to find out how your existing data will perform, or when you want to Explain and Compare the access paths for dynamic SQL from subsystems running on different versions of DB2.</td>
</tr>
<tr>
<td>Application groups and application profiles</td>
<td>This feature of SQL Performance enables you to define the plans, programs, and users that make up an application group (using your own criteria) and the application groups that make up an application profile. Application groups and application profiles can be defined by each user or at the system level. Public access to an application group can be limited to NONE, READ, or UPDATE. This feature allows you to create collections of plans, programs, users, correlation IDs, or a combination of these elements in groups that you want to analyze for application performance.</td>
</tr>
</tbody>
</table>
Table 2  Features and benefits of SQL Performance (part 2 of 4)

<table>
<thead>
<tr>
<th>Feature</th>
<th>Benefit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Archive directory</td>
<td>This feature of SQL Performance houses historical data collected for each output group enabled for archiving. The archived data sets can be dynamically allocated for online or batch reporting based on qualifiers that you provided. Archived data can also be loaded into the Performance Advisor Database or used as a source of SQL text for Workload Access Path Compare or Workload Index Advisor.</td>
</tr>
<tr>
<td>Complete SQL capture</td>
<td>This feature of SQL Performance enables you to capture data for each SQL statement (OPEN, FETCH, CLOSE, and so on) executed during a collection period. SQL Performance provides the information needed to tune application performance, while Explain-based products provide only part of the information needed.</td>
</tr>
<tr>
<td>DB2 current status</td>
<td>SQL Performance can display current information about a DB2 subsystem and active threads quickly and easily. From an active thread, you can also zoom down to the text of the SQL statement currently executing and Explain it, if needed. This feature is very helpful for diagnosing a system-level problem, such as a long-running transaction or batch job.</td>
</tr>
<tr>
<td>Exception Advisor component</td>
<td>Exception Advisor looks at performance trends to identify the root causes of exceptions. SQL tuners that run on DB2 subsystems have access to large quantities of performance data. As a result, tuning efforts can be quite time-consuming. By narrowing the focus of the tuning effort to those statements that exceed target thresholds, Exception Advisor can often identify the root cause of an exception immediately. The Exception Advisor is not looking at predicted performance such as Explain data but at actual performance history. Because Exception Advisor uses existing data, you incur no further cost for data collection.</td>
</tr>
<tr>
<td>Explain function</td>
<td>This feature of SQL Performance enables you to Explain dynamic and static SQL statements, providing both statistical and textual information about the access path, with suggestions on how to improve SQL statement performance.</td>
</tr>
<tr>
<td>Fully functional administrative facility</td>
<td>SQL Performance product administrators can define Data Collector parameters and user privileges completely. If multiple System and SQL Performance products are installed, all products can be controlled from one console. This feature enables central control of the major functions of SQL Performance (and other System and SQL Performance products, if installed) from a single point.</td>
</tr>
<tr>
<td>Graphical data reporting option</td>
<td>This feature of SQL Performance enables you to choose whether reports are displayed in a traditional, numeric format or a graphical format. The solution displays statistical reports in an easy-to-read, understandable graphical format, or in a numeric format.</td>
</tr>
<tr>
<td>Index component</td>
<td>This component of the SQL Performance solution extends the capability of APPTUNE object analysis by collecting and reporting on column usage data for SQL statements; also extends the capability of the Explain function by comparing access paths after making changes to indexes in a cloned database. The Index Component offers users a way to obtain accurate, real-time performance information about DB2 indexes. Because the Index Component presents data at the object level, you can review the index access data to evaluate the performance of your indexes and identify candidates for index improvements.</td>
</tr>
</tbody>
</table>
## IN-SQL measurement

This feature of SQL Performance enables you to measure only the resources consumed during the execution of an SQL statement. IN-SQL measurements exclude the time spent on associated DB2 housekeeping tasks. This feature provides the most accurate data for measuring the cost of an SQL statement. Measurements based on IN-DB2 time are less accurate and might not reflect the actual cost of an SQL statement.

## Intuitive interface

SQL Performance has a highly intuitive user interface that conforms to ISPF standards and includes a comprehensive online Help facility. This feature allows you to use the product with unparalleled ease and minimal supporting documentation.

## Object reports

This feature of SQL Performance enables you to collect and analyze information about access, usage, and response times for DB2 buffer pools, databases, page sets, tables, and indexes.

## Performance Advisor component

This component of the SQL Performance solution enables you to identify performance trends, compare recent performance with a baseline, identify usage patterns, and generate recommendations for improved performance.

As DBAs look for ways to improve database performance, their efforts often do not target areas that will show the highest return on their time investment. Some performance improvement areas that Performance Advisor targets are:

- reorganizing objects
- eliminating obsolete indexes
- identifying degraded objects

## Product compatibility

SQL Performance works with the following BMC Software products or solutions to perform common functions:

- MainView for DB2 – Data Collector
- OPERTUNE for DB2
- Pool Advisor for DB2
- System Performance for DB2

This ability further automates performance analysis functions and saves time when you are using SQL Performance with one or more of these BMC Software products or solutions.

## REORG Advisor component

The REORG Advisor feature of Performance Advisor generates performance improvement recommendations by identifying REORG candidates based on physical characteristics reported in RTS combined with performance degradations as shown in the APPTUNE statistics.

## SQL error reporting

This feature of SQL Performance enables you to identify SQL errors and view the corresponding SQLCA data and SQL text. Using SQL Performance, you can capture SQL error data automatically without having to change existing programs, and to determine which errors occurred when and to whom.

## SQL export function

This feature of SQL Performance exports SQL statement data for additional processing with a third-party analysis tool. It provides additional compatibility between SQL Performance and analysis tools from other vendors, extending your existing investment in software and training.

---

### Table 2  Features and benefits of SQL Performance (part 3 of 4)

<table>
<thead>
<tr>
<th>Feature</th>
<th>Benefit</th>
</tr>
</thead>
<tbody>
<tr>
<td>IN-SQL measurement</td>
<td>This feature of SQL Performance enables you to measure only the resources consumed during the execution of an SQL statement. IN-SQL measurements exclude the time spent on associated DB2 housekeeping tasks. This feature provides the most accurate data for measuring the cost of an SQL statement. Measurements based on IN-DB2 time are less accurate and might not reflect the actual cost of an SQL statement.</td>
</tr>
<tr>
<td>Intuitive interface</td>
<td>SQL Performance has a highly intuitive user interface that conforms to ISPF standards and includes a comprehensive online Help facility. This feature allows you to use the product with unparalleled ease and minimal supporting documentation.</td>
</tr>
<tr>
<td>Object reports</td>
<td>This feature of SQL Performance enables you to collect and analyze information about access, usage, and response times for DB2 buffer pools, databases, page sets, tables, and indexes.</td>
</tr>
<tr>
<td>Performance Advisor</td>
<td>This component of the SQL Performance solution enables you to identify performance trends, compare recent performance with a baseline, identify usage patterns, and generate recommendations for improved performance.</td>
</tr>
<tr>
<td>component</td>
<td></td>
</tr>
<tr>
<td>Product compatibility</td>
<td>SQL Performance works with the following BMC Software products or solutions to perform common functions:</td>
</tr>
<tr>
<td></td>
<td>- MainView for DB2 – Data Collector</td>
</tr>
<tr>
<td></td>
<td>- OPERTUNE for DB2</td>
</tr>
<tr>
<td></td>
<td>- Pool Advisor for DB2</td>
</tr>
<tr>
<td></td>
<td>- System Performance for DB2</td>
</tr>
<tr>
<td></td>
<td>This ability further automates performance analysis functions and saves time when you are using SQL Performance with one or more of these BMC Software products or solutions.</td>
</tr>
<tr>
<td>REORG Advisor component</td>
<td>The REORG Advisor feature of Performance Advisor generates performance improvement recommendations by identifying REORG candidates based on physical characteristics reported in RTS combined with performance degradations as shown in the APPTUNE statistics.</td>
</tr>
<tr>
<td>SQL error reporting</td>
<td>This feature of SQL Performance enables you to identify SQL errors and view the corresponding SQLCA data and SQL text. Using SQL Performance, you can capture SQL error data automatically without having to change existing programs, and to determine which errors occurred when and to whom.</td>
</tr>
<tr>
<td>SQL export function</td>
<td>This feature of SQL Performance exports SQL statement data for additional processing with a third-party analysis tool. It provides additional compatibility between SQL Performance and analysis tools from other vendors, extending your existing investment in software and training.</td>
</tr>
</tbody>
</table>
### SQL Performance features

#### Chapter 1 Overview of SQL Performance

SQL-level statistics: Using SQL Performance, you can obtain statistics and data at the SQL statement level about accounting, buffer, I/O, and lock activities. This feature provides you with the information needed to identify a costly SQL statement quickly and to undertake effective tuning.

Support for multiple DB2 subsystems: Using SQL Performance, you can analyze all of the DB2 subsystems across the sysplex where SQL Performance is installed, or support SQL Performance from a single Data Collector. This feature provides concurrent access to multiple DB2 subsystems and enables you to work with multiple BMC Software products using only one started task.

Support for static and dynamic SQL: SQL Performance captures performance data for both static and dynamic SQL statements and enables you to view both as a single workload. This feature allows you to capture data for all SQL statements—including dynamic SQL statements that usually cannot be captured—to determine the tuning needs of applications and users.

LOGSET archiving: Using SQL Performance, you can perform a comprehensive historical analysis for a specified period of time.

Variable and fixed-length collection intervals: This feature of SQL Performance enables the SQL Performance administrator to establish a set interval for data collection and to vary this interval as needed (for problem diagnosis, for example).

What-If Index analysis: This feature of SQL Performance simulates the effects of adding, dropping, or updating statistics for an index that uses cloned structures. You can also save DDL for your What-If Index changes to a data set, and then import those changes on another statement to see the effects of the changes on that statement. The What-If Index process is cumulative, so incremental changes followed by Explains enable you to see the effects of each change.

Workload Access Path Compare component: This component of the SQL Performance solution enables you to analyze and predict access path changes for both dynamic SQL and static SQL, compare access paths for a workload on different DB2 subsystems, and compare access paths for a workload on the same DB2 subsystem after changes have been made. You can then use the comparison results to predict changes in access paths within an application before migrating to a new release of DB2 or deploying a new release of an application. This component can share workload definitions and results with Workload Index Advisor.

Workload Index Advisor component: This component of SQL Performance makes it easy to define a workload, extract the SQL text, and recommend indexes. Using the online interface, you identify the source of the SQL statements and use qualifiers and object definitions to define the workload for which you want to provide index recommendations. This component can share workload definitions and results with Workload Access Path Compare.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Benefit</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL-level statistics</td>
<td>Using SQL Performance, you can obtain statistics and data at the SQL statement level about accounting, buffer, I/O, and lock activities. This feature provides you with the information needed to identify a costly SQL statement quickly and to undertake effective tuning.</td>
</tr>
<tr>
<td>Support for multiple DB2 subsystems</td>
<td>Using SQL Performance, you can analyze all of the DB2 subsystems across the sysplex where SQL Performance is installed, or support SQL Performance from a single Data Collector. This feature provides concurrent access to multiple DB2 subsystems and enables you to work with multiple BMC Software products using only one started task.</td>
</tr>
<tr>
<td>Support for static and dynamic SQL</td>
<td>SQL Performance captures performance data for both static and dynamic SQL statements and enables you to view both as a single workload. This feature allows you to capture data for all SQL statements—including dynamic SQL statements that usually cannot be captured—to determine the tuning needs of applications and users.</td>
</tr>
<tr>
<td>LOGSET archiving</td>
<td>Using SQL Performance, you can perform a comprehensive historical analysis for a specified period of time.</td>
</tr>
<tr>
<td>Variable and fixed-length collection intervals</td>
<td>This feature of SQL Performance enables the SQL Performance administrator to establish a set interval for data collection and to vary this interval as needed (for problem diagnosis, for example).</td>
</tr>
<tr>
<td>What-If Index analysis</td>
<td>This feature of SQL Performance simulates the effects of adding, dropping, or updating statistics for an index that uses cloned structures. You can also save DDL for your What-If Index changes to a data set, and then import those changes on another statement to see the effects of the changes on that statement. The What-If Index process is cumulative, so incremental changes followed by Explains enable you to see the effects of each change.</td>
</tr>
<tr>
<td>Workload Access Path Compare component</td>
<td>This component of the SQL Performance solution enables you to analyze and predict access path changes for both dynamic SQL and static SQL, compare access paths for a workload on different DB2 subsystems, and compare access paths for a workload on the same DB2 subsystem after changes have been made. You can then use the comparison results to predict changes in access paths within an application before migrating to a new release of DB2 or deploying a new release of an application. This component can share workload definitions and results with Workload Index Advisor.</td>
</tr>
<tr>
<td>Workload Index Advisor component</td>
<td>This component of SQL Performance makes it easy to define a workload, extract the SQL text, and recommend indexes. Using the online interface, you identify the source of the SQL statements and use qualifiers and object definitions to define the workload for which you want to provide index recommendations. This component can share workload definitions and results with Workload Access Path Compare.</td>
</tr>
</tbody>
</table>
SQL Performance for DB2 architecture

SQL Performance has the following main functions and components:

- Data Collector
- Report Manager
- Sysplex support

These functions and components help you to identify problems and tune costly SQL statements. Figure 1 illustrates the architecture of SQL Performance.

**Figure 1**  SQL Performance for DB2 (AFD) architecture

In addition, SQL Performance has some shared infrastructure components.

**Data Collector**

At the heart of SQL Performance’s data collection capabilities is the component called the Data Collector, which provides user access to performance data. Data Collectors are defined in the DOMPLEX option set. You can define any number of Data Collectors on each MVS system (only if each Data Collector is defined to only one
DOMPLEX). Each Data Collector can monitor any number of DB2 subsystems and can support up to 999 concurrent users. In addition, the Data Collector can issue commands on behalf of SQL Performance users to DB2, MVS, and OPERTUNE for DB2.

The Data Collector’s unique method of gathering information eliminates the need for expensive SQL traces by retrieving data directly from DB2 control blocks and trace data sets to satisfy requests from users. This collection method creates records of summarized accounting statistics, object access statistics, SQL text, and SQL error data—and writes them to LOGSET files.

Report Manager

The Report Manager is your interface to SQL Performance. The Report Manager’s main function is to take the records that are collected by the Data Collector and generate reports. All common SQL Performance reports use the Report Manager for display.

The Report Manager also provides direct links to the APPTUNE and SQL Explorer components of SQL Performance.

Sysplex support

SQL Performance is designed to run in a sysplex environment. One Data Collector from each MVS image in the sysplex can be defined in a DOMPLEX group. Each Data Collector can monitor all DB2s on its same MVS image and all Data Collectors in the DOMPLEX can communicate with each other, resulting in the data from all of the DB2s across the DOMPLEX being available to all DOMPLEX users. All users can view online reports that contain data from any or all DB2s in the DOMPLEX and commands can be issued to any DB2 in the DOMPLEX. Figure 2 on page 32 illustrates sysplex communication in an SQL Performance DOMPLEX.
SQL Performance administrators have the authority to define DOMPLEXes using the DOMPLEX Option Sets function of the Administration menu. For a complete explanation of DOMPLEX option sets, see the System and SQL Performance for DB2 Administrator Guide.

Shared infrastructure components

DB2 Product Configuration, NGL, and the Data Collector run within the DBC subsystem. Figure 3 on page 33 illustrates the architecture of the shared components. For more information about the functions of the components, see “Shared infrastructure components” on page 25.
For more information about managing these components, see the *System and SQL Performance for DB2 Administrator Guide*.

**SQL Performance administration**

SQL Performance uses profiles and option sets to control access to SQL Performance and related features in DB2. They are also used to set default values for subsystem-wide activities. Administrative functions are available only to users with administration authority (granted in the User Profile).

**User Profiles**

Each SQL Performance user is identified by a *User Profile*, which restricts access to SQL Performance functions and access for monitoring DB2. In addition, the User Profile is used to set defaults for display characteristics and function keys.

Individual profiles can be created and tailored to individual needs, or profile characteristics can be shared by many users by creating a Master Profile. The SQL Performance administrator can grant users the ability to change their own profiles or restrict their access to portions of the profile.
DOMPLEX and filter option sets

You can define the following types of option sets for managing SQL Performance:

- DOMPLEX option sets
- filter option sets

DOMPLEX option sets

In SQL Performance, the collection of data, the summarization of data, and the filtering of data are specific to DB2. Each DB2 subsystem that can be monitored is defined in a DOMPLEX option set. The DOMPLEX option set also defines the DB2 subsystems to be monitored and the LOGSET files used by the DOMPLEX. The collection, summarization, and filtering options set in the DOMPLEX option set apply to all programs and plans that are run on the DB2 subsystem.

DOMPLEX parameters also include a set of values that determine how APPTUNE is configured. You can modify these values to suit the needs of your site. These options allow you to specify the following operating characteristics:

- whether users are allowed to issue commands to DB2 and MVS from APPTUNE
- whether APPTUNE uses the DB2 user authorization table (SYSUSERAUTH) to enforce security for DB2 commands and traces
- whether hiperspace is used to stage records during the report-viewing process
- whether APPTUNE panels will be displayed using both upper- and lowercase characters or using uppercase characters only
- the style of date displayed on panels where the date occurs
- the symbol used to the left of the fractional portion of a number with decimal places

For a complete explanation of all common administrative functions, see the System and SQL Performance for DB2 Administrator Guide.

Filter option sets

The collection options set in the DOMPLEX option set apply to all programs and plans that are run on the DB2 subsystem. You can use filter option sets to set filtering options that apply to specific combinations of programs, plans, users, correlation IDs, and DB2 subsystems.
You can associate a specific filter option set to a DB2 subsystem in the DB2 definition in the DOMPLEX option set. You can specify only one filter option set to be used with a DB2 subsystem at a time, but many DB2 subsystems can use the same filter option set.

Table 3 lists the different collection options that you can specify through filters.

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>collection options</td>
<td>You can choose the types of data that will be collected by each DB2 subsystem. You can choose to collect any or all of the following types of data:</td>
</tr>
<tr>
<td></td>
<td>- buffer pool</td>
</tr>
<tr>
<td></td>
<td>- lock</td>
</tr>
<tr>
<td></td>
<td>- SQL text</td>
</tr>
<tr>
<td></td>
<td>- number of SQL statements</td>
</tr>
<tr>
<td></td>
<td>- object</td>
</tr>
<tr>
<td></td>
<td>You can also choose not to collect any data.</td>
</tr>
<tr>
<td>collection keys</td>
<td>You can choose to summarize data by the following values:</td>
</tr>
<tr>
<td></td>
<td>- program name</td>
</tr>
<tr>
<td></td>
<td>- plan name</td>
</tr>
<tr>
<td></td>
<td>- user ID</td>
</tr>
<tr>
<td></td>
<td>- correlation ID</td>
</tr>
<tr>
<td></td>
<td>- collection ID</td>
</tr>
<tr>
<td></td>
<td>- connection ID</td>
</tr>
<tr>
<td></td>
<td>- consistency token</td>
</tr>
<tr>
<td></td>
<td>- client application</td>
</tr>
<tr>
<td></td>
<td>- client user ID</td>
</tr>
<tr>
<td></td>
<td>- client workstation</td>
</tr>
<tr>
<td></td>
<td>- requesting location</td>
</tr>
<tr>
<td></td>
<td>- implicit qualifier</td>
</tr>
<tr>
<td></td>
<td>- thread type</td>
</tr>
<tr>
<td></td>
<td>- section number</td>
</tr>
<tr>
<td></td>
<td>- statement number</td>
</tr>
<tr>
<td></td>
<td>- dynamic SQL detail</td>
</tr>
<tr>
<td>resource-saving options</td>
<td>You can use the following techniques to limit the amount of data collected, subsequently reducing overhead.</td>
</tr>
<tr>
<td></td>
<td>- bypass timings and exception checks for fetches after the first fetch in cursor</td>
</tr>
<tr>
<td></td>
<td>- ignore literal values for numbers and strings in dynamic</td>
</tr>
<tr>
<td></td>
<td>All dynamic SQL statements that are otherwise the same are considered as one SQL statement.</td>
</tr>
<tr>
<td></td>
<td>- sample data instead of collecting all of it and extrapolate the results</td>
</tr>
</tbody>
</table>
SQL Performance security

Because some SQL Performance functions can affect DB2 performance, security mechanisms are provided to the SQL Performance administrator to control access to these functions.

Access to SQL Performance is controlled through IBM® RACF® or a similar MVS security system.

Authority to issue DB2 commands is controlled through the following mechanisms:

- User Profile options
- the DB2 catalog table, if the Enforce security via DB2 authorization table global option is set to Y
- an exit that allows you to specify a DB2 primary authorization ID that differs from the user ID
- the DB2 RACF®@ATH secondary authorization exit

You can also use the APPFILT command at any time to set or modify the filter for a DB2 subsystem.

For complete instructions for defining filter option sets (including host variables and exception processing), or for information about setting collection and summarization options, see the System and SQL Performance for DB2 Administrator Guide.

### SQL Performance security

You can choose whether to generate exception records and messages, issue WTO codes, or capture details for specific negative SQL codes.

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>exception thresholds and options</td>
<td>You can specify thresholds for the following values that will cause exception records to be created:</td>
</tr>
<tr>
<td></td>
<td>- elapsed time</td>
</tr>
<tr>
<td></td>
<td>- CPU time</td>
</tr>
<tr>
<td></td>
<td>- number of getpages</td>
</tr>
<tr>
<td></td>
<td>- number of synchronous I/Os</td>
</tr>
<tr>
<td></td>
<td>- deadlocks</td>
</tr>
<tr>
<td></td>
<td>- timeouts</td>
</tr>
<tr>
<td></td>
<td>- host variables</td>
</tr>
<tr>
<td></td>
<td>- exception WTOs</td>
</tr>
<tr>
<td></td>
<td>- exception-only rollups</td>
</tr>
<tr>
<td></td>
<td>- efficiency filters</td>
</tr>
<tr>
<td>negative SQL codes</td>
<td>You can choose whether to generate exception records and messages, issue WTO codes, or capture details for specific negative SQL codes.</td>
</tr>
</tbody>
</table>
Authorization for the following tasks is controlled through User Profile options:

- monitoring DB2
- accessing Data Collectors
- issuing commands to the Data Collector, MVS, and OPERTUNE

For more information about security in SQL Performance, see the *System and SQL Performance for DB2 Installation Guide*. 
Getting started with SQL Performance

This chapter presents the following topics:

- Logging on .................................................. 39
- Navigation .................................................. 40
- Panel layout ............................................... 41
- Online report layout ................................. 43
- System and SQL Performance for DB2 main menu ....................... 45
- Online assistance .................................. 47

Logging on

As part of the SQL Performance installation, one of the following logon mechanisms is created:

- a menu option for logging on under ISPF
- a CLIST for logging on under TSO

Select the ISPF option or execute the CLIST to log on to SQL Performance. Contact your SQL Performance administrator for the proper procedure to use at your site.

When you log on, the System and SQL Performance for DB2 logo panel (Figure 4 on page 40) is displayed briefly while the product initializes, followed by the System and SQL Performance for DB2 main menu (Figure 7 on page 46). This menu is displayed when multiple System and SQL Performance for DB2 products are installed.
Navigation

SQL Performance’s interface has been designed for quick and easy navigation through the product’s panels and reports. After you familiarize yourself with SQL Performance, you should be able to navigate throughout the product with minimal assistance from the supporting documentation.

Action codes are used to navigate between common reports. Action codes are listed in each common report header. To use an action code, type the one-letter code on the plus sign (+) at the left of a line of data, and press Enter. Another report containing related data is displayed.

**NOTE**

As you navigate in SQL Performance, you will cross over to APPTUNE panels and reports and SQL Explorer and Explain panels. Navigation in APPTUNE is identical to navigation in SQL Performance common panels and reports. Navigation in SQL Explorer and Explain panels is governed by those components or functions. For navigation conventions, see the component documentation.
SQL Performance panels have a number of common characteristics, including various standard elements that are displayed on all SQL Performance panels and online reports and the use of confirmation dialogs.

Figure 5 shows the common attributes that are shared by SQL Performance panels.

**Figure 5**  SQL Performance panels

---

**Table 4** describes the common elements of SQL Performance panels.

**Table 4**  Panel elements  (part 1 of 3)

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>panel ID</td>
<td>The panel ID displayed in the upper left corner of each panel. Display of the panel ID is optional. The default value is set in the User Profile. You can use the PANELID command to suspend the default temporarily.</td>
</tr>
<tr>
<td>environment identifier</td>
<td>The environment identifier is displayed to the right of the panel ID, separated from it by a slash (/). Possible values are</td>
</tr>
<tr>
<td>panel description</td>
<td>Each SQL Performance panel displays a description in the center of the top line of the panel.</td>
</tr>
</tbody>
</table>
Panel layout

Table 4  Panel elements (part 2 of 3)

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>time or line count indicator</td>
<td>On nonscrollable panels, the time is displayed in the upper right corner in the format hh:mm:ss, where:</td>
</tr>
<tr>
<td></td>
<td>■ hh is hours in the range 00–23</td>
</tr>
<tr>
<td></td>
<td>■ mm is minutes in the range 00–59</td>
</tr>
<tr>
<td></td>
<td>■ ss is seconds in the range 00–59</td>
</tr>
<tr>
<td></td>
<td>On scrollable panels, the current line number and the total number of lines is displayed in the format LINE nnn OF nnn.</td>
</tr>
<tr>
<td></td>
<td>If the panel is scrollable, the scroll amount (in the format Scroll ===&gt; ___) is also displayed at the end of the Command line. Each time you begin a SQL Performance session, the default scroll amount is retrieved from your User Profile and displayed here. You can change the scroll amount at any time, and the new value will remain in effect until you either change it or end your session. The following values are valid:</td>
</tr>
<tr>
<td></td>
<td>■ PAGE - Data is scrolled one full page at a time. A full page varies in size depending on the number of scrollable lines on the panel or report. For example, if there are 10 scrollable lines on the panel, data is scrolled 10 lines at a time.</td>
</tr>
<tr>
<td></td>
<td>■ HALF - Data is scrolled a half-page at a time. A half-page varies in size depending on the number of scrollable lines on the panel or report. For example, if there are 12 scrollable lines on the panel, data is scrolled 6 lines at a time.</td>
</tr>
<tr>
<td></td>
<td>■ CSR - The position of the cursor determines the number of lines scrolled. When scrolling down, position the cursor on the line to be displayed at the top of the scrollable area, and press F8. When scrolling up, place the cursor on the line to be displayed at the bottom of the scrollable area, and press F7. If CSR is specified and the scrolling keys are used when the cursor is not positioned on a scrollable portion of the panel, scrolling defaults to PAGE.</td>
</tr>
<tr>
<td></td>
<td>■ GRP - Data is scrolled one repeating group at a time. The first line of the next repeating group is displayed on the first line of the scrollable area. This value is valid only when viewing reports with repeating groups. If this value is specified, and there are no repeating groups on the panel, scrolling defaults to PAGE.</td>
</tr>
<tr>
<td></td>
<td>■ nnnn - Data is scrolled nnnn lines at a time, where nnnn is any number in the range 1 to 9999.</td>
</tr>
<tr>
<td>command line</td>
<td>The Command line can be displayed directly beneath the panel ID and description or at the bottom of the panel directly above the function keys. The default value for the position of the Command line is set in the User Profile.</td>
</tr>
</tbody>
</table>
SQL Performance online reports have a number of common characteristics, including various standard elements that are displayed on all online reports. Figure 6 shows the common elements that are shared by SQL Performance online reports.

**Figure 6  Common online report elements**

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>function keys</td>
<td>The function keys are displayed on the bottom two lines of the panel. Display of function keys is controlled by a parameter in the User Profile. The default values for each key can also be modified in the User Profile. The following defaults are shipped with SQL Performance:</td>
</tr>
</tbody>
</table>
|               | F1 - Help  F10 - Left  F19 - Up  
|               | F2 - Split  F11 - Right  F20 - Down  
|               | F3 - End  F12 - Cancel  F21 - Expand All  
|               | F4 - Sort A (ascending)  F13 - Home  F22 - Left  
|               | F5 - Sort D (descending)  F14 - Keys  F23 - Right  
|               | F6 - Showcmds  F15 - End  F24 - Retrieve  
|               | F7 - Up  F16 - Terse  
|               | F8 - Down  F17 - Rfind  
|               | F9 - Swap  F18 - Filter  
| messages      | SQL Performance messages are displayed directly below the Command line when the Command line is displayed at the top of the panel, and directly above the Command line when the Command line is displayed at the bottom of the panel. |
Table 5 describes the common characteristics of SQL Performance online reports.

### Table 5  Report elements (part 1 of 2)

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>report name</td>
<td>The name (report ID) of the report.</td>
</tr>
<tr>
<td>report description</td>
<td>A description of the report.</td>
</tr>
<tr>
<td>report format</td>
<td>The format in which data is displayed on the report; can be numeric (DATA) or graphical (GRAPH).</td>
</tr>
<tr>
<td>date</td>
<td>The current date in either the USA/ISO format (mm/dd) or European format (dd/mm), where mm is month in the range 01–12, and dd is day in the range 01–31.</td>
</tr>
<tr>
<td>time</td>
<td>The current time in the format hh:mm:ss, where:</td>
</tr>
<tr>
<td></td>
<td>● hh is hours in the range 00–23</td>
</tr>
<tr>
<td></td>
<td>● mm is minutes in the range 00–59</td>
</tr>
<tr>
<td></td>
<td>● ss is seconds in the range 00–59</td>
</tr>
<tr>
<td>source</td>
<td>The source of data (data set or Data Collector) for this report. If the source is a data set, the value displayed is DATA SET. If the source is a Data Collector, the Data Collector subsystem ID is displayed with the current status of the Data Collector.</td>
</tr>
<tr>
<td>status</td>
<td>If the source for this report is a Data Collector, one of the following values is displayed showing the status of the Data Collector:</td>
</tr>
<tr>
<td></td>
<td>● DOWN – The Data Collector is not active.</td>
</tr>
<tr>
<td></td>
<td>● ACTIVE – The Data Collector is active and available for use.</td>
</tr>
<tr>
<td></td>
<td>● STARTING – The Data Collector is starting but is not yet available.</td>
</tr>
<tr>
<td></td>
<td>● STOPPING – The Data Collector is stopping and is not available.</td>
</tr>
<tr>
<td>intvl interval start time</td>
<td>The interval of analysis. If the source is a Data Collector, the value mm/dd hh:mm is displayed, indicating the start time of the interval, where (date) mm is month in the range 01–12, dd is day in the range 01–31, (time) hh is hours in the range 00–23, and mm is minutes in the range 00–59.</td>
</tr>
<tr>
<td>actions</td>
<td>The action codes you can select to zoom to other workload analysis reports or to expand the current report. To zoom or expand, type the corresponding action code over the plus sign (+) beside the relevant object, and press Enter.</td>
</tr>
</tbody>
</table>

**Note:** The line containing the Source and Intvl fields is suppressed in TERSE mode. Additional information on TERSE is available in online Help. Type HELP TERSE on the Command line, and press Enter. For detailed information about the online Help facility, see “Online assistance” on page 47.
The System and SQL Performance for DB2 main menu (Figure 7 on page 46) provides access to all major System and SQL Performance product functions.

**NOTE**
The System and SQL Performance for DB2 main menu is only displayed when you have installed both the SQL Performance and System Performance for DB2 solutions.
The main menu contains the following options:

- **Use **System Performance Solution** to access the features and functions of the System Performance solution components.

- **Use** the options under SQL Performance Solution to access the features and functions of the SQL Performance solution components.

- **Use DOMPLEXes** to select a DOMPLEX for monitoring DB2 or to change from one active DOMPLEX to another.

- **Use Session Status** to view information about the status of the currently-active DB2 subsystems.

- **Use User Options** to view and modify the options used to tailor your session to your needs.

- **Use Log Operations** to create and modify user-defined collections of plans, programs, correlation IDs, and users representing the workload associated with a specific business function or individual.
Online assistance

- Use Administration to modify user profiles and option sets and to view zap maintenance.

**NOTE**

This option is displayed on your menu only if you have administration authority.

- Use Help to learn about the online Help facility, to see an overview of the System and SQL Performance products, and to explore online Help topics.

**NOTE**

This applies to the online Help system used by all System and SQL Performance products except OPER TUNE for DB2 and SQL Explorer for DB2.

- Use Exit to terminate your session.

- Use Summary of Changes to see a list of new features and functions in the current (and previous) product releases.

- Use About the System and SQL Performance Products to view copyright and trademark information about System and SQL Performance and to obtain Internet and e-mail addresses for BMC Software Sales and Customer Support.

The main menu that is displayed depends on the System and SQL Performance for DB2 products and solutions that are installed and active at your site. If multiple products or solutions are installed, the main menu displayed will reflect the active product mix. Figure 7 on page 46 is an example of the main menu that is displayed when all System and SQL Performance products for DB2 are installed.

Online assistance

Wherever you go in SQL Performance, online assistance is only a keystroke away. Every panel, report, and field has traditional, context-sensitive online Help that is available by pressing F1 (Help).

Online Help for a report provides a basic explanation of the report, the action codes that can be used, and static descriptions of the values in the report fields. Action codes are available on all SQL Performance reports.
You can access online Help from the APPTUNE and Index Component Main Menu to get either an explanation about how to use online Help or an overview of the product. The overview also includes lists of the following major Help topics:

- commands
- messages
- reports
- panels
- tutorial topics

Each major topic branches to additional items that provide more detail.

SQL Performance online Help is context-sensitive. Move the cursor to a text, input, or output area of any panel or report, and press F1 for Help that is specific to that field, panel, or report.

You can also use the HELP command from any panel or report to get Help on any topic you specify. For a complete explanation of the HELP command, see the APPTUNE for DB2 User Guide.
Managing performance with Performance Advisor

This chapter presents the following topics:

Overview of Performance Advisor .................................................. 49
Why you need Performance Advisor ............................................. 50
  Reorganizing objects ............................................................... 50
  Eliminating obsolete indexes ................................................... 51
  Identifying degraded objects .................................................... 51
Task summary ............................................................................. 52
Setting up Performance Advisor ................................................... 52
  Creating tables ......................................................................... 53
  Configuring automatic data collection ....................................... 54
  Loading data into Performance Advisor PADB ......................... 55
  Purging old data from Performance Advisor PADB ................... 58
  Reporting based on the contents of tables ................................. 58
Using REORG Advisor ............................................................... 60
  Integrating REORG Advisor with BMC DASD MANAGER PLUS. . 60

Overview of Performance Advisor

Using the Performance Advisor component, you can create and maintain a Performance Advisor Database (PADB) in DB2. PADB tables contain the following types of data:

- statement and object performance data collected by APPTUNE
- errors, exceptions, SQL text, and index usage patterns identified by APPTUNE
- daily snapshots of real-time statistics (RTS) collected by DB2
- recommendations for improving system and application performance
Why you need Performance Advisor

With this data, you can

- identify performance trends
- compare recent performance against a baseline
- identify usage patterns
- implement performance improvements

You can maintain one central PADB (*centralized configuration*), or a PADB at each DB2 subsystem or data sharing group where data is collected (*localized configuration*).

**Why you need Performance Advisor**

DBAs commonly look for ways to improve database performance. However, their efforts frequently fail to target areas that would produce the highest return on their time investment. Performance Advisor effectively targets the following critical areas:

- reorganizing objects
- eliminating obsolete indexes
- identifying degraded objects

**Reorganizing objects**

Many DBAs reorganize application objects based on a schedule, or by analyzing statistics and reorganizing when designated thresholds are reached. However, for some types of workload, disorganized objects do not degrade application performance. Reorganizing objects that do not degrade performance is a waste of resources.

Performance Advisor addresses this problem with a feature called REORG Advisor (REXX EXEC program IODADREO). REORG Advisor analyzes

- physical characteristics reported in RTS
- performance degradations shown in the APPTUNE data (in the table containing daily object performance statistics)

REORG Advisor then identifies objects that are candidates for reorganization based on two criteria:

- the object’s level of disorganization
- the object’s level of performance degradation since the last reorganization
That is, REORG Advisor recommends an object for reorganization only if it is disorganized and is experiencing performance degradation.

REORG Advisor loads candidate objects into a REORG candidate table. Columns in the table reflect the calculations performed in analyzing the objects. The REORG_CANDIDATE column contains Y or N to reflect REORG recommendations. You can query this table to get a list of recommendations. Alternatively, you can use built-in integration with the DASD MANAGER product’s BMCTRIG utility to trigger automatic reorganizations. In either case, reorganizing only the recommended objects avoids wasting valuable CPU time.

Eliminating obsolete indexes

Over years of service and multiple versions of DB2, many applications accumulate a lot of indexes. It is not uncommon for some tables to have up to 6 indexes. Due to changes in the optimizer logic over time, some of these indexes are likely to be obsolete. Using the PMDB, you can easily identify indexes that have not been used in a while. Figure 8 shows an example query.

Figure 8  Example index query

<table>
<thead>
<tr>
<th>SQL command</th>
</tr>
</thead>
<tbody>
<tr>
<td>SELECT DBNAME, OBNAME, PARTITN FROM BMCSFTWR.INDEX_USAGE WHERE READTM IS NULL OR READTM &lt; CURRENT_TIMESTAMP - 90 DAYS;</td>
</tr>
</tbody>
</table>

Dropping an obsolete index produces CPU performance benefits every time you execute an INSERT or DELETE command on the associated table, or update columns that were in the dropped index.

Identifying degraded objects

Monitors see what is happening in real time. As such, they do a poor job of letting you know that performance is degrading over time. In contrast, using data in the PADB, you can identify objects that are degrading before the degradation becomes problematic. Catching degraded objects early allows you to manage and tune them, resulting in improved performance. Figure 9 on page 52 shows an example query that identifies degraded objects.
Task summary

You must perform several tasks to set up and use the Performance Advisor component. Figure 10 is a high-level overview of these tasks. For more information, see Appendix A, “Performance Advisor reference.”

Figure 10  Performance Advisor task overview

Setting up Performance Advisor

Complete the following tasks to prepare to use Performance Advisor:

- “Creating tables” on page 53
- “Configuring automatic data collection” on page 54
- “Loading data into Performance Advisor PADB” on page 55

SQL Performance for DB2 includes several samples in the LLQSAMP library (SAMP library), where LLQ is DB, XX, BB, and UBB.
Creating tables

SQL Performance for DB2 provides a sample job, PASETUP, that you can use to create the tables for the PADB. You can find PASETUP and the sample members that PASETUP uses to create the tables in the SAMP library. Table 6 provides a list of the sample members that you can use to create each table.

Table 6  Performance Advisor sample members for creating tables

<table>
<thead>
<tr>
<th>SAMP member</th>
<th>Creates table(s) containing</th>
</tr>
</thead>
<tbody>
<tr>
<td>IODDDLIN</td>
<td>APPTUNE statistics accumulated by collection interval</td>
</tr>
<tr>
<td>IODDDLBA</td>
<td>APPTUNE statistics aggregated for baseline comparisons</td>
</tr>
<tr>
<td>IODDDLDA</td>
<td>APPTUNE statistics accumulated daily</td>
</tr>
<tr>
<td>IODDDLWE</td>
<td>APPTUNE statistics aggregated weekly</td>
</tr>
<tr>
<td>IODDDLMO</td>
<td>APPTUNE statistics aggregated monthly</td>
</tr>
<tr>
<td>IODDDDLEV</td>
<td>APPTUNE statistics representing exception and error events</td>
</tr>
<tr>
<td>IODDDLTE</td>
<td>aggregated SQL text collected by APPTUNE</td>
</tr>
<tr>
<td>IODDDLIL</td>
<td>aggregated index usage information collected by APPTUNE</td>
</tr>
<tr>
<td>IODDDLRT</td>
<td>real-time statistics accumulated daily, used by all releases of DB2</td>
</tr>
<tr>
<td>IODDDLRT9</td>
<td>real-time statistics accumulated daily, for DB2 Version 9 and later releases</td>
</tr>
<tr>
<td>IODDDLRA</td>
<td>REORG candidates for the REORG Advisor feature</td>
</tr>
<tr>
<td>IODDDLXA</td>
<td>rules for Exception Advisor</td>
</tr>
</tbody>
</table>

Notice that some tables contain aggregated data and others do not. The baseline, weekly, monthly, SQL text, and index usage tables aggregate input data with data already in the table. When you load and purge data, you will have different procedures for aggregated and nonaggregated tables. For more information, see “Loading data into Performance Advisor PADB” on page 55 and “Purging old data from Performance Advisor PADB” on page 58.

**NOTE**
You can create Performance Advisor tables in one central location (centralized PADB) or on each DB2 subsystem or data sharing group (localized PADB).
**Before you begin**

Have the following information available:

- DB2 subsystem ID
- DB2 LOAD library name
- DB2 DSNEXIT library name
- SQL Performance high-level qualifier
- which tables you want to create
- ID you want to use as the table owner
- whether you want to rename tables or remove any columns that are not designated as NOT NULL

**To create Performance Advisor tables**

1. Review the instructions in SAMP member PASETUP.

2. Edit PASETUP (as instructed in the member) to create the tables you want.

3. *(optional)* Edit referenced members to rename tables, change the creator, or remove unneeded columns.

**NOTE**

You can remove a column that is not designated as NOT NULL. If you rename tables or remove columns you must make adjustments when you load data in the PADB. For more information, see “Loading data into Performance Advisor PADB” on page 55.

4. Run PASETUP to create the tables.

5. Review the tables to make sure the results match your expectations.

**Configuring automatic data collection**

1. Copy the NGLARCH member from SAMPLIB to your SYS1.PROCLIB (or equivalent) started task.

   Doing so allows for the automatic submission of the NGL9ARCH utility each time a Data Collector log file becomes full or each time the Data Collector SWITCH command is issued. NGLARCH creates an archive data set. This data set will contain all data from the log file that was filled or switched from.
2 At the DOMPLEX option set panel for your DOMPLEX, expand **DB2 Monitor List** to define the SQL statistics collection interval for each DB2 subsystem.

For best results, align your interval definitions so that an interval ends at midnight each day. This approach aligns the daily, weekly, and monthly tables. Otherwise, the table aggregations will include data from the end of the aggregation period until the end of the interval. For example, if your SQL statistics collection interval ends at 2 A.M. instead of midnight, your daily tables will cover the period from 2 A.M. on the indicated day until 2 A.M. the following day; however, the INTVTIM column for this table will show a time of midnight.

3 Set up an automatic Data Collector SWITCH.

A log file switch should occur when the last SQL statistics collection interval concludes each day and the APPTUNE UNLOAD process completes. The switch tells the NGLARCH started task to populate your archive data sets with data for the Performance Advisor tables (typically, shortly after midnight), and can be implemented using an automated Data Collector SWITCH command. Implement this using the system automation tool of your choice.

---

**TIP**

You can set the switch with an automated Data Collector SWITCH command using the system automation tool of your choice. You can also accomplish this by submitting a job that runs the DOMBSWIT utility.

---

**Loading data into Performance Advisor PADB**

This task uses the DMDBMERG utility. For reference information about the utility, see Appendix A, “Performance Advisor reference.”

**Before you begin**

The first time you generate data for aggregated tables, save the output data set containing load utility records (DMDBMERG utility DB2LOAD DD statement) for use as input on the next execution of the utility (DMDBMERG utility LOADIN DD statement).

To facilitate this process, define a set of generation data groups (GDGs) to hold at least two generations of data sets with load utility records. Use SAMP member IODGDG to set this up.
To load APPTUNE statistics into the PADB

1. Load the data you want using the following SAMP jobs as models:
   - IODLODBA (baseline)
   - IODLODDA (daily)
   - IODLODEV (events)
   - IODLODIN (interval)
   - IODLODIU (index usage)
   - IODLODMO (monthly)
   - IODLODTE (text)
   - IODLODWE (weekly)

2. Modify the QUALIFIER statement to reflect the data you want to load, or remove the QUALIFIER statement to load all data. For more information, see “QUALIFIER statement” on page 213.

3. After the initial run of each job, add the LOADIN DD statement with the previous generation of the DB2LOAD data set.

4. Modify the PURGE_DAYS statement, if present, to reflect the number of days for which you want to maintain data.

5. Modify the TBCREATOR statement to reflect the owner of your tables.

6. Modify the TABLE statement to reflect the tables you are maintaining. Do not change the table names on this statement.

7. The examples in SAMP use INTERVAL( START(-1) FOR(1) ). This INTERVAL statement uses entries in the COPYDIR (directory of archive data sets) to select the archive data sets that contain data for the previous day and makes it unnecessary to code specific input data sets or a specific date-time range. For more information, see “Using COPYDIR” on page 57.

8. If you have removed columns or changed table names, complete the following steps:
   - A. Run the GENPERF step of the SAMP jobs once by itself.
   - B. Edit and save the resulting LOADCTL output to reflect the columns and table names.
   - C. Use this modified result as the SYSIN DD of the LOADPERF step from this point forward.
Loading data into Performance Advisor PADB

Chapter 3  Managing performance with Performance Advisor  57

Using COPYDIR

The COPYDIR data set is a VSAM file that is created when you install the products. COPYDIR maintains information about the archive data sets that the NGLARCH started task creates. Archive data set entries are retained until all directory slots are used or entries are removed. When all of the available slots are filled, the oldest entry is replaced by the newest entry.

When an INTERVAL statement is used in the DMDBMERG utility, a list of data sets is passed to the utility based on data in the COPYDIR. If DMDBMERG attempts to allocate an uncataloged archive data set the utility will fail. You can use the DOMARCB utility to remove uncataloged entries from the COPYDIR and reorganize the COPYDIR to improve performance of DMDBMERG. For more information, see the System and SQL Performance Administrator Guide.

To load real-time statistics into the PADB

1. For a localized PADB, where you maintain the statistics on each originating system, run SAMP job PARTSUL once daily on each system. For data-sharing groups, run this on only one member of the group.

2. For a centralized PADB, where you maintain the statistics in one central location, run SAMP jobs PARTSUC1 and PARTSUC2 as follows:

   A. Run PARTSUC1 once daily on each system. For data-sharing groups, run this on only one member of the group.

   B. Run PARTSUC2 once daily at your centralized location, modifying SYSREC to capture all data unloaded by PARTSUC1.

To populate the REORG candidate table

1. Configure REORG Advisor thresholds and options. For more information about the configuration options that are available, see the instructions in member IODPAREO.

2. Submit a daily job to execute REORG Advisor by submitting a job based on SAMP member IODPAREO.

TIP
Submit the jobs once daily after you have modified them for your needs.
Purging old data from Performance Advisor PADB

A maintenance task for your PADB is purging older data from the PADB.

For aggregated tables, the SAMP members for loading aggregated tables include the specification of PURGE_DAYS. Any data older than the specified PURGE_DAYS value is automatically removed and no longer appears when the data is loaded to the tables.

For non-aggregated tables and daily real-time statistics accumulations, use a DELETE SQL statement to purge old data from the tables based on the EVNTTIME column for the following tables:

- STMT_ERRORS
- STMT_EXCEPTIONS
- STMT_EXCEPTIONS_HV
- STMT_EXCEPTIONS_OB

For other tables, use a DELETE SQL statement to purge old data based on the INTVTIME column. Figure 11 shows examples of DELETE SQL statements for purging data.

**Figure 11  Example DELETE SQL statements**

```sql
DELETE FROM BMCSFTWR.DAILY_OBJ_STATISTICS
WHERE INTVTIME < CURRENT_TIMESTAMP - 30 DAYS;

DELETE FROM BMCSFTWR.STMT_ERRORS
WHERE EVNTTIME > CURRENT_TIMESTAMP - 30 DAYS;
```

Reporting based on the contents of tables

You can generate batch reports using DB2 LOAD utility records generated by DMDBMERG as input. See an example of this in SAMP member IODRPTLD.

To generate batch reports based on DMDBMERG LOAD utility records, specify a LOADIN DD statement identifying the input data.
You can also query the tables to learn about the execution and performance characteristics of your system. SAMP library member QUERIES contains sample queries you can use.

**NOTE**
The following examples assume a localized database, where the tables contain data for a single DB2 instance (SSID or data sharing group). If you use a centralized database, you can use the SMFID, LDB2NAME, QWHSSID, and QWHADESIGN columns to differentiate between DB2 instances.

For example, to get a daily picture of object performance as indicated by getpages per synchronous I/O, issue the following query:

```sql
SELECT INTVTIME, DBNAME, OBNAME, PARTITN,
    CASE WHEN SYNCIO=0 THEN GETPAGES
     ELSE GETPAGES/SYNCIO
    END AS "GETPAGES PER SYNCIO"
FROM BMCSFTWR.DAILY_OBJ_STATISTICS
ORDER BY DBNAME, OBNAME, PARTITN, INTVTIME;
```

To find out which users and statements accessed a particular object, issue the following query:

```sql
SELECT INTVTIME, AUTHID, PROGRAM, TEXTHASH
FROM STMT_STATISTICS_OB
WHERE DBNAME='MYDB' AND OBNAME='MYTABLE';
```

To see the associated SQL text, issue the following query:

```sql
SELECT SQLTEXT
FROM BMCSFTWR.STMT_TEXT
WHERE TEXTHASH='0123456789ABCDEF0123'
ORDER BY SQLTEXT#;
```

To see a list of indexes that have not been read in the past 90 days, issue the following query:

```sql
SELECT DBNAME, OBNAME, PARTITN
FROM BMCSFTWR.INDEX_USAGE
WHERE READTM IS NULL
    OR READTM < CURRENT_TIMESTAMP - 90 DAYS;
```
Using REORG Advisor

The REORG Advisor is a REXX® EXEC program, IODADREO, that identifies potential candidate objects for REORG, based on two criteria:

- the level of disorganization of the object
- the level of performance degradation for the object since the last REORG occurred.

The advisor helps you avoid performing unnecessary reorganizations for tables and indexes that are disorganized, yet do not show degraded performance since the time of the last REORG. To make these recommendations, the advisor uses data from the daily real-time stats tables (BMCTABLESPACESTATS, BMCV8TBLSPACESTATS, and BMCINDEXSPACESTATS), and the daily object statistics table (DAILY_OBJ_STATISTICS).

The advisor creates and maintains a REORG candidate table. This table contains columns that reflect the calculations performed in analyzing the objects and a column, REORG_CANDIDATE, containing a value of Y or N to reflect REORG recommendations. You can query this table to get a list of recommendations or you can use integration with DASD MANAGER BMCTRIG to provide the recommendations and trigger automatic REORGs.

Verify that plan DSNREXX and its packages have been bound. A bind job for DB2 REXX is available in DB2 install DSNSAMP library member DSNTIJRX.

To see a list of REORG recommendations, issue the following query on the REORG candidate table:

```sql
SELECT LDB2NAME, DBNAME, PSNAME, PART, TYPE
FROM BMCSFTWR.REORG_CANDIDATE_TABLE
WHERE REORG_CANDIDATE = 'Y';
```

Integrating REORG Advisor with BMC DASD MANAGER PLUS

If you also have BMC DASD MANAGER PLUS for DB2, you can integrate the REORG Advisor feature with the BMCTRIG feature. The following sections give you more information about integrating these features.

For more information, see the DASD MANAGER technical bulletin “You can use performance data from SQL Performance for DB2 to help DASD MANAGER PLUS evaluate when to run a reorganization” and the DASD MANAGER PLUS for DB2 User Guide.
Working with object sets

This chapter presents the following topics:

Overview of object sets .................................................. 61
Description of object set patterns .................................. 63
Inclusion versus exclusion of objects from the workload .... 64
Defining an object set .................................................... 65
Managing object set definitions ................................. 68
  Selecting an object set to use .................................. 68
  Editing an object set ................................................. 68
  Renaming an object set ........................................... 70
  Copying an object set ............................................... 71
  Viewing an object set ............................................... 71
Delet ing an object set .................................................. 72

Overview of object sets

Object sets contain naming patterns that identify the objects for which you want to extract SQL. Within an object set, you can create naming patterns for

- client IDs (CL)
- correlation IDs (CR)
- DBRM names (DM)
- package names (PG)
- plan names (PL)
- user IDs (US)

SQL Performance uses these naming patterns combined with source parameters in which you define such things as the location and source format to construct a workload from which SQL will be extracted (as shown in Figure 12 on page 62).
The object set definition is applied to the workload source to extract SQL. The source types include:

- APPTUNE archived trace data sets
- BMC Performance Database
- DB2 catalog
- DB2 statement cache
- a DBRM library
- SQL text data set

Each source type supports different object types. For example, if the source from which you are extracting SQL is the DB2 statement cache, the types of objects for which you can define naming patterns include DBRM names, package names, and user IDs.

**NOTE**

You do not need to create object set definitions if your source is a SQL text data set or DBRM source.
**Description of object set patterns**

Table 7 describes the types of objects for which you can specify naming patterns within an object set and the source type that support that naming pattern.

<table>
<thead>
<tr>
<th>Object type</th>
<th>Description</th>
<th>Applicable sources</th>
</tr>
</thead>
<tbody>
<tr>
<td>client IDs (CL)</td>
<td>The name pattern of the client IDs is composed of the following values:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ <strong>Client ID</strong>: The name pattern for a client ID. This value can contain up to 16 characters.</td>
<td>■ APPTUNE trace data sets (^a)</td>
</tr>
<tr>
<td></td>
<td>■ <strong>Workstation</strong>: The name pattern for the workstation name to associate with the client ID. This value can contain up to 18 characters.</td>
<td>■ BMC Performance Database tables (^a)</td>
</tr>
<tr>
<td></td>
<td>■ <strong>Application</strong>: The name pattern for the application name to associate with the client ID value. This value can contain up to 32 characters.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>You can use wildcard characters (*) as part of the pattern for each value.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Format</strong>: <code>user.workstation.application</code></td>
<td></td>
</tr>
<tr>
<td>correlation IDs (CR)</td>
<td>The name pattern for the correlation ID.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>This value can contain up to 12 mixed-case characters. You can use a wildcard character (*) as part of the pattern.</td>
<td>■ APPTUNE trace data sets (^a)</td>
</tr>
<tr>
<td></td>
<td><strong>Format</strong>: <code>name</code></td>
<td>■ BMC Performance Database tables (^a)</td>
</tr>
<tr>
<td>DBRM names (DM)</td>
<td>The identifier for a DBRM. This identifier is composed of the following values:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ <strong>Plan name</strong>: The name pattern for a plan name. This value can contain up to 8 uppercase characters. Leave the field blank for an implied wildcard.</td>
<td>■ APPTUNE trace data sets (^a)</td>
</tr>
<tr>
<td></td>
<td>■ <strong>DBRM name</strong>: The name pattern for the DBRM name. This value can contain up to 8 uppercase characters.</td>
<td>■ BMC Performance Database tables (^a)</td>
</tr>
<tr>
<td></td>
<td>You can use wildcard characters (*) as part of the pattern for each value.</td>
<td>■ DB2 catalog</td>
</tr>
<tr>
<td></td>
<td><strong>Format</strong>: <code>planName.dbrmName</code></td>
<td>■ DB2 statement cache</td>
</tr>
</tbody>
</table>
Inclusion versus exclusion of objects from the workload

You can create naming patterns for objects that you want to include and exclude from a workload. You use a plus sign (+) to mark patterns for objects that you want to include within the workload, and a minus sign (-) to mark patterns to exclude.

Table 7  Object types for object definitions (part 2 of 2)

<table>
<thead>
<tr>
<th>Object type</th>
<th>Description</th>
<th>Applicable sources</th>
</tr>
</thead>
</table>
| package names | The identifier for a package. This identifier is composed of the following values:  
| (PG)          | - Collection ID: The name pattern for a collection ID to associate with the package name. This value can contain up to 128 characters. You must include mixed-case characters in quotes.  
|               | - Package name: The name for a package. This value can contain up to 8 characters. You must include mixed-case characters in quotes.  
|               | - Version: The name pattern for the version to associate with the package name. This value can contain up to 122 characters. This value can be mixed case and can contain blanks if you include them in quotes.  
|               |   You can use the keyword LAST in place of the version to obtain the most recent version of the package.  
|               |   You can use wildcard characters (*) as part of the pattern for each value.  
|               |   Format: collID.packageName.version                                           | - APPTUNE trace data sets  
|               |                                                                             | - BMC Performance Database tables  
|               |                                                                             | - DB2 catalog  
|               |                                                                             | - DB2 statement cache                                                          |
| plan name (PL)| The name of the plan. This value can contain up to 8 uppercase characters. You can use a wildcard character (*) as part of the pattern.  
|               | Format: name                                                                 | - APPTUNE trace data sets  
|               |                                                                             | - BMC Performance Database tables  
|               |                                                                             | - DB2 catalog  
| user IDs (US) | The name pattern for a user ID. This value can contain up to 8 uppercase characters. You can use a wildcard character (*) as part of the pattern.  
|               | Format: name                                                                 | - APPTUNE trace data sets  
|               |                                                                             | - BMC Performance Database tables  
|               |                                                                             | - DB2 statement cache                                                          |

To use this object type, the associated collection option must have been set to Y (Yes) in the APPTUNE filter parameters while the APPTUNE Data Collector monitored the SQL. For more information about filtering, see the System and SQL Performance for DB2 Administrator Guide.
Defining an object set

This procedure explains how to define an object set that contains the naming patterns that identify the objects to include in the workload.

Before you begin

Review the information in “Overview of object sets” on page 61 to determine the naming patterns that you will need to define.

To define an object set

1. On the Performance Advisor panel, select 1 (Workload Access Path Compare) or 2 (Workload Index Advisor) and press Enter.

2. On the Workload Access Path Compare Menu or Index Advisor Menu, select 1 to define your workload.

The Extract SQL for Workload panel (Figure 13) is displayed. The top of this panel includes options that allow you to select, view, or create workloads.

---

**EXAMPLE**

Assume that you had the following plan that included the specified packages:

**Plan:** ACCOUNT
**Packages:** PAYROLL.%, ACCTPAY.%, ACCTREC.%, and ESCROW.%

If you want to include all of the packages in the workload with the exception of the PAYROLL.% packages, you could specify the following inclusion and exclusion rules:

- **+ PL ACCOUNT**
- **- PG PAYROLL**

Doing so includes the ACCTPAY, ACCTREC, and ESCROW packages in the workload, but excludes the PAYROLL package.
3 On the Extract SQL for Workload panel, type S next to Select to edit an Object Set Definition.

4 In the Object Set Data Set field, enter the name of the data set that will contain your definitions for object sets.

If the data set does not exist, SQL Performance allocates it.

5 Ensure that the Object Set Name field is blank and press Enter.

6 At the Edit Object Set Specification panel (Figure 14), enter the name for this set of naming patterns in the Current Object Set Name field.

**Figure 14  Object Set Specification panel (PSSWC180)**

7 In Remarks field, enter a description of the objects.

8 Use one of the following methods to create a new name pattern to identify the objects that you want to include in the workload.

**TIP**

The Insert from popup option guides you through a step-by-step process of creating a name pattern. The Insert option allows you to create name patterns in a free-form list.

For information about the values you enter to create a name pattern, see Table 7 on page 63.

- to use the popup method to create a new name pattern

1. In the Act field, type ?.

The Enter Name Pattern Specification panel (Figure 15 on page 67) is displayed.
Figure 15   Enter Name Specification panel (PSSWC185)

2. In the Include/Exclude field, type 1 (Include) or 2 (Exclude) to indicate whether you want to include or exclude objects that match that pattern from the workload.

3. In the Pattern Type field, enter the number corresponding to the object type of the object for which you are creating a name pattern and press Enter to display the appropriate Enter <objectType> Specification panel.

4. Enter the values that will make up the name pattern and press Enter.

- insert a new line to create a new name pattern

1. In the Act field, type I and press Enter to create a new line.

2. In the Incl/Excl field on the new line, enter + (include) or - (exclude) to indicate whether you want to include or exclude objects that match the name pattern from the workload.

3. In the Obj Type field, enter the two-letter code to indicate the type of object for which you are creating a naming pattern.

4. In the Name or Name Pattern field, enter the name pattern of the objects you want to include and press Enter.

9 After you finish adding objects and naming patterns, press PF3 to return to the previous panel.
Managing object set definitions

You can manage object set definitions as needed to keep them up-to-date with the objects that you need to define within the workload. You can edit the object set to include or remove object types and name patterns. You can also create, rename, and delete object sets.

Selecting an object set to use

1. On the Performance Advisor panel, select 1 (Workload Access Path Compare) or 2 (Workload Index Advisor) and press Enter.

2. On the Workload Access Path Compare Menu or Index Advisor Menu, select 1 to define your workload and press Enter to display the Extract SQL for Workload panel (Figure 13 on page 65).

3. On the Extract SQL for Workload panel, type S next to Select to edit an Object Set Definition.

4. In the Object Set Data Set field, type the name of the data set that contains your definitions for object sets.

5. In the Object Set Name field, type the name of the object set or enter the % wildcard character and press Enter.

6. If you specified a wildcard character, type S next to the object set that you want to use on the Object Set List panel and press Enter.

You return to the Extract SQL for Workload panel.

Editing an object set

1. On the Performance Advisor panel, select 1 (Workload Access Path Compare) or 2 (Workload Index Advisor) and press Enter.

2. On the Workload Access Path Compare Menu or Index Advisor Menu, select 1 to define your workload and press Enter.

3. On the Extract SQL for Workload panel, type S next to Select to edit an Object Set Definition.
4 In the **Object Set Data Set** field, enter the name of the data set that contains your definitions for object sets.

5 On the Extract SQL for Workload panel (Figure 13 on page 65), perform one of the following operations:

- If you know the name of the object set, enter the name in the **Object Set Name** field, and press **Enter**.

- If you want to pick the name of the object set from a list, type % and press **Enter**.

  On the Object Set List panel (Figure 16), enter **E** next to the object set you want to edit and press **Enter**.

**Figure 16 Object Set List panel (PSSWC175)**

6 On the Edit Object Set Specification panel (Figure 14 on page 66), edit the object set as needed:

<table>
<thead>
<tr>
<th>Action</th>
<th>Steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>change an existing object by typing over a line</td>
<td>Change the existing values in the following fields by typing over them:</td>
</tr>
<tr>
<td></td>
<td>- <strong>Incl/Excl</strong></td>
</tr>
<tr>
<td></td>
<td>- <strong>Obj Type</strong></td>
</tr>
<tr>
<td></td>
<td>- <strong>Name or Name Pattern</strong></td>
</tr>
</tbody>
</table>
| I - insert a new object by inserting a line in the list | 1. In the **Act** field, type **I**.  
2. In the **Incl/Excl** field, enter + (include) or - (exclude) to indicate whether you want to include or exclude objects that match the name pattern from the workload.  
3. In the **Obj Type** field, enter the two-letter code to indicate the type of object for which you are creating a naming pattern. (For information, see Table 7 on page 63.)  
4. In the **Name or Name Pattern** field, enter the name pattern of the objects you want to include or exclude and press **Enter**. |
Renaming an object set

<table>
<thead>
<tr>
<th>Action</th>
<th>Steps</th>
</tr>
</thead>
</table>
| ? - insert a new object by using pop-up menus | 1. In the Act field, type ?.  
2. In the Enter Name Pattern Specification panel, type 1 (Include) or 2 (Exclude) to indicate whether you want to include or exclude objects that match that pattern from the workload.  
3. In the Pattern Type field, enter the number corresponding to the object type of the object for which you are creating a name pattern and press Enter.  
4. On the Enter <objectType> Specification panel, enter the values that will make up the name pattern. (For information, see Table 7 on page 63.)  
5. Press Enter. |
| D - delete an object           | In the Act field next to the object specification that you want to delete, type D and press Enter.                               |
| R - repeat an object           | In the Act field next to the object specification that you want to repeat, type R and press Enter. You can modify the Incl/Excl, Object Type, and Name or Name Pattern fields by typing over them. |

**Renaming an object set**

1. On the Performance Advisor panel, select 1 (Workload Access Path Compare) or 2 (Workload Index Advisor) and press Enter.

2. On the Workload Access Path Compare Menu or Index Advisor Menu, select 1 to define your workload and press Enter.

3. On the Extract SQL for Workload panel (Figure 13 on page 65), type S next to Select to edit an Object Set Definition.

4. In the Object Set Name field, enter %.

5. In the Object Set Data Set field, enter the name of the data set that contains the object set definition that you want to rename and press Enter.

6. On the Object Set List panel (Figure 16 on page 69), enter R next to the object set you want to rename and press Enter.

7. On the Rename Object Set Name panel, enter a new name for the object set and press Enter.

**NOTE**

Do not enter a name of an object set that already exists. You cannot overwrite an existing object set when renaming one.
Copying an object set

1. On the Performance Advisor panel, select 1 (Workload Access Path Compare) or 2 (Workload Index Advisor) and press Enter.

2. On the Workload Access Path Compare Menu or Index Advisor Menu, select 1 to define your workload and press Enter.

3. On the Extract SQL for Workload panel (Figure 13 on page 65), type S next to Select to edit an Object Set Definition.

4. In the Object Set Name field, enter %.

5. In the Object Set Data Set field, enter the name of the data set that contains your definitions for object sets and press Enter.

6. On the Object Set List panel (Figure 16 on page 69), enter C next to the object set you want to copy and press Enter.

7. In the Copy Object Set panel, enter the name for the copy of the object set in the Specify TO Object Set Name field.

8. If you want the copy of the object set to overwrite one that already exists with the same name in the object data set, type Y in the Replace field.

9. Provide remarks to describe the object set, as necessary.

10. Press Enter to complete.

Viewing an object set

1. On the Performance Advisor panel, select 1 (Workload Access Path Compare) or 2 (Workload Index Advisor) and press Enter.

2. On the Workload Access Path Compare Menu or Index Advisor Menu, select 1 to define your workload and press Enter.

3. On the Extract SQL for Workload panel (Figure 13 on page 65), type S next to Select to edit an Object Set Definition.

4. In the Object Set Name field, enter %.

5. In the Object Set Data Set field, enter the name of the data set that contains your definitions for object sets and press Enter.
Deleting an object set

6 On the Object Set List panel (Figure 16 on page 69), enter V next to the object set that you want to view and press Enter.

7 On the View Object Set panel, review the name patterns that make up that object set.

If the name pattern is longer than what can fit in the list, enter Z next to the line for that pattern and press Enter to view the entire pattern.

8 When you finishing reviewing the object set, press F3 until you return to the Extract SQL for Workload panel.

Deleting an object set

1 On the Performance Advisor panel, select 1 (Workload Access Path Compare) or 2 (Workload Index Advisor) and press Enter.

2 On the Workload Access Path Compare Menu or Index Advisor Menu, select 1 to define your workload and press Enter.

3 On the Extract SQL for Workload panel (Figure 13 on page 65), type S next to Select to edit an Object Set Definition.

4 In the Object Set Name field, enter %.

5 In the Object Set Data Set field, enter the name of the data set that contains your definitions for object sets and press Enter.

6 On the Object Set List panel (Figure 16 on page 69), enter D next to the object set you want to delete and press Enter.
Comparing access paths for workloads

This chapter presents the following topics:

Overview of workload comparisons . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 73
Benefits ........................................................................................................................................... 74
Workload Access Path Compare process . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 75
Comparing results ....................................................................................................................... 77
Extracting and explaining a workload ...................................................................................... 78
Explaining a workload a second time ...................................................................................... 84
Comparing two or more workloads ......................................................................................... 87
Working with workload comparison reports . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 90
Sample batch report for Workload Access Path Compare ..................................................... 91
Specifying report options for workload comparisons ........................................................... 96
Customizing the comparison report ......................................................................................... 98

Overview of workload comparisons

The Workload Access Path Compare component of the SQL Performance solution identifies changed access paths in a user-defined workload. Workload Access Path Compare analyzes SQL statements and provides reports that indicate specific differences between statements, including such things as differences in statistics, SQL text, and indexes that might have caused changes to the access paths.

The Workload Access Path Compare component makes it easy to define a workload, and then extract and explain the SQL text. Using the online interface, you specify the source of the SQL statements and use qualifiers to define the workload through the creation of object sets.
The Workload Access Path Compare component can gather and compare workload data (including column distribution statistics and SQL text) from the following sources:

- APPTUNE archived trace data sets
- BMC Performance Database
- DB2 catalog
- DB2 statement cache
- A DBRM library
- SQL text data set

After you define the workload, the Workload Access Path Compare component generates a job that extracts and explains the SQL. For each workload, the Workload Access Path Compare component saves the results in the following files that are registered in the HLQ.REPOS repository. (HLQ represents a workload high-level qualifier that you specify.)

- `HLQ.SQLTXT.seqNumber` contains the SQL text that the Extract process identifies.
- `HLQ.OBJECT.seqNumber` contains all objects and their statistics that the Explain process identifies.
- `HLQ.PATH.seqNumber` contains all access paths that the Explain process identifies.

**Benefits**

With the Workload Access Path Compare component, you can perform the following activities:

- Analyze and predict access path changes for both dynamic SQL and static SQL
- Compare access paths for a workload on different DB2 subsystems

For example, you can compare the same application running on different versions of DB2 or on different subsystems with the same version of DB2. Performing the Explain operation on a different subsystem does not require running your application on the second system. Thus, you can predict changes or problems before moving an application to another subsystem.
Workload Access Path Compare process

- compare access paths for a workload on the same DB2 subsystem after changes have been made at a different point in time

  For example, you can analyze changes to access paths in a local subsystem when environmental variables (such as statistics) or application options have been changed. Use this approach to check the effects of such variables as adding or removing indexes, running RUNSTATS, or performing a new BIND. Workload Access Path Compare stores the resulting data in the \textit{HLQ.OBJECT.seqNumber} and \textit{HLQ.PATH.seqNumber} files.

- use the comparison results to predict changes in access paths within an application before migrating to a new release of DB2

  After the migration, you can compare the predicted changes against the actual changes.

- use the comparison results to predict changes in access paths before deploying a new release of an application

  After deploying the release, you can compare the predicted changes against the actual changes.

- use the comparison results of a BIND with EXPLAIN(YES) to a dynamic explain to decide whether to perform selective rebinds to prevent the access path from getting worse during rebind

- use the comparison results to decide whether to perform selective BIND REPLACE operations at compile time to eliminate the need to bind if no SQL changes were found

\textbf{Workload Access Path Compare process}

The Workload Access Path Compare process (illustrated in Figure 17 on page 76) encompasses the following basic tasks:

1. Define object sets to identify objects in the workload (as described in Chapter 4, “Working with object sets”).

2. Extract and explain a workload (page 78).

3. Explain the same workload on a different subsystem, with a different set of environmental factors, or at a different point in time (page 84).

4. Compare results and report any differences (page 87).
Figure 17  Workload Access Path Compare process (two different subsystems)
Comparing results

The Workload Access Path Compare component allows you to compare the access paths of up to 10 workloads. You can also view the access paths and SQL text of a single workload.

Usually, you specify one workload to use as a baseline (0) and one other workload (1) for comparison. If you specify more than the baseline and a single workload, the comparisons are made as follows:

- Workload 0 is compared to Workload 1.
- Workload 1 is compared to Workload 2.
- Workload 2 is compared to Workload 3, and so on.

Workload Access Path Compare uses an exclusive analysis and calculation process to identify matching SQL statements to be compared. SQL statements from two workload extract files (HLQ.SQLTXT.seqNumber) are analyzed to locate matching statements based on

- program name
- statement type
- reference to the same tables

Next, statement pairs that match are analyzed at the character (statement text) level. The Workload Access Path Compare component identifies the longest common sequence of shared characters and divides that length by the average length of the two SQL statements. The Workload Access Path Compare component reports on the differences in access paths and other statistics for the pairs of statements with the highest degree of similarity.

The initial summary displays a list of matching SQL statements for which differences have been found:

- In the summary of the online report, statements are listed in descending order by difference in cost, measured in timerons.

- In the summary of the batch report, you can specify to sort statements by difference in cost, weighted cost, or by plan or program. All details are included with each statement. See Figure 26 on page 91.
Extracting and explaining a workload

Perform this task to use the Workload Access Path Compare component to define a workload and then extract and explain the SQL text for a workload from a specific source type.

Before you begin

- Create the object set definition that identifies the objects for the sources that you want to include in the workload (as described in Chapter 4, “Working with object sets”). You do not need to create object set definitions if your source is a SQL text data set or DBRM library source.

- Gather the information for the source type of the workload for which you are extracting data. For more information, see “Description of source type information” on page 223.

- Consider the following for APPTUNE trace data set sources:

  Workload Access Path Compare extracts static and dynamic SQL from APPTUNE archived trace data sets that contain the following data classes:

  - APSTMT (BMC IFCID 005), used to extract the SQL text
  - (optional) APSTACC (BMC IFCID 307), used to determine the execution count

  **NOTE**

  If you want to compare statements based upon certain types of collection data, you must ensure that the collection keys are set in APPTUNE before the SQL statements are executed.

To extract and explain SQL from a workload

1. Access the Workload Access Path Compare component:

   A  On the SQL Performance for DB2 main menu, select A (Performance Advisors) and press Enter.

   B  On the Performance Advisors panel, select 1 (Workload Access Path Compare) and press Enter.

      The Workload Access Path Compare Main Menu is displayed (Figure 18 on page 79).
Figure 18  Workload Access Path Compare Main Menu (PSSWC100)

PSSWC100 -------------- Workload Access Path Compare Menu -------------------
Command ===> 
Select one of the following options and then press ENTER to continue.

1  1. Extract and Explain SQL - Define workload
2. Explain SQL             - Explain SQL from Extract step
3. Compare access paths    - Compare Explain results & generate report

Specify workload high level qualifier : 
HLQ: RDAVEM.WLC       (Max 26 characters)

C  Select 1 (Extract and Explain SQL).

D  Specify a workload high-level qualifier to be used for the repository and output files that contain the extracted SQL text.

This value must be a valid data set name qualifier and cannot exceed 26 characters. Low-level qualifiers will be appended and the data sets will be created if they do not already exist.

E  Press Enter to display the Extract SQL for Workload panel (Figure 19).

Figure 19  Extract SQL for Workload (PSSWC105)

PSSWC105 ------------------ Extract SQL for Workload ------------------------
Command ===> 
  _ Select to edit an Object Set Definition
  Object Set Name : MYOBJECTSET       (blank to add new, % to show list)
  Object Set Data Set : RDAVEM.TEST.OBJSETDB

Select one of the following Workload sources and press Enter to continue
Object Set Types Used

  _ DB2 catalog                   PL, PG, DM
  _ APPTUNE trace data           PL, PG, DM, CL, US, CR
  _ DBRM library                 None
  _ SQL Text data set            None
  _ Statement CACHE              PG, DM, US

Specify workload high level qualifier : 
HLQ: RDAVEM.WLC       (Max 26 characters)
2 Identify the source for the workload from which you want to extract SQL:

A In the Object Set Name field, use one of the following responses:

- If you want to explain SQL for a DBRM library or a SQL text data set, ignore this field.
- If you want to explain SQL for any other source type, enter the name of the object set that contains the naming patterns you want to use for this workload.

B In the Object Set Data Set field, use one of the following responses:

- If you want to explain SQL for a DBRM library or a SQL text data set, ignore this field.
- For any other source type, enter the name of the data set that contains your object set definitions.

C Select the source for the workload for which you want to extract SQL to explain.

D If necessary, update the high-level qualifier for the workload in the HLQ field.

3 Specify the parameters for the source type that you selected.

Depending on which source type you selected, various panels are displayed. You use the panels to specify information about the source of the SQL text that you want to extract for this workload.
A  Complete the panels for your source type (as listed in Table 8).

Table 8  Source Workload Extract panels by source

<table>
<thead>
<tr>
<th>Source type</th>
<th>Panels to complete</th>
<th>Related info</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2 catalog</td>
<td>■ SQL Workload Source Extract from DB2 catalog</td>
<td>Table 27 on page 224</td>
</tr>
<tr>
<td>APPTUNE trace data</td>
<td>■ SQL Workload Source Extract from APPTUNE Trace Data</td>
<td>Table 28 on page 224</td>
</tr>
<tr>
<td></td>
<td>■ SQL Workload Source - Archive Trace Data</td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ SQL Workload Source - Add Archive Trace Data</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> The SQL Workload Source - Archive Trace Data panel is displayed only if</td>
<td></td>
</tr>
<tr>
<td></td>
<td>you specified Y in the <strong>List APPTUNE data set</strong> field on the previous panel. On</td>
<td></td>
</tr>
<tr>
<td></td>
<td>this panel, you select the trace data sets that you want to use and then press</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>PF3</strong> to continue.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>To add a data set that is not displayed in the list, type <strong>A</strong> in any action</td>
<td></td>
</tr>
<tr>
<td></td>
<td>field or on the Command line and enter the data set name in the SQL Workload</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Source - Add Archive Trace Data panel.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>You might want to add an additional archive trace data set if it was not</td>
<td></td>
</tr>
<tr>
<td></td>
<td>registered in the COPYDIR when created. The added archived trace data sets exist</td>
<td></td>
</tr>
<tr>
<td></td>
<td>only in the list for the SQL extraction and are not saved in the APPTUNE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>archive directory.</td>
<td></td>
</tr>
<tr>
<td>DBRM library</td>
<td>■ SQL Workload Source Extract from DBRM library</td>
<td>Table 29 on page 226</td>
</tr>
<tr>
<td>BMC Performance Database</td>
<td>■ Extract SQL for Performance Advisor Database</td>
<td>Table 30 on page 226</td>
</tr>
<tr>
<td></td>
<td>■ Extract parameters for Performance Advisor Database</td>
<td></td>
</tr>
<tr>
<td>SQL text data set</td>
<td>■ SQL Workload Extract from SQL Text Data Set</td>
<td>Table 31 on page 229</td>
</tr>
<tr>
<td>Statement cache</td>
<td>■ SQL Workload Source Extract from DB2 Statement Cache</td>
<td>Table 32 on page 230</td>
</tr>
</tbody>
</table>

B  Press **Enter** to display the SQL Workload Explain panel (Figure 20).

Figure 20  SQL Workload Explain (PSSWC120)

```
PSSWC120 --------------------------- SQL Workload Explain ---------------------------
Command ===>

Specify SQL workload qualifiers :
Explain type : XD  1. XD = Explain dynamic
                   Plan Table Owner : BMC (USERID,authid)
                   2. XS = Explain static from bind with EXPLAIN(YES)

Explain SSID : DEBF
Remarks      : Extract and Explain V8V9DIFF on DEBF
```
4 Specify the type of Explain operation that you want to perform.

A In the Explain type field, type XD to select a dynamic Explain operation or XS to select static Explain operation.

**NOTE**
Dynamic Explains (XD) use the DB2 optimizer to identify the access paths. This process evaluates access paths, SQL text, and key catalog statistics in real time.

Static Explains (XS) report the access path information that was derived from the owner.PLAN_TABLE when you performed a bind with EXPLAIN(YES). This type of explain is only valid for static SQL.

B If you selected dynamic Explain, in the Plan Table Owner field, enter the user ID or authorization ID for the plan tables to be used.

The default is BMC.

**TIP**
For a dynamic Explain, the DB2 optimizer stores information in the plan tables of a specified owner. After reading and reporting on the access paths, the product deletes the rows from the plan tables. To avoid creating multiple empty plan tables, all users should use the same plan table owner (such as BMC). Be aware that users must have authority to perform a SET CURRENT SQLID command to point to a plan table owner that is different from the current AUTHID. If the user does not have this authority, the product will use the current AUTHID as the owner of the plan tables.

BMC recommends that you point to BMC or empty plan tables. If your plan table has a lot of rows, you should add the recommended indexes in the DAADB2IX member in the SAMP library to avoid performance problems.

C (optional) In the Remarks field, enter text to help you identify the workload.

You can enter up to 40 alphanumeric characters.
D Press Enter to display the Batch Job panel (Figure 21).

Figure 21 Batch Job panel (PSSPB000)

<table>
<thead>
<tr>
<th>Field</th>
<th>Action</th>
</tr>
</thead>
</table>
| JCL Data Set   | Specify the data set to which Workload Access Path Compare will write the generated JCL.  
|                | You can type the name of a partitioned or sequential file, or you can specify the TEMP keyword. If you specify TEMP, Workload Access Path Compare uses the data set that is specified in ZTEMPF.  
|                | Note: If any of the data sets do not exist, the Allocate Data Set panel is displayed and allows you to allocate a new data set. |
| Current Counter| Enter any value from 1 through 9999 to replace the #### suffix in the PDS member name.  
|                | If needed, Workload Access Path Compare adds leading zeros to fill the four-digit suffix. |
| Set JCL Options| (optional) Specify Y to review or change options for creating JCL.  
|                | For information about setting JCL options, see the SQL Explorer for DB2 User Guide. |

5 Specify the batch JCL options and submit the job.

When you perform an Extract and Explain, an Explain, or a Compare operation, Workload Access Path Compare generates a job for you to submit. The Batch Job panel enables you to set options and build and edit the JCL before executing the jobs.

A Complete the Batch Job options as described in Table 9.
Explaining a workload a second time

In this task, you explain a workload for a second time, either on a different DB2 subsystem or after making other changes to the system.

To explain a workload again

1. On the SQL Performance for DB2 main menu, select A (Performance Advisors) and press Enter.

2. On the Performance Advisors panel, select 1 (Workload Access Path Compare) and press Enter.

   The Workload Access Path Compare Main Menu is displayed (Figure 18 on page 79).

3. Specify that you want to explain SQL that has already been extracted:

   A. Select 2 (Explain SQL).

   B. Verify the workload high-level qualifier contains the value that was used when the SQL was extracted.
Explaining a workload a second time

Chapter 5 Comparing access paths for workloads 85

C Press Enter to display the SQL Workload Compare List for Explain panel (Figure 22).

This panel lists all workloads that have been previously extracted by either the workload comparison or index recommendation process and that exist in the specified workload HLQ repository.

Figure 22  SQL Workload Compare List for Explain (PSSWC115)

4 Select a workload:

A Type S beside the relevant workload.

B (optional) If you want to view more information about a workload, type V in the adjacent action field and press Enter.

The workload qualifiers that were used when the workload was extracted will be displayed. Press PF3 to return to the workload list.

C Press Enter to display the SQL Workload Explain panel (Figure 20 on page 81).

5 Specify Explain parameters:

A In the Explain type field, type XD to select a dynamic Explain operation or XS to select a static Explain operation.

NOTE

Dynamic Explains (XD) use the DB2 optimizer to identify the access paths. This process evaluates access paths, SQL text, and key catalog statistics in real time.

Static Explains (XS) report the access path information that was derived from the owner PLAN_TABLE when you performed a bind with EXPLAIN(YES). This type of explain is only valid for static SQL.
Explaining a workload a second time

B If you selected dynamic Explain, in the **Plan Table Owner** field, enter the user ID or authorization ID for the plan tables to be used.

The default is BMC.

---

**TIP**

For a dynamic Explain, the DB2 optimizer stores information in the plan tables of a specified owner. After reading and reporting on the access paths, the product deletes the rows from the plan tables. To avoid creating multiple empty plan tables, all users should use the same plan table owner (such as BMC). Be aware that users must have authority to perform a SET CURRENT SQLID command to point to a plan table owner that is different from the current AUTHID. If the user does not have this authority, the product will use the current AUTHID as the owner of the plan tables.

BMC recommends that you point to BMC or empty plan tables. If your plan table has a lot of rows, you should add the recommended indexes in the DAADB2IX member in the SAMP library to avoid performance problems.

C In the **Explain SSID** field, specify the DB2 subsystem on which the Explain operation will be performed.

This subsystem does not have to be the same DB2 on which the extracted SQL is located.

D *(optional)* In the **Remarks** field, enter text to help you identify the workload.

You can enter up to 40 alphanumeric characters.

E Press **Enter**.

6 Use the Batch Job panel (Figure 21 on page 83) to generate, edit, and submit JCL to perform the Explain operation.

For more information about the options on the Batch Job panel, see Table 9 on page 83 for more information.
Comparing two or more workloads

In this task, you compare two or more Explain results and generate a report. You must extract and explain the SQL before running the comparison reports.

To compare access paths and generate reports

1. On the SQL Performance for DB2 main menu, select **A** (Performance Advisors) and press **Enter**.

2. On the Performance Advisors panel, select **1** (Workload Access Path Compare) and press **Enter**.

3. Generate a workload comparison:
   
   A. Select **3** (Compare access paths).
   
   B. Verify the workload high-level qualifier is the same as the one used when performing the procedure “Explaining a workload a second time” on page 84.
   
   C. Press **Enter** to display the SQL Workload Compare panel (Figure 23).

This panel lists all workloads that have been previously explained and that exist for the specified workload high-level qualifier.

Figure 23   SQL Workload Compare (PSSWC130)
Comparing two or more workloads

4 Specify report options and select the workload files:

A In the Processing mode field, specify O to run the comparison online or specify B to run the comparison in batch.

NOTE
Consider using batch processing for extremely large workloads. Online processing of large workloads can take a significant amount of time.

B (optional) To specify report options, type any character in the Report options field.

For more information about setting the options, see “Specifying report options for workload comparisons” on page 96.

C Select a workload to use as a baseline by typing 0 (zero) in the action field beside the workload.

D (optional) Select up to nine additional workloads to compare by typing digits from 1 through 9 in the action field beside the comparison workloads.

If you do not enter additional workloads, no comparison will be made but you can still view the baseline in the report.

For information about how comparisons are made, see “Comparing results” on page 77.

E Press Enter.

The next panel varies depending on your entry in the Processing mode field.

- If you specified B (batch) in the Processing mode field, the Batch Job panel is displayed.

Use this panel to generate, edit, and submit JCL to perform the comparison and generate the batch report. For more information about the options on the Batch Job panel, see Table 9 on page 83 for more information.

For information about the resulting batch report, see “Sample batch report for Workload Access Path Compare” on page 91.

- If you specified O (online) in the Processing mode field, the SQL Workload Compare Report panel is displayed.
5 If you generated an online report, review the report on the SQL Workload Compare Report panel (Figure 24). Table 10 describes how to use the report.

**Figure 24 SQL Workload Compare Report (PSSWC140)**

Table 10  **Reviewing an online report (part 1 of 2)**

<table>
<thead>
<tr>
<th>Action</th>
<th>Steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>S - show detailed</td>
<td>Type S next to the statement for which you want to show detailed</td>
</tr>
<tr>
<td>information for a</td>
<td>information and press Enter. The SQL Workload Compare Detail Report</td>
</tr>
<tr>
<td>workload</td>
<td>panel (Figure 25) is displayed.</td>
</tr>
<tr>
<td>**H - hide detailed</td>
<td>Type H in next to the entry for which you want to hide details and</td>
</tr>
<tr>
<td>information for a</td>
<td>press Enter.</td>
</tr>
<tr>
<td>workload</td>
<td></td>
</tr>
</tbody>
</table>
Working with workload comparison reports

This section describes how to work with the workload comparison reports that are generated. It includes the following topics:

- “Sample batch report for Workload Access Path Compare” on page 91
- “Specifying report options for workload comparisons” on page 96
- “Customizing the comparison report” on page 98

### Table 10  Reviewing an online report (part 2 of 2)

<table>
<thead>
<tr>
<th>Action</th>
<th>Steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>T - show the SQL text for a statement</td>
<td>Type T next to the statement for which you want to show SQL text and press Enter.</td>
</tr>
<tr>
<td>show information for all entries</td>
<td>Type S ALL on the command line and press Enter.</td>
</tr>
<tr>
<td>hide information for all entries</td>
<td>Type H ALL on the command line and press Enter.</td>
</tr>
</tbody>
</table>

**NOTE**
Fields may scroll off your visible viewing area to the right. Use PF10 and PF11 to scroll to the left or the right.

For information about fields on the reports, see “Workload Access Path Compare and Index Advisor report fields” on page 230.
Sample batch report for Workload Access Path Compare

Figure 26 shows a sample of the batch version of the compare report sorted by cost difference.

**TIP**

In the batch report, differences are marked with an asterisk (*) on the left side of the changed item.

If the width of a field is too short to display all characters, one of the following events occurs:

- For numbers, the product attempts to show the number up to the decimal point. If this is not possible, asterisks (*) are used to fill the entire field.
- For characters, the product truncates the character text from the right.

You can change the width of the fields by customizing the report layout. For more information, see “Changing the layout of the compare report” on page 234.

Figure 26  Batch Workload Access Path Compare report (part 1 of 5)
**Figure 26  Batch Workload Access Path Compare report (part 2 of 5)**

```
FROM V8V9DFTB
WHERE COLUMN_3_CHAR_12 = 'ABC'
FOR FETCH ONLY

<table>
<thead>
<tr>
<th>DB PL M TB AC MT</th>
<th>INDEX</th>
<th>O UJOG UJOG F F SQ MOD C R T TYPE</th>
<th>CREATOR Creator</th>
<th>TY D G P D G P D PG PG</th>
</tr>
</thead>
<tbody>
<tr>
<td>XD01 1 1 0 1 I *</td>
<td>0</td>
<td>0</td>
<td>IS 0</td>
<td></td>
</tr>
<tr>
<td>XD00 1 1 0 1 I</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LABL BL NO D NO TY CL TABLE</th>
<th>INDEX</th>
<th>O UJOG UJOG F F SQ MOD C R T TYPE</th>
<th>CREATOR Creator</th>
<th>TY D G P D G P D PG PG</th>
</tr>
</thead>
<tbody>
<tr>
<td>XD01</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XD00</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| LABL SSID SSID NAME COLLID VERSION STMTNO COST COUNT COST DIFF IN |
|-----------------------|---------|------------------|--------|--------|------------------|
| XD01 DEDR DEDR V8V9DICI DEFR V8V9DICI V1.01 304* 8 1* 8 7 PATH STATS INDEX |
| XD00 DEBF DEBF V8V9DICI V8V9DICI V1.01 304 1 1 1 |

<table>
<thead>
<tr>
<th>SRC EXPLAIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>XD01 C 2011-01-20 14:25.36.940000</td>
</tr>
<tr>
<td>XD00 C 2011-01-20 14:22.43.460000</td>
</tr>
</tbody>
</table>

DECLARE CUR-V8V9DICI-THREE CURSOR FOR
SELECT COLUMN_20
FROM V8V9DFTB
WHERE COLUMN_20_SML_SHORT = 1
    AND COLUMN_3_CHAR_12 = 'ABC'
    AND KEY1_INTEGER_SHORT = 32123
    AND COLUMN_17 = 34
    AND COLUMN_19 = 'WHATS UP DOC'
    AND COLUMN_20 = 23
FOR FETCH ONLY

<table>
<thead>
<tr>
<th>DB PL M TB AC MT</th>
<th>INDEX</th>
<th>O UJOG UJOG F F SQ MOD C R T TYPE</th>
<th>CREATOR Creator</th>
<th>TY D G P D G P D PG PG</th>
</tr>
</thead>
<tbody>
<tr>
<td>XD01 1 1 0 1 I *</td>
<td>0</td>
<td>0</td>
<td>IS 0</td>
<td></td>
</tr>
<tr>
<td>XD00 1 1 0 1 I</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LABL BL NO D NO TY CL TABLE</th>
<th>INDEX</th>
<th>O UJOG UJOG F F SQ MOD C R T TYPE</th>
<th>CREATOR Creator</th>
<th>TY D G P D G P D PG PG</th>
</tr>
</thead>
<tbody>
<tr>
<td>XD01</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XD00</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| LABL SSID SSID NAME COLLID VERSION STMTNO COST COUNT COST DIFF IN |
|-----------------------|---------|------------------|--------|--------|------------------|
| XD01 DEDR DEDR V8V9DICI DEFR V8V9DICI V1.01 297* 10 1* 10 5 PATH STATS INDEX |
| XD00 DEBF DEBF V8V9DICI V8V9DICI V1.01 297 5 1 5 |
```
Figure 26  Batch Workload Access Path Compare report  (part 3 of 5)
### Figure 26  Batch Workload Access Path Compare report (part 4 of 5)

<table>
<thead>
<tr>
<th>INDEX</th>
<th>U C C</th>
<th>BUFFER</th>
<th>FIRST</th>
<th>FULL</th>
<th>CLUSTER</th>
</tr>
</thead>
<tbody>
<tr>
<td>LBAL NAME</td>
<td>R G D</td>
<td>NLEAF</td>
<td>NLVL</td>
<td>POOL</td>
<td>KEYCARD</td>
</tr>
<tr>
<td>----------</td>
<td>------</td>
<td>--------</td>
<td>------</td>
<td>------</td>
<td>---------</td>
</tr>
<tr>
<td>XD00 COLUMN_2_SML_SHORT</td>
<td>2 SMALLINT</td>
<td>2</td>
<td>N</td>
<td>COLUMN_2_SML_SHORT</td>
<td>2</td>
</tr>
<tr>
<td>XD00 COLUMN_3_CHAR_12</td>
<td>3 CHAR</td>
<td>12</td>
<td>N</td>
<td>COLUMN_3_CHAR_12</td>
<td>3</td>
</tr>
<tr>
<td>XD00 COLUMN_4</td>
<td>4 CHAR</td>
<td>7</td>
<td>N</td>
<td>COLUMN_4</td>
<td>4</td>
</tr>
<tr>
<td>XD00 COLUMN_5</td>
<td>5 CHAR</td>
<td>4</td>
<td>N</td>
<td>COLUMN_5</td>
<td>5</td>
</tr>
<tr>
<td>XD00 COLUMN_6_VARCHAR55</td>
<td>6 VARCHAR</td>
<td>55</td>
<td>N</td>
<td>COLUMN_6_VARCHAR55</td>
<td>6</td>
</tr>
<tr>
<td>XD00 COLUMN_9</td>
<td>7 SMALLINT</td>
<td>2</td>
<td>N</td>
<td>COLUMN_9</td>
<td>7</td>
</tr>
<tr>
<td>XD00 COLUMN_10</td>
<td>8 INTEGER</td>
<td>4</td>
<td>N</td>
<td>COLUMN_10</td>
<td>8</td>
</tr>
<tr>
<td>XD00 COLUMN_11</td>
<td>9 FLOAT</td>
<td>4</td>
<td>N</td>
<td>COLUMN_11</td>
<td>9</td>
</tr>
<tr>
<td>XD00 COLUMN_12</td>
<td>10 FLOAT</td>
<td>8</td>
<td>N</td>
<td>COLUMN_12</td>
<td>10</td>
</tr>
<tr>
<td>XD00 COLUMN_13</td>
<td>11 DATE</td>
<td>7</td>
<td>N</td>
<td>COLUMN_13</td>
<td>11</td>
</tr>
<tr>
<td>XD00 COLUMN_14</td>
<td>12 TIME</td>
<td>3</td>
<td>N</td>
<td>COLUMN_14</td>
<td>12</td>
</tr>
<tr>
<td>XD00 COLUMN_15</td>
<td>13 TIMESTMP</td>
<td>10</td>
<td>N</td>
<td>COLUMN_15</td>
<td>13</td>
</tr>
<tr>
<td>XD00 COLUMN_16</td>
<td>14 VARCHAR</td>
<td>30</td>
<td>N</td>
<td>COLUMN_16</td>
<td>14</td>
</tr>
<tr>
<td>XD00 COLUMN_17</td>
<td>15 INTEGER</td>
<td>4</td>
<td>N</td>
<td>COLUMN_17</td>
<td>15</td>
</tr>
<tr>
<td>XD00 COLUMN_18</td>
<td>16 CHAR</td>
<td>20</td>
<td>Y</td>
<td>COLUMN_18</td>
<td>16</td>
</tr>
<tr>
<td>XD00 COLUMN_19</td>
<td>17 CHAR</td>
<td>24</td>
<td>Y</td>
<td>COLUMN_19</td>
<td>17</td>
</tr>
<tr>
<td>XD00 COLUMN_20</td>
<td>18 INTEGER</td>
<td>4</td>
<td>Y</td>
<td>COLUMN_20</td>
<td>18</td>
</tr>
<tr>
<td>XD00 COLUMN_21</td>
<td>19 TIMESTMP</td>
<td>10</td>
<td>Y</td>
<td>COLUMN_21</td>
<td>19</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>INDEX</th>
<th>U C C</th>
<th>BUFFER</th>
<th>FIRST</th>
<th>FULL</th>
<th>CLUSTER</th>
</tr>
</thead>
<tbody>
<tr>
<td>LBAL NAME</td>
<td>R G D</td>
<td>NLEAF</td>
<td>NLVL</td>
<td>POOL</td>
<td>KEYCARD</td>
</tr>
<tr>
<td>----------</td>
<td>------</td>
<td>--------</td>
<td>------</td>
<td>------</td>
<td>---------</td>
</tr>
<tr>
<td>XD01 V8V9DFTB_INDEX3</td>
<td>*U *<em>Y</em></td>
<td>-1*</td>
<td>-1 BPO</td>
<td>*</td>
<td>-1*</td>
</tr>
<tr>
<td>XD01 V8V9DFTB_INDEX1</td>
<td>U N Y</td>
<td>10</td>
<td>2 BPO</td>
<td>2036</td>
<td>2036</td>
</tr>
<tr>
<td>XD01 V8V9DFTB_INDEX2</td>
<td>D N N</td>
<td>35</td>
<td>2 BPO</td>
<td>213</td>
<td>2036</td>
</tr>
<tr>
<td>XD01 V8V9DFTB_INDEX3</td>
<td>O Y N</td>
<td>79</td>
<td>2 BPO</td>
<td>1</td>
<td>2036</td>
</tr>
<tr>
<td>XD01 V8V9DFTB_INDEX4</td>
<td>D N Y</td>
<td>6</td>
<td>2 BPO</td>
<td>614</td>
<td>614</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>KEY</th>
<th>CO CO CO INDEX</th>
<th>LBAL COLUMN</th>
<th>NO SQ O NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>xedo COLUMN_9</td>
<td>7</td>
<td>1 A V8V9DFTB_INDEX3</td>
<td></td>
</tr>
<tr>
<td>xedo COLUMN_2_SML</td>
<td>2</td>
<td>2 A V8V9DFTB_INDEX3</td>
<td></td>
</tr>
<tr>
<td>xedo COLUMN_3_CHAR</td>
<td>3</td>
<td>3 A V8V9DFTB_INDEX3</td>
<td></td>
</tr>
<tr>
<td>xedo COLUMN_4</td>
<td>4</td>
<td>4 A V8V9DFTB_INDEX3</td>
<td></td>
</tr>
<tr>
<td>xedo COLUMN_5</td>
<td>5</td>
<td>5 A V8V9DFTB_INDEX3</td>
<td></td>
</tr>
<tr>
<td>xedo KEY1_INTEGER</td>
<td>1</td>
<td>6 A V8V9DFTB_INDEX3</td>
<td></td>
</tr>
<tr>
<td>xedo COLUMN_10</td>
<td>8</td>
<td>7 A V8V9DFTB_INDEX3</td>
<td></td>
</tr>
<tr>
<td>xedo COLUMN_11</td>
<td>9</td>
<td>8 A V8V9DFTB_INDEX3</td>
<td></td>
</tr>
<tr>
<td>xedo COLUMN_12</td>
<td>10</td>
<td>9 A V8V9DFTB_INDEX3</td>
<td></td>
</tr>
<tr>
<td>xedo COLUMN_13</td>
<td>11</td>
<td>10 A V8V9DFTB_INDEX3</td>
<td></td>
</tr>
<tr>
<td>xedo COLUMN_14</td>
<td>12</td>
<td>11 A V8V9DFTB_INDEX3</td>
<td></td>
</tr>
<tr>
<td>xedo COLUMN_15</td>
<td>13</td>
<td>12 A V8V9DFTB_INDEX3</td>
<td></td>
</tr>
<tr>
<td>xedo COLUMN_16</td>
<td>14</td>
<td>13 A V8V9DFTB_INDEX3</td>
<td></td>
</tr>
<tr>
<td>xedo COLUMN_17</td>
<td>15</td>
<td>14 A V8V9DFTB_INDEX3</td>
<td></td>
</tr>
<tr>
<td>xedo COLUMN_18</td>
<td>16</td>
<td>15 A V8V9DFTB_INDEX3</td>
<td></td>
</tr>
<tr>
<td>xedo COLUMN_19</td>
<td>17</td>
<td>16 A V8V9DFTB_INDEX3</td>
<td></td>
</tr>
<tr>
<td>xedo COLUMN_20</td>
<td>18</td>
<td>17 A V8V9DFTB_INDEX3</td>
<td></td>
</tr>
<tr>
<td>xedo COLUMN_21</td>
<td>19</td>
<td>18 A V8V9DFTB_INDEX3</td>
<td></td>
</tr>
<tr>
<td>xedo COLUMN_22</td>
<td>20</td>
<td>19 A V8V9DFTB_INDEX3</td>
<td></td>
</tr>
<tr>
<td>xedo COLUMN_23</td>
<td>21</td>
<td>20 A V8V9DFTB_INDEX3</td>
<td></td>
</tr>
<tr>
<td>xedo COLUMN_24</td>
<td>22</td>
<td>21 A V8V9DFTB_INDEX3</td>
<td></td>
</tr>
<tr>
<td>xedo COLUMN_25</td>
<td>23</td>
<td>22 A V8V9DFTB_INDEX3</td>
<td></td>
</tr>
</tbody>
</table>

1'BATCH COMPARE REPORT DEBF TO DEDR V8V9DIFF PG'
### Figure 26  Batch Workload Access Path Compare report (part 5 of 5)

<table>
<thead>
<tr>
<th>LABEL NAME</th>
<th>FREQUENCY VALUE</th>
<th>T</th>
<th>CARDF GROUP</th>
</tr>
</thead>
<tbody>
<tr>
<td>XD00 COLUMN_2_SML</td>
<td>0.232318 8000</td>
<td>F</td>
<td>-1</td>
</tr>
<tr>
<td>XD00 COLUMN_2_SML</td>
<td>0.033399 A328</td>
<td>F</td>
<td>-1</td>
</tr>
<tr>
<td>XD00 COLUMN_2_SML</td>
<td>0.018173 AEE0</td>
<td>F</td>
<td>-1</td>
</tr>
<tr>
<td>XD00 COLUMN_2_SML</td>
<td>0.017682 04D0</td>
<td>F</td>
<td>-1</td>
</tr>
<tr>
<td>XD00 COLUMN_2_SML</td>
<td>0.012279 A710</td>
<td>F</td>
<td>-1</td>
</tr>
<tr>
<td>XD00 COLUMN_2_SML</td>
<td>0.010314 A260</td>
<td>F</td>
<td>-1</td>
</tr>
<tr>
<td>XD00 COLUMN_2_SML</td>
<td>0.009823 8ED8</td>
<td>F</td>
<td>-1</td>
</tr>
<tr>
<td>XD00 COLUMN_2_SML</td>
<td>0.009332 90B3</td>
<td>F</td>
<td>-1</td>
</tr>
<tr>
<td>XD00 COLUMN_2_SML</td>
<td>0.008841 8520</td>
<td>F</td>
<td>-1</td>
</tr>
<tr>
<td>XD00 COLUMN_3_CHA</td>
<td>0.001965 40C1C1C2D5F0F0F0F0F4F6F5</td>
<td>F</td>
<td>-1</td>
</tr>
<tr>
<td>XD00 COLUMN_3_CHA</td>
<td>0.001965 40C1C1C2D5F0F0F0F0F4F6F8</td>
<td>F</td>
<td>-1</td>
</tr>
<tr>
<td>XD00 COLUMN_3_CHA</td>
<td>0.001965 40C1C1C2D5F0F0F0F0F4F7F1</td>
<td>F</td>
<td>-1</td>
</tr>
<tr>
<td>XD00 COLUMN_3_CHA</td>
<td>0.001965 40C1C1C2D5F0F0F0F0F4F7F4</td>
<td>F</td>
<td>-1</td>
</tr>
<tr>
<td>XD00 COLUMN_3_CHA</td>
<td>0.001965 40C1C1C2D5F0F0F0F0F4F7F7</td>
<td>F</td>
<td>-1</td>
</tr>
<tr>
<td>XD00 COLUMN_3_CHA</td>
<td>0.001965 40C1C1C2D5F0F0F0F0F4F8F0</td>
<td>F</td>
<td>-1</td>
</tr>
<tr>
<td>XD00 COLUMN_3_CHA</td>
<td>0.001965 40C1C1C2D5F0F0F0F0F4F8F3</td>
<td>F</td>
<td>-1</td>
</tr>
<tr>
<td>XD00 COLUMN_3_CHA</td>
<td>0.001965 40C1C1C2D5F0F0F0F0F4F8F6</td>
<td>F</td>
<td>-1</td>
</tr>
<tr>
<td>XD00 COLUMN_3_CHA</td>
<td>0.001965 40C1C1C2D5F0F0F0F0F4F8F9</td>
<td>F</td>
<td>-1</td>
</tr>
<tr>
<td>XD00 COLUMN_3_CHA</td>
<td>0.001965 40C1C1C2D5F0F0F0F0F4F9F2</td>
<td>F</td>
<td>-1</td>
</tr>
<tr>
<td>XD00 COLUMN_9</td>
<td>1.000000 8000</td>
<td>F</td>
<td>-1</td>
</tr>
</tbody>
</table>

**REPORT SUMMARY**

- Statements with differences: 3
- Total statements: 3
- Total tables: 1
Specifying report options for workload comparisons

When you perform a Compare operation, the SQL Workload Compare Report Options panel enables you to set report options before generating the report.

The value that you set for the layout data set applies to both the online and the batch report. All other options apply only to the batch report.

To specify report options

1 To display the SQL Workload Compare Report Options panel, type any character in the Report options field on the SQL Workload Compare panel (Figure 23 on page 87).

2 In the Layout data set field, specify the name of the data set in which the report layout is stored.

This layout determines how the report fields are displayed as part of the Report generation. If left blank, the report layout defaults to the HLQ.samplib(PSSREPB) member. The name of this library will vary depending on your installation options. The name could be hlq.BBSSAMP, hlq.PSSSAMP, or hlq.BMCSAMP. For more information about report layouts, see “Changing the layout of the compare report” on page 234.
3 In the **Edit Layout data set** field, specify whether you want to edit the layout data set.

4 Specify options for the batch report:

**A** *(optional)* In the **Generate report title** field, specify whether to add a report title to the batch report:

- **Y** adds a user-specified title to the report. You must enter a title (step 4B).
- **N** produces a batch report with no title.

**B** If you specified **Y** in step 4A, in the **Title** field, specify a title to be added to the batch report.

You can enter up to 65 alphanumeric characters for the report title.

**C** In the **Summary report** field, specify whether to produce a summary report of the comparison results:

- **Y** creates a summary at the beginning of the report. You must specify the sort order (step 4D).
- **N** produces a batch report with no summary.

**D** If you specified **Y** in step 4C, in the **Sort order** field, specify how to order the results of the batch Summary report:

- **1** *(Cost)* sorts in descending order by SQL statement cost in timerons.
- **2** *(Weighted cost)* sorts in descending order by SQL statement cost, multiplied by the number of executions.

  Weighted cost can only be calculated when you are comparing data from the APPTUNE archived trace data sets. For other sources, the number of executions is set to 1.

- **3** *(Plan/Program Cost)* sorts in descending order by SQL statement costs, summarized by plan and program.

**E** Specify what information to include in the Detail report section:

- In the **Show SQL** field, specify whether to include the SQL text in the report.
- In the **Show obj** field, specify whether to display the DB2 objects used in the SQL statements in the report.
Customizing the comparison report

You can customize reports by creating a copy of the HLQ.samplib(PSSREPB) member and modifying the layout in that file. For information about PSSREPB and the layout, see “Changing the layout of the compare report” on page 234. For information about the fields that make up the report, see “Workload Access Path Compare and Index Advisor report fields” on page 230.
Chapter 6

Recommending indexes for workloads

This chapter presents the following topics:

Overview of the Workload Index Advisor ............................................ 99
  Benefits .......................................................................................... 100
  Workload Index Advisor process ..................................................... 101
Extracting SQL and recommending an index ..................................... 103
Recommending an index for a workload a second time .................... 108
Reviewing index recommendations ................................................... 110
Working with index recommendation reports .................................... 114
  Sample batch report for Index Advisor ........................................... 114
  Specifying report options for Index Advisor reports ...................... 117
  Customizing index recommendation reports .................................. 119

Overview of the Workload Index Advisor

SQL Performance for DB2 provides the Workload Index Advisor to help create the right indexes to minimize the cost of running a set of SQL statements. Workload Index Advisor uses the Explain capability to extract and explain the SQL from a user-defined workload and then analyzes the indexes to provide estimated costs and recommendations for additional indexes. Workload Index Advisor then validates the results by testing the recommendations with virtual indexes.

The Workload Index Advisor component makes it easy to define a workload, extract the SQL text, and recommend indexes. Using the online interface, you identify the source of the SQL statements and use qualifiers and object definitions to define the workload for which you want to provide index recommendations.
Index Advisor can gather workload data from the following sources:

- APPTUNE archived trace data sets
- BMC Performance Database
- the DB2 catalog
- the DB2 statement cache
- DBRM libraries
- SQL text data sets

After you define the workloads, Index Advisor generates a job that extracts and recommends indexes for the SQL. For each workload, Index Advisor saves the results in the following files:

- The `HLQ.OBJECT.sequenceNumber` file contains objects identified during the Recommend Index process.
- The `HLQ.PATH.sequenceNumber` file contains access paths identified during the Recommend Index process.
- The `HLQ.SQLTXT.sequenceNumber` file contains SQL text identified during the Extract process.
- The `HLQ.RIX.sequenceNumber` file contains Index Advisor report information.

These files are registered in the `HLQ.REPOS` repository, where `HLQ` represents a workload high-level qualifier that you specify.

**Benefits**

Workload Index Advisor provides the following features and benefits:

- analyzes access paths of SQL statements in user-defined object sets and recommends new indexes
- identifies indexes to keep or ones that were not seen in processing the workload
- considers existing indexes when providing recommendations
- uses virtual indexes to validate results
- does not create indexes on your system until you want them
- quickly identifies the best indexes to use for workloads containing dynamic and static SQL
allows analysis across multiple DB2s

- works with Workload Compare feature of the SQL Performance solution

- provides a snapshot of access paths, SQL text, and object statistics saved for use in comparing before and after schema changes

- provides a column usage report

**Workload Index Advisor process**

The Workload Index Advisor process (illustrated in Figure 28 on page 102) encompasses the following basic tasks:

1. Define object sets that identify the source objects from which you want to extract SQL for analysis (as described in Chapter 4, “Working with object sets”).

2. Extract the SQL and generate a recommendation for indexes for the workload (page 103).

3. Generate an index recommendation for the same workload on a different subsystem, with a different set of environmental factors, or at a different point in time (page 108).

4. Review the report for the recommendations (page 110).
Figure 28  Index Advisor process

Object sets and sources

- DB2A
  - Catalog
  - STMT_TEXT
  - STMT_CACHE
- OBJSET File
- Archive trace data sets
- Performance Advisor Database

SQL file data set

EXTRACT and RECOMMINDEX

SQLTXT_FILE
From IXA and WLC

RECOMMINDEX

DB2A
For Explain

OBJECT_FILE
PATH_FILE (w/V-I)
RIX_FILE

Online display

REVIEW IXA REPORT

Batch report

WLC COMPARE REPORT
Extracting SQL and recommending an index

Use this procedure to extract SQL from one of the supported sources and generate an index recommendation for the associated workload.

Before you begin

- Create the object set definition that identifies the objects for the sources that you want to include in the workload (as described in Chapter 4, “Working with object sets”). You do not need to create object set definitions if your source is a SQL text data set or DBRM library source.

- Gather the information for the source type of the workload for which you are extracting data. For more information, see “Description of source type information” on page 223.

- Consider the following for APPTUNE trace data set sources:

  Workload Access Path Compare extracts static and dynamic SQL from APPTUNE archived trace data sets that contain the following data classes:

  — APSTMT (BMC IFCID 005), used to extract the SQL text
  — (optional) APSTACC (BMC IFCID 307), used to determine the execution count

  **NOTE**

  If you want to compare statements based upon certain types of collection data, you must ensure that the collection keys are set in APPTUNE before the SQL statements are executed.

To extract SQL and recommend an index

1. Access the Workload Index Advisor.

   A. On the SQL Performance for DB2 main menu, select A (Performance Advisors) and press Enter.

   B. On the Performance Advisors panel, select 2 (Workload Index Advisor) and press Enter.

   The Index Advisor Menu is displayed (Figure 29 on page 104).
Extracting SQL and recommending an index

Figure 29  Index Advisor Menu (PSSWC100)

PSSWI100 ---------------------- Index Advisor Menu ----------------------
Command ===>

Select one of the following options and then press ENTER to continue.

1 1. Extract SQL and            - Define workload, Create SQL file and
    Recommmindex               Recommend Indexes
2. Recommmindex               - Recommend Indexes
3. Review Recommendations    - Generate Index Advisor reports

Specify workload high level qualifier :
HLQ: PSS.V101TEST.WLCIX          (Max 26 characters)

C  Select 1 (Extract SQL and Recommindex).

D  Specify a workload high-level qualifier to be used for the repository and output files that contain the extracted SQL text.

This value must be a valid data set name qualifier and cannot exceed 26 characters. Low-level qualifiers will be appended and the data sets will be created if they do not already exist.

E  Press Enter to display the Extract SQL for Workload panel (Figure 19 on page 79).

2  Identify the sources for the workloads from which you want to recommend indexes:

A  In the Object Set Name field, use one of the following responses:

- If you want to explain SQL for a DBRM library or a SQL text data set, ignore this field.

- If you want to explain SQL for any other source type, enter the name of the object set that contains the naming patterns you want to use for this workload.

For more information about creating object sets, see Chapter 4, “Working with object sets.”.
In the **Object Set Data Set** field, use one of the following responses:

- If you want to explain SQL for a DBRM library or a SQL text data set, ignore this field.
- For any other source type, enter the name of the data set that contains your object set definitions.

Select the source type for the workload for which you want to extract SQL to explain.

If necessary, update the high-level qualifier for the workload in the **HLQ** field.

Specify the parameters for the source type that you selected.

Depending on which source type you selected, various panels are displayed. You can use them to specify the source of the SQL text that you want to extract for this workload.

Complete the panels for your source type (as listed in Table 11).

### Table 11  **Source Workload Extract panels by source**

<table>
<thead>
<tr>
<th>Source type</th>
<th>Panels to complete</th>
<th>Related info</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2 catalog</td>
<td>■ SQL Workload Source Extract from DB2 catalog</td>
<td>Table 27 on page 224</td>
</tr>
<tr>
<td>APPTUNE trace data</td>
<td>■ SQL Workload Source Extract from APPTUNE Trace Data</td>
<td>Table 28 on page 224</td>
</tr>
<tr>
<td></td>
<td>■ SQL Workload Source - Archive Trace Data</td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ SQL Workload Source - Add Archive Trace Data</td>
<td></td>
</tr>
<tr>
<td><strong>Note:</strong> The SQL Workload Source - Archive Trace Data panel is displayed only if you specified Y in the <strong>List APPTUNE data set</strong> field on the previous panel. On this panel, you select the trace data sets that you want to use and then press <strong>PF3</strong> to continue.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>To add a data set that is not displayed in the list, type A in any action field or on the Command line and enter the data set name in the SQL Workload Source - Add Archive Trace Data panel.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>You might want to add an additional archive trace data set if it was not registered in the COPYDIR when created. The added archived trace data sets exist only in the list for the SQL extraction and are not saved in the APPTUNE archive directory.</td>
<td></td>
</tr>
<tr>
<td>DBRM library</td>
<td>■ SQL Workload Source Extract from DBRM library</td>
<td>Table 29 on page 226</td>
</tr>
<tr>
<td>BMC Performance Database</td>
<td>■ Extract SQL for Performance Advisor Database</td>
<td>Table 30 on page 226</td>
</tr>
<tr>
<td></td>
<td>■ Extract parameters for Performance Advisor Database</td>
<td></td>
</tr>
<tr>
<td>SQL text data set</td>
<td>■ SQL Workload Extract from SQL Text Data Set</td>
<td>Table 31 on page 229</td>
</tr>
<tr>
<td>Statement cache</td>
<td>■ SQL Workload Source Extract from DB2 Statement Cache</td>
<td>Table 32 on page 230</td>
</tr>
</tbody>
</table>
After you have finished entering your source information, press Enter to display the Recommindex panel (Figure 30).

**Figure 30  Recommindex panel (PSSWC120)**

PSSWC120 --------------- Recommindex ------------------------------
Command ==>

Index Advisor Report Options :

Specify SQL workload qualifiers :
Explain type : XD  1. XD = Explain dynamic

Plan Table Owner : USERID (USERID,authid)

2. XS = Explain static from bind with EXPLAIN(YES)

Explain SSID : DEDR
Remarks :

4 At the Recommindex panel, specify the Explain parameters for the operation that you want to perform:

A Specify whether you want to change the settings for reports in the **Index Advisor Report Options** field.

If you specify Y (Yes) in this field, see “Specifying report options for Index Advisor reports” on page 117 for instructions.

B In the **Explain type** field, type XD to select a dynamic Explain operation or XS to select static Explain operation.

**NOTE**

Dynamic Explains (XD) use the DB2 optimizer to identify the access paths. This process evaluates access paths, SQL text, and key catalog statistics in real time.

Static Explains (XS) report the access path information that was stored in the owner.PLAN_TABLE when you performed a bind with EXPLAIN(YES). This type of explain is only valid for static SQL.
C If you selected dynamic Explain, in the **Plan Table Owner** field, enter the user ID or authorization ID for the plan tables to be used.

The default is BMC.

**TIP**
For a dynamic Explain, the DB2 optimizer stores information in the plan tables of a specified owner. After reading and reporting on the access paths, the product deletes the rows from the plan tables. To avoid creating multiple empty plan tables, all users should use the same plan table owner (such as BMC). Be aware that users must have authority to perform a SET CURRENT SQLID command to point to a plan table owner that is different from the current AUTHID. If the user does not have this authority, the product will use the current AUTHID as the owner of the plan tables.

BMC recommends that you point to BMC or empty plan tables. If your plan table has a lot of rows, you should add the recommended indexes in the DAADB2IX member in the SAMP library to avoid performance problems.

D In the Explain SSID field, enter the name of the DB2 subsystem on which the index recommendation operation will be performed.

E *(optional)* In the **Remarks** field, enter text to help you identify the workload.

You can enter up to 40 alphanumeric characters.

F Press **Enter** to display the Batch Job panel *(Figure 21 on page 83).*

5 Specify the batch JCL options.

**NOTE**
When you perform an Extract on a workload to recommend an index, Index Advisor generates a JCL job for you to submit. This process is also used when comparing workloads with the Workload Access Path Compare feature.

A Complete the Batch Job options as described in Table 9 on page 83.

B Press **Enter** to complete the actions that you specified.
Recommending an index for a workload a second time

In this task, you recommend an index for a workload for a second time, either on a different DB2 subsystem or after making other changes.

To recommend an index again

1. On the SQL Performance for DB2 main menu, select A (Performance Advisors) and press Enter.

2. On the Performance Advisors panel, select 2 (Workload Index Advisor) and press Enter.

   The Index Advisor Menu is displayed (Figure 29 on page 104).

3. Choose to recommend an index:

   A. Select 2 (Recommindex).

   B. Verify the workload high-level qualifier is the same as that used when performing the procedure “Extracting SQL and recommending an index” on page 103.

   C. Press Enter to display the SQL Workload List for Recommindex panel (Figure 31).

   This panel lists all workloads that have been previously extracted by either the workload comparison or index recommendation process and that exist in the specified workload HLQ repository.

Figure 31  SQL Workload List for Recommindex (PSSWC115)

<table>
<thead>
<tr>
<th>Command ===</th>
<th>Scroll ===</th>
<th>CSR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select a workload to process :</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Actions: S V</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Explain SQL Explain Source</td>
<td></td>
<td></td>
</tr>
<tr>
<td>type source DB2 DB2 Remarks</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------</td>
<td>--------</td>
<td>--------</td>
</tr>
<tr>
<td>XD CATALOG DEDR DEDR WLC VERSION=LAST ON DEDR - TRY#2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>XD CATALOG DEDK DEDK IxA V8V9DIFF ON DEDK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>XD CATALOG DEDR DEDR IxA V8V9DIFF ON DEDR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>XD CATALOG DEDR DEDR WLC V8V9DIFF ON DEDR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>XD CATALOG DEBF DEBF WLC V8V9DIFF ON DEBF</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*********** Bottom of data ******************
4 On the SQL Workload List for Recommindex panel, type S next to the workload for which you want to recommend an index and press Enter.

**TIP**

From the SQL Workload List for Recommindex panel, you can also view a workload by typing V in the adjacent action field and press Enter. Doing so displays the workload qualifiers that were used when the workload was extracted. You can press PF3 to return to the workload list.

5 On the Recommindex panel (Figure 30 on page 106), specify the Explain parameters for the operation that you want to perform:

A Specify whether you want to change the settings for reports in the Index Advisor Report Options field.

If you specify Y (Yes) in this field, see “Specifying report options for Index Advisor reports” on page 117 for instructions.

B In the Explain type field, type XD to select a dynamic Explain operation or XS to select a static Explain operation.

**NOTE**

Dynamic Explains (XD) use the DB2 optimizer to identify the access paths. This process evaluates access paths, SQL text, and key catalog statistics in real time.

Static Explains (XS) report the access path information that was derived from the owner.PLAN_TABLE when you performed a bind with EXPLAIN(YES). This type of explain is only valid for static SQL.

C If you selected dynamic Explain, in the Plan Table Owner field, enter the user ID or authorization ID for the plan tables to be used.

The default is BMC.

**TIP**

For a dynamic Explain, the DB2 optimizer stores information in the plan tables of a specified owner. After reading and reporting on the access paths, the product deletes the rows from the plan tables. To avoid creating multiple empty plan tables, all users should use the same plan table owner (such as BMC). Be aware that users must have authority to perform a SET CURRENT SQLID command to point to a plan table owner that is different from the current AUTHID. If the user does not have this authority, the product will use the current AUTHID as the owner of the plan tables.

BMC recommends that you point to BMC or empty plan tables. If your plan table has a lot of rows, you should add the recommended indexes in the DAADB2IX member in the SAMP library to avoid performance problems.
D In the Explain SSID field, specify the DB2 subsystem on which the Explain operation will be performed.

This subsystem does not have to be the same DB2 on which the extracted SQL is located.

E (optional) In the Remarks field, enter text to help you identify the workload.

You can enter up to 40 alphanumeric characters.

F Press Enter to display the Batch Job panel (Figure 21 on page 83).

6 Specify the batch JCL options.

When you perform an Extract on a workload to recommend an index, Index Advisor generates a JCL job for you to submit. This process is also used when comparing workloads with the Workload Access Path Compare feature.

A Complete the Batch Job options as described in Table 9 on page 83.

B Press Enter to complete the actions that you specified.

Reviewing index recommendations

In this task, you review the index recommendations for workloads. For each index that is analyzed, the Recommend field indicates the recommendation for that index. Possible values include:

- Create — Index Advisor recommends creating this index after analyzing the predicates in the workload. The optimizer used this index in at least one of the statements in the workload and a cost savings was observed.

- Keep — Index Advisor recommends keeping this existing index. It was used by the optimizer while explaining at least one statement in the workload.

- Not seen — Index Advisor recommended this index after reviewing the results from analyzing the predicates or it is an existing index. However, the optimizer did not use it in any statements in the workload.

- No savings — Index Advisor recommended this index after reviewing the results from analyzing the predicates in the workload. However, the overall cost increased when the index was created, so no savings is associated with creating this index.
Reviewing index recommendations

The Savings field indicates the potential cost savings of using the index. For information about additional fields, see “Workload Access Path Compare and Index Advisor report fields” on page 230.

Figure 32  Index Recommendation Report

To review index recommendations

1 On the SQL Performance for DB2 main menu, select A (Performance Advisors) and press Enter.

2 On the Performance Advisors panel, select 2 (Workload Index Advisor) and press Enter.

3 Choose to review index recommendations:

   A Select 3 (Review Recommendations).

   B Verify that the workload high-level qualifier for the repository matches the one used when performing the recommendation process.

   C Press Enter to display the Index Advisor Workloads panel (Figure 33 on page 112).

This panel lists all workloads for which indexes have been generated and that exist in the specified workload repository.
**Figure 33  Index Advisor Workloads (PSSWC245)**

Specify a workload

Actions: S V D

Explain SQL  Explain Source

type  source  DB2  DB2  Remarks

_ XD  SQLFILE  DEDR  DEDR  IXA - SELECT FROM SYSPACKAGES - TEST
_ XD  CATALOG  DEDK  DEDK  IXA - SQL TEXT FROM COMPANY
_ XD  SQLFILE  DEDR  DEDR  IXA - SQL TEXT - NO DUPE CURSORS
_ XD  SQLFILE  DEDR  DEDR  IXA - SQL TEXT FROM COMPANY

4 Review the index recommendations for a workload:

A In the Action field for the workload for which you want to display the results of your index analysis, enter S and press Enter.

**TIP**

From the Index Advisor Workloads panel, you can also

- View more information about the workload by entering V and press Enter.
- Permanently delete a workload from the repository by entering D next to the workload and pressing Enter.

B On the Recommindex: Index Recommendation Report panel (Figure 34), specify the action that you want to perform:

**Figure 34  Recommindex: Index Recommendation Report panel (PSSWC140)**
To show the details for a workload, type `S` in the action column next to the object for which you want to show details and press Enter.

To hide the details for a workload, type `H` in the action column next to the object for which you want to hide details and press Enter.

To show the SQL text for a statement, type `T` in the action column next to the statement and press Enter.

The following figure shows an example of an expanded statement.

![Expanded statement example]

To show details for all workloads, type `S ALL` in the command line.

To hide details for all workloads, type `H ALL` in the command line.

**NOTE**

Fields may scroll off your visible viewing area to the right. Use **PF10** and **PF11** to scroll to the left or the right.

For information about fields on the reports, see “Workload Access Path Compare and Index Advisor report fields” on page 230.
Working with index recommendation reports

This section describes how to work with index recommendation reports that are generated. It includes information about

- “Sample batch report for Index Advisor” on page 114
- “Specifying report options for Index Advisor reports” on page 117
- “Customizing index recommendation reports” on page 119

Sample batch report for Index Advisor

Figure 35 shows a sample of a batch Index Advisor report.

**TIP**

If the width of a field is too short to display all characters, one of the following events occurs:

- For numbers, the product attempts to show the number up to the decimal point. If this is not possible, asterisks (*) are used to fill the entire field.

- For characters, the product truncates the character text from the right.

You can change the width of the fields by customizing the report layout. For more information, see “Changing the layout of the compare report” on page 234.

![Sample Index Advisor report](part1of3)
Figure 35  Sample of Index Advisor report (part 2 of 3)
Figure 35  Sample of Index Advisor report  (part 3 of 3)

```sql
DECLARE CUR-V8V9DIFF-ONE CURSOR FOR
SELECT COLUMN_3_CHAR_12
FROM V8V9DFTB
WHERE COLUMN_3_CHAR_12 = 'ABC'
FOR FETCH ONLY

INDEX NAME SAVINGS #STMTS RECOMMEND R G D NLEAF NLVL POOL KEYCARD KEYCARD RATIO
--------------------------------- -------- ------ --------- - - - --------- ---- -------- --------- --------- ---------
V8V9DFTB_INDEX3           0      1 KEEP      U Y N        79    2 BP0              1      2036  0.623772

EXPL SRC STMT EXEC WEIGHTED SRC
SSID SSID NAME COLLID VERSION STMTNO COST COUNT COST TYPE
1'INDEX RECOMMENDATION REPORT'
3 0
```

```sql
DECLARE CUR-V8V9DIFF-THREE CURSOR FOR
SELECT COLUMN_20
FROM V8V9DFTB
WHERE COLUMN_2_SML_SHORT = 1 AND COLUMN_3_CHAR_12 = 'ABC' AND KEY1_INTEGER_SHORT = 32123 AND COLUMN_17 = 34 AND COLUMN_19 = 'WHATS UP DOC' AND COLUMN_20 = 23
FOR FETCH ONLY
```
Specifying report options for Index Advisor reports

When you perform a Recommend Index operation, the Recommend Index panel enables you to set report options before generating the report.

The value that you set for the layout data set applies to both the online and the batch report. All other options apply only to the batch report.

To specify report options

1. At the Recommend Index panel (Figure 30 on page 106), enter Y (Yes) in the Index Advisor Report Options field to specify that you want to change the settings for reports.

   The Index Advisor Reports Options panel is displayed.

   Figure 36  Index Advisor Report Options (PSSWC275)

   PSSWC275 ---------------- Index Advisor Report Options -----------------------   
   Command ===>   
   Specify options and press PF3 to return to index analysis.   
   Layout data set : AFDQA.V101TEST.CNTL(PSSREPB)   
   Edit Layout data set : N (Y/N)   
   --------------------- Batch Report Options -------------------------------   
   Generate report title : Y (Y/N)   
   Title : INDEX RECOMMENDATION REPORT

2. In the Layout data set field, specify the name of the data set in which the report layout is stored.

   This layout determines how the report fields are displayed as part of the Report generation. If left blank, the report layout defaults to the HLQ.samplib(PSSREPB) member. The name of this library will vary depending on your installation options. The name could be hlq.BBSAMP, hlq.PSSSAMP, or hlq.BMCSAMP. For more information about report layouts, see “Changing the layout of the compare report” on page 234.

3. If you want to edit the layout data set, complete the following steps:

   A. In the Edit Layout data set field, specify Y (Yes) to update the layout data set.

   The Layout Data Set panel (Figure 37 on page 118) is displayed.
Specifying report options for Index Advisor reports

**Figure 37  Layout Data Set panel (PSSWCLDS)**

<table>
<thead>
<tr>
<th>PSSWCLDS</th>
<th>---------------------------</th>
<th>Layout Data Set</th>
<th>-----------------------------</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command ===</td>
<td></td>
<td>Layout Data Set 'AFDOA.V101DBC.CNTL(PSSREPB)'</td>
<td></td>
</tr>
<tr>
<td>Default Data Set</td>
<td>'AFDOA.V101DBC.CNTL(PSSREPB)'</td>
<td>(Used if layout data set is blank)</td>
<td></td>
</tr>
</tbody>
</table>

**B** In the Layout Data Set Field, enter the name of the data set that you want to edit.

**C** Complete one of the following actions:

<table>
<thead>
<tr>
<th>Action</th>
<th>Steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edit the layout data set</td>
<td>1. In the Option field, type 1 and press Enter.</td>
</tr>
<tr>
<td></td>
<td>2. Edit the layout data set.</td>
</tr>
<tr>
<td></td>
<td>3. Press PF3 to return to the Layout Data Set panel.</td>
</tr>
<tr>
<td></td>
<td>1. In the Option field, type 2 and press Enter.</td>
</tr>
<tr>
<td></td>
<td>2. Browse the layout data set.</td>
</tr>
<tr>
<td></td>
<td>3. Press PF3 to return to the Layout Data Set panel.</td>
</tr>
<tr>
<td>Browse the layout data set</td>
<td>In the Option field, type 3 and press Enter. You return to the</td>
</tr>
<tr>
<td>Use the default data set</td>
<td>Layout Data Set panel.</td>
</tr>
<tr>
<td>Use and edit the default data set</td>
<td>1. In the Option field, type 4 and press Enter.</td>
</tr>
<tr>
<td></td>
<td>2. Edit the layout data set.</td>
</tr>
<tr>
<td></td>
<td>3. Press PF3 to return to the Layout Data Set panel.</td>
</tr>
</tbody>
</table>

**D** Press PF3 to return to the Index Advisor Reports Options panel.

**4** In the Generate report title field, specify whether you want batch reports to include a report title that you specify.

**5** If you set Generate report title field to Y (Yes), enter the title of the report in the Title field. You can enter up to 65 alphanumeric characters for the report title.

**6** Press PF3 to return to the Recommend index panel.
Customizing index recommendation reports

You can customize reports by creating a copy of the HLQ`samplib`(PSSREPB) member and modifying the layout in that file. For information about PSSREPB and the layout, see “Changing the layout of the compare report” on page 234. For information about the fields that make up the report, see “Workload Access Path Compare and Index Advisor report fields” on page 230.
Chapter 7 Exploring advice for exceptions

This chapter presents the following topics:

- Overview of the Exception Advisor ........................................ 122
- Benefits ............................................................................. 122
- Exception Advisor input ......................................................... 123
- Exception Advisor process ....................................................... 123
- Running the Exception Advisor batch job ................................. 124
  - Rules ............................................................................. 124
  - Exception Advisor example ................................................. 125
  - Exception Advisor parameters ........................................... 130
- Viewing the Exception Advisor reports online ......................... 135
- Exception Advisor output ....................................................... 138
  - Exception information ...................................................... 138
  - History of statement performance ..................................... 139
  - Comparison graphs .......................................................... 140
  - SQL text and host variables .............................................. 141
  - Advice ........................................................................... 142
Overview of the Exception Advisor

The Performance Advisor component of SQL Performance for DB2 maintains a DB2 Performance Advisor Database (PADB). Using the PADB, Exception Advisor identifies the root causes of previously triggered exception executions and makes recommendations.

Exception Advisor examines the data accompanying the exception and compares that data to past execution statistics for the same statement in baseline or aggregated tables. You can adjust the analysis without changing code. Exception Advisor bases its recommendations on observed conditions.

**NOTE**
The Performance Advisor component is available only as part of the SQL Performance solution (not as a stand-alone component).

Benefits

SQL tuners that run on IBM® DB2 subsystems have access to large quantities of performance data. As a result, tuning efforts can be quite time-consuming. By narrowing the focus of the tuning effort to those statements that exceed target thresholds, Exception Advisor can often identify the root cause of an exception immediately. The Exception Advisor is not looking at predicted performance such as Explain data but at actual performance history. Because Exception Advisor uses existing data, you incur no further cost for data collection.

With Exception Advisor, you can

- set exception definitions for plan, program, user ID, DB2, correlation ID or any combination
- specify exception thresholds
- specify input parameters for the Exception Advisor job
- modify the Rules data set
Exception Advisor input

Exception Advisor requires the following input:

- IODPAEXC batch job
- user-specified parameters or defaults
- rules data set
- PADB tables

Exception Advisor process

The Exception Advisor process encompasses the following basic tasks:

1. Set up exception definitions in APPTUNE Administration panels.
   For more information about setting up exception definitions, see the System and SQL Performance for DB2 Administrator Guide.

2. After the Data Collector collects performance data (including exception records), load the collected performance data into the PADB.
   For more information about creating and maintaining a PADB in DB2, see Chapter 3, “Managing performance with Performance Advisor.”

3. Run the Exception Advisor job.
   See “Running the Exception Advisor batch job” on page 124.

4. Use online Exception Advisor or batch reports in tuning efforts.
   See “Viewing the Exception Advisor reports online” on page 135.
Running the Exception Advisor batch job

Use this procedure to run the Exception Advisor batch job.

Before you begin

- Ensure you have set up exception definitions in APPTUNE Administration panels.
  
  For more information about setting up exception definitions, see the System and SQL Performance for DB2 Administrator Guide.

- Ensure you have loaded the collected exception data into the PADB.
  
  For more information about creating and maintaining a PADB in DB2, see Chapter 3, “Managing performance with Performance Advisor.”

The IODPAEXC Exception Advisor job is in the *.*SAMP installation library. IODPAEXC runs the Exception Advisor REXX exec. The user parameters are set and documented in the Exception Advisor job.

To run the Exception Advisor batch job

1 Edit the rules data set (IODRUEXC) found in the *.*SAMP installation library.
   
   For more information on editing the rules data set, see “Rules” on page 124.

2 Edit and submit the IODPAEXC Exception Advisor job is in the *.*SAMP installation library. See the following sections for more information on the Exception Advisor job:
   
   - “Exception Advisor example” on page 125
   - “Exception Advisor parameters” on page 130

Rules

The rules data set (IODRUEXC) is located in the *.*SAMP installation library.

The IODPAEXC job references the rules data set in the following DD statement:

```
//RULES DD DISP=SHR, DSN=BMC-SAMP(IODRUEXC)
```
The following example shows part of the rules data set:

```plaintext
* RULE:Timeouts per Escalation;
  RATIO=QTXADEA+QTXATIM/QTXALEX+QTXALES; > 0
  HISTORY:Threshold
  This statement experienced a timeout or deadlock, most likely due to
  lock escalation.
  *
RULE:SyncIO Wait percentage;PERCENTAGE=SYNCWAIT/ELAPTIME; > 30
This execution spent a large percentage of time performing
synchronous I/Os. If the getpage count has risen, it can indicate
a need to rebind the program, update catalog statistics for accessed
objects, or reorganize affected objects. Also check for RID list
failures for the statement and buffer pool performance for accessed
objects.
*
...```

**TIP**

You can create your own rules data set with new or modified rules by using IODRUEXC as a
model. Ensure that you copy your changed member into the ?HLQ?.UBBSAMP data set so
that it is not overwritten when you apply maintenance.

---

**Exception Advisor example**

The following example shows the Exception Advisor job. The input parameters are
explained in comments.

```plaintext
//PAEXC        JOB  (ACCT),'PERFORMANCE ADVISOR', <== MODIFY
//             MSGCLASS=X,                        <== CHECK
//             CLASS=A                            <== CHECK
//*
//******************************************************************/
//*                          INSTRUCTIONS
//*
//*----> CHANGE: ?BMC-CLIST? = BMC CLIST DATA SET
//*              ?BMC-SAMP?  = BMC SAMPLIB DATA SET
//*              ?DBID?      = DB2 SSID FOR PA TABLES
//*              ?DSNLOAD?   = DB2 LOAD LIBRARY
//*              ?DSNEXIT?   = DB2 DSNEXIT LIBRARY
//*
//******************************************************************/
//EXCADV   EXEC PGM=IKJEFT01,DYNAMNBR=20,TIME=1440
//STEPLIB  DD  DISP=SHR,DSN=?DSNLOAD?
//         DD  DISP=SHR,DSN=?DSNEXIT?
//SYSTSPRT DD  SYSOUT=* 
//SYSPRINT DD  SYSOUT=* 
//SYSTSIN  DD  * 
//RULES    DD  DISP=SHR,DSN=?BMC-SAMP?(IODRUEXC)
//SYSSIN  DD  *
```
**EXCEPTION ADVISOR options**

The following options control various aspects of the rexx exec that reports on SQL executions that exceed CPU or elapsed time thresholds.

To get the "IF OMITTED" setting for a particular option, leave the option out by removing or commenting the line. To get the "IF NULL" setting for a particular option, specify nothing to the right of the equal sign.

**exception.startdate**: the first date for which exceptions will be evaluated. Use format "YYYY-MM-DD" or supported SQL expressions such as CURRENT_DATE - 1 DAY. Enclose in quotes to imbed blanks.

- IF OMITTED: "CURRENT_DATE - 1 DAY"
- IF NULL: Unlimited start

**exception.stopdate**: the last date for which exceptions will be evaluated. Use format "YYYY-MM-DD" or supported SQL expressions such as CURRENT_DATE - 1 DAY. Enclose in quotes to imbed blanks.

- IF OMITTED: "CURRENT_DATE - 1 DAY"
- IF NULL: Date of most recent record in statement exception table

**exception.filter**: can be used to limit evaluation by SMFID, DB2 SSID, PLAN, PROGRAM, etc. Specify a WHERE predicate to be applied to the exception table. Enclose in double quotes and use plus sign + for continuation if needed (with double quotes only at beginning and end).

Example:

```
exception.filter=WHERE PROGRAM='PROGRAM1' +
                     AND PLAN='PLAN1'
```

- IF OMITTED: No filtering
- IF NULL: No filtering

**exception.level**: set to SUMMARY or DETAIL. For SUMMARY, all exceptions of same type from same statement will be evaluated as a single exception. For DETAIL, each exception will be evaluated individually.

- IF OMITTED: DETAIL
- IF NULL: DETAIL

**exception.printreports**: 0 or 1. For 0, reports will not be printed as the advisor runs (see exception.updateresults for information on sending results to DB2 tables for online viewing).

- IF OMITTED: 1
- IF NULL: 1

---

**exception.startdate=** "CURRENT_DATE - 1 DAY"

**exception.stopdate=** "CURRENT_DATE - 1 DAY"

**exception.filter=** WHERE PROGRAM='PROGRAM1' +
                     AND PLAN='PLAN1'

**exception.level=** DETAIL

**exception.printreports=** 1
exception.tabcreator: default table owner/creator if subsequent table specifications do not include the owner/creator.
* IF OMITTED: BMCSFTWR
* IF NULL: BMCSFTWR
exception.tabcreator=BMCSFTWR

exception.tab: table that holds statement exception information. This table must exist and be populated with data from the DMDBMERG utility. If specified without owner, exception.tabcreator is the owner.
* IF OMITTED: STMT_EXCEPTIONS
* IF NULL: STMT_EXCEPTIONS
exception.tab=STMT_EXCEPTIONS

exception.texttab: table that holds SQL text. This table is optional; if present it must be populated with data from the DMDBMERG utility. If specified without owner, exception.tabcreator is the owner.
* IF OMITTED: STMT_TEXT
* IF NULL: No text reporting
exception.texttab=STMT_TEXT

exception.hvtab: table that holds host variable value information. This table is optional; if present it must be populated with data from the DMDBMERG utility. If specified without owner, exception.tabcreator is the owner.
* IF OMITTED: STMT_EXCEPTIONS_HV
* IF NULL: No host variable reporting
exception.hvtab=STMT_EXCEPTIONS_HV

exception.obtab: table that holds exception object information. This table is optional; if present it must be populated with data from the DMDBMERG utility. If specified without owner, exception.tabcreator is the owner.
* IF OMITTED: STMT_EXCEPTIONS_OB
* IF NULL: No object reporting
exception.obtab=STMT_EXCEPTIONS_OB

exception.baseline: table that holds baseline statement statistics. This is used to compare performance characteristics of an exceptional statement execution with statistics for the same statement in a baseline. If specified without owner, exception.tabcreator is the owner.
* IF OMITTED: BASELINE_STMT_STATISTICS
* IF NULL: No baseline reporting
exception.baseline=BASELINE_STMT_STATISTICS
**exception.monthly**: table that holds monthly statement statistics. This is used to compare performance characteristics of an exceptional statement execution with statistics for the same statement in a monthly table.

* If specified without owner, exception.tabcreator is the owner.

* **OMITTED**: MONTHLY_STMT_STATISTICS
* **NULL**: No monthly reporting

**exception.monthly**=MONTHLY_STMT_STATISTICS

**exception.weekly**: table that holds weekly statement statistics. This is used to compare performance characteristics of an exceptional statement execution with statistics for the same statement in a weekly table.

* If specified without owner, exception.tabcreator is the owner.

* **OMITTED**: WEEKLY_STMT_STATISTICS
* **NULL**: No weekly reporting

**exception.weekly**=WEEKLY_STMT_STATISTICS

**exception.daily**: table that holds daily statement statistics. This is used to compare performance characteristics of an exceptional statement execution with statistics for the same statement in a daily table.

* If specified without owner, exception.tabcreator is the owner.

* **OMITTED**: DAILY_STMT_STATISTICS
* **NULL**: No daily reporting

**exception.daily**=DAILY_STMT_STATISTICS

**exception.interval**: table that holds interval statement statistics. This is used to compare performance characteristics of an exceptional statement execution with statistics for the same statement in an interval table.

* **OMITTED**: No interval reporting
* **NULL**: No interval reporting

**exception.interval**=

**exception.nondml**: processing for statements other than SELECT, OPEN, FETCH, CLOSE, INSERT, UPDATE, DELETE.

* The following values are supported:
* 0 = do not report exceptions for these statements
* 1 = report exceptions but do not report history
* 2 = report exceptions and history

* **OMITTED**: 1
* **NULL**: 1

**exception.nondml**=1

**exception.starthist**: expressed in the same form as starttime and stoptime, this determines how far back history will be reported.

* **OMITTED**: No limit
* **NULL**: No limit

**exception.starthist**=
exception.updateresults=1

exception.purgeresults="CURRENT_DATE - 7 DAYS"

exception.runtab=XADVISOR_RUNS

exception.ruletab=XADVISOR_RULES

exception.evnttab=XADVISOR_EVENTS

exception.valuetab=XADVISOR_VALUES

exception.histtab=XADVISOR_HISTORY_TITLES
Exception Advisor parameters

Table 12 describes input parameters that control various aspects reported on by the Exception Advisor. The parameters are also explained in the comments of the Exception Advisor job.

**NOTE**

The Exception Advisor parameters have two default value behaviors. One value is the default if the parameter is not specified (omitted). The other default behavior is used when the parameter is specified without a value (NULL).

- **omitted**
  
  To use the omitted value for a parameter, leave the option out by removing or commenting out the line for the parameter.

- **NULL**
  
  To use the null value for a parameter, include the parameter, but specify nothing to the right of the equal sign.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Omitted value</th>
<th>Null value</th>
</tr>
</thead>
<tbody>
<tr>
<td>exception.baseline</td>
<td>table that contains baseline statement statistics</td>
<td>BASELINE_STMT_STATISTICS</td>
<td>no baseline reporting</td>
</tr>
<tr>
<td></td>
<td>Exception Advisor uses this value to compare performance characteristics of</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>an exceptional statement execution with statistics for the same statement</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>in a baseline.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>exception.daily</td>
<td>table that contains daily statement statistics</td>
<td>DAILY_STMT_STATISTICS</td>
<td>no daily reporting</td>
</tr>
<tr>
<td></td>
<td>Exception Advisor uses this value to compare performance characteristics of</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>an exceptional statement execution with statistics for the same statement</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>in a daily table.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table 12  Exception Advisor parameters (part 2 of 5)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Omitted value</th>
<th>Null value</th>
</tr>
</thead>
<tbody>
<tr>
<td>exception.evnttab</td>
<td>if updateresults is 1, table that holds basic information about exceptions</td>
<td>XADVISOR_EVENTS</td>
<td>XADVISOR_EVENTS</td>
</tr>
<tr>
<td>exception.filter</td>
<td>use to limit the statements to be evaluated</td>
<td>no filtering</td>
<td>no filtering</td>
</tr>
<tr>
<td></td>
<td>Specify a WHERE predicate to be applied to the exception table. Enclose the predicate in double quotation marks. Use a plus sign (+) for continuation if needed.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Example</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>exception.filter=&quot;WHERE PROGRAM='PROGRAM1' + AND PLAN='PLAN1'&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>You can use any columns in the STATS_EXCEPTION table for filtering.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>exception.hisetab</td>
<td>if updateresults is 1, table that holds statement history entries</td>
<td>XADVISOR_HISTORY_ENTRIES</td>
<td>XADVISOR_HISTORY_ENTRIES</td>
</tr>
<tr>
<td>exception.histtab</td>
<td>if updateresults is 1, table that holds header information for statement history</td>
<td>XADVISOR_HISTORY_TITLES</td>
<td>XADVISOR_HISTORY_TITLES</td>
</tr>
<tr>
<td>exception.hvtab</td>
<td>table that contains host variable value information</td>
<td>STMT_EXCEPTIONS_HV</td>
<td>no host variable reporting</td>
</tr>
<tr>
<td>exception.interval</td>
<td>table that contains interval statement statistics</td>
<td>no interval reporting</td>
<td>no interval reporting</td>
</tr>
<tr>
<td></td>
<td>Exception Advisor uses this value to compare performance characteristics of an exceptional statement execution with statistics for the same statement in an interval table.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table 12  Exception Advisor parameters (part 3 of 5)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Omitted value</th>
<th>Null value</th>
</tr>
</thead>
<tbody>
<tr>
<td>exception.level</td>
<td>whether to evaluate multiple exceptions individually or together as one:</td>
<td>DETAIL</td>
<td>DETAIL</td>
</tr>
<tr>
<td></td>
<td>• SUMMARY evaluates all exceptions of same type from the same statement as a single exception.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• DETAIL evaluates each exception individually.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>exception.monthly</td>
<td>table that contains monthly statement statistics</td>
<td>MONTHLY_STMT_STATISTICS</td>
<td>no monthly reporting</td>
</tr>
<tr>
<td></td>
<td>Exception Advisor uses this value to compare performance characteristics of an exceptional statement execution with statistics for the same statement in a monthly table.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>exception.nondml</td>
<td>whether to process statements other than SELECT, OPEN, FETCH, CLOSE, INSERT, UPDATE, and DELETE</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>The following values are supported:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 0 = do not report exceptions for these statements</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 1 = report exceptions but do not report history</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 2 = report exceptions and history</td>
<td></td>
<td></td>
</tr>
<tr>
<td>exception.obtab</td>
<td>table that contains exception object information</td>
<td>STMT_EXCEPTIONS_OB</td>
<td>no object reporting</td>
</tr>
<tr>
<td></td>
<td>This table is optional; if present it must be populated with data from the DMDBMERG utility. If specified without owner, exception.tabcreator is the owner.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>exception.printreports</td>
<td>whether to process reports as online Exception Advisor table output</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>• 0 = reports will not be printed as the advisor runs</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 1 = reports are generated as advisor output</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>The exception.updateresults parameter controls sending results to DB2 tables for online viewing.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table 12  Exception Advisor parameters (part 4 of 5)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Omitted value</th>
<th>Null value</th>
</tr>
</thead>
<tbody>
<tr>
<td>exception.purgeresults</td>
<td>if updateresults is in effect, this specifies the age at which existing rows in the XADVISOR results tables are purged.</td>
<td>CURRENT_DATE - 7 DAYS</td>
<td>No rows deleted</td>
</tr>
<tr>
<td>exception.ruletab</td>
<td>if updateresults is 1, table that holds advice related to defined rules (from //RULES DD) If specified without owner, exception.tabcreator is the owner.</td>
<td>XADVISOR_RULES</td>
<td>XADVISOR_RULES</td>
</tr>
<tr>
<td>exception.runstart</td>
<td>table that holds information pertaining to this and other executions of the advisor, such as number and time range of exceptions reported. If specified without owner, exception.tabcreator is the owner.</td>
<td>XADVISOR_RUNS</td>
<td>XADVISOR_RUNS</td>
</tr>
<tr>
<td>exception.startdate</td>
<td>the first date for which exceptions will be evaluated Use format yyyy-mm-dd or supported SQL expressions such as CURRENT_DATE - 1 DAY. Blank specifies no limit.</td>
<td>CURRENT_DATE - 1 DAY</td>
<td>unlimited start</td>
</tr>
<tr>
<td>exception.starthist</td>
<td>the date at which to start reporting historical data use the same format as the exception start date</td>
<td>no limit</td>
<td>no limit</td>
</tr>
<tr>
<td>exception.stopdate</td>
<td>the last date for which exceptions will be evaluated Use format yyyy-mm-dd or supported SQL expressions such as CURRENT_DATE - 1 DAY. Blank specifies Date of most recent record in statement exception table.</td>
<td>CURRENT_DATE - 1 DAY</td>
<td>date of most recent record in statement exception table</td>
</tr>
<tr>
<td>exception.tab</td>
<td>table that contains statement exception information This table must exist and be populated with data from the DMDBMERG utility. If specified without an owner, exception.tabcreator is the owner.</td>
<td>STMT_EXCEPTIONS</td>
<td>STMT_EXCEPTIONS</td>
</tr>
<tr>
<td>exception.tabcreator</td>
<td>default table owner or creator if subsequent table specifications do not include the owner or creator</td>
<td>BMCSFTWR</td>
<td>BMCSFTWR</td>
</tr>
</tbody>
</table>
### Table 12  Exception Advisor parameters (part 5 of 5)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Omitted value</th>
<th>Null value</th>
</tr>
</thead>
<tbody>
<tr>
<td>exception.texttab</td>
<td>table that contains SQL text</td>
<td>STMT_TEXT</td>
<td>no text reporting</td>
</tr>
<tr>
<td></td>
<td>This table is optional; if present it must be populated with data from the DMDBMERG utility. If specified without an owner, exception.tabcreator is the owner.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>exception.updateresults</td>
<td>whether to update the ADVISOR tables with the results of this run for subsequent online viewing</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>- 0 = do not update tables with results</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- 1 = update tables with results</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>If 1 is specified, exception.level must be set to DETAIL.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>exception.valuetab</td>
<td>updateresults is 1, table that holds exception values that exceeded thresholds</td>
<td>XADVISOR_VALUES</td>
<td>XADVISOR_VALUES</td>
</tr>
<tr>
<td></td>
<td>If specified without owner, exception.tabcreator is the owner.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>exception.weekly</td>
<td>table that contains weekly statement statistics</td>
<td>WEEKLY_STMT_STATISTICS</td>
<td>no weekly reporting</td>
</tr>
<tr>
<td></td>
<td>Exception Advisor uses this value to compare performance characteristics of an exceptional statement execution with statistics for the same statement in a weekly table.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Viewing the Exception Advisor reports online

Use this procedure to view the Exception Advisor online reports.

Before you begin

Ensure you have run the Exception Advisor batch job. For more information about running the Exception Advisor batch job, see “Running the Exception Advisor batch job” on page 124.

To view the Exception Advisor online reports

1. Access the Exception Advisor.
   
   A. On the SQL Performance for DB2 main menu, select A (Performance Advisors) and press Enter.
   
   B. On the Performance Advisors panel, select B (Exception Advisor) and press Enter.

   The Exception Advisor panel is displayed (Figure 38).

   Figure 38 Exception Advisor panel (IODEWCMB/I)

<table>
<thead>
<tr>
<th>IODEWCMB/I</th>
<th>Exception Advisor</th>
</tr>
</thead>
<tbody>
<tr>
<td>IODEWCMB/I</td>
<td>Exception Advisor</td>
</tr>
<tr>
<td>Command =====&gt; ________________________________________________________________</td>
<td></td>
</tr>
<tr>
<td>Current Data Collector : BA1D   Status : ACTIVE   Data Collection : READY</td>
<td></td>
</tr>
<tr>
<td>DB2 subsystem where Performance Advisor Database resides: DEFF (blank for list)</td>
<td></td>
</tr>
<tr>
<td>Name of Exception Advisor Run Table--Specify synonym, table, view, alias, creator.table, creator.view, or creator.alias:</td>
<td></td>
</tr>
<tr>
<td>BMCAFD_XARUN ________________________________________________________________</td>
<td></td>
</tr>
<tr>
<td>(blank for synonym BMCAFD_XARUN)</td>
<td></td>
</tr>
</tbody>
</table>

2. Use the Exception Advisor panel to specify the location of your Exception Advisor Run Table.

   This table records information about recent executions of the Exception Advisor. In order to view the Exception Advisor reports, you must specify a name that identifies the table, along with the name of the DB2 where the table resides. The DB2 that you specify must be accessible from the DOMPLEX to which the current Data Collector belongs.
A The **Current Data Collector** field shows the subsystem ID of the currently selected Data Collector. If there is no Data Collector selected, this field is blank.

B The **Status** field shows the status of the currently selected Data Collector. The possible values are:

- ACTIVE
- DOWN
- STOPPING
- INVALID

If there is no Current Data Collector subsystem selected, this field is blank.

**NOTE**

A status of INVALID indicates one of the following:

- A Data Collector has been defined using an SSID that already exists on the MVS system (a DB2 SSID, for example). If this is the case, define a new Data Collector using a valid SSID and delete the invalid Data Collector.

- The selected Data Collector is for a level of the product prior to the level currently being used. The Data Collector must be migrated to the current level before you can use it.

C In the **DB2 subsystem where Performance Advisor Database resides** field, specify the subsystem name, or leave blank and press Enter to select the subsystem list of DB2 subsystems known to the current Data Collector.

D In the **Name of Exception Advisor Run Table** field, specify the fully-qualified name of a table, view, or alias, or the synonym that defines the Exception Advisor Run Table to which the Exception Advisor recorded its results. This is the exception.runtab value from your Exception Advisor batch job.

E Press Enter to view the EXCEPTION ADVISOR-EXCEPTIONS panel (Figure 39 on page 137).
On the EXCEPTION ADVISOR-EXCEPTIONS panel, you can view more information about the exceptions.

On exception lines that start with a + (plus) sign, the following commands can be executed by typing over the + with the letter and pressing Enter:

- **R**–See the rules for the exception.
- **S**–See the SQL text for the exception.
- **V**–See the host variables associated with the exception.
- **O**–See the objects associated with the exception.
- **1**–See more information on the exception.

On exception detail lines that start with a . (period), the following commands can be executed by typing over the . with the letter and pressing Enter:

- **A**–See advice related to this threshold.
- **H**–See exception history related to this threshold.
Exception Advisor output

The batch and online Exception Advisor produce the same output. The Exception Advisor produces detailed or summary reports that provide the following items:

- exception information
- history of statement performance
- comparison graphs
- SQL text and host variables
- advice

Exception information

The exception information includes the following items:

- time of exception (detail), or number and time range of exceptions (summary)
- statement identifiers, such as system, subsystem, plan, program, statement type, section, and statement number
- reason for the exception (elapsed or CPU time), elapsed time, and CPU time
  In summary mode, the avg/max/min values across all exceptions are shown.
- value of any rule threshold exceeded and the advice associated with that rule for the exception or group of exceptions
- value of any statistics from rules that are designated as always reported (such as GETPAGE COUNT in the following examples)
The following example shows an exception summary report:

```
Exception summary: 10 exceptions 2010-01-20-01.07.01.313464
through 2010-01-20-20.54.59.918471

DSDHQ (SYSM/DHQ6 8.1) Member DHQ6 of group DSDHQ
Plan SPRTST00 DBRM SPRTST00 STATIC DELETE Section 700

Reason for exception: CPU TIME
Elapsed min/max/avg: 9.883207/15.287155/12.206514
CPU min/max/avg: 2.165759/2.337422/2.239000
GETPAGE COUNT is 191788

```

The following example shows an exception detail report:

```
***Exception detail: 2010-01-20-08.05.31.344934

DSDHZ (SYSM/DHZ6 9.1) Member DHZ6 of group DSDHZ
Plan DSNTEP2 DBRM DSNTEP2 DYNAMIC OPEN Section 1

Statement 1801
Reason for exception: CPU TIME Elapsed time: 4.772373 CPU time: 0.963468
SYNCIO WAIT PERCENTAGE is 52.203
*ADVICE: This execution spent a large percentage of time performing synchronous I/Os. If the getpage count has risen, it can indicate a need to rebind the program, update catalog statistics for accessed objects, or reorganize affected objects. Also check for RID list failures for the statement and buffer pool performance for accessed objects.

GETPAGE COUNT is 43216

```

History of statement performance

The following example shows a portion of the performance history section of an elapsed time exception report:

```
+-------------------+-------------------+--------------+-------+---+-----+-----+
| Elapsed time      | Average History level | Statement     | Type   | Calls | Selects |
|-------------------+-------------------+--------------+-------+---+-----+-----+---|
| >                 | 0.000021 MONTHLY 2009-06 | SELECT       | 4000  | 4000 |
| >>>>>>            | 1.853088 MONTHLY 2009-10 | SELECT       | 25892 | 25892 |
| >>>>>>            | 1.783560 WEEKLY 2009-10-11 | SELECT       | 1581  | 1581  |
| >>>>>>            | 2.011105 WEEKLY 2009-10-18 | SELECT       | 10466 | 10466 |
| >>>>>>            | 1.741576 WEEKLY 2009-10-25 | SELECT       | 13845 | 13845 |
| >>>>>>            | 1.349979 WEEKLY 2009-11-01 | SELECT       | 6394  | 6394  |
| >>>>>>            | 1.349979 MONTHLY 2009-11 | SELECT       | 6394  | 6394  |
| >>>>>>            | 1.349979 MONTHLY 2009-11 | SELECT       | 214   | 214   |
| >>>>>>            | 2.009972 MONTHLY 2009-12 | SELECT       | 214   | 214   |
| >>>>>>            | 0.304685 WEEKLY 2009-12-13 | SELECT       | 53548 | 5548  |
| >>>>>>            | 0.351906 WEEKLY 2009-12-20 | SELECT       | 2817  | 2817  |
| >>>>>>>            | 2.009972 WEEKLY 2009-12-27 | SELECT       | 214   | 214   |
| >>>>>>>            | 6.893685 XCPT:01-20-00.59.38.988630 | SELECT | 1     | 1     |

+-------------------+-------------------+--------------+-------+---+-----+-----+
```
Comparison graphs

Exception Advisor provides graphical comparisons of elapsed and CPU times. Exception Advisor also compares other performance indicators if they are present in current or historical records for the exception statement.

The following example shows CPU time and elapsed time comparisons:

<table>
<thead>
<tr>
<th>Historical comparison of CPU time with Elapsed time</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU time</td>
</tr>
<tr>
<td>-----------------+-----------------+----------+-------------+-----------------+-------+-------</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

The following example shows a CPU time exception comparison of historical CPU time with SYNCIO WAIT PERCENTAGE and GETPAGE COUNT. Note the correlation between high CPU time and high GETPAGE COUNT:

<table>
<thead>
<tr>
<th>Historical comparison of CPU time with SYNCIO WAIT PERCENTAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU time</td>
</tr>
<tr>
<td>-----------------+-----------------+------+-------------+-----------------+-------+-------</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Historical comparison of CPU time with GETPAGE COUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU time</td>
</tr>
<tr>
<td>-----------------+-----------------+----------+-------------+-----------------+-------+-------</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

The following example shows a historical comparison of CPU time with LOCK WAIT PERCENTAGE.

**TIP**

This example shows no correlation between CPU time and LOCK WAIT PERCENTAGE. That would suggest that locking problems are not a significant contributor to the statement exception CPU usage.
### SQL text and host variables

Both summary and detail reports display SQL text. The following example shows some SQL text from the exception report:

```sql
SQL text:
UPDATE AFDQA.QATBL01
SET DATETIME_EFF = CURRENT_TIMESTAMP
WHERE COL_A = 13
```
The following example shows a detail report with host variables, SQL text, and objects:

**Host variables:**

<table>
<thead>
<tr>
<th>#</th>
<th>(TYPE)</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td># 1</td>
<td>INTEGER</td>
<td>38</td>
</tr>
</tbody>
</table>

**Accessed objects:**

<table>
<thead>
<tr>
<th>Object name</th>
<th>(Type)</th>
<th>BP Name</th>
<th>Getpages</th>
<th>Getpage time</th>
<th>%elap</th>
<th>SyncIOs</th>
<th>SyncIO time</th>
<th>%elap</th>
</tr>
</thead>
<tbody>
<tr>
<td>AFDQA.QATBL01</td>
<td>T</td>
<td>BP0</td>
<td>23263</td>
<td>0.001641</td>
<td>0.0</td>
<td>1</td>
<td>6.074652</td>
<td>88.1</td>
</tr>
</tbody>
</table>

**SQL text:**

```sql
FROM AFDQA.QATBL01
WHERE COL_A = :H
```

---

**Advice**

Advice contained in Rules member IODRUEXEC is triggered by thresholds. Sometimes the advice can be very specific and point to an immediate solution:

*ADVICE:* The number of parallel groups did not reach the planned level due to buffer pool limitations. Increase the size of the buffer pool in which objects accessed by this statement reside.

In other cases, the advice might suggest additional research paths:

*ADVICE:* This execution spent a high percentage of time waiting on locks or latches. You can use EXPLAIN to determine the locking characteristics of this statement; you can get details on locking activity with MainView for DB2.

The following examples show a rule and the corresponding advice that Exception Advisor issued after a threshold was surpassed:
Rule example

RULE:SyncIO Wait percentage;PERCENTAGE=SYNCHWT/ELAPTIME; > 30
This execution spent a large percentage of time performing
synchronous I/Os. If the getpage count has risen, it can indicate
a need to rebind the program, update catalog statistics for accessed
objects, or reorganize affected objects. Also check for RID list
failures for the statement and buffer pool performance for accessed
objects.

Advice example

*ADVICE: This execution spent a large percentage of time performing
synchronous I/Os. If the getpage count has risen, it can indicate a
need to rebind the program, update catalog statistics for accessed
objects, or reorganize affected objects. Also check for RID list
failures for the statement and buffer pool performance for accessed
objects.
Index Component reports

This chapter presents the following topics:

Index Component report set .......................................................... 145
  Report hierarchies ........................................................................ 146
  Action codes ............................................................................... 150
  Reporting intervals ..................................................................... 150
Using panels .................................................................................. 150
Using online reports ...................................................................... 151
Viewing reports ............................................................................. 152
  Application groups and application profiles ................................ 153
  IN-SQL elapsed time measurement ............................................. 153
  Explain processing ...................................................................... 155
  Additional reporting features ..................................................... 156
Batch reporting .............................................................................. 157
  Report format ............................................................................ 157
  Batch utilities ............................................................................. 158
  Batch report qualifiers ................................................................ 159

Index Component report set

The Index Component reporting process invokes a series of input panels and reports based on the selection criteria specified. The overall structure of these panels and reports is designed to provide increasingly specific information until an SQL statement is identified. The options provided on these panels and reports allow you to specify reporting criteria appropriate for your site.
Figure 40 shows a hierarchy of reports in SQL Performance. Index Component initial reports IODTAPPL, IODDSTAT, IODTGETV, and IODIGETV link to a series of additional reports. For more information, see

- “Application Group Getpage Volume reports” on page 147
- “Table Getpage Volume reports” on page 148
- “Index Getpage Volume reports” on page 149

Figure 40  **Index Component reports**

Application Group Getpage Volume reports

Figure 41 shows the navigation from the Application Group Getpage Volume report.

Figure 41  IODTAPPL reports
Table Getpage Volume reports

Figure 42 shows the navigation from the Table Getpage Volume report.

Figure 42  IODTGETV reports
Index Getpage Volume reports

Figure 43 shows the navigation from the Index Getpage Volume report.

Figure 43   IODIGETV reports
Action codes

Action codes are used to navigate from reports to other reports. Action codes are listed in the report header. To use an action code, type the code over the plus sign (+) on the left margin of the report, and press Enter.

Reporting intervals

The data used by SQL Performance for Workload Analysis reporting can come from either the currently selected Data Collector or a data set.

When the source is a Data Collector, the current activity as well as the records in the current trace data set are available for reporting. The product creates trace records and writes them to the trace data sets periodically whenever the Data Collector is active. You control the frequency at which these records are cut by specifying options on the SQL Statistics Collection Interval panel.

When the source is a data set, all records in the data set are available for reporting. You can use either archived trace data sets or EXPORT data sets (created by copying the data from a report online to a data set using the EXPORT command).

Each time you use the Workload Analysis reports, you must select a time period over which reporting will take place. That time period will consist of one or more “intervals” from the data in either the current trace data sets or in the source data set.

For more information about reporting intervals, see the online Help. Type HELP APINT on the Command line of common panels, and press Enter.

Using panels

To help you navigate through SQL Performance, some panels display current reporting criteria in the panel header. In the example shown in Figure 44 on page 151, current reporting criteria (in bold print) is displayed in the header of the SQL Workload Analysis Menu. When you change reporting criteria, the information in the menu header changes accordingly.

Specific information about fields displayed on panels is provided in the Index Component’s online Help facility. Move the cursor to any input or output field, and press F1 (Help). For detailed information about using online Help, see “Online assistance” on page 47.
Using online reports

As you travel from one report to another, identifiers located in the report header are displayed to help you keep track of the selections you made on your way to the current report. In Figure 45, the identifiers (in bold print) show the statement list for the table RDADMB.DMBTBDYN1, which is running on DB2 subsystem DECX.

Using online reports

Figure 44  Example of navigation—SQL Workload Analysis Menu (AS QUEWAM1)

Figure 45  Example of navigation—Statement List report
Viewing reports

The reporting process begins at the APPTUNE and Index Component Main Menu (Figure 46), from which you can invoke the SQL Workload Analysis Menu (Figure 47) to specify data collection criteria.

Figure 46  APPTUNE and Index Component Main Menu (IODESELCl)

Figure 47  SQL Workload Analysis Menu (ASQEWMAl)
The SQL Workload Analysis Menu (Figure 47 on page 152) displays the current reporting criteria. You can begin reporting immediately using the criteria displayed, or you can change the reporting criteria in the following ways:

- specify the initial report to be displayed

  After the initial report is displayed, you can go to any or all of the remaining reports to view the same data from a different perspective.

- select an application profile to use

- select the time period to be used for reporting

- specify the source of the data used—archived data set or Data Collector

Index Component reports are initially sorted in descending order by the percentage of elapsed time consumed by each object, but you can change the order using the SORT command.

As you zoom between reports, identifiers located in the report header are displayed to help you keep track of the selections you made on your way to the current report. After a report is generated, you can return to it any time during a reporting session.

**Application groups and application profiles**

SQL Performance provides the ability to define application groups and application profiles. Application groups contain the plans, programs, users, DB2s, and correlation IDs you select, and application profiles contain the application groups you define. This enables you to select the aspects of a DB2 workload you want to include (or exclude) for application performance analysis. For more information, see the APPTUNE for DB2 User Guide.

**IN-SQL elapsed time measurement**

To provide a more accurate method of determining the cost of an SQL statement, the Index Component calculates the resources consumed during the execution of an SQL statement by measuring an event called IN-SQL elapsed time. This event (IN-SQL elapsed time) provides the most accurate data available for determining the cost of executing an SQL statement because it excludes the time spent on associated DB2 housekeeping tasks and measures only the time spent executing an SQL statement. Measurements based on IN-DB2 time are less focused and overestimate the actual cost of SQL statement execution.
Figure 48 illustrates the execution of an SQL statement within the context of class 2 (IN-DB2) elapsed time and the context of class 1 (in-application) elapsed time. The SQL statement on the left in Figure 48 depicts the three distinct phases of SQL statement execution—prologue time, IN-SQL elapsed time, and epilogue time.

**Figure 48  SQL statement execution measurement**

![Diagram showing the execution of an SQL statement with prologue time, IN-SQL elapsed time, and epilogue time, along with class 1 and class 2 elapsed times.]

**Prologue time**

The first phase of activity during the execution of an SQL statement. Examples of tasks performed during prologue time include the following:

- thread creation/reuse
- plan allocation
- authorization checking
- determination of implicit qualifier to use for unqualified tables

**IN-SQL elapsed time**

The second phase of activity during the execution of an SQL statement, when an SQL query is satisfied. Examples of tasks performed during IN-SQL elapsed time include the following:

- parsing of dynamic SQL statements
- logical and physical I/O processing to access data
- required data set allocation and deallocation
- log writes
Epilogue time

The third phase of activity during the execution of an SQL statement. Examples of tasks performed during epilogue time include the following:

- return of result set
- IFCID creation
- commit processing
- rollback processing
- thread termination

Explain processing

To complement workload analysis, SQL Performance includes an integrated, robust Explain function. This Explain function allows you to dynamically Explain dynamic or static SQL statements. The Explain function provides both statistical and textual information about the access path along with suggestions on how to improve SQL statement performance. This Explain function also allows you to export an SQL statement to an ISPF edit session for modification and return it to the SQL Performance environment to be explained.

An EXPLAIN command can be issued from any SQL statement reported on the SQL Statement Text report or you can use the Explain Object Specification panel (PSSPA115).

Explain output is displayed in the Explain Results panel (Figure 49) and includes helpful information about improving the performance of the SQL statement. You can use action codes to display the plan table summary, catalog statistics, and index key columns.

Figure 49   Explain Results panel (PSSPW200)
You can also issue historical Explains (static Explains) and retrieve SQL text from the catalog from the SQL Statement Text report. Further, you can issue an EXPLAIN COMPARE on a static SQL statement from the SQL workload. This process performs a dynamic Explain and a static Explain, matching the STMTNO from the catalog to QUERYNO in the PLAN_TABLE.

For more information about Explain, see the APPTUNE for DB2 User Guide and the SQL Explorer for DB2 User Guide.

**Additional reporting features**

The following features are available to enhance SQL Performance’s reporting capability:

**EXPORT command**

While viewing a report, you can use the EXPORT command to save all of the data collected for that report and write the records to a data set you specify. Later, you can view the exported data online or produce batch reports using the export data set as the source. All SQL Performance data is written to trace data sets, but the EXPORT command allows you to save specific data to your own data set.

**Report logging**

Report logging enables you to save copies of reports and other screen images in a data set for later viewing, printing, or both. You can log individual reports, portions of reports, and screen images manually using the LOG command, or you can use the Log Options panel to automatically log all reports. When you log reports automatically, images of entire reports are written to the report log data set specified.

**Session status**

Session status allows you to display some key values concerning your user session that are useful to monitor periodically. You can use the STATUS command to display the Session Status panel. The Session Status panel reports storage usage data concerning your session, and information about the reports currently active under your user ID.

For more information about SQL Performance reports, see the online Help for any individual report (type HELP <reportName>) on the Command line of any panel or report, and press Enter.
Batch reporting

The Index Component provides the ability to produce most of its reports in both online and batch formats, so you can schedule routine reporting activities as needed. In addition, APPTUNE provides several utilities to help you process batch reports.

Report format

While most reports are available in batch or online, some reports are produced in an easy-to-read format designed especially for batch reporting. In cases requiring analysis of large volumes of data, a printed report is often easier to read than an online report that requires extensive scrolling.

Batch Subsystem GETPAGE Volume (IODDSTAT)

The Batch Subsystem GETPAGE Volume report provides overview of GETPAGE and I/O activity on all active DB2s in the sysplex. The logical DB2 name is the name of a data sharing group when data sharing is active or the DB2 subsystem ID when no data sharing is taking place.

Batch Index GETPAGE Volume (IODIGETV)

The Batch Index GETPAGE Volume report provides an overview of GETPAGE and I/O activity for all indexes on the sysplex, ordered by the volume of GETPAGEs associated with the indexes. Only indexes with GETPAGE activity are reported. The report is sorted by the total number of GETPAGEs in descending order.

Batch Table Information for an Index (IODITABL)

The Batch Table Information for an Index report shows statistics for the table associated with the selected index and any other indexes associated with the same table.

Batch Statement List (IODSTMTS)

The Batch Statement List report lists all of the SQL statements associated with the selected table. This report displays one line of statistics for each SQL statement, sorted by the number of GETPAGE requests, with the largest number reported first.
Batch utilities

The batch utilities provided with APPTUNE perform the following tasks:

- produce printed reports from batch jobs using active or archived trace data sets as input
- print all entries from a specified report log containing logged reports and screen images
- archive the current trace data set, switch to the next available VSAM trace data set, and make periodic copies of trace data sets for long-term storage

For more information, see the APPTUNE for DB2 Reference manual.
Table 13 shows the qualifiers that you can use for each of the reports.

### Table 13  Report qualifiers used by the Index Component batch reports

<table>
<thead>
<tr>
<th>Report</th>
<th>Report description</th>
<th>DB2</th>
<th>Application Group</th>
<th>Operator ID</th>
<th>Comm ID/MP number</th>
<th>Corr ID/WP ID</th>
<th>Plan name</th>
<th>DBRM/PG/Server</th>
<th>Buffer pool ID</th>
<th>Database name</th>
<th>Page set</th>
<th>Table name</th>
<th>Table creator</th>
</tr>
</thead>
<tbody>
<tr>
<td>IODDSTAT</td>
<td>Subsystem Getpage Volume</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>IODIGETV</td>
<td>Index Getpage Volume</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>IODITABL</td>
<td>Table Information for an Index</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>IODSTMTS</td>
<td>Statement List</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>IODSTXDT</td>
<td>Table/Index Breakdown</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>IODTAGPL</td>
<td>Application Group Getpage Volume</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>IODTGETV</td>
<td>Table Getpage Volume</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Overview of the Index Component

The Index Component of SQL Performance extends the capability of APPTUNE object analysis by collecting and reporting on column usage data for SQL statements. It also extends the capability of the Explain function by comparing access paths after making changes to indexes in a cloned database.

The Index Component offers users a way to obtain accurate, real-time performance information about DB2 indexes. Because the Index Component presents data at the object level, you can review the index access data to evaluate the performance of your indexes and identify candidates for index improvements. You can use the Index Component to analyze index activity from the following views:

- subsystem
- object
- application group
Preparing to use What-If Index

You can initiate Index Component reports by using an initial index report, or through the Object Analysis report (SQMCACTO) in APPTUNE for DB2. For more information, see Chapter 8, “Index Component reports.”

**NOTE**
To use the Index Component, you must select Y for Object data in the filter option set.
For more information about the filter option set, see the System and SQL Performance for DB2 Administration Guide.

The tasks in this chapter demonstrate different approaches to analyzing index performance.

### Preparing to use What-If Index

In this task, you specify a data source, select an interval, and specify an initial report for your analysis.

**NOTE**
The tasks in this chapter describe a single method for invoking Explain and What-If Index. Using SQL Performance, you can invoke Explain and What-If Index from any location in the product where SQL text is visible.

1. On the System and SQL Performance for DB2 main menu, select APPTUNE and Index Component, and press Enter.

The APPTUNE and Index Component Main Menu is displayed (Figure 50).

![Figure 50 APPTUNE and Index Component Main Menu (DOMEPNL3)](image)
2 Select SQL Workload, and press Enter.

The SQL Workload Analysis Menu is displayed (Figure 51). This menu displays the criteria that are used to select data for workload analysis reporting, and offers options to begin reporting and to define new reporting criteria.

Figure 51 SQL Workload Analysis Menu (DOMEPNL3)

3 Select Data source, DB2(s) and press Enter to display the Data Source panel (Figure 52).

Figure 52 Data Source panel (DOMEPNL3)
4 On the Data Source panel, specify data source information for reporting, and press F3.

**WARNING**

Index analysis is sensitive to certain objects and their statistics, and to the SQL executed against those objects. Because of this, BMC Software recommends that you limit your index reporting to one DB2 subsystem or data sharing group at a time. Reporting on index information gathered from multiple subsystems or from subsystems that are not related by data sharing could be misleading.

5 At the SQL Workload Analysis menu (Figure 51 on page 163), select **Time interval** and press **Enter**.

The Select Analysis Interval report is displayed (Figure 53).

6 To select an interval, type S beside a single interval or beside the first and last intervals in the range, and press **Enter**. Press F3 when finished.

7 At SQL Workload Analysis menu (Figure 51 on page 163), select **Initial report**, and press **Enter**.

The SQL Workload Initial Analysis Level panel is displayed (Figure 54 on page 165).
Identifying What-If Index candidates

In this task, you use the Index Component to identify a table that has both a significant amount of getpage activity and a minimal amount of index getpage activity associated with it. Such a table could benefit from the addition of an index.

1. Use “Preparing to use What-If Index” on page 162 to define options for reporting.

2. Examine the Table Getpage Volume report (Figure 55 on page 166) for an overview of getpage and I/O activity for all tables (and their indexes) on the sysplex, ordered by the volume of getpages that are associated with the tables.
Identifying What-If Index candidates

Each table has one line of statistics. Only tables with getpage activity are reported. The Getpage Index % values provide information about the efficiency of the indexes on the tables. A large Getpage Index Number value with a low Getpage Index % value could be a good candidate for What-If Index analysis.

For table RDADMB.DMBTBDYN1, the value in the Getpage Number column shows that this table had getpage requests and that the percentage of total getpages for this table that can be attributed to its indexes is 0.0%.

Figure 55 Table Getpage Volume report (IODTGETV)

3 To view table statistics, type T over the plus sign (+) beside the table, and press Enter.

The Table Statistics report is displayed (Figure 56 on page 167). This report shows the physical characteristics of the table, along with RUNSTAT statistics that you can use to analyze the index.
Identifying What-If Index candidates

Chapter 9 Using the Index Component

4 To view index information for the table, type N over the asterisk (*) beside the table name, and press Enter.

The Defined Indexes for a Table report is displayed (Figure 57). This report helps you to determine whether indexes exist for a table. In this example, the table has no defined indexes.

Figure 57 Defined Indexes for a Table report (IODTSTIN)
5 Press F3 until you return to the Table Getpage Volume report (Figure 58).

Figure 58  Table Getpage Volume report (IODTGETV)

6 To view a CRUD Matrix for the table, type M over the plus sign (+) beside a table, and press Enter.

The Table CRUD Matrix report is displayed (Figure 59). This report shows the columns of the selected table and the manner in which they have been accessed. In this example, COL01 was used a number of times in an indexable predicate but was never updated. This column could be a good candidate to include in an index.

Figure 59  Table CRUD Matrix report (IODTCRUD)
7 Press F3 to return to the Table Getpage Volume report (Figure 60).

Figure 60 Table Getpage Volume report (IODTGETV)

8 To view statement information for the table, type an S over the plus sign (+) beside a table name, and press Enter.

The Statement List report is displayed (Figure 61). This report shows all statements that use the specified table.

Figure 61 Statement List report (IODSTMTS)
To view SQL text for a statement, type T over the plus sign (+) beside a statement, and press Enter.

The Statement Text report is displayed (Figure 62).

**Figure 62 Statement Text report (IODSTXDH)**

From this report, you can begin using What-If Index to evaluate the effects of various index changes. For more information, see "Using the What-If Index function."

---

**Using the What-If Index function**

In this task, you use the What-If Index function of SQL Performance to simulate the effects of adding, dropping, or updating statistics for an index using cloned structures. You will also save DDL for your What-If Index changes to a data set, and then import those changes on another statement to see the effects of the changes on that statement. This task contains the following procedures:

- “Adding an index by using What-If Index” on page 171
- “Dropping an index by using What-If Index” on page 177
- “Updating statistics on an index by using What-If Index” on page 180
- “Saving What-If Index DDL changes” on page 183
- “Importing What-If Index changes to another statement for a table” on page 186
- “Adding an expression to a What-If Index” on page 191
Adding an index by using What-If Index

This task explains how to use the What-If Index feature to simulate the effects of adding an index.

Before you begin

- Use “Preparing to use What-If Index” on page 162 to specify options in preparation for your analysis.
- Use “Identifying What-If Index candidates” on page 165 to locate a table that could benefit from the addition of an index.

To add an index

1. On the Statement Text report (Figure 63), type X (for Explain Stmt) over the plus sign (+) beside the SQL statement text to be Explained, and press Enter.

![Figure 63 Statement Text report (IODSTXDH)]

NOTE
The procedures in this task use examples that illustrate the cumulative nature of the What-If Index process. In actual practice, you could perform What-If Index actions in any order after creating cloned objects on the Explain results.
The Explain Results panel (Figure 64) is displayed.

**Figure 64  Explain Results panel (PSSPW200)**

2 To clone the objects for that statement, type **C** in the space beside the statement and press **Enter**.

The cloned objects appear on the Explain Results panel (Figure 65). Notice that the objects will be created in database *PSSnnnnn* (where *nnnn* is a generated random number). The objects will be created when you perform an Explain.

**NOTE**
If you are using DB2 Version 9 and create a clone, the SEGSIZE of the clone defaults to 4. In earlier versions of DB2, the SEGSIZE value defaulted to 0. This change in behavior can impact the cost in the “What-If” analysis.
3 To specify a What-If Index, type W in the space beside the table and press Enter.

The Specify Index Attributes panel is displayed (Figure 66). Specifying a What-If Index (W) on the table will display default values for the new index as a starting point. Specifying a W on an existing index in the clone will use that index definition as a starting point for your new index.

**Figure 66 Specify Index Attributes panel (PSSPWIXA)**

4 Specify attributes for a new index to be added to the table, select Specify key columns and Update index and key column statistics, and press Enter.

The Select Key Columns panel is displayed (Figure 67).

---

**NOTE**

You can type S beside the statement on this panel, and press Enter to show nicely formatted text. Doing so is helpful when selecting index columns if you have a long statement and want to review the contents of the predicates. Type S, and press Enter again to hide this information.
To select the key columns for the index, specify sequence numbers (SEQ) and order (ascending (A) or descending (D)).

**NOTE**

If you are using DB2 Version 9 or later, you have the following additional options:

- You can also specify random (R) next to a column in the Order field. Doing so defines a random key column in a similar way to using the RANDOM option with the CREATE INDEX and ALTER INDEX statements within DB2 Version 9.1.
- You can add indexes on expressions on the Select Key Columns panel. For more information, see “Adding an expression to a What-If Index” on page 191.

When you finish entering key column information, press Enter to display the Update Index Access Path Statistics panel (Figure 68 on page 175).

Some of the values on this panel are gleaned from existing statistics.
Figure 68  Update Index Access Path Statistics panel (PSSPWIXS)

PSSPWIXS  Update Index Access Path Statistics
Command ===>

Specify statistics and press END to continue

Name ........ myindex

First Key Card  -1
Full Key Card  -1
Cluster Ratio  80 (0-100)
Leaf Pages ... 20
Number of Levels 2

/ Update first key column statistics

7  Update index and key column statistics:

A  (optional) On the Update Index Access Path Statistics panel, specify statistical values as needed.

B  Press Enter to display the Update First Key Column Statistics panel (Figure 69).

Figure 69  Update First Key Column Statistics panel (PSSPWED)

PSSPWED  UPDATE_FIRST_KEY_COLUMN_STATISTICS
Command ===>  Scroll ===> CSR

FREQUENCY VALUE - VARCHAR 15 -
****** ****************************************** Top of Data ******************************************
000001 0.500000000000 CCCCCCCCCCCCC
****** ****************************************** Bottom of Data ******************************************

C  Edit the default values as needed (Figure 70).

Figure 70  Edited first key column statistics

PSSPWED  UPDATE_FIRST_KEY_COLUMN_STATISTICS
Command ===>  Scroll ===> CSR

FREQUENCY VALUE - VARCHAR 15 -
****** ****************************************** Top of Data ******************************************
000001 0.500000000000 ALLISTER
000002 0.300000000000 BAKER
000003 0.200000000000 ELDON
****** ****************************************** Bottom of Data ******************************************

D  Press F3 until the Explain Results panel is displayed (Figure 71 on page 176).

Notice that the new index is displayed in the list. At this point, the index has not yet been created; it will be created when you perform the Explain.
Adding an index by using What-If Index

NOTE

You can make any number of changes before performing the Explain. The What-If Index process is cumulative, so incremental changes followed by Explains will enable you to see the effects of each change. When you perform a dynamic Explain on a clone, the DB2 optimizer uses the cloned objects as they exist at that point in time. The Index Component will compare each successive What-If Index Explain to the previous Explain.

Figure 71   Explain Results panel showing new index

NOTE

8 To perform a dynamic Explain, type XD in the space beside the clone and press Enter.

The results of the Explain are displayed (Figure 72). The WI01 label represents the Explain on the statement for the cloned objects. The asterisk and highlighting make it easier to visually identify the differences between the Explains.

Figure 72   Explain Results panel after new index is created
To show the cloned objects, type `S` in the space beside the clone and press `Enter`.

The cloned objects are displayed (Figure 73).

Figure 73  Display of cloned objects

After showing the cloned objects, you can make additional changes to the clone and perform additional What-If Index Explains. To see the effects of one or more changes, type an `XD` in the space beside the clone, and press `Enter`.

The results of your changes are displayed in the Explain Results panel. You can now perform additional What-If Index operations such as adding another index, dropping an index, updating statistics for an index, and so on.

Dropping an index by using What-If Index

This task describes how to use the What-If Index feature to simulate dropping an index.

Before you begin

Perform the following procedures:

- “Preparing to use What-If Index” on page 162
- “Identifying What-If Index candidates” on page 165
- “Adding an index by using What-If Index” on page 171
Dropping an index by using What-If Index

The examples in this section are continued from the task, “Adding an index by using What-If Index” on page 171.

To drop an index

1. On the Explain Results panel (Figure 74), type D in the space beside the index that you want to drop, and press Enter.

Figure 74 Specifying an index to be dropped in What-If Index

The index to be dropped from the clone is removed from the display (Figure 75 on page 179). At this point, the index has not yet been dropped even though it is no longer displayed on the panel; it will be dropped when you perform the Explain.

NOTE

You can make any number of changes before performing the Explain. The What-If Index process is cumulative, so incremental changes followed by Explains will enable you to see the effects of each change. When you perform a dynamic Explain on a clone, the DB2 optimizer uses the cloned objects as they exist at that point in time. The Index Component will compare each successive What-If Index Explain to the previous Explain.
To perform a dynamic Explain, type `XD` in the space beside the clone, and press Enter.

The results of the Explain are displayed (Figure 76). The `WI02` label represents the Explain on the cloned objects with new index `myindex` and having dropped index `XEMP2`.

You can now perform additional What-If Index operations such as adding another index, updating statistics for an index, and so on.
Updating statistics on an index by using What-If Index

This task explains how to update statistics for an index in your What-If Index analysis.

Before you begin

Perform the following procedures to create an index by using cloned objects:

- “Preparing to use What-If Index” on page 162
- “Identifying What-If Index candidates” on page 165
- “Adding an index by using What-If Index” on page 171

The examples in this section are continued from the previous task, “Dropping an index by using What-If Index” on page 177.

To update statistics

1. At the Explain Results panel (Figure 77), type U in the space beside the index and press Enter.

Figure 77 Specifying statistics to be updated for an index using What-If Index
The Update Index Access Path Statistics panel is displayed (Figure 78).

**Figure 78**  Update Index Access Path Statistics panel (PSSPWIXS)

<table>
<thead>
<tr>
<th>PSSPWIXS</th>
<th>Update Index Access Path Statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command</td>
<td>====&gt;</td>
</tr>
<tr>
<td></td>
<td>Specify statistics and press END to continue</td>
</tr>
<tr>
<td></td>
<td>Name . . . . . . myindex</td>
</tr>
<tr>
<td></td>
<td>First Key Card -1</td>
</tr>
<tr>
<td></td>
<td>Full Key Card -1</td>
</tr>
<tr>
<td></td>
<td>Cluster Ratio 80 (0-100)</td>
</tr>
<tr>
<td></td>
<td>Leaf Pages . . . 20</td>
</tr>
<tr>
<td></td>
<td>Number of Levels 2</td>
</tr>
</tbody>
</table>

_ Update first key column statistics_

2 Update the statistical values as needed (Figure 79).

**NOTE**

To update statistics for the first key column, select the Update first key column statistics option as shown in Figure 68 on page 175. You can modify this information on the Update First Key Column Statistics panel that is displayed (Figure 69 on page 175).

**Figure 79**  Update Index Access Path Statistics panel with updated statistics

<table>
<thead>
<tr>
<th>PSSPWIXS</th>
<th>Update Index Access Path Statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command</td>
<td>====&gt;</td>
</tr>
<tr>
<td></td>
<td>Specify statistics and press END to continue</td>
</tr>
<tr>
<td></td>
<td>Name . . . . . . myindex</td>
</tr>
<tr>
<td></td>
<td>First Key Card 100</td>
</tr>
<tr>
<td></td>
<td>Full Key Card 100</td>
</tr>
<tr>
<td></td>
<td>Cluster Ratio 90 (0-100)</td>
</tr>
<tr>
<td></td>
<td>Leaf Pages . . . 20</td>
</tr>
<tr>
<td></td>
<td>Number of Levels 2</td>
</tr>
</tbody>
</table>

_ Update first key column statistics_
3 Press F3 to display the Explain Results panel (Figure 80).

**Figure 80  Explain Results panel showing updated statistics**

<table>
<thead>
<tr>
<th>FILE</th>
<th>COMMANDS</th>
<th>OPTIONS</th>
<th>HELP</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSSPW200</td>
<td>Explain Results for SQLTEXT</td>
<td>Command ===&gt;</td>
<td>Scroll ===&gt; CSR</td>
</tr>
<tr>
<td>Actions:</td>
<td>S H R RS RW RI XD XS W P T C D U IM SA</td>
<td>More: &gt;</td>
<td></td>
</tr>
<tr>
<td>XD CLONE</td>
<td>Objects in database PSS60364</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STMTNO</td>
<td>SQL-STATEMENT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>117</td>
<td>SELECT * FROM PXB.EMP WHERE LASTNAME = 'BAKER'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CREATOR</td>
<td>TBNAME</td>
<td>CARD</td>
<td>NPAGES</td>
</tr>
<tr>
<td>PXB</td>
<td>EMP</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>CREATOR</td>
<td>IXNAME</td>
<td>1STKEY</td>
<td>FULLKEY</td>
</tr>
<tr>
<td>PXB</td>
<td>myindex</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>LBL</td>
<td>STMTNO</td>
<td>COST*RATE</td>
<td>SQL-STATEMENT</td>
</tr>
<tr>
<td>WI02</td>
<td>117</td>
<td>344.638184</td>
<td>SELECT * FROM PXB.EMP WHERE LASTNAME = 'BAKER'</td>
</tr>
<tr>
<td>WI01*</td>
<td>117</td>
<td>344.638184</td>
<td>SELECT * FROM PXB.EMP WHERE LASTNAME = 'BAKER'</td>
</tr>
<tr>
<td>XD01</td>
<td>117</td>
<td>1012.256592</td>
<td>SELECT * FROM PXB.EMP WHERE LASTNAME = 'BAKER'</td>
</tr>
<tr>
<td>COST*RATE</td>
<td>QB</td>
<td>PL</td>
<td>MIX</td>
</tr>
<tr>
<td>WI02</td>
<td>344.63818</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>WI01*</td>
<td>344.63818</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>XD01</td>
<td>1012.25656</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>******************************** Bottom of Data **********************************</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4 To perform a dynamic Explain on the cloned object, type XD in the space beside the clone, and press Enter.

The results of the Explain are displayed (Figure 81 on page 183). The WI03 label represents the Explain on the cloned objects after the following changes:

- index *myindex* added (see “Adding an index by using What-If Index” on page 171)
- index *XEMP2* dropped (see “Dropping an index by using What-If Index” on page 177)
- statistics updated on index *myindex* (see “Updating statistics on an index by using What-If Index” on page 180)
5 You can now perform additional What-If Index operations such as adding another index, dropping an index, and so on.

## Saving What-If Index DDL changes

This task explains how to save changes (except text changes) made on indexes are the session up to that point. These changes can include DDL for adding or dropping indexes, or SQL statements to update index statistics. Changes are saved to a PDS that you can import during a What-If Index operation to another statement that references the objects, or that you can use in an environment where you want the actual changes to take effect.

### NOTE

Because What-If Index changes are cumulative and are saved at a given point in time, BMC Software recommends that you save your DDL after each change is made. Some changes could have negative effects on the access path, but at incremental points will enable you to revert to an earlier version before such changes were made.

### Before you begin

Perform the following procedures to create an index by using cloned objects:

- “Preparing to use What-If Index” on page 162
- “Identifying What-If Index candidates” on page 165
- “Adding an index by using What-If Index” on page 171
The examples in this section are continued from the previous task, “Updating statistics on an index by using What-If Index” on page 180.

**To save What-If Index DDL changes**

1. On the Explain Results panel (Figure 82), type SA in the space beside the clone for which you want to save the DDL, and press Enter.

![Figure 82 Explain Results panel (PSSPW200)](image)

When you specify an SA command on a cloned object, all changes (except text changes) made on indexes are saved for the session up to that point.

The Save DDL panel is displayed (Figure 83).

![Figure 83 Save DDL panel (PSSPWD00)](image)
2 Specify a partitioned data set (including member) or sequential data set in which to save your DDL, specify Y (Yes) for the Generate DDL and Browse Data Set options, and press Enter.

The DDL is saved to the data set that you specified, and the saved DDL is opened in browse mode (Figure 84).

**NOTE**

In DDL that is saved from this process, DROP INDEX statements are prefixed with an “X” that must be removed before you can execute the statement.
Importing What-If Index changes to another statement for a table

This task explains how to import saved changes to another statement for a table.

**Before you begin**

Perform the following procedures to create an index by using cloned objects:

- “Preparing to use What-If Index” on page 162
- “Identifying What-If Index candidates” on page 165
- “Adding an index by using What-If Index” on page 171

You will also need previously saved DDL containing What-If Index changes for importing, as described in “Saving What-If Index DDL changes” on page 183.

The examples in this section are continued from the previous task, “Saving What-If Index DDL changes” on page 183.

**To import What-If Index changes**

1. On the Explain Results panel (Figure 85), type C in the space beside a statement, and press Enter to clone the objects for that statement.

**Figure 85 Explain Results panel (PSSPW200)**

![Figure 85 Explain Results panel (PSSPW200)](image-url)
The cloned objects are displayed (Figure 86).

Figure 86  Explain Results panel showing cloned objects

2 Type IM in the space beside the clone for which you want to import your saved DDL changes, and press Enter.

The Import DDL panel is displayed (Figure 87). When you specify an IM command, you can browse a data set of previously saved index changes and import and apply one of the changes to your current cloned objects. This enables you to test the index changes on different statements to evaluate their effects.

NOTE
You do not need to alter the “X” DROP INDEX statement to use it with the IM command. The index will not be included in the cloned objects.
3 Specify a partitioned data set (including member) or sequential data set in which you saved your DDL, specify Y (Yes) for the Browse Data Set and Import DDL options, and press Enter.

A list of the members in the specified data set is displayed (Figure 88).

4 To browse a data set, type S in the space beside the data set member, and press Enter.

The member that you selected is displayed (Figure 89).

5 Verify whether this is the member you want to import, then press F3.

The list of data set members is displayed again (Figure 88).
6 Repeat step 4 and step 5 until you locate the member you want to import, then press F3.

The Import DDL Member Selection List panel is displayed (Figure 90).

**Figure 90  Import DDL Member Selection List panel (PSSPMEML)**

7 To import the member, type S in the space beside that DDL member, then press Enter.

The panel displays a message when the import process is complete. (Figure 91)

**Figure 91  Message displayed when import process is complete**
8 Press F3 until the Explain Results panel is displayed (Figure 92).

Notice that index *myindex* is now displayed in the list with the statistics from the imported DDL.

**Figure 92  Explain Results panel (PSSPW200)**

---

9 To perform a dynamic Explain on the clone, type XD in the space beside the clone and press Enter.

You can now compare the What-If Explain (*WI01*) to the previous dynamic Explain (*XD01*). In this example (Figure 93), the imported changes had no effect on this statement.

**Figure 93  Results of dynamic Explain on imported DDL**

---

---
Adding an expression to a What-If Index

This procedure describes how to add an index on an expression in your What-If Index analysis. Adding an index to an expression is supported when using DB2 Version 9 and later.

**NOTE**
You can access the What-If Index from multiple points within the product. This example shows you accessing the What-If Index functions by accessing the Ad Hoc SQL option on the Explain Specification panel.

1. At the Explain Object Specification panel, select Ad Hoc SQL and press Enter.

   An editing panel (Figure 94) is displayed.

   **Figure 94**  Explain Edit panel

   ```
   EDIT       EXPLAIN
   Command ===>                                                       Scroll ===> CSR
   ****** ********************************* Top of Data***************************
   000001  SELECT  EMPNO
   000002  FROM    PSS.EMP
   000003  WHERE   UPPER(LASTNAME, 'EN_US') = 'JOE'
   000004    AND   UPPER(FIRSTNME, 'EN_US') = 'JOHN'
   ```

2. Enter the SQL statements that you want to use and press F3.

   The Explain or Execute Parameters panel (Figure 95) is displayed.

   **Figure 95**  Explain or Execute Parameters panel (PSSPA117)

   ```
PSSPA117 ----------------- Explain or Execute Parameters -------------------
Command ===>
Specify the options below and press ENTER to continue.
Option . . . 1  1. Explain
             2. Execute
             3. Edit
Qualifier Name SYSIBM
```
At the Explain or Execute Parameters panel, select **Explain** and press **Enter**.

The Explain Results panel (Figure 96) is displayed.

**Figure 96**  Explain Results panel showing an ad hoc query

To create a clone of the object upon which you want to perform the What-If analysis, type **C** in the space beside the object, and press **Enter**.

The Explain Results panel (Figure 97) shows the clone.

**Figure 97**  Explain Results panel showing the clone

To specify a What-If Index, type **W** in the space beside the table and press **Enter**.

The Specify Index Attributes panel is displayed (Figure 66 on page 173).

Specify attributes for a new index to be added to the table, select **Specify key columns**, and press **Enter**.

The Select Key Columns panel (Figure 67 on page 174) is displayed. If the table already had some indexed expressions, those expressions are displayed on the panel.
7 To add an expression to the key columns list, perform the following steps:

A In the SEQ field next to the column that you want to include in the expression, type E and press Enter.

B In the editing panel (Figure 98), enter the expression upon which you want to create an index.

Figure 98 Explain Edit panel showing an expression

C Press F3 to return to the Select Key Columns panel.

The Select Key Columns (Figure 99) displays the expression that you created.

Figure 99 Select Key Columns showing an expression

8 To select the key columns for the index, specify sequence numbers (SEQ) and order (ascending (A) or descending (D)).

NOTE
You can also specify random (R) next to a column in the Order field. Doing so defines a random key column in a similar way to using the RANDOM option with the CREATE INDEX and ALTER INDEX statements within DB2 Version 9.1.
9 Press F3 to return to the Explain Results panel.

The new index is displayed in the list. At this point, the index has not yet been created; it will be created when you perform the Explain.

10 To perform a dynamic Explain, type XD in the space beside the clone and press Enter.

The results of the Explain are displayed. The WI01 label represents the Explain on the statement for the cloned objects. The asterisk and highlighting make it easier to visually identify the differences between the Explains.

11 To show the cloned objects, type S in the space beside the cloned object and press Enter. Then, type S next to the index and press Enter.

The Explain Results panel (Figure 100) shows the detailed information.

**Figure 100  Explain Results panel showing an index on expression**
Explaining and Comparing dynamic SQL from trace data sets

This chapter presents the following topics:

Overview of the Explain and Compare functions ........................................ 195
Comparing dynamic SQL from different versions of DB2 .......................... 196
Comparing dynamic SQL from different periods of time .......................... 199

Overview of the Explain and Compare functions

The PSSEXTRC function of SQL Performance can help you to analyze dynamic SQL from APPTUNE archive trace data sets. Specifically, the PSSEXTRC function uses BMC IFCID 005 records. This function is useful when you plan to migrate to a different version of DB2 and want to find out how your existing data will perform, or when you want to Explain and Compare the access paths for dynamic SQL from subsystems running on different versions of DB2.

NOTE
This process can be used only in batch mode.

The Workload Access Path Compare Component provides an automated process to perform these functions. For more information, see Chapter 5, “Comparing access paths for workloads.”
Comparing dynamic SQL from different versions of DB2

In this task, you will Explain and Compare access paths for dynamic SQL from a DB2 Version 8 subsystem against the access path of that same SQL on a DB2 Version 9 subsystem.

1. Collect dynamic SQL data on a DB2 Version 8 subsystem by activating a filter with SQL TEXT DATA = Y or D specified.

   **TIP**
   To reduce the amount of data in your Data Collector trace data sets, capture SQL from only your intended version 8 subsystems and turn off data collection for all of your other DB2 subsystems whenever possible.

2. Execute the dynamic SQL on the DB2 Version 8 subsystem.

3. Issue the SWITCH command for the output group containing the APSTMT class (or specify ALL output groups). Examine the resulting archive task output to determine the name of the archive created. This archive will be input to the PSSEXTRJ jobs.

   For detailed instructions, see the APPTUNE for DB2 User Guide.

4. Run PSSEXTRJ on the DB2 Version 8 subsystem, using the archived trace data set created in step 3.

   PSSEXTRJ performs a dynamic Explain on all dynamic SQL that is captured from the DB2 Version 8 subsystem. The job requires you to enter the input parameters shown in Table 14.

<table>
<thead>
<tr>
<th>Input parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSIDEX</td>
<td>subsystem where the dynamic SQL will be executed</td>
</tr>
<tr>
<td>SSID</td>
<td>subsystem where Explain will be performed</td>
</tr>
<tr>
<td>CREATOR(^a)</td>
<td>value that you create to identify the dynamic SQL to be Explained</td>
</tr>
<tr>
<td>PLAN</td>
<td>product plan; for example, DAAPRDD1</td>
</tr>
<tr>
<td>RETAIN</td>
<td>number of history records to be retained for the package or DBRM</td>
</tr>
</tbody>
</table>

\(^a\) You must specify the same value for CREATOR for PSSEXTRJ in step 4 and step 7 of this task, and for PSSCOMPJ in step 8.
If PSSEXTRJ finds no dynamic SQL that matches the given DB2 subsystem, the job will issue an error report (Figure 101). Verify the value of the SSIDEX parameter and the contents of your archived trace data set and resubmit the job.

**Figure 101  Example of a PSSEXTRJ error report**

| LSCX503 **** WARNING **** ERRNO = ENFOUND |
| Generated in READ_SMF(MAIN), offset 000092 |
| File never created, open failed. |
| Interrupted while: Opening file "IN" |

5 Run PSSHUNLD to unload the requested Explain records, which were created in PSSEXTRJ from the DB2 Version 8 history tables.

Before you submit PSSHUNLD, update member PSSHUNL1 to unload records from the appropriate history tables and where OBJOWNR is the same value that you entered for CREATOR in step 4.

**NOTE**
The value that you enter for OBJOWNR must be identical to the value you enter for CREATOR in the PSSEXTRJ and PSSCOMPJ jobs.

6 Run PSSHLOAD to load the DB2 Version 8 history records into the appropriate DB2 Version 9 history tables. This is required because Compare can read from only one set of history tables.

Before you submit PSSHLOAD, update member PSSHLOD1 to load the DB2 Version 8 history records into the appropriate DB2 Version 9 history tables.

7 Run PSSEXTRJ on the DB2 Version 9 subsystem, using the same archived trace data set that was used when you ran the job on the DB2 Version 8 subsystem in step 4.

PSSEXTRJ performs a dynamic Explain on all dynamic SQL that was captured from the DB2 Version 8 subsystem. The job requires you to enter the input parameters shown in Table 15.

<p>| Table 15  PSSEXTRJ input parameters for DB2 Version 9 subsystem (part 1 of 2) |</p>
<table>
<thead>
<tr>
<th>Input parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSIDEX</td>
<td>subsystem where dynamic SQL will be executed</td>
</tr>
<tr>
<td>SSID</td>
<td>subsystem where Explain will be performed</td>
</tr>
<tr>
<td>CREATOR&lt;sup&gt;a&lt;/sup&gt;</td>
<td>value that you create to identify the dynamic SQL to be Explained</td>
</tr>
<tr>
<td></td>
<td>This value can be 1 to 8 characters in length.</td>
</tr>
</tbody>
</table>
Run PSSCOMPJ on the DB2 Version 9 subsystem to Compare the loaded DB2 Version 8 history records to the DB2 Version 9 history records. The job requires you to enter the input parameters shown in Table 16.

Table 16  PSSCOMPJ input parameters for DB2 Version 9 subsystem

<table>
<thead>
<tr>
<th>Input parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPNEW</td>
<td>version of an Explain history (in this case, the Explain from the DB2 Version 9 subsystem) to be compared to an older history</td>
</tr>
<tr>
<td>COMPOLD</td>
<td>version of an Explain history (in this case, the Explain from the DB2 Version 8 subsystem) to be compared to a newer history</td>
</tr>
<tr>
<td>CREATOR&lt;sup&gt;a&lt;/sup&gt;</td>
<td>value that you create to identify the dynamic SQL to be Explained</td>
</tr>
<tr>
<td>OBJECT&lt;sup&gt;b&lt;/sup&gt;</td>
<td>type of object to be Compared: DBRM or package containing the dynamic SQL</td>
</tr>
<tr>
<td>NAME</td>
<td>object name of first DBRM or package</td>
</tr>
<tr>
<td>COLLID</td>
<td>collection ID of object</td>
</tr>
<tr>
<td>VERSION</td>
<td>version of object</td>
</tr>
</tbody>
</table>

<sup>a</sup> You must specify the same value for CREATOR for PSSEXTRJ in step 4 and step 7 of this task, and for PSSCOMPJ in step 8.

<sup>b</sup> BMC Software recommends that you specify % (percent sign) for this parameter, in order to process all dynamic SQL that was Explained from the trace data. This is because trace data sets usually have a large number of packages and DBRMs that you would need to process.

---

Table 15  PSSEXTRJ input parameters for DB2 Version 9 subsystem (part 2 of 2)

<table>
<thead>
<tr>
<th>Input parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLAN</td>
<td>product plan; for example, DAAPRDD1</td>
</tr>
<tr>
<td>RETAIN</td>
<td>number of history records to be retained for the package or DBRM</td>
</tr>
</tbody>
</table>

<sup>a</sup> You must specify the same value for CREATOR for PSSEXTRJ in step 4 and step 7 of this task, and for PSSCOMPJ in step 8.
Comparing dynamic SQL from different periods of time

In this task, you will Explain and Compare access paths for dynamic SQL that was collected for different periods of time from a single subsystem of your choice.

1. Collect dynamic SQL data on a DB2 Version 8 subsystem by activating a filter with SQL TEXT DATA = Y or D specified.

   **TIP**
   To reduce the amount of data in your Data Collector trace data sets, capture SQL from only your intended current subsystems and turn off data collection for all of your other DB2 subsystems whenever possible.

2. Execute the dynamic SQL on the subsystem of your choice.

3. Issue the SWITCH command for the output group containing the APSTMT class (or specify ALL output groups). Examine the resulting archive task output to determine the name of the archive created. This archive will be input to the PSSEXTRJ jobs.

   For detailed instructions, see the *APPTUNE for DB2 User Guide*.

4. Run PSSEXTRJ on the subsystem, using the archived trace data set that was created in step 3.

   PSSEXTRJ performs a dynamic Explain on all dynamic SQL that is captured from the subsystem you specified. The job requires you to enter the input parameters shown in Table 17.

<table>
<thead>
<tr>
<th>Table 17</th>
<th>PSSEXTRJ input parameters  (part 1 of 2)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input parameter</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>SSIDEX</td>
<td>subsystem where dynamic SQL will be run</td>
</tr>
<tr>
<td>SSID</td>
<td>subsystem where Explain will be performed</td>
</tr>
<tr>
<td>CREATORa</td>
<td>value that you create to identify the dynamic SQL to be Explained</td>
</tr>
<tr>
<td></td>
<td>This value can be 1 to 8 characters in length.</td>
</tr>
</tbody>
</table>
Comparing dynamic SQL from different periods of time

200 SQL Performance for DB2 User Guide

If PSSEXTRJ finds no dynamic SQL that matches the given DB2 subsystem, the job will issue an error report (Figure 102). Verify the value of the SSIDEX parameter and the contents of your archived trace data set and resubmit the job.

After a period of time, statistics on the object are likely to have changed. Because of this, you might want to run the Explain again on the same trace data set, in order to Compare the access paths.

Run PSSEXTRJ on the specified subsystem, using the same archived trace data set that was used when you ran the job in step 4.

PSSEXTRJ performs a dynamic Explain on all dynamic SQL that was captured from the subsystem you specified. The job requires you to enter the input parameters shown in Table 18.

<table>
<thead>
<tr>
<th>Input parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSIDEX</td>
<td>subsystem where dynamic SQL will be executed</td>
</tr>
<tr>
<td>SSID</td>
<td>subsystem where Explain will be performed</td>
</tr>
<tr>
<td>CREATOR(^a)</td>
<td>value that you create to identify the dynamic SQL to be Explained</td>
</tr>
<tr>
<td>PLAN</td>
<td>product plan; for example, DAAPRDD1</td>
</tr>
<tr>
<td>RETAIN</td>
<td>number of history records to be retained for the package or DBRM</td>
</tr>
</tbody>
</table>

\(^a\) You must specify the same value for CREATOR for PSSEXTRJ in step 4 and step 6 of this task, and for PSSCOMPJ in step 7.

---

Table 17 PSSEXTRJ input parameters (part 2 of 2)

<table>
<thead>
<tr>
<th>Input parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLAN</td>
<td>product plan; for example, DAAPRDD1</td>
</tr>
<tr>
<td>RETAIN</td>
<td>number of history records to be retained for the package or DBRM</td>
</tr>
</tbody>
</table>

\(^a\) You must specify the same value for CREATOR for PSSEXTRJ in step 4 and step 6 of this task, and for PSSCOMPJ in step 7.

---

Figure 102 Example of a PSSEXTRJ error report

LSCX503 **** WARNING **** ERRNO = ENFOUND
Generated in FOPEN called from line 738 of READ_SMF(MAIN) , offset 000092
File never created, open failed.
Interrupted while: Opening file "IN"
Once the product history tables contain at least two entries, you can compare the access paths from those periods of time. Run PSSCOMPJ on the specified subsystem. The job requires you to enter the input parameters shown in Table 19.

Table 19  PSSCOMPJ input parameters

<table>
<thead>
<tr>
<th>Input parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| COMPNEW         | version of an Explain history (in this case, the most recent Explain) to be compared to a later history  
This value is typically set to 0. |
| COMPOLD         | version of an Explain history (in this case, the earlier Explain) to be compared to an earlier history  
This value is typically set to -1. |
| CREATOR<sup>a</sup> | value that you create to identify the dynamic SQL to be Explained  
This value can be 1 to 8 characters in length. |
| OBJECT<sup>b</sup> | type of object to be Compared: DBRM or package containing the dynamic SQL |
| NAME            | object name of first DBRM or package  
BMC Software recommends that you specify % (percent sign) for this parameter, in order to process all dynamic SQL that was Explained from the trace data. This is because trace data sets usually have a large number of packages and DBRMs that you would need to process. |
| COLLID          | collection ID of object  
BMC Software recommends that you specify % (percent sign) for this parameter, in order to process all dynamic SQL that was Explained from the trace data. |
| VERSION         | version of object  
BMC Software recommends that you specify % (percent sign) for this parameter, in order to process all dynamic SQL that was Explained from the trace data. |

<sup>a</sup> You must specify the same value for CREATOR for PSSEXTRJ in step 4 and step 6 of this task, and for PSSCOMPJ in step 7.

<sup>b</sup> If your dynamic SQL comes from both packages and DBRMs, you must run PSSCOMPJ twice: Once with OBJECT=DBRM and a second time with OBJECT=PACKAGE.
Performance Advisor reference

Several utilities, tables, and samples are shipped with the Performance Advisor component of SQL Performance for DB2. The samples reside in the LLQSAMP library (SAMP library), where LLQ is DB, XX, BB, and UBB.

For more information about setting up and using Performance Advisor, see Chapter 3, “Managing performance with Performance Advisor.”

This appendix contains the following topics:

Performance Advisor utilities ............................................. 203
  DMDBMERG ................................................................. 204
  DMDBRTSU ................................................................. 215
Performance Advisor tables .............................................. 218
Sample members .............................................................. 220

Performance Advisor utilities

The Performance Advisor component contains the following utilities:

- DMDBMERG allows you to maintain Performance Advisor tables that are based on APPTUNE data.

- DMDBRTSU generates UNLOAD and LOAD utility control statements to maintain the Performance Advisor real-time stats (RTS) tables.
The DMDBMERG utility allows you to maintain Performance Advisor tables that are based on APPTUNE data. DMDBMERG generates control statements for input into a LOAD utility and generates load records representing data for one or more Performance Advisor tables.

The required input to DMDBMERG consists of archive files containing APPTUNE trace data. These files are produced by the Data Collector when an active trace data set is full or switched. By default the archive files are located and dynamically allocated by DMDBMERG based on the interval and Performance Advisor tables requested. You can override this default setting by specifying a TRACEIN DD card. If this is the only input to the utility, the resulting LOAD control statements cause the generated load records to be added to the table(s) as new rows.

For some tables, you must update existing rows. In these cases, DMDBMERG also requires as input a LOADIN file containing load records generated by the previous DMDBMERG execution for the same table(s). When LOADIN is present, the resulting LOAD control statements cause the referenced table(s) to be entirely replaced by the generated load records.

Typically, a job that executes DMDBMERG in one step would then execute a DB2 LOAD utility such as DSNUTILB or BMC LOAD PLUS for DB2 as the next step to maintain Performance Advisor tables.

Figure 103 shows sample JCL and instructions to run the DMDBMERG utility.

---

Figure 103  DMDBMERG JCL (part 1 of 4)

```
//DMDBMERG     JOB  (ACCT),'GENERATE PA TABLES',  <== MODIY
//             MSGCLASS=X,                        <== CHECK
//             CLASS=A                            <== CHECK
//*
//*
//*  DMDBMERG -- GENERATE DATA FOR PERFORMANCE ADVISOR TABLES
//*
//*  DMDBMERG PRODUCES LOAD CONTROL STATEMENTS AND LOAD RECORDS, WHICH
//*  SERVE AS INPUT TO A LOAD UTILITY TO POPULATE AND MAINTAIN
//*  PERFORMANCE ADVISOR TABLES.
//*
//*  THE SYNTAX OF THE CONTROL STATEMENTS AND THE USE OF OPTIONAL
//*  DD STATEMENTS ARE ALSO DOCUMENTED IN THE SQL PERFORMANCE USER
//*  GUIDE.
//*
//*  NOTE - THE DATE EXAMPLES ARE IN USA FORMAT. MODIFY MM/DD/YY
//*  TO MATCH THE FORMAT SPECIFIED IN THE GLOBAL OPTIONS PANEL.
//*
//*
```
Figure 103  DMDBMERG JCL (part 2 of 4)

```plaintext
/*
// TABLES    EXEC PGM=DMDBMERG
// STEPLIB    DD DISP=SHR,DSN=?BMC-HLQ?.BBLINK
// SYSPRINT   DD SYSOUT=*               ** MESSAGES               **
// TRACEIN    DD DISP=SHR,DSN=?TRACEDSN? ** OPTIONAL SPECIFICATION  **
//           ** OF INPUT ARCHIVE TRACE  **
//           ** DATA. BY DEFAULT THE  **
//           ** DATA IS DYNAMICALLY    **
//           ** LOCATED.              **
// LOADIN     DD DISP=SHR,DSN=?LOADDSN?  ** OPTIONAL SPECIFICATION  **
//           ** OF INPUT LOAD RECORDS **
//           ** PRODUCED AS DB2LOAD ON **
//           ** PREVIOUS RUN OF DMDBMERG.**
//           ** IF LOADIN IS SPECIFIED, **
//           ** DATA FROM THESE RECORDS **
//           ** IS MERGED WITH ARCHIVE  **
//           ** INPUT AND THE RESULT IS **
//           ** USED TO REPLACE THE TABLE.**
//           ** OTHERWISE DATA FROM     **
//           ** ARCHIVE INPUT IS ADDED TO **
//           ** THE TABLE.              **
// TRACEWRK   DD DISP=(NEW,PASS),        ** STAGED SORT OUTPUT. IF **
//           UNIT=WORK,               ** DD DUMMY, TABLE OUTPUT **
//           SPACE=(CYL,(5,5),RLSE),  ** WILL NOT BE PRODUCED **
//           DCB=(RECFM=VBS,LRECL=32760,BLKSIZE=8192)
// DB2LOAD    DD DISP=(NEW,PASS).       ** DB2 LOAD-FORMAT DATA TO BE **
//           UNIT=WORK,DSN=&LOAD,     ** PASSED AS INPUT TO A DB2  **
//           SPACE=(CYL,(5,5),RLSE). ** LOAD UTILITY               **
//           DCB=(RECFM=VB,LRECL=32752,BKSIZE=32756)
// LOADCTL    DD DISP=(NEW,PASS).       ** DB2 LOAD UTILITY CONTROL  **
//           UNIT=WORK,DSN=&CNTL.    ** STATEMENTS                   **
//           SPACE=(CYL,(5,5),RLSE).**
//           DCB=(RECFM=FB,LRECL=80,BKSIZE=3120)
// SORTWK01   DD DISP=(NEW,PASS),UNIT=WORK,SPACE=(CYL,(10,10),RLSE)
// SORTWK02   DD DISP=(NEW,PASS),UNIT=WORK,SPACE=(CYL,(10,10),RLSE)
// SORTWK03   DD DISP=(NEW,PASS),UNIT=WORK,SPACE=(CYL,(10,10),RLSE)
// SORTWK04   DD DISP=(NEW,PASS),UNIT=WORK,SPACE=(CYL,(10,10),RLSE)
// SYSOUT     DD SYSOUT=*               ** SORT MESSAGES           **
```

Appendix A  Performance Advisor reference  205
Figure 103  DMDBMERG JCL (part 3 of 4)

```bash
//* OPTIONAL DDNAME SRCTRACE TO GET DIAGNOSTIC INFORMATION ABOUT
//* INPUT DATA
//*
//* SRCTRACE DD SYSOUT=*
//*
//* OPTIONAL DDNAMES FOR VIRTUAL FILES TO OVERRIDE INSTALLATION
//* AND USER OPTIONS. ADJUST R0001VFL (SELECT FILE) BASED ON
//* NUMBER OF INPUT RECORDS, R0001VFG (GROUP FILE) BASED ON
//* NUMBER OF INPUT RECORDS AND DEGREE OF RECORD REDUCTION.
//*
//* R0001VFL DD DISP=(NEW,PASS),UNIT=WORK,SPACE=(CYL,(10,10),RLSE)
//* R0001VFG DD DISP=(NEW,PASS),UNIT=WORK,SPACE=(CYL,(10,10),RLSE)
//*
//*-----------------------------------------------------------------*
//* SUPPORTED CONTROL STATEMENTS:
//*
//* INTERVAL(                   - GLOBAL (DEFAULT) INTERVAL
//*   START(MM/DD/YY,HH:MM:SS)  - INTERVAL STARTING DATE/TIME
//*     END(MM/DD/YY,HH:MM:SS)) - INTERVAL ENDING DATE/TIME
//* -OR-
//* INTERVAL(                   - GLOBAL (DEFAULT) INTERVAL
//*   START(MM/DD/YY,HH:MM:SS)  - INTERVAL STARTING DATE/TIME
//*     FOR(    DAYS,HH:MM:SS)) - INTERVAL DURATION DAYS/TIME
//* DATE/TIME CAN BE RELATIVE (E.G. -1 FOR YESTERDAY)
//* INTERVAL( START(-1) FOR(1) ) SPECIFIES ALL DAY YESTERDAY
//*-----------------------------------------------------------------
//* APPPROF(PROFILE.OWNER)      - APPLICATION PROFILE USED WHEN
//*                               GROUP REPORTING IS SPECIFIED
//*-----------------------------------------------------------------
//* QUALIFIER(     - GLOBAL (DEFAULT) QUALIFIERS FOR restrictING
//*                DB2(XXXX)  - " BY DB2 SSID NAMED XXXX
//*              ,PLAN(    )  - " BY PLAN
//*           ,PROGRAM(    )  - " BY DBRM/PACKAGE ID
//*              ,CONN(    )  - " BY CONNECT-IDS
//*              ,OPER(    )  - " BY OPER ID
//*           ,COLL(    )  - " BY COLLECTION ID
//*            ,APGRP(   )  - " BY APPLICATION PROFILE-APPTUNE ONLY
//*        )             - END OF QUALIFIER KEYWORDS
//*-----------------------------------------------------------------
//* PURGE_DAYS(DAYS)    - APPLIES TO LOADIN DATA,
//*                       PURGE IF OLDER THAN N DAYS
//*-----------------------------------------------------------------
//* FIRST_DAY(SUNDAY)   - SPECIFIES THE FIRST DAY OF THE WEEK FOR
//*                       WEEKLY TABLES--SPECIFY SUNDAY, MONDAY,
//*                       TUESDAY, WEDNESDAY, THURSDAY, FRIDAY,
//*                       OR SATURDAY: DEFAULT IS SUNDAY
//*-----------------------------------------------------------------
//* TBCREATOR(XXXXXXXX) - SPECIFIES THE TABLE OWNER FOR GENERATED
//*                       LOAD UTILITY CONTROL STATEMENTS;
//*                       DEFAULT IS BMCSFTWR
```
Figure 103  DMDBMERG JCL (part 4 of 4)

```bash
// *-------------------------------------------------------------------*
// * CCSID(NNNN)         - SPECIFIES THE EBCDIC CHARACTER SET TO BE *
// * USED FOR CONVERSION FROM/TO UNICODE;                             *
// * DEFAULT IS 0037                                                   *
// *-------------------------------------------------------------------*
// * TABLE(NAME(XXXXXXXX,YYYYYYYY,...))   - SPECIFIES TABLE(S) FOR WHICH *
// * DATA WILL BE GENERATED                                           *
// *-------------------------------------------------------------------*

/** SUPPORTED TABLES: **
/** OBJ_STATISTICS          OBJECT STATISTICS BY INTERVAL
/** STMT_STATISTICS          STATEMENT STATISTICS BY INTERVAL
/** STMT_STATISTICS_OB       OBJ STATS PER STATEMENT BY INTVL
/** WKLD_STATISTICS          WORKLOAD STATISTICS BY INTERVAL
/** DAILY_OBJ_STATISTICS     OBJECT STATISTICS BY DAY
/** DAILY_STMT_STATISTICS    STATEMENT STATISTICS BY DAY
/** DAILY_WKLD_STATISTICS    WORKLOAD STATISTICS BY DAY
/** WEEKLY_OBJ_STATISTICS    OBJECT STATISTICS BY WEEK
/** WEEKLY_STMT_STATISTICS   STATEMENT STATISTICS BY WEEK
/** WEEKLY_WKLD_STATISTICS   WORKLOAD STATISTICS BY WEEK
/** MONTHLY_OBJ_STATISTICS   OBJECT STATISTICS BY MONTH
/** MONTHLY_STMT_STATISTICS  STATEMENT STATISTICS BY MONTH
/** MONTHLY_WKLD_STATISTICS  WORKLOAD STATISTICS BY MONTH
/** BASELINE_OBJ_STATISTICS  OBJECT STATISTICS BASELINE
/** BASELINE_STMT_STATISTICS STATEMENT STATISTICS BASELINE
/** BASELINE_WKLD_STATISTICS WORKLOAD STATISTICS BASELINE
/** STMT_TEXT                SQL TEXT
/** STMT_ERRORS              STATEMENT ERRORS
/** STMT_EXCEPTIONS          STATEMENT EXCEPTIONS
/** STMT_EXCEPTIONS_OB       OBJECT STATISTICS FROM EXCEPTIONS
/** INDEX_USAGE              INDEX ACCESS TIMES
/**

//SYSIN DD *
INTERVAL( START(-1) FOR(1) )
PURGE_DAYS(365)
TBCREATOR(BMCSFTWR)
TABLE (NAME( OBJ_STATISTICS, -
                      STMT_STATISTICS, -
                      WKLD_STATISTICS ) )

```
Table 20 describes the data sets that are processed by DMDBMERG.

### Table 20  DMDBMERG data sets (part 1 of 2)

<table>
<thead>
<tr>
<th>DDNAME</th>
<th>Description</th>
<th>DCB Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2LOAD</td>
<td>DB2 load format data will be written to this data set, which can be passed to a load utility in another step to load the data.</td>
<td>RECFM=VB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LRECL = 32752</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BLKSIZE = 32756</td>
</tr>
<tr>
<td>R0001VFG</td>
<td>(optional) data set used to hold group records for each table if there are too many records to fit in the internal buffers</td>
<td>not applicable</td>
</tr>
<tr>
<td></td>
<td>This value can be adjusted according to the number of selected records, and the degree to which the data is reduced. For example, a table summarizing by hour requires fewer group records than a table summarizing by day. If not specified, the file is dynamically allocated by using the space allocation that is specified in your User Profile and the unit that is specified in your installation options. Default values are generally sufficient, but can be customized in the Data Collector administration panels. To size the data set: 1. Multiply 32 KB (maximum size of each record) with the number of records to calculate a total size of the input records. 2. Adjust this figure to the track size and approximate sizes of the initial allocation.</td>
<td></td>
</tr>
<tr>
<td>R0001VFL</td>
<td>(optional) data set used to hold selected records for each table if there are too many records to fit in the internal buffers</td>
<td>not applicable</td>
</tr>
<tr>
<td></td>
<td>This value can be adjusted according to the number of selected records. If not specified, the file is dynamically allocated by using the space allocation that is specified in your User Profile and the unit that is specified in your installation options. Default values are generally sufficient, but can be customized in the Data Collector administration panels. To size the data set: 1. Multiply 32 KB (maximum size of each record) with the number of records to calculate a total size of the input records. 2. Adjust this figure to the track size and approximate sizes of the initial allocation.</td>
<td></td>
</tr>
<tr>
<td>SORTWKnn</td>
<td>(optional) interim data sets for the Sort utility working storage</td>
<td>not applicable</td>
</tr>
<tr>
<td></td>
<td>This data set might be required by a sort program if large amounts of trace data are being processed. Refer to your site’s sort program documentation for details about the specification of this DD statement.</td>
<td></td>
</tr>
</tbody>
</table>
### Table 20  DMDBMERG data sets (part 2 of 2)

<table>
<thead>
<tr>
<th>DDNAME</th>
<th>Description</th>
<th>DCB Attributes</th>
</tr>
</thead>
</table>
| SRCTRACE    | optional data set containing a listing of all records found in the input archive files | - RECFM = FBA  
- LRECL = 121  
- BLKSIZE = any multiple of LRECL |
|             | A single line is generated for each valid record, including an indication of whether the record is used to generate table data. |                                                      |
| STEPLIB     | Data Collector load library                                                  | not applicable                                      |
| SYSIN       | DMDBMERG control statements                                                  | not applicable                                      |
| SYSPRINT    | SORT messages                                                               | not applicable                                      |
| SYSOUT      | DMDBMERG messages and statistics                                            |                                                      |
| LOADCTL     | Output LOAD utility control statements for input as SYSIN to a subsequent LOAD utility step. | - RECFM = FB  
- LRECL = 80  
- BLKSIZE = any multiple of LRECL |
| TRACEIN     | (optional) trace input data sets to override automatic archive selection    | not applicable                                      |
|             | The input data sets can be any of the following data sets: archived data sets or EXPORT data sets. |                                                      |
|             | You must either include the TRACEIN data set or specify the INTERVAL control statement. |                                                      |
| TRACEWRK    | interim data set for holding the records that were selected and sorted for table data | - RECFM = VBS  
- LRECL = 32760  
- BLKSIZE = 8192 |
|             | The amount of required space increases as the amount of selected data increases. Data is selected if any table requires the record, based on IFCID type, date and time, and qualifiers. |                                                      |
The control statements listed in Table 21 can be used with the DMDBMERG utility.

**Table 21  DMDBMERG SYSIN control statements**

<table>
<thead>
<tr>
<th>Control Statement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTERVAL</td>
<td>Use this control statement to specify a time interval for the data to be included in tables. For more information, see Table 22. You must either include the TRACEIN data set or specify the INTERVAL control statement.</td>
</tr>
<tr>
<td>APPPROF (profile.owner)</td>
<td>Use this control statement to specify the application profile for group reporting.</td>
</tr>
<tr>
<td>QUALIFIER (type)</td>
<td>Use this control statement to use qualifier values to restrict the data that is selected for tables.</td>
</tr>
<tr>
<td>PURGE_DAYS (nn)</td>
<td>Use this control statement to purge data older than the specified number of days.</td>
</tr>
<tr>
<td>FIRST_DAY (Sunday)</td>
<td>Use this control statement to specify the first day of the week for weekly tables.</td>
</tr>
<tr>
<td>TBCREATOR (name)</td>
<td>Use this control statement to specify the table owner for generated load utility control statements.</td>
</tr>
<tr>
<td>CCSID (mmnn)</td>
<td>Use this control statement to specify the EBCDIC character set to be used for conversion to or from unicode.</td>
</tr>
<tr>
<td>TABLE</td>
<td>Use this control statement to produce tables.</td>
</tr>
</tbody>
</table>

**INTERVAL statement**

The INTERVAL statement specifies the time interval that applies to the statistics gathered by DMDBMERG for the PMDB. Table 22 on page 211 describes the keywords and options for each keyword.
Table 22 INTERVAL statement and parameter keywords (part 1 of 3)

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Value</th>
<th>Description of value</th>
</tr>
</thead>
<tbody>
<tr>
<td>START</td>
<td>specifies the beginning date and time for the data to be included</td>
<td>date</td>
<td>beginning date of the interval</td>
</tr>
<tr>
<td>(optional)</td>
<td>This keyword must be specified if FOR or END is specified. The START</td>
<td></td>
<td>When specified without a time, the time defaults to midnight (12:00 A.M.) on the specified date (midnight = beginning of day).</td>
</tr>
<tr>
<td></td>
<td>date and time must be prior to the date and time that the job is run.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>If no START keyword is specified the start time defaults to the timestamp</td>
<td>-nn</td>
<td>number of days prior to the current date to be used for the beginning date</td>
</tr>
<tr>
<td></td>
<td>of the first record in the TRACEIN data set.</td>
<td></td>
<td>Valid values include any number from -0 through -60. For example, if -5 is specified on 03/06/2005 (USA format), the start date will be 03/01/2005. Use -0 to indicate the current date.</td>
</tr>
<tr>
<td></td>
<td>If there are multiple DB2s being reported that do not all share the same</td>
<td>time</td>
<td>beginning time for the data to be included</td>
</tr>
<tr>
<td></td>
<td>collection interval, the start time for each DB2 can be different.</td>
<td></td>
<td>Valid values are any time in the range 00:00:00–23:59:59 or 00:00–23:59.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td></td>
<td>When specified without a date, the date defaults to the first date upon which that time occurs in the archive data set. For example, if a time of 8:00:00 is specified and the timestamp of the first record in the data set is 01/01/2005 10:00:00 (USA format), the date defaults to 01/02/2005.</td>
</tr>
<tr>
<td></td>
<td>There are two DB2s being reported:</td>
<td>date,time</td>
<td>When both a date and time are specified, the selected data begins with the first record having a timestamp that is equal to or greater than the specified date and time.</td>
</tr>
<tr>
<td></td>
<td>■ DB2A has a collection interval of 24 hours, starting at midnight</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ DB2B has a collection interval of 8 hours (intervals starting at midnight</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>8:00 A.M. and 4:00 P.M.)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>The report is activated at 10:00 A.M. The interval for DB2A starting at</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>midnight is active, so midnight is the start time for DB2A. The interval</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>for DB2B starting at 8:00 A.M. is active, so 8:00 A.M. is the start time</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>for DB2B.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Note: Records are cut at the SQL statistics collection interval that is</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>specified in the DOMPLEX option set. If you specify START and END</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>times that span intervals, you might not get all of the data that you</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>expect. To avoid this, run the Select Analysis Interval report against the</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>data set to see the actual times to use.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Valid abbreviation: S</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
END (optional) specifies the ending date and time for the data to be included.

END and FOR are two different ways of specifying the ending time of reports. If both keywords are specified, END is ignored.

If no END or FOR keyword is specified the end time defaults to the timestamp of the last record in the TRACEIN data set.

The END date and time should be prior to the date and time that the job is run. If both the START and END dates and times are later than the date and time that the job is run, an error results, and no report is produced. If only the END time is later than the date and time that the job is run, the report is produced, but the date-time combination is flagged with a dollar sign ($) in the text of the control statements and the BMC24189 warning message is produced, stating that the end time is later than the run time.

Note: APPTUNE cuts records at the APPTUNE SQL statistics collection interval that is specified on the DB2 Definition panel of the DOMPLEX option set. If you specify START and END times that span intervals, you might not get all of the data that you expect. To avoid this, run the Select Analysis Interval report against the data set to see the actual times to use.

Valid abbreviation: E

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Value</th>
<th>Description of value</th>
</tr>
</thead>
<tbody>
<tr>
<td>date</td>
<td>ending date of the interval</td>
<td>-nn</td>
<td>number of days prior to the current date to be used for the ending date</td>
</tr>
<tr>
<td>time</td>
<td>ending time for the data to be included</td>
<td>-nn,time</td>
<td>When both a number of days and a time are specified, the date represented by -nn is calculated and the selected data ends with the last record having a timestamp equal to or less than the calculated date and specified time.</td>
</tr>
<tr>
<td>date,time</td>
<td>When both a date and time are specified, the selected data ends with the last records in the archive data set having a timestamp equal to or less than the specified date and time.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>for</td>
<td>two different ways of specifying the ending time of reports. If both keywords are specified, END is ignored.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The QUALIFIER statement uses qualifier values to filter the data to be included in the PADB. Table 23 describes the keywords for the QUALIFIER statement.

### Table 23 QUALIFIER statement and parameter keywords (part 1 of 3)

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Value</th>
<th>Description of value</th>
</tr>
</thead>
<tbody>
<tr>
<td>INCLUDE / EXCLUDE</td>
<td>filters the records that are selected</td>
<td>INCLUDE</td>
<td>includes only records with the qualifier values that are specified in this statement</td>
</tr>
<tr>
<td>(optional)</td>
<td>You can specify either INCLUDE or EXCLUDE for each qualifier type, but you cannot specify both of them. INCLUDE is the default.</td>
<td>EXCLUDE</td>
<td>excludes records containing the qualifier values that are specified in this statement</td>
</tr>
</tbody>
</table>

### QUALIFIER statement

The QUALIFIER statement uses qualifier values to filter the data to be included in the PADB. Table 23 describes the keywords for the QUALIFIER statement.
### Table 23 QUALIFIER statement and parameter keywords (part 2 of 3)

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Value</th>
<th>Description of value</th>
</tr>
</thead>
<tbody>
<tr>
<td>type value</td>
<td>type of qualifiers to be included or excluded</td>
<td>PLAN</td>
<td>plan name</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Valid abbreviation: P</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DB2^a</td>
<td>DB2 SSID</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Valid abbreviation: U</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CORRELATION</td>
<td>Correlation ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Valid abbreviation: CORR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CONNECTION</td>
<td>Connection ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Valid abbreviations: CONN, N</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OPERATOR</td>
<td>original operator ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Valid abbreviations: OPER, OPID, O</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BP</td>
<td>buffer pool ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Valid abbreviation: B</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DBNAME</td>
<td>database name</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Valid abbreviation: DBN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DBRM or PACKAGE^b</td>
<td>database request module or package name</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Valid abbreviation: G</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TBNAME^bc</td>
<td>table name</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Valid abbreviation: TBN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TBCREATOR</td>
<td>table creator</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Valid abbreviation: TBC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PAGESET</td>
<td>page set name</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Valid abbreviation: PGS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>APPGROUP</td>
<td>application group</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Valid abbreviations: APGRP, AG</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BP</td>
<td>buffer pool ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Valid abbreviation: B</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DBNAME</td>
<td>database name</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Valid abbreviation: DBN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DBRM, PROGRAM, or PACKAGE</td>
<td>database request module, program name, or package name</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Valid abbreviation: G</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TBNAME^c</td>
<td>table name</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Valid abbreviation: TBN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TBCREATOR</td>
<td>table creator</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Valid abbreviation: TBC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PAGESET</td>
<td>page set name</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Valid abbreviation: PGS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>APPGROUP</td>
<td>application group</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Valid abbreviations: APGRP, AG</td>
</tr>
</tbody>
</table>

Note: For a list of appropriate qualifiers for each report, see Table 6 on page 53.
The DMDBRTSU utility generates UNLOAD and LOAD utility control statements to maintain the Performance Advisor real-time statistics (RTS) tables. The utility executes with an input parameter specifying a local DB2 subsystem that is running at the time of execution. Depending on the release level of the targeted DB2 subsystem, DMDBRTSU generates statements and data to maintain BMCINDEXSPACESTATS and BMCTABLESPACE tables (for DB2 Version 9 and later) or to maintain BMCINDEXSPACESTATS and BMCV8TBLSPACESTATS tables (for DB2 Version 8).

Run this utility once a day on each managed DB2 location to maintain a set of tables containing a daily history of real-time statistics. SAMP member PARTSUL helps you run this utility to maintain historical RTS tables in a localized PADB. SAMP members PARTSUC1 and PARTSUC2 help you run this utility to maintain the RTS tables in a centralized PADB.

Typically a job that executes DMDBRTSU in one step would then execute an UNLOAD utility such as DSNUTILB or BMC UNLOAD PLUS for DB2 as the second step to unload real-time statistics from DB2 tables. A third step to load data into the BMC tables using DSNUTILB or BMC LOAD PLUS for DB2 would follow, either in the same job if your PADB is localized, or in a separate job on your central DB2 instance if your PADB is centralized. The BMC tables are defined to receive RTS data.

---

**Table 23 QUALIFIER statement and parameter keywords (part 3 of 3)**

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Value</th>
<th>Description of value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>value or values to be included or excluded</td>
<td>not applicable</td>
<td>not applicable</td>
</tr>
<tr>
<td></td>
<td>At least one value is required for each specified qualifier type. Multiple values must be separated by commas (,). Wildcards are permitted.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

a The EXCLUDE keyword cannot be specified with DB2.

b The length of this field can be up to 128 characters and can be continued on the next line by adding a hyphen character to the end of the table name field as the last character of the line, and then continuing the name field on column 1 of the next line. For example:

```
TBNAME(LONGTABLENAMEFORDOMPLEXANDSHORTCOLSANDEVERYTHING)
```

c The length of this field can be up to 128 characters and can be continued on the next line by adding a hyphen character to the end of the table name field as the last character of the line, and then continuing the name field on column 1 of the next line. For example:

```
TBNAME(LONGTABLENAMEFORDOMPLEXANDSHORTCOLSANDEVERYTHING)
```
on a daily basis, so the utility should run shortly after midnight, generating a
timestamp of the previous day, or shortly before midnight, generating a timestamp of
the current day. For data-sharing groups, run DMDBRTSU on only one member of
the data-sharing group.

Figure 104 shows sample JCL and instructions to run the DMDBRTSU utility.

Figure 104  DMDBRTSU JCL (part 1 of 2)

```
//DMDBRTSU JOB (ACCT),'BMC RTS TABLES',     <= MODIFY
// MSGCLASS=X,                                             <= CHECK
// CLASS=A                                                 <= CHECK
//*
//** DMDBRTSU -- GENERATE UNLOAD/LOAD STATEMENTS FOR REAL-TIME STATS
//**
//** DMDBRTSU PRODUCES UNLOAD CONTROL STATEMENTS TO UNLOAD DB2 REAL-
//** TIME STATS, AND CORRESPONDING LOAD CONTROL STATEMENTS TO LOAD
//** THESE STATS INTO HISTORICAL TABLES OF DAILY SNAPSHOTS.
//**
//** SEE BBSAMP MEMBERS PARTSUL AND PARTSUC1 FOR CURRENT EXAMPLES
//** THAT USE DMDBRTSU TO MAINTAIN THESE TABLES IN A LOCALIZED AND
//** CENTRALIZED PERFORMANCE MANAGEMENT DATABASE (PMDB), RESPECTIVELY.
//**
//**         ?DB2-SSID?            = SOURCE DB2, CURRENTLY RUNNING
//RTSU  EXEC PGM=DMDBRTSU,PARM='?DB2-SSID?'
//STEPLIB DD DISP=SHR,DSN=?BMC-HLQ?.BBLINK
//TABLDEF DD DUMMY                         ** LEAVE AS DUMMY **
//UNLDCTL DD DISP=(,PASS),DSN=&&UNLDCTL,   ** UNLOAD CONTROL STMTS **
//       DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=3120),
//       SPACE=(TRK,(3,1),RLSE),
//       UNIT=SYSDA
//LOADCTL DD DISP=(,PASS),DSN=&&LOADCTL,   ** LOAD CONTROL STMTS **
//       DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=3120),
//       SPACE=(TRK,(3,1),RLSE),
//       UNIT=SYSDA
//** SYSIN STATEMENTS CONTROL THE GENERATION OF UNLOAD AND LOAD      *
//** UTILITY CONTROL STATEMENTS, AS DESCRIBED IN BBSAMP MEMBERS        *
//** PARTSUL AND PARTSUC1.                                             *
//SYSIN DD *
*********************************************************************
*                                                                 *
* UNLOAD CONTROL STATEMENTS - DB2 V8                              *
*                                                                 *
*********************************************************************
U8 UNLOAD DATA...
*********************************************************************
*                                                                 *
Table 24 describes the data sets that are processed by DMDBRTSU.

<table>
<thead>
<tr>
<th>DDNAME</th>
<th>Description</th>
<th>DCB Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>STEPLIB</td>
<td>Data Collector load library</td>
<td>not applicable</td>
</tr>
<tr>
<td>TABLDEF</td>
<td>not used</td>
<td>set to DUMMY</td>
</tr>
<tr>
<td>UNLDCTL</td>
<td>holds utility statements for unloading real-time stats from DB2 tables</td>
<td>RECFM=FB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LRECL=80</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BLKSIZE=any, multiple of LRECL</td>
</tr>
<tr>
<td>LOADCTL</td>
<td>holds utility statements for loading real-time stats into BMC historical tables</td>
<td>RECFM=FB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LRECL=80</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BLKSIZE=any, multiple of LRECL</td>
</tr>
<tr>
<td>SYSIN</td>
<td>model utility statements, subject to DB2 release, containing tokens for date, DB2 environment</td>
<td>not applicable</td>
</tr>
</tbody>
</table>

The SYsin for DMDBRTSU controls the generation of UNLOAD and LOAD statements, with substitutions for date, MVS ID, DB2 SSID, DB2 data-sharing group name, and logical DB2 name. For more information, see the SAMP members PARTSUL, PARTSUC1, and PARTSUC2.

The only modification you may need to make to the statements in SAMP depends on the time of day you run this utility. Ideally you would run the utility just after midnight each day in order to collect real-time stats for the previous day. If, instead, you run the utility just before midnight, change occurrences of **DATE** in the SYsin statements to **DATE**, so that the resulting stats are identified with the current day.
DMDBRTSU issues the following return codes:

<table>
<thead>
<tr>
<th>Return code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>successful execution of DMDBRTSU</td>
</tr>
<tr>
<td>8</td>
<td>specified DB2 subsystem was not found or is not active No control statements are generated.</td>
</tr>
</tbody>
</table>

### Performance Advisor tables

Table 25 lists the tables that Performance Advisor uses and the APPTUNE IFCIDs used as the data source.

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>IFCID 004</th>
<th>IFCID 005</th>
<th>IFCID 007</th>
<th>IFCID 008</th>
<th>IFCID 009</th>
<th>IFCID 010</th>
<th>IFCID 011</th>
<th>IFCID 307</th>
<th>IFCID 310</th>
</tr>
</thead>
<tbody>
<tr>
<td>OBJ_STATISTICS</td>
<td>Statistics per object per defined collection interval</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DAILY_OBJ_STATISTICS</td>
<td>Statistics per object per day</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WEEKLY_OBJ_STATISTICS</td>
<td>Statistics per object per week</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MONTHLY_OBJ_STATISTICS</td>
<td>Statistics per object per month</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BASELINE_OBJ_STATISTICS</td>
<td>Statistics per object representing acceptable performance</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>STMT_STATISTICS</td>
<td>Statistics per statement per defined collection interval</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DAILY_STMT_STATISTICS</td>
<td>Statistics per statement per day</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WEEKLY_STMT_STATISTICS</td>
<td>Statistics per statement per week</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MONTHLY_STMT_STATISTICS</td>
<td>Statistics per statement per month</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BASELINE_STMT_STATISTICS</td>
<td>Statistics per statement representing acceptable performance</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WKLD_STATISTICS</td>
<td>Statement statistics aggregated by non-statement collection key (plan, program, user, and so on) per defined collection interval</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table 25  Performance Advisor tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>IFCID 004</th>
<th>IFCID 005</th>
<th>IFCID 007</th>
<th>IFCID 008</th>
<th>IFCID 009</th>
<th>IFCID 010</th>
<th>IFCID 307</th>
<th>IFCID 310</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAILY_WKLD_STATISTICS</td>
<td>Statement statistics aggregated by non-statement collection key (plan, program, user, and so on) per day</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WEEKLY_WKLD_STATISTICS</td>
<td>Statement statistics aggregated by non-statement collection key (plan, program, user, etc.) per week</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MONTHLY_WKLD_STATISTICS</td>
<td>Statement statistics aggregated by non-statement collection key (plan, program, user, etc.) per month</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BASELINE_WKLD_STATISTICS</td>
<td>Statement statistics aggregated by non-statement collection key (plan, program, user, etc.) representing acceptable performance</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>STMT_STATISTICS_OB</td>
<td>Statistics per object per statement per defined collection interval</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STMT_TEXT</td>
<td>SQL statement text</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>STMT_ERRORS</td>
<td>SQL executions finishing with SQLCODE &lt; 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>STMT_EXCEPTIONS</td>
<td>SQL executions exceeding defined thresholds</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>STMT_EXCEPTIONS_HV</td>
<td>Host variable values associated with statement exceptions</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>STMT_EXCEPTIONS_OB</td>
<td>Object statistics associated with statement exceptions</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>INDEX_USAGE</td>
<td>indexes with dates last accessed for read, change, update, delete</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>BMCTABLESPACESTATS&lt;sup&gt;a&lt;/sup&gt;</td>
<td>Table space real-time stats per day</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMCV8TBLSpacESTATS&lt;sup&gt;a&lt;/sup&gt;</td>
<td>Table space real-time stats per day (DB2 Version 8)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMCINDEXSPACESTATS&lt;sup&gt;a&lt;/sup&gt;</td>
<td>Index space real-time stats per day</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<sup>a</sup> The data source is DB2 real-time statistics.
Sample members

Several members of the SAMP library help you set up and maintain your PADB tables. For more information, see the comments in each member.

Table 26 Performance Advisor samples (part 1 of 2)

<table>
<thead>
<tr>
<th>SAMP member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DMDBMERG</td>
<td>sample job to generate data for Performance Advisor tables</td>
</tr>
<tr>
<td>DMDBRTSU</td>
<td>sample job to generate unload and load statements for real-time stats</td>
</tr>
<tr>
<td>INSCACT</td>
<td>corrective action inserts for DASD MANAGER tables</td>
</tr>
<tr>
<td>INSOBJS</td>
<td>insert object sets for DASD MANAGER tables</td>
</tr>
<tr>
<td>INSTHRSH</td>
<td>insert thresholds for DASD MANAGER tables</td>
</tr>
<tr>
<td>IODDDLBA</td>
<td>creates table with APPTUNE data aggregated for baseline comparisons</td>
</tr>
<tr>
<td>IODDDLDA</td>
<td>creates table with APPTUNE data accumulated daily</td>
</tr>
<tr>
<td>IODDDLDB</td>
<td>creates a database</td>
</tr>
<tr>
<td>IODDDLDR</td>
<td>drops a database</td>
</tr>
<tr>
<td>IODDDLEV</td>
<td>creates table with APPTUNE data representing exception and error events</td>
</tr>
<tr>
<td>IODDDLIN</td>
<td>creates table with APPTUNE data by workload interval</td>
</tr>
<tr>
<td>IODDDLIU</td>
<td>creates table with aggregated index usage information collected by APPTUNE</td>
</tr>
<tr>
<td>IODDDLTE</td>
<td>creates table with aggregated SQL text collected by APPTUNE</td>
</tr>
<tr>
<td>IODDLR9</td>
<td>creates table with real-time statistics accumulated daily, for DB2 Version 8 and later releases</td>
</tr>
<tr>
<td>IODDDLRA</td>
<td>creates REORG candidate table</td>
</tr>
<tr>
<td>IODDLRT</td>
<td>creates table with real-time statistics accumulated daily, used by all releases of DB2</td>
</tr>
<tr>
<td>IODDLRE</td>
<td>REORG candidates for the REORG Advisor feature</td>
</tr>
<tr>
<td>IODDDLMO</td>
<td>creates table with APPTUNE data aggregated monthly</td>
</tr>
<tr>
<td>IODDDLWE</td>
<td>creates table with APPTUNE data aggregated weekly</td>
</tr>
<tr>
<td>IODGDG</td>
<td>creates generation data groups</td>
</tr>
<tr>
<td>IODLODBA</td>
<td>sample job to aggregate APPTUNE baseline statistics</td>
</tr>
<tr>
<td>IODLODDA</td>
<td>sample job to accumulate APPTUNE daily statistics</td>
</tr>
<tr>
<td>IODLODEV</td>
<td>sample job to accumulate APPTUNE exception and error events</td>
</tr>
<tr>
<td>IODLODIM</td>
<td>sample job to accumulate APPTUNE statistics by collection interval</td>
</tr>
<tr>
<td>IODLODIU</td>
<td>sample job to aggregate APPTUNE index usage data</td>
</tr>
<tr>
<td>IODLODOMO</td>
<td>sample job to aggregate APPTUNE monthly statistics</td>
</tr>
<tr>
<td>IODLODTE</td>
<td>sample job to aggregate APPTUNE text data</td>
</tr>
<tr>
<td>SAMP member</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>IODLODWE</td>
<td>sample job to aggregate APPTUNE monthly statistics</td>
</tr>
<tr>
<td>IODPAREO</td>
<td>sample job to run APPTUNE REORG advisor</td>
</tr>
<tr>
<td>IODRPTLD</td>
<td>sample job to run reports from APPTUNE DB2LOAD data</td>
</tr>
<tr>
<td>PACOLMNS</td>
<td>column definitions for Performance Advisor tables</td>
</tr>
<tr>
<td>PARTSUC1</td>
<td>updates historical RTS tables in a centralized PADB</td>
</tr>
<tr>
<td>PARTSUC2</td>
<td>updates historical RTS tables in a centralized PADB</td>
</tr>
<tr>
<td>PARTSUL</td>
<td>updates historical RTS tables in a localized PADB</td>
</tr>
<tr>
<td>PASETUP</td>
<td>creates tables with APPTUNE data you specify</td>
</tr>
<tr>
<td>QUERIES</td>
<td>sample SQL queries to be run against Performance Advisor tables</td>
</tr>
<tr>
<td>TRIGEXCP</td>
<td>sample job to run BMCSTATS</td>
</tr>
<tr>
<td>TRIGFORC</td>
<td>sample job to run BMCTRIG</td>
</tr>
</tbody>
</table>
Workload Access Path Compare and Workload Index Advisor reference

This appendix provides reference information for Workload Access Path Compare Path and Workload Index Advisor.

Description of source type information .................................................. 223
Working with reports ................................................................. 230
   Workload Access Path Compare and Index Advisor report fields ........ 230
   Changing the layout of the compare report ................................. 234
Return codes for Workload Access Path Compare and Index Advisor ....... 239
   Workload Access Path Compare return codes ............................. 239
   Index Advisor return codes .................................................. 240
Working tables used by Index Advisor ............................................. 240

Description of source type information

The tables in this section describe values that you enter while defining the type of source that holds the SQL to be extract for the Workload Access Path Compare component or Workload Index Advisor. The following table lists the applicable required by each source.

<table>
<thead>
<tr>
<th>Source</th>
<th>Worksheet</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2 catalog</td>
<td>Table 27 on page 224</td>
</tr>
<tr>
<td>APPTUNE trace data sets</td>
<td>Table 28 on page 224</td>
</tr>
<tr>
<td>DBRM libraries</td>
<td>Table 29 on page 226</td>
</tr>
<tr>
<td>Performance Advisor database</td>
<td>Table 30 on page 226</td>
</tr>
<tr>
<td>SQL text data sets</td>
<td>Table 31 on page 229</td>
</tr>
<tr>
<td>DB2 statement cache</td>
<td>Table 32 on page 230</td>
</tr>
</tbody>
</table>
Table 27 describes the information that you enter when identifying DB2 catalogs as the source of the extracted SQL.

Table 27 Information for DB2 catalog sources

<table>
<thead>
<tr>
<th>Field name</th>
<th>What you Enter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL Workload Source Extract from DB2 catalog (PSSWC110)</td>
<td>Specify SSID for Extract</td>
<td>Specify the subsystem ID of the DB2 system from which you want to extract SQL text.</td>
</tr>
<tr>
<td></td>
<td>Specify SSID for Extract</td>
<td>The DB2 catalog on this subsystem will be used as the source when performing the Extract and Explain processes for the workload comparisons or the Extract and Recommindex processes for the Index Advisor.</td>
</tr>
</tbody>
</table>

Table 28 describes the fields that you enter when identifying APPTUNE trace data sets as the source of the extracted SQL.

Table 28 Information for APPTUNE trace data sources

<table>
<thead>
<tr>
<th>Field name</th>
<th>What you Enter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL Workload Source Extract from APPTUNE Trace Data (PSSWC111)</td>
<td>Specify SSID for Extract</td>
<td>Specify the DB2 subsystem on which the SQL text was executed.</td>
</tr>
<tr>
<td></td>
<td>Specify SSID for Extract</td>
<td>The Workload Compare and Index Advisor feature will use the APPTUNE archived trace data sets for this subsystem as the source when performing the Extract and Explain processes for the workload comparisons. You can use a wildcard character in this field.</td>
</tr>
<tr>
<td>Ignore literals in dynamic SQL</td>
<td>Specify whether to ignore literal values for numbers and strings in dynamic SQL when determining matches.</td>
<td>Enter Y (Yes) to ignore literal values in the text of a dynamic SQL statement. All dynamic SQL statements that are otherwise the same will be considered to be the same SQL statement.</td>
</tr>
<tr>
<td></td>
<td>Ignore literals in dynamic SQL</td>
<td>Enter N (No) to honor literal values. All dynamic SQL statements that are otherwise the same but have different literal values will be considered to be different SQL statements.</td>
</tr>
<tr>
<td>List APPTUNE trace data sets</td>
<td>Specify whether to choose an archive trace data set from a list.</td>
<td>Enter Y (Yes) to display a list of archived trace data sets that contain SQL statements (BMC IFCID 005 from the APSTMT data class) from the APPTUNE archive directory for the specified DB2 subsystem.</td>
</tr>
<tr>
<td></td>
<td>List APPTUNE trace data sets</td>
<td>Enter N (No) to not list data sets. If you specify No, you must enter an archived trace data set in the Archive trace DSN field.</td>
</tr>
</tbody>
</table>
Table 28  Information for APPTUNE trace data sources

<table>
<thead>
<tr>
<th>Field name</th>
<th>What you Enter</th>
<th>Description</th>
</tr>
</thead>
</table>
| Archive directory search for approximate time of SQL execution | Specify the beginning date or both the beginning and ending date and time when the SQL was executed and written to the active trace data set. | - Valid date and time formats are `mm/dd/yy, hh:mm:ss` or `mm/dd/yy`.  
  - The Workload Access Path Compare and Index Advisor features use this information to locate the correct archive trace data set(s) when searching the APPTUNE archive directory for the SQL text.  
  **Note:** The time range that you specify is not related to the time interval used for reporting within APPTUNE. It is only used to specify the approximate time that the SQL was executed. |
| Archive trace DSN | Specify the name of an APPTUNE archived trace data set to be used to extract the SQL text. | - You must specify this value if you specified No in the List APPTUNE trace data sets field.  
  - This data set does not need to be registered in the APPTUNE archive directory, but it must contain BMC IFCID 005 records. |
| SQL Workload Source - Archive Trace Data (PSSWC117) | **Action** | Specify S to select a trace set in the list or A to add an additional archived trace data set. |
| SQL Workload Source - Add Archive Trace Data (PSSWC118) | **DSN** | Specify the name of an APPTUNE archived trace data set to be used for the extraction. |
| | | - Workload Compare automatically selects adjacent data sets when those data sets contain SQL data from the same workload interval.  
  - If you add new archived trace data sets, the added data sets exist only in the extraction and are not saved in the APPTUNE archive directory.  
  - The added data sets exist only in the list for the SQL extraction and are not saved in the APPTUNE archive directory.  
  - You might need to add an additional archive trace data set if it was not registered in the COPYDIR at the time that it was created. |
Table 29 describes the fields that you enter when identifying DBRM libraries as the source of the extracted SQL.

Table 29  Information for DBRM libraries

<table>
<thead>
<tr>
<th>Field name</th>
<th>What you Enter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL Workload Source Extract from DBRM Library (PSSWC114)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Specify SSID for Explain</td>
<td>Specify the DB2 subsystem on which the SQL text from the DBRM library is to be explained.</td>
<td>Objects do not have to be bound on this subsystem, but the tables must exist.</td>
</tr>
<tr>
<td>Table qualifier</td>
<td>Specify a table qualifier.</td>
<td>This table qualifier is used to qualify any unqualified SQL text in the Explain process.</td>
</tr>
<tr>
<td>Specify PDS with member name or wildcard member</td>
<td>Specify the data set and member name of the DBRM library.</td>
<td>You can use wildcards in the member name. Examples:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MYUSERID.TEST.DBRMLIB(TEST)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MYUSERID.PROD.DBRMLIB(APROD*)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MYUSERID.PROD.DBRMLIB(*)</td>
</tr>
<tr>
<td></td>
<td>The wildcard character can be anywhere within the name (leading, trailing, or in the middle).</td>
<td></td>
</tr>
</tbody>
</table>

Table 30 describes the information that you enter when identifying a Performance Advisor Database (PADB) as the source of the extracted SQL.

Table 30  Information for Performance Advisor Database sources

<table>
<thead>
<tr>
<th>Field name</th>
<th>What you Enter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extract SQL for Performance Advisor Database (PSSWC160)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PADB Table Creator a</td>
<td>Specify the name of the owner or creator that was used when the Performance Advisor Database was created.</td>
<td>If the creator name has changed from the default value, edit the field to contain the updated value.</td>
</tr>
<tr>
<td>PADB STMT Text Table Name a</td>
<td>Specify the SQL text table name for the Performance Advisor Database</td>
<td>The default name is STMT_TEXT. If the SQL text table name has changed from the default value, edit the field to contain the updated value.</td>
</tr>
</tbody>
</table>
Table 30  Information for Performance Advisor Database sources

<table>
<thead>
<tr>
<th>Field name</th>
<th>What you Enter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PADB Tables *</td>
<td>Type S next to the table to use as the source for the SQL text extract process and edit the name of the table.</td>
<td>- If the name of the table has changed from the default value, edit the table name to the updated value.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- You can choose from the following tables:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- DAILY_STMT_STATISTICS — The daily STMT statistics table contains SQL statement statistics per statement per day.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- WEEKLY_STMT_STATISTICS — The weekly STMT statistics table contains SQL statement statistics per statement per week.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- MONTHLY_STMT_STATISTICS — The monthly STMT statistics table contains SQL statement statistics per statement per month.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- DAILY_STMT_SUMMARY — The daily STMT summary table contains SQL statement statistics per statement per day. This table contains a subset of columns from the Daily STMT Statistics table and generally loads faster.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- WEEKLY_STMT_SUMMARY — The weekly STMT summary table contains SQL statement statistics per statement per week. This table contains a subset of columns from the Weekly STMT Statistics table and generally loads faster.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- MONTHLY_STMT_SUMMARY — The monthly STMT summary table contains SQL statement statistics per statement per month. This table contains a subset of columns from the Monthly STMT Statistics table and generally loads faster.</td>
</tr>
<tr>
<td>DB2 SSID where the PADB resides</td>
<td>Specify the subsystem ID of the DB2 subsystem where the Performance Advisor Database is installed.</td>
<td>- You cannot use wildcards in this field.</td>
</tr>
<tr>
<td>DB2 SSID where the data was</td>
<td>Specify the subsystem ID of the DB2 subsystem from which statement text and statistics were originally collected.</td>
<td>- This value is used as a search parameter for the data in the Performance Advisor Database.</td>
</tr>
<tr>
<td>collected</td>
<td></td>
<td>- You can use a wildcard character (&quot;*&quot; or &quot;) as part of the name (for example, DBN&quot;).</td>
</tr>
</tbody>
</table>
Table 30  Information for Performance Advisor Database sources

<table>
<thead>
<tr>
<th>Field name</th>
<th>What you Enter</th>
<th>Description</th>
</tr>
</thead>
</table>
| Last N <time> | Specify the number of time periods (from 1 through 10) for which you want to retrieve statement statistics and SQL text. | - The time periods correspond to workload intervals in the Performance Advisor table that you selected.  
- The default value of 1 retrieves the statement statistics and SQL text for the last workload interval time.  
- Units of measurement are as follows:  
  - DAILY_STMT_STATISTICS - n days  
  - WEEKLY_STMT_STATISTICS - n weeks  
  - MONTHLY_STMT_STATISTICS - n months  
  - DAILY_STMT_SUMMARY - n days  
  - WEEKLY_STMT_SUMMARY - n weeks  
  - MONTHLY_STMT_SUMMARY - n months |
| Approximate Start Date | Specify the approximate start date to use to locate the rows in the Performance Advisor Database table. | The Extract process will use the workload interval time closest to the specified date. The specified time will be greater than or equal to the workload interval time (INTVTIME) in the specified table. Valid date formats are:  
  - mm/dd/yy  
  - yy-mm-dd  
  - mm/dd/yyyy  
  - yyyy-mm-dd |
| For Duration | Specify the duration that will define the ending time for extracting data | - The duration label that is displayed depends on the table that was selected on the previous panel.  
- For example, if you selected the DAILY statistics table, the duration label will be DAYS. If you selected the WEEKLY statistics table is selected, the duration label will be WEEKS.  
**Note:** If you omit both the LAST N and Approximate Start Date field from the batch parameters, the default value will be LAST 1 and the time period will depend on the table selected (as described above). |

a If the Performance Database names for objects have changed from those distributed by BMC, you can edit the names on the panel by typing the new name in place of the BMC name.  
If the length of the new name exceeds the space allowed on the panel, press PF6 (field-level zoom) on the table name field to enter the new name on the zoomed panel.
Table 31 describes the fields that you enter when identifying a SQL text data set as the source of the extracted SQL.

**Table 31 Information for SQL text data set sources**

<table>
<thead>
<tr>
<th>Field name</th>
<th>What you Enter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Specify SSID for Explain</strong></td>
<td>Specify the DB2 subsystem on which the SQL text from the data set is to be explained.</td>
<td>The objects do not have to be bound on this subsystem.</td>
</tr>
</tbody>
</table>
| **Table qualifier**             | Specify the table qualifier to use to qualify any unqualified SQL text in the Explain operation. | You can change the table qualifiers for different SQL statements referencing unqualified table names with the SQL text data set by including the following statement prior to the SQL text:  

```
-- TBQUAL = qualifier
```

The Extract process will use the qualifier for the SQL statements that follow until it encounters another TBQUAL statement or reaches the end of the file. To reset the table qualifier to the value specified on the panel and remove the qualifier from use for subsequent statements, specify the keyword QUAL as the qualifier value:

```
-- TBQUAL = QUAL
```

| **Specify Data Set Name**       | Specify the data set name of the file that contains the SQL text.              | Include the member name if the data set is a PDS (for example, MYUSERID.TEST.SQL(TEST)).                                              |
| **File Format**                 | Specify the file type of the data set from which SQL text is to be extracted. | Valid values include:                                                                                                                  |
|                                 |                                                                               | ■ 1 (COBOL) - A COBOL program source code file that contains static SQL.                                                              |
|                                 |                                                                               | ■ 2 (PL/I) - A PL/I program source code file that contains static SQL.                                                                   |
|                                 |                                                                               | ■ 3 (C) - A C/C++ program source code file that contains static SQL.                                                                      |
|                                 |                                                                               | ■ 4 (Assembler) - An assembler language source code file that contains static SQL.                                                        |
|                                 |                                                                               | ■ 5 (FORTRAN) - A FORTRAN program source code file that contains static SQL.                                                              |
|                                 |                                                                               | ■ 6 (SQL text only) - An SQL text file that contains SQL statements terminated by a semi-colon (;).                                        |
|                                 |                                                                               | The default format is SQL text.                                                                                                          |
Table 32 describes the information that you enter when identifying the DB2 statement cache as the source of the extracted SQL.

### Table 32  Information for DB2 statement cache sources

<table>
<thead>
<tr>
<th>Field name</th>
<th>What you Enter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SQL Workload Source Extract from DB2 Statement Cache (PSSWC280)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Specify SSID for Extract</td>
<td>Specify the subsystem ID of the DB2 system from which you want to extract SQL text.</td>
<td>The DB2 statement cache on this subsystem will be accessed as the source using the DSN_STATEMENT_CACHE_TABLE of the user ID associated with the runner of the batch job. This Explain table will be created if the DSN_STATEMENT_CACHE_TABLE does not already exist. The runner of the batch job should have should have authority to issue an EXPLAIN STMTCACHE ALL statement.</td>
</tr>
</tbody>
</table>

## Working with reports

This section describes how to work with the workload compare and recommendindex reports. It includes information about

- “Workload Access Path Compare and Index Advisor report fields” on page 230
- “Changing the layout of the compare report” on page 234

### Workload Access Path Compare and Index Advisor report fields

This section lists the fields found on the Workload Access Path Compare or Index Advisors reports in alphabetic order by field label name. Each row indicates whether the field is used in Workload Access Path Compare (WLC) or Index Advisor (IA), or both is shown in the first column. The report also lists the DB2 column name, the DB2 table or other source from which it comes, and the section of the report in which it can be found.

The field labels are controlled by the HLQ.LLQSAMP(PSSREPB) layout data set member. The labels shown below are the default labels that are shipped with SQL Performance. For information about customizing the layout of your reports, see “Changing the layout of the compare report” on page 234.
Table 33  Workload Access Path Compare and Index Advisor report field labels (from PSSREPB layout data set)  (part 1 of 4)

<table>
<thead>
<tr>
<th>Apply</th>
<th>Report field label</th>
<th>DB2 column name</th>
<th>DB2 table or other source</th>
<th>Report section</th>
</tr>
</thead>
<tbody>
<tr>
<td>WLC</td>
<td>AC DG</td>
<td>ACCESS_DEGREE</td>
<td>PLAN_TABLE</td>
<td>PATH</td>
</tr>
<tr>
<td>WLC</td>
<td>AC PG</td>
<td>ACCESS_PGROUP_ID</td>
<td>PLAN_TABLE</td>
<td>PATH</td>
</tr>
<tr>
<td>WLC</td>
<td>AC TY</td>
<td>ACCESSTYPE</td>
<td>PLAN_TABLE</td>
<td>PATH</td>
</tr>
<tr>
<td>Both</td>
<td>BUFFER POOL</td>
<td>BPOOL</td>
<td>SYSTABLESPACE</td>
<td>TABL</td>
</tr>
<tr>
<td>Both</td>
<td>BUFFER POOL</td>
<td>BPOOL</td>
<td>SYSINDEXES</td>
<td>INDX</td>
</tr>
<tr>
<td>Both</td>
<td>C</td>
<td>COMPRESS</td>
<td>SYSTABLESPACE</td>
<td>TABL</td>
</tr>
<tr>
<td>Both</td>
<td>C D</td>
<td>CLUSTERED</td>
<td>SYSINDEXES</td>
<td>INDX</td>
</tr>
<tr>
<td>WLC</td>
<td>C F</td>
<td>COLUMN_FN_EVAL</td>
<td>PLAN_TABLE</td>
<td>PATH</td>
</tr>
<tr>
<td>Both</td>
<td>C G</td>
<td>CLUSTERING</td>
<td>SYSINDEXES</td>
<td>INDX</td>
</tr>
<tr>
<td>Both</td>
<td>CARDF</td>
<td>CARDF</td>
<td>SYSTABLES</td>
<td>TABL</td>
</tr>
<tr>
<td>WLC</td>
<td>CARDF</td>
<td>CARDF</td>
<td>SYSTABLES</td>
<td>TABL</td>
</tr>
<tr>
<td>WLC</td>
<td>CARDF</td>
<td>CARDF</td>
<td>SYSTABLES</td>
<td>TABL</td>
</tr>
<tr>
<td>Both</td>
<td>CLUSTER RATIO</td>
<td>CLUSTERRATIOF</td>
<td>SYSINDEXES</td>
<td>INDX</td>
</tr>
<tr>
<td>Both</td>
<td>CO NO</td>
<td>COLNO</td>
<td>SYSKEYS</td>
<td>KEYC</td>
</tr>
<tr>
<td>Both</td>
<td>CO SQ</td>
<td>COLSEQ</td>
<td>SYSKEYS</td>
<td>KEYC</td>
</tr>
<tr>
<td>Both</td>
<td>COL LEN</td>
<td>LENGTH</td>
<td>SYSCOLUMNS</td>
<td>COLS</td>
</tr>
<tr>
<td>Both</td>
<td>COL NUM</td>
<td>COLNO</td>
<td>SYSCOLUMNS</td>
<td>COLS</td>
</tr>
<tr>
<td>Both</td>
<td>COLCARDF</td>
<td>COLCARDF</td>
<td>SYSCOLUMNS</td>
<td>COLS</td>
</tr>
<tr>
<td>Both</td>
<td>COLLID</td>
<td>CLID</td>
<td>SYSPACKAGE, APPTUNE</td>
<td>STMT</td>
</tr>
<tr>
<td>WLC</td>
<td>COLUMN GROUP</td>
<td>COLGROUPCOLNO</td>
<td>SYSCOLDIST</td>
<td>DIST</td>
</tr>
<tr>
<td>Both</td>
<td>COLUMN NAME</td>
<td>NAME</td>
<td>SYSCOLUMNS</td>
<td>COLS</td>
</tr>
<tr>
<td>WLC</td>
<td>COLUMN NAME</td>
<td>NAME</td>
<td>SYSCOLUMNS</td>
<td>DIST</td>
</tr>
<tr>
<td>Both</td>
<td>COLUMN TYPE</td>
<td>COLTYPE</td>
<td>SYSCOLUMNS</td>
<td>COLS</td>
</tr>
<tr>
<td>WLC</td>
<td>COLUMN VALUE</td>
<td>COLVALUE</td>
<td>SYSCOLDIST</td>
<td>DIST</td>
</tr>
<tr>
<td>WLC</td>
<td>COMP UJOG</td>
<td>SORTC_UNIQUE</td>
<td>PLAN_TABLE</td>
<td>PATH</td>
</tr>
<tr>
<td>WLC</td>
<td>CORR NAME</td>
<td>CORRELATION_NAME</td>
<td>PLAN_TABLE</td>
<td>PATH</td>
</tr>
<tr>
<td>Both</td>
<td>DBNAME</td>
<td>DBNAME</td>
<td>SYSTABLES</td>
<td>TABL</td>
</tr>
<tr>
<td>WLC</td>
<td>DIFF IN</td>
<td>DIFFIN</td>
<td>none</td>
<td>STMT</td>
</tr>
<tr>
<td>Both</td>
<td>E S</td>
<td>ENCODING_SCHEME</td>
<td>SYSTABLESPACE</td>
<td>TABL</td>
</tr>
<tr>
<td>Both</td>
<td>EDIT PROC</td>
<td>EDPROC</td>
<td>SYSTABLES</td>
<td>TABL</td>
</tr>
<tr>
<td>IA</td>
<td>EQUAL</td>
<td>EQUAL</td>
<td>SQL text</td>
<td>COLS</td>
</tr>
<tr>
<td>Both</td>
<td>EXEC COUNT</td>
<td>COUNT</td>
<td>none</td>
<td>STMT</td>
</tr>
<tr>
<td>Both</td>
<td>EXPL SSID</td>
<td>EXPLSSID</td>
<td>none</td>
<td>STMT</td>
</tr>
<tr>
<td>Both</td>
<td>EXPLAIN TIME</td>
<td>BIND_TIME</td>
<td>PLAN_TABLE</td>
<td>PATH</td>
</tr>
<tr>
<td>Both</td>
<td>EXPLAIN TIME</td>
<td>BIND_TIME</td>
<td>PLAN_TABLE</td>
<td>STMT</td>
</tr>
<tr>
<td>Both</td>
<td>EXTRACT TIME</td>
<td>TIMESTAMP</td>
<td>SYSPLAN, SYSPACKAGE,</td>
<td>STMT</td>
</tr>
</tbody>
</table>

Appendix B  Workload Access Path Compare and Workload Index Advisor reference  231
### Table 33  Workload Access Path Compare and Index Advisor report field labels (from PSSREPB layout data set) (part 2 of 4)

<table>
<thead>
<tr>
<th>Apply</th>
<th>Report field label</th>
<th>DB2 column name</th>
<th>DB2 table or other source</th>
<th>Report section</th>
</tr>
</thead>
<tbody>
<tr>
<td>IA</td>
<td>FETCH</td>
<td>FETCH</td>
<td>SQL text</td>
<td>COLS</td>
</tr>
<tr>
<td>Both</td>
<td>FIRST KEYCARD</td>
<td>FIRSTKEYCARD</td>
<td>SYSINDEXES</td>
<td>INDX</td>
</tr>
<tr>
<td>Both</td>
<td>FREE PAGE</td>
<td>FREEPAGE</td>
<td>SYSTABLEPART</td>
<td>TABL</td>
</tr>
<tr>
<td>WLC</td>
<td>FREQUENCY</td>
<td>FREQUENCYF</td>
<td>SYSCOLDIST</td>
<td>DIST</td>
</tr>
<tr>
<td>Both</td>
<td>FULL KEYCARD</td>
<td>FULLKEYCARDF</td>
<td>SYSINDEXES</td>
<td>INDX</td>
</tr>
<tr>
<td>WLC</td>
<td>GROUP MEMBER</td>
<td>GROUP_MEMBER</td>
<td>PLAN_TABLE</td>
<td>PATH</td>
</tr>
<tr>
<td>IA</td>
<td>GRPBY</td>
<td>GROUPBY</td>
<td>SQL text</td>
<td>COLS</td>
</tr>
<tr>
<td>WLC</td>
<td>HIGH VALUE</td>
<td>HIGHVALUE</td>
<td>SYSCOLDIST</td>
<td>DIST</td>
</tr>
<tr>
<td>Both</td>
<td>HIGH2KEY</td>
<td>HIGH2KEY</td>
<td>SYSCOLUMNS</td>
<td>COLS</td>
</tr>
<tr>
<td>WLC</td>
<td>HINT USED</td>
<td>HINT_USED</td>
<td>PLAN_TABLE</td>
<td>PATH</td>
</tr>
<tr>
<td>WLC</td>
<td>I O</td>
<td>INDEXONLY</td>
<td>PLAN_TABLE</td>
<td>PATH</td>
</tr>
<tr>
<td>WLC</td>
<td>INDEX</td>
<td>ACCESSNAME</td>
<td>PLAN_TABLE</td>
<td>PATH</td>
</tr>
<tr>
<td>Both</td>
<td>INDEX CREATOR</td>
<td>CREATOR</td>
<td>SYSINDEXES</td>
<td>INDX</td>
</tr>
<tr>
<td>Both</td>
<td>INDEX CREATOR</td>
<td>IXCREATOR</td>
<td>SYSKEYS</td>
<td>KEYC</td>
</tr>
<tr>
<td>WLC</td>
<td>INDEX CREATOR</td>
<td>ACCESSCREATOR</td>
<td>PLAN_TABLE</td>
<td>PATH</td>
</tr>
<tr>
<td>Both</td>
<td>INDEX NAME</td>
<td>NAME</td>
<td>SYSINDEXES</td>
<td>INDX</td>
</tr>
<tr>
<td>Both</td>
<td>INDEX NAME</td>
<td>IXNAME</td>
<td>SYSKEYS</td>
<td>KEYC</td>
</tr>
<tr>
<td>IA</td>
<td>IXABC</td>
<td>INDEXABLE</td>
<td>SQL text</td>
<td>COLS</td>
</tr>
<tr>
<td>WLC</td>
<td>J C</td>
<td>MERGE_JOIN_COLS</td>
<td>PLAN_TABLE</td>
<td>PATH</td>
</tr>
<tr>
<td>WLC</td>
<td>J D</td>
<td>JOIN_DEGREE</td>
<td>PLAN_TABLE</td>
<td>PATH</td>
</tr>
<tr>
<td>WLC</td>
<td>J T</td>
<td>JOIN_TYPE</td>
<td>PLAN_TABLE</td>
<td>PATH</td>
</tr>
<tr>
<td>WLC</td>
<td>JN PG</td>
<td>JOIN_PGROUP_ID</td>
<td>PLAN_TABLE</td>
<td>PATH</td>
</tr>
<tr>
<td>Both</td>
<td>KEY COLUMN</td>
<td>COLNAME</td>
<td>SYSKEYS</td>
<td>KEYC</td>
</tr>
<tr>
<td>Both</td>
<td>L R</td>
<td>LOCKRULE</td>
<td>SYSTABLESPACE</td>
<td>TABL</td>
</tr>
<tr>
<td>WLC</td>
<td>LCK MOD</td>
<td>TSLOCKMODE</td>
<td>PLAN_TABLE</td>
<td>PATH</td>
</tr>
<tr>
<td>WLC</td>
<td>LOW VALUE</td>
<td>LOWVALUE</td>
<td>SYSCOLDIST</td>
<td>DIST</td>
</tr>
<tr>
<td>Both</td>
<td>LOW2KEY</td>
<td>LOW2KEY</td>
<td>SYSCOLUMNS</td>
<td>COLS</td>
</tr>
<tr>
<td>WLC</td>
<td>M E</td>
<td>METHOD</td>
<td>PLAN_TABLE</td>
<td>PATH</td>
</tr>
<tr>
<td>WLC</td>
<td>MT CL</td>
<td>MATCHCOLS</td>
<td>PLAN_TABLE</td>
<td>PATH</td>
</tr>
<tr>
<td>WLC</td>
<td>MX SQ</td>
<td>MIXOPSEQ</td>
<td>PLAN_TABLE</td>
<td>PATH</td>
</tr>
<tr>
<td>Both</td>
<td>N</td>
<td>NULLS</td>
<td>SYSCOLUMNS</td>
<td>COLS</td>
</tr>
<tr>
<td>WLC</td>
<td>NACTIVE PAGES</td>
<td>NACTIVE</td>
<td>SYSTABSTATS</td>
<td>TABS</td>
</tr>
<tr>
<td>Both</td>
<td>NACTIVEF</td>
<td>NACTIVEF</td>
<td>SYSTABLESPACE</td>
<td>TABL</td>
</tr>
<tr>
<td>Both</td>
<td>NAME</td>
<td>PROG</td>
<td>SYSPLAN, SYSPACKAGE, APPTUNE trace data</td>
<td>STMT</td>
</tr>
<tr>
<td>WLC</td>
<td>NEW UJOG</td>
<td>SORTN_UNIQUE</td>
<td>PLAN_TABLE</td>
<td>PATH</td>
</tr>
<tr>
<td>Both</td>
<td>NLEAF</td>
<td>NLEAF</td>
<td>SYSINDEXES</td>
<td>INDX</td>
</tr>
<tr>
<td>Both</td>
<td>NLVL</td>
<td>NLEVELS</td>
<td>SYSINDEXES</td>
<td>INDX</td>
</tr>
</tbody>
</table>
Table 33  Workload Access Path Compare and Index Advisor report field labels (from PSSREPB layout data set) (part 3 of 4)

<table>
<thead>
<tr>
<th>Apply</th>
<th>Report field label</th>
<th>DB2 column name</th>
<th>DB2 table or other source</th>
<th>Report section</th>
</tr>
</thead>
<tbody>
<tr>
<td>WLC</td>
<td>NM CO</td>
<td>NUMCOLUMNS</td>
<td>SYSCOLDIST</td>
<td>DIST</td>
</tr>
<tr>
<td>WLC</td>
<td>NPAGES</td>
<td>NPAGES</td>
<td>SYSTABSTATS</td>
<td>TABS</td>
</tr>
<tr>
<td>Both</td>
<td>NPAGESF</td>
<td>NPAGESF</td>
<td>SYSTABLES</td>
<td>TABL</td>
</tr>
<tr>
<td>IA</td>
<td>#STMTS</td>
<td>NUMSTMTS</td>
<td>none</td>
<td>INDX</td>
</tr>
<tr>
<td>Both</td>
<td>O</td>
<td>ORDERING</td>
<td>SYSKEYS</td>
<td>KEYC</td>
</tr>
<tr>
<td>WLC</td>
<td>OPT HINT</td>
<td>OPTHINT</td>
<td>SYSPACKAGE</td>
<td>PATH</td>
</tr>
<tr>
<td>IA</td>
<td>OTHER</td>
<td>OTHER</td>
<td>SQL text</td>
<td>COLS</td>
</tr>
<tr>
<td>WLC</td>
<td>P A</td>
<td>PRIMARY_ACCESTYPE</td>
<td>PLAN_TABLE</td>
<td>PATH</td>
</tr>
<tr>
<td>WLC</td>
<td>P F</td>
<td>PREFETCH</td>
<td>PLAN_TABLE</td>
<td>PATH</td>
</tr>
<tr>
<td>WLC</td>
<td>P M</td>
<td>PARALLELISM_MODE</td>
<td>PLAN_TABLE</td>
<td>PATH</td>
</tr>
<tr>
<td>WLC</td>
<td>P R</td>
<td>PAGE_RANGE</td>
<td>PLAN_TABLE</td>
<td>PATH</td>
</tr>
<tr>
<td>WLC</td>
<td>PA QB</td>
<td>PARENT_QBLOCKNO</td>
<td>PLAN_TABLE</td>
<td>PATH</td>
</tr>
<tr>
<td>WLC</td>
<td>PART</td>
<td>PARTITION</td>
<td>SYSTABLESPACE</td>
<td>TABS</td>
</tr>
<tr>
<td>Both</td>
<td>PARTS</td>
<td>PARTITIONS</td>
<td>SYSTABLESPACE</td>
<td>TABL</td>
</tr>
<tr>
<td>Both</td>
<td>PCT COMP</td>
<td>PCTROWCOMP</td>
<td>SYSTABLES</td>
<td>TABL</td>
</tr>
<tr>
<td>Both</td>
<td>PCT FREE</td>
<td>PCTFREE</td>
<td>SYSTABLEPART</td>
<td>TABL</td>
</tr>
<tr>
<td>Both</td>
<td>PCT PAGES</td>
<td>PCTPAGES</td>
<td>SYSTABLES</td>
<td>TABL</td>
</tr>
<tr>
<td>WLC</td>
<td>PCT PGS</td>
<td>PCTPAGES</td>
<td>SYSTABLES</td>
<td>TABS</td>
</tr>
<tr>
<td>WLC</td>
<td>PCT ROW</td>
<td>PCTROWCOMP</td>
<td>SYSTABLES</td>
<td>TABS</td>
</tr>
<tr>
<td>WLC</td>
<td>PL NO</td>
<td>PLANNO</td>
<td>PLAN_TABLE</td>
<td>PATH</td>
</tr>
<tr>
<td>WLC</td>
<td>QB BL</td>
<td>QBLOCKNO</td>
<td>PLAN_TABLE</td>
<td>PATH</td>
</tr>
<tr>
<td>WLC</td>
<td>QBLOCK TYPE</td>
<td>QBLOCK_TYPE</td>
<td>PLAN_TABLE</td>
<td>PATH</td>
</tr>
<tr>
<td>WLC</td>
<td>QNO</td>
<td>QUANTILENO</td>
<td>SYSCOLDIST</td>
<td>DIST</td>
</tr>
<tr>
<td>IA</td>
<td>RANGE</td>
<td>RANGE</td>
<td>SQL text</td>
<td>COLS</td>
</tr>
<tr>
<td>IA</td>
<td>RECOMMEND</td>
<td>RECOMMEND</td>
<td>none</td>
<td>INDX</td>
</tr>
<tr>
<td>IA</td>
<td>SAVINGS</td>
<td>SAVINGS</td>
<td>none</td>
<td>INDX</td>
</tr>
<tr>
<td>WLC</td>
<td>SC PG</td>
<td>SORTC_PGROUP_ID</td>
<td>PLAN_TABLE</td>
<td>PATH</td>
</tr>
<tr>
<td>Both</td>
<td>SCALE</td>
<td>SCALE</td>
<td>SYSCOLUMNS</td>
<td>COLS</td>
</tr>
<tr>
<td>Both</td>
<td>SEG SIZE</td>
<td>SEGSIZE</td>
<td>SYSTABLESPACE</td>
<td>TABL</td>
</tr>
<tr>
<td>WLC</td>
<td>SN PG</td>
<td>SORTN_PGROUP_ID</td>
<td>PLAN_TABLE</td>
<td>PATH</td>
</tr>
<tr>
<td>IA</td>
<td>SORT1</td>
<td>ORDERBY1</td>
<td>SQL text</td>
<td>COLS</td>
</tr>
<tr>
<td>IA</td>
<td>SORT2</td>
<td>ORDERBY2</td>
<td>SQL text</td>
<td>COLS</td>
</tr>
<tr>
<td>IA</td>
<td>SORT3</td>
<td>ORDERBY3</td>
<td>SQL text</td>
<td>COLS</td>
</tr>
<tr>
<td>Both</td>
<td>SRC SSID</td>
<td>SSID</td>
<td>none</td>
<td>STMT</td>
</tr>
<tr>
<td>Both</td>
<td>SRC TYPE</td>
<td>CTYP</td>
<td>APPTUNE trace data</td>
<td>STMT</td>
</tr>
<tr>
<td>IA</td>
<td>STAG1</td>
<td>STAGE1</td>
<td>SQL text</td>
<td>COLS</td>
</tr>
<tr>
<td>Both</td>
<td>STATSTIME</td>
<td>STATSTIME</td>
<td>SYSTABLES</td>
<td>TABL</td>
</tr>
<tr>
<td>Both</td>
<td>STATSTIME</td>
<td>STATSTIME</td>
<td>SYSINDEXES</td>
<td>INDX</td>
</tr>
<tr>
<td>Both</td>
<td>STATSTIME</td>
<td>STATSTIME</td>
<td>SYSCOLUMNS</td>
<td>COLS</td>
</tr>
</tbody>
</table>
The appearance of the online and batch reports is controlled by a layout data set member. The Workload Access Path Compare component and Workload Index Advisor are shipped with a default layout member PSSREPB in the HLQ.LLQSAMP data set. The product does not allow customization of the default PSSREPB member, but you can copy the default layout to a new data set name or member and edit that member. You can save one or more layout members.

A sample of the default member is shown in Figure 105 on page 236.

**NOTE**

The name of HLQ.LLQSAMP will vary depending on your installation options and might be HLQ.BBSAMP, HLQ.PSSSAMP, or HLQ.BMCSAMP.
To customize a default layout member

1 Copy the PSSREPB member.

The new default layout member can reside in any data set.

2 Edit the newly created member and save your changes.

**WARNING**

DO NOT make changes to the delivered PSSREPB member in the HLQ.LLQSAMP data set.

DO NOT change the values in the COL_NAME column or the section names in the brackets in your new layout member.

You can change the following values:

<table>
<thead>
<tr>
<th>value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WIDTH</td>
<td>Specify the width of a column.</td>
</tr>
<tr>
<td></td>
<td>If the display width of a field is too short to show all of the values</td>
</tr>
<tr>
<td></td>
<td>in the field, one of the following events occurs:</td>
</tr>
<tr>
<td></td>
<td>■ For numbers, the product attempts to show the number up to the</td>
</tr>
<tr>
<td></td>
<td>decimal point. If this is not possible, asterisks (*) are used to</td>
</tr>
<tr>
<td></td>
<td>fill the entire field.</td>
</tr>
<tr>
<td></td>
<td>■ For characters, the product truncates the character text from</td>
</tr>
<tr>
<td></td>
<td>the right.</td>
</tr>
<tr>
<td>SHOW</td>
<td>Specify whether to display a column value on the report.</td>
</tr>
<tr>
<td></td>
<td>■ S - SHOW the column</td>
</tr>
<tr>
<td></td>
<td>■ H - HIDE the column</td>
</tr>
<tr>
<td>COLUMN HEADING</td>
<td>Specify the heading of the report column.</td>
</tr>
<tr>
<td></td>
<td>If the value that you enter in the column heading is longer than the</td>
</tr>
<tr>
<td></td>
<td>width of the column, the column will be the width of the heading.</td>
</tr>
<tr>
<td></td>
<td>To display a two word heading on two lines, type the two words separated by</td>
</tr>
<tr>
<td></td>
<td>a space. For example, typing “EXPL SSID” will show on the report as</td>
</tr>
<tr>
<td></td>
<td>EXPL</td>
</tr>
<tr>
<td></td>
<td>SSID</td>
</tr>
<tr>
<td></td>
<td>To display a two word heading on the same line, separate the words with an</td>
</tr>
<tr>
<td></td>
<td>underscore. For example, typing “EXPL_SSID” will show on the report as</td>
</tr>
</tbody>
</table>
|                |     EXPL_SSID
3 Specify the new layout data set in the Report Options panel.

To set report options, see “Workload Access Path Compare and Index Advisor report fields” on page 230.

Figure 105 Default layout data set member (PSSREPB) (part 1 of 4)

```
* SQL PERFORMANCE V10R1
* THIS IS THE LAYOUT FILE USED TO DISPLAY RESULTS
* FROM BATCH AND ONLINE WORKLOAD COMPARE.
* AN ASTERISK IN COLUMN 1 INDICATES A COMMENT LINE.
* SECTION NAMES ARE IN BRACKETS "" AND CANNOT BE CHANGED.
* EACH SECTION HAS FIVE COLUMNS AS FOLLOWS:
*   WIDTH = THE NUMBER OF CHARACTERS TO USE FOR DISPLAYING DATA.
*   SHOW = WHETHER TO SHOW OR HIDE THE DATA IN THIS COLUMN.
*          S = SHOW THE DATA
*          H = HIDE THE DATA
*   COL_NAME = THE NAME ASSOCIATED WITH THE SOURCE
*          OF THE DATA. THE NAMES IN THIS COLUMN CANNOT BE CHANGED.
*   COLUMN HEADING1 = THE NAME TO USE AS THE FIRST COLUMN HEADING
*   COLUMN HEADING2 = THE NAME TO USE AS THE SECOND COLUMN HEADING
* THE SECTIONS MAY APPEAR IN ANY ORDER IN THIS LAYOUT.
* THE ORDER OF THE COLUMNS IS IMPLIED BY THE ORDER IN WHICH THEY APPEAR
* IN THE LIST. ANY COLUMNS THAT ARE HIDDEN WILL BE SKIPPED.
*
ÝSTMT"*
* THIS IS INFORMATION FOR AN SQL STATEMENT.
*WIDTH SHOW  COL_NAME            COLUMN HEADING
  4 S  EXPLSSID         EXPL SSID
  4 S  SSID             SRC SSID
  8 S  PROG             NAME
 12 S  CLID             COLLID
 26 S  VERSION          VERSION
  6 S  STMTNO           STMTNO
  7 S  TOTAL_COST       STMT COST
  7 S  COUNT            EXEC COUNT
  8 S  WEIGHTED         WEIGHTED COST
  8 S  DIFF             WEIGHTED DIFF
 24 S  DIFFIN           DIFF IN
  1 S  CTYP             SRC TYPE
 26 H  TIMESTAMP        EXTRACT TIME
 26 S  BIND_TIME        EXPLAIN TIME
ÝPATH"*
* THIS IS INFORMATION FOR AN ACCESS PATH STEP.
*WIDTH SHOW  COL_NAME            COLUMN HEADING
  2 S  QBLOCKNO         QB BL
  2 S  PLANNO           PL NO
  1 S  METHOD           M E
  2 S  TABNO            TB NO
  2 S  ACESSTYPE        AC TY
```
### Figure 105  Default layout data set member (PSSREPB)  (part 2 of 4)

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2 S</td>
<td>MATCHCOLS</td>
<td>MT CL</td>
</tr>
<tr>
<td>12 S</td>
<td>TNAME</td>
<td>TABLE</td>
</tr>
<tr>
<td>12 S</td>
<td>ACCESSNAME</td>
<td>INDEX</td>
</tr>
<tr>
<td>2 S</td>
<td>INDEXONLY</td>
<td>I O</td>
</tr>
<tr>
<td>4 S</td>
<td>SORTN</td>
<td>NEW UJOG</td>
</tr>
<tr>
<td>4 S</td>
<td>SORTC</td>
<td>COMP UJOG</td>
</tr>
<tr>
<td>1 S</td>
<td>PREFETCH</td>
<td>P F</td>
</tr>
<tr>
<td>1 S</td>
<td>COLUMN_FN_EVAL</td>
<td>C F</td>
</tr>
<tr>
<td>2 S</td>
<td>MIXOPSEQ</td>
<td>MX SQ</td>
</tr>
<tr>
<td>3 S</td>
<td>TSLOCKMODE</td>
<td>LCK MOD</td>
</tr>
<tr>
<td>1 S</td>
<td>MERGE_JOIN_COLS</td>
<td>J C</td>
</tr>
<tr>
<td>1 S</td>
<td>PAGE_RANGE</td>
<td>P R</td>
</tr>
<tr>
<td>1 S</td>
<td>JOIN_TYPE</td>
<td>J T</td>
</tr>
<tr>
<td>6 S</td>
<td>QBLOCK_TYPE</td>
<td>QBLOCK TYPE</td>
</tr>
<tr>
<td>8 S</td>
<td>CREATOR</td>
<td>TABLE CREATOR</td>
</tr>
<tr>
<td>8 S</td>
<td>ACCESSCREATOR</td>
<td>INDEX CREATOR</td>
</tr>
<tr>
<td>2 S</td>
<td>TABLE_TYPE</td>
<td>TB TY</td>
</tr>
<tr>
<td>2 S</td>
<td>ACCESS_DEGREE</td>
<td>AC DG</td>
</tr>
<tr>
<td>2 S</td>
<td>ACCESS_PGROUP_ID</td>
<td>AC PG</td>
</tr>
<tr>
<td>1 S</td>
<td>JOIN_DEGREE</td>
<td>J D</td>
</tr>
<tr>
<td>2 S</td>
<td>JOIN_PGROUP_ID</td>
<td>JN PG</td>
</tr>
<tr>
<td>2 S</td>
<td>SORTC_PGROUP_ID</td>
<td>SC PG</td>
</tr>
<tr>
<td>2 S</td>
<td>SORTN_PGROUP_ID</td>
<td>SN PG</td>
</tr>
<tr>
<td>1 S</td>
<td>PARALLELISM_MODE</td>
<td>P M</td>
</tr>
<tr>
<td>4 S</td>
<td>CORRELATION_NAME</td>
<td>CORR NAME</td>
</tr>
<tr>
<td>6 S</td>
<td>GROUP_MEMBER</td>
<td>GROUP MEMBER</td>
</tr>
<tr>
<td>1 S</td>
<td>WHEN_OPTIMIZE</td>
<td>W O</td>
</tr>
<tr>
<td>26 H</td>
<td>BIND_TIME</td>
<td>EXPLAIN TIME</td>
</tr>
<tr>
<td>4 S</td>
<td>OPTHINT</td>
<td>OPT HINT</td>
</tr>
<tr>
<td>4 S</td>
<td>HINT_USED</td>
<td>HINT USED</td>
</tr>
<tr>
<td>1 S</td>
<td>PRIMARY_ACCESSSTYPE</td>
<td>P A</td>
</tr>
<tr>
<td>2 S</td>
<td>PARENT_QBLOCKNO</td>
<td>PA QB</td>
</tr>
</tbody>
</table>

* THIS IS INFORMATION FOR A DB2 TABLE.

*WIDTH SHOW   COL_NAME | COLUMN HEADING
8 H   CREATOR | TABLE CREATOR
18 S   NAME | TABLE NAME
8 H   DBNAME | DBNAME
8 H   TSNAM | TSNAM
8 H   EDPROC | EDIT PROC
3 S   PCTPAGES | PCT PAGES
3 S   PCTROWCOMP | PCT COMP
9 S   CARDF | CARDF
9 S   NPAGESF | NPAGESF
1 S   ENCODING_SCHEMA | E S
1 S   LOCKRULE | L R
4 S   FREEPAGE | FREE PAGE
3 S   PCTFREE | PCT FREE
8 S   BPOOL | BUFFER POOL
1 S   COMPRESS | C
3 S   SEGSIZE | SEG SIZE
4 S   PARTITIONS | PARTS
9 S   NACTIVEF | NACTIVEF
26 S   STATSTIME | STATSTIME
5 S   PCTUPD | % UPDATE
### YCOLS

* This is information for a column in a table.

<table>
<thead>
<tr>
<th>*WIDTH SHOW</th>
<th>COL_NAME</th>
<th>COLUMN HEADING</th>
</tr>
</thead>
<tbody>
<tr>
<td>18 S</td>
<td>NAME</td>
<td>COLUMN NAME</td>
</tr>
<tr>
<td>3 S</td>
<td>COLNO</td>
<td>COL NUM</td>
</tr>
<tr>
<td>5 S</td>
<td>FETCH</td>
<td>FETCH</td>
</tr>
<tr>
<td>5 S</td>
<td>UPDATE</td>
<td>UPDAT</td>
</tr>
<tr>
<td>5 S</td>
<td>GROUPBY</td>
<td>GRPBY</td>
</tr>
<tr>
<td>5 S</td>
<td>ORDERBY1</td>
<td>SORT1</td>
</tr>
<tr>
<td>5 S</td>
<td>ORDERBY2</td>
<td>SORT2</td>
</tr>
<tr>
<td>5 S</td>
<td>ORDERBY3</td>
<td>SORT3</td>
</tr>
<tr>
<td>5 S</td>
<td>EQUAL</td>
<td>EQUAL</td>
</tr>
<tr>
<td>5 S</td>
<td>RANGE</td>
<td>RANGE</td>
</tr>
<tr>
<td>5 S</td>
<td>OTHER</td>
<td>OTHER</td>
</tr>
<tr>
<td>5 S</td>
<td>STAGE1</td>
<td>STAG1</td>
</tr>
<tr>
<td>5 S</td>
<td>INDEXABLE</td>
<td>IXABL</td>
</tr>
<tr>
<td>8 S</td>
<td>COLTYPE</td>
<td>COLUMN TYPE</td>
</tr>
<tr>
<td>5 S</td>
<td>LENGTH</td>
<td>COL LEN</td>
</tr>
<tr>
<td>3 S</td>
<td>SCALE</td>
<td>SCALE</td>
</tr>
<tr>
<td>1 S</td>
<td>NULLS</td>
<td>N</td>
</tr>
<tr>
<td>18 S</td>
<td>NAME</td>
<td>COLUMN NAME</td>
</tr>
<tr>
<td>3 S</td>
<td>COLNO</td>
<td>COL NUM</td>
</tr>
<tr>
<td>26 H</td>
<td>STATSTIME</td>
<td>STATSTIME</td>
</tr>
<tr>
<td>16 S</td>
<td>HIGH2KEY</td>
<td>HIGH2KEY</td>
</tr>
<tr>
<td>16 S</td>
<td>LOW2KEY</td>
<td>LOW2KEY</td>
</tr>
<tr>
<td>9 S</td>
<td>COLCARDF</td>
<td>COLCARDF</td>
</tr>
</tbody>
</table>

### YINDEX

* This is information for an index on a table.

<table>
<thead>
<tr>
<th>*WIDTH SHOW</th>
<th>COL_NAME</th>
<th>COLUMN HEADING</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 H</td>
<td>CREATOR</td>
<td>INDEX CREATOR</td>
</tr>
<tr>
<td>18 S</td>
<td>NAME</td>
<td>INDEX NAME</td>
</tr>
<tr>
<td>8 S</td>
<td>SAVINGS</td>
<td>SAVINGS</td>
</tr>
<tr>
<td>6 S</td>
<td>NUMSTMTS</td>
<td>#STMTS</td>
</tr>
<tr>
<td>9 S</td>
<td>RECOMMEND</td>
<td>RECOMMEND</td>
</tr>
<tr>
<td>1 S</td>
<td>UNIQUERULE</td>
<td>U R</td>
</tr>
<tr>
<td>1 S</td>
<td>CLUSTERING</td>
<td>C G</td>
</tr>
<tr>
<td>1 S</td>
<td>CLUSTERED</td>
<td>C D</td>
</tr>
<tr>
<td>9 S</td>
<td>NLEAF</td>
<td>NLEAF</td>
</tr>
<tr>
<td>4 S</td>
<td>NLEVELS</td>
<td>NLVL</td>
</tr>
<tr>
<td>8 S</td>
<td>BPOOL</td>
<td>BUFFER POOL</td>
</tr>
<tr>
<td>9 S</td>
<td>FIRSTKEYCARDF</td>
<td>FIRST KEYCARD</td>
</tr>
<tr>
<td>9 S</td>
<td>FULLKEYCARDF</td>
<td>FULL KEYCARD</td>
</tr>
<tr>
<td>9 S</td>
<td>CLUSTERRATIOF</td>
<td>CLUSTER RATIO</td>
</tr>
<tr>
<td>26 H</td>
<td>STATSTIME</td>
<td>STATSTIME</td>
</tr>
</tbody>
</table>

### YDIST

* This is information for column distribution statistics.

<table>
<thead>
<tr>
<th>*WIDTH SHOW</th>
<th>COL_NAME</th>
<th>COLUMN HEADING</th>
</tr>
</thead>
<tbody>
<tr>
<td>12 S</td>
<td>NAME</td>
<td>COLUMN NAME</td>
</tr>
<tr>
<td>9 S</td>
<td>FREQUENCYF</td>
<td>FREQUENCY</td>
</tr>
<tr>
<td>32 S</td>
<td>COLVALUE</td>
<td>COLUMN VALUE</td>
</tr>
<tr>
<td>1 S</td>
<td>TYPE</td>
<td>T</td>
</tr>
<tr>
<td>9 S</td>
<td>CARDF</td>
<td>CARDF</td>
</tr>
<tr>
<td>2 H</td>
<td>NUMCOLUMNS</td>
<td>NM CO</td>
</tr>
<tr>
<td>16 S</td>
<td>COLGROUPCOLNO</td>
<td>COLUMN GROUP</td>
</tr>
<tr>
<td>3 H</td>
<td>QUANTILENO</td>
<td>QNO</td>
</tr>
</tbody>
</table>
Appendix B Workload Access Path Compare and Workload Index Advisor reference 239

Return codes for Workload Access Path Compare and Index Advisor

This section describes the return codes that are generated for these components.

Workload Access Path Compare return codes

The return code alerts you when certain events occur during the execution of a Workload Access Path Compare job.

Table 34 on page 240 lists the return codes for the Workload Access Path Compare function and gives information about the events that cause Workload Access Path Compare to issue a particular return code.
Table 34  Return codes for the Workload Access Path Compare function

<table>
<thead>
<tr>
<th>Condition</th>
<th>Return code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Everything matched and no differences existed, so there is nothing to</td>
<td>0</td>
</tr>
<tr>
<td>report.</td>
<td></td>
</tr>
<tr>
<td>This return code occurs under the following conditions:</td>
<td>1</td>
</tr>
<tr>
<td>■ Nothing matched, so nothing could be compared.</td>
<td></td>
</tr>
<tr>
<td>■ Nothing printed in the Detail report (when the workload contains</td>
<td></td>
</tr>
<tr>
<td>archived trace data).</td>
<td></td>
</tr>
<tr>
<td>A change in access path, indexes, stats, or text occurred, but the cost</td>
<td>2</td>
</tr>
<tr>
<td>did not go up so the change was not adverse</td>
<td></td>
</tr>
<tr>
<td>■ Added or deleted statements were printed in the Summary report.</td>
<td></td>
</tr>
<tr>
<td>A change in an access path occurred and the cost went up, so the change</td>
<td>4</td>
</tr>
<tr>
<td>is adverse.</td>
<td></td>
</tr>
</tbody>
</table>

Index Advisor return codes

The return code alerts you when certain events occur during the execution of an Index Advisor job.

Table 34 lists the return codes for the Index Advisor function and gives information about the events that cause Index Advisor to issue a particular return code.

Table 35  Return codes for the Index Advisor function

<table>
<thead>
<tr>
<th>Condition</th>
<th>Return code</th>
</tr>
</thead>
<tbody>
<tr>
<td>The function ran successfully.</td>
<td>0</td>
</tr>
<tr>
<td>A problem occurred with a related negative SQL</td>
<td>8</td>
</tr>
<tr>
<td>code.</td>
<td></td>
</tr>
<tr>
<td>A security violation occurred.</td>
<td>&gt; 8</td>
</tr>
</tbody>
</table>

Working tables used by Index Advisor

The Index Advisor component uses the following tables as working tables when analyzing indexes:

- DSN_SORT_TABLE
- DSN_SORTKEY_TABLE
- DSN_VIRTUAL_INDEXES
Table 36 shows the DSN_SORT_TABLE.

<table>
<thead>
<tr>
<th>Column name</th>
<th>From DB2 catalog table</th>
<th>NULL?</th>
<th>Type</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>QUERYNO</td>
<td>PLAN_TABLE</td>
<td>NOT NULL</td>
<td>INTEGER</td>
<td>4</td>
</tr>
<tr>
<td>QBLOCKNO</td>
<td>PLAN_TABLE</td>
<td>NOT NULL</td>
<td>SMALLINT</td>
<td>2</td>
</tr>
<tr>
<td>PLANNO</td>
<td>not applicable</td>
<td>NOT NULL</td>
<td>SMALLINT</td>
<td>2</td>
</tr>
<tr>
<td>APPLNAME</td>
<td>PLAN_TABLE</td>
<td>NOT NULL</td>
<td>CHAR</td>
<td>8</td>
</tr>
<tr>
<td>PROGNAME</td>
<td>PLAN_TABLE</td>
<td>NOT NULL</td>
<td>VARCHAR</td>
<td>128</td>
</tr>
<tr>
<td>COLLID</td>
<td>not applicable</td>
<td>NOT NULL</td>
<td>VARCHAR</td>
<td>128</td>
</tr>
<tr>
<td>SORTC</td>
<td>not applicable</td>
<td>NOT NULL</td>
<td>CHAR</td>
<td>5</td>
</tr>
<tr>
<td>SORTN</td>
<td>not applicable</td>
<td>NOT NULL</td>
<td>CHAR</td>
<td>5</td>
</tr>
<tr>
<td>SORTNO</td>
<td>not applicable</td>
<td>NOT NULL</td>
<td>SMALLINT</td>
<td>2</td>
</tr>
<tr>
<td>KEYSIZE</td>
<td>not applicable</td>
<td>NOT NULL</td>
<td>SMALLINT</td>
<td>2</td>
</tr>
<tr>
<td>ORDERCLASS</td>
<td>not applicable</td>
<td>NOT NULL</td>
<td>INTEGER</td>
<td>4</td>
</tr>
<tr>
<td>EXPLAIN_TIME</td>
<td>not applicable</td>
<td>NOT NULL</td>
<td>TIMESTMP</td>
<td>10</td>
</tr>
</tbody>
</table>

Table 37 shows the DSN_SORTKEY_TABLE.

<table>
<thead>
<tr>
<th>Column name</th>
<th>From DB2 catalog table</th>
<th>NULL?</th>
<th>Type</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>QUERYNO</td>
<td>PLAN_TABLE</td>
<td>NOT NULL</td>
<td>INTEGER</td>
<td>4</td>
</tr>
<tr>
<td>QBLOCKNO</td>
<td>PLAN_TABLE</td>
<td>NOT NULL</td>
<td>SMALLINT</td>
<td>2</td>
</tr>
<tr>
<td>PLANNO</td>
<td>PLAN_TABLE</td>
<td>NOT NULL</td>
<td>SMALLINT</td>
<td>2</td>
</tr>
<tr>
<td>APPLNAME</td>
<td>PLAN_TABLE</td>
<td>NOT NULL</td>
<td>CHAR</td>
<td>8</td>
</tr>
<tr>
<td>PROGNAME</td>
<td>PLAN_TABLE</td>
<td>NOT NULL</td>
<td>VARCHAR</td>
<td>128</td>
</tr>
<tr>
<td>COLLID</td>
<td>PLAN_TABLE</td>
<td>NOT NULL</td>
<td>VARCHAR</td>
<td>128</td>
</tr>
<tr>
<td>SORTNO</td>
<td>not applicable</td>
<td>NOT NULL</td>
<td>SMALLINT</td>
<td>2</td>
</tr>
<tr>
<td>ORDERNO</td>
<td>not applicable</td>
<td>NOT NULL</td>
<td>SMALLINT</td>
<td>2</td>
</tr>
<tr>
<td>EXPTYPE</td>
<td>not applicable</td>
<td>NOT NULL</td>
<td>CHAR</td>
<td>3</td>
</tr>
<tr>
<td>TEXT</td>
<td>not applicable</td>
<td>NOT NULL</td>
<td>VARCHAR</td>
<td>128</td>
</tr>
<tr>
<td>TABNO</td>
<td>PLAN_TABLE</td>
<td>NOT NULL</td>
<td>SMALLINT</td>
<td>2</td>
</tr>
<tr>
<td>COLNO</td>
<td>not applicable</td>
<td>NOT NULL</td>
<td>SMALLINT</td>
<td>2</td>
</tr>
<tr>
<td>DATATYPE</td>
<td>not applicable</td>
<td>NOT NULL</td>
<td>CHAR</td>
<td>18</td>
</tr>
<tr>
<td>LENGTH</td>
<td>not applicable</td>
<td>NOT NULL</td>
<td>INTEGER</td>
<td>4</td>
</tr>
<tr>
<td>CCSID</td>
<td>not applicable</td>
<td>NOT NULL</td>
<td>INTEGER</td>
<td>4</td>
</tr>
<tr>
<td>ORDERCLASS</td>
<td>not applicable</td>
<td>NOT NULL</td>
<td>INTEGER</td>
<td>4</td>
</tr>
<tr>
<td>EXPLAIN_TIME</td>
<td>not applicable</td>
<td>NOT NULL</td>
<td>TIMESTMP</td>
<td>10</td>
</tr>
</tbody>
</table>
Table 38 shows the DSN_VIRTUAL_INDEXES table.

**Table 38  DSN_VIRTUAL_INDEXES**

<table>
<thead>
<tr>
<th>Column name</th>
<th>From DB2 catalog table</th>
<th>NULL?</th>
<th>Type</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBCREATOR</td>
<td>not applicable</td>
<td>NOT NULL</td>
<td>VARCHAR</td>
<td>128</td>
</tr>
<tr>
<td>TBNAME</td>
<td>not applicable</td>
<td>NOT NULL</td>
<td>VARCHAR</td>
<td>128</td>
</tr>
<tr>
<td>IXCREATOR</td>
<td>SYSIBM.SYSINDEXES</td>
<td>NOT NULL</td>
<td>VARCHAR</td>
<td>128</td>
</tr>
<tr>
<td>IXNAME</td>
<td>SYSIBM.SYSINDEXES</td>
<td>NOT NULL</td>
<td>VARCHAR</td>
<td>128</td>
</tr>
<tr>
<td>ENABLE</td>
<td>not applicable</td>
<td>NOT NULL</td>
<td>CHAR</td>
<td>1</td>
</tr>
<tr>
<td>MODE</td>
<td>not applicable</td>
<td>NOT NULL</td>
<td>CHAR</td>
<td>1</td>
</tr>
<tr>
<td>UNIQUERULE</td>
<td>SYSIBM.SYSINDEXES</td>
<td>NOT NULL</td>
<td>CHAR</td>
<td>1</td>
</tr>
<tr>
<td>COLCOUNT</td>
<td>not applicable</td>
<td>NOT NULL</td>
<td>SMALLINT</td>
<td>2</td>
</tr>
<tr>
<td>CLUSTERING</td>
<td>SYSIBM.SYSINDEXES</td>
<td>NOT NULL</td>
<td>CHAR</td>
<td>1</td>
</tr>
<tr>
<td>NLEAF</td>
<td>SYSIBM.SYSINDEXES</td>
<td>NOT NULL</td>
<td>INTEGER</td>
<td>4</td>
</tr>
<tr>
<td>NLEVELS</td>
<td>SYSIBM.SYSINDEXES</td>
<td>NOT NULL</td>
<td>SMALLINT</td>
<td>2</td>
</tr>
<tr>
<td>INDEXTYPE</td>
<td>not applicable</td>
<td>NOT NULL</td>
<td>CHAR</td>
<td>1</td>
</tr>
<tr>
<td>PGSIZE</td>
<td>not applicable</td>
<td>NOT NULL</td>
<td>SMALLINT</td>
<td>2</td>
</tr>
<tr>
<td>FIRSTKEYCARDF</td>
<td>SYSIBM.SYSINDEXES</td>
<td>NOT NULL</td>
<td>FLOAT</td>
<td>8</td>
</tr>
<tr>
<td>FULLKEYCARDF</td>
<td>not applicable</td>
<td>NOT NULL</td>
<td>FLOAT</td>
<td>8</td>
</tr>
<tr>
<td>CLUSTERRATIOF</td>
<td>SYSIBM.SYSINDEXES</td>
<td>NOT NULL</td>
<td>FLOAT</td>
<td>8</td>
</tr>
<tr>
<td>PADDED</td>
<td>not applicable</td>
<td>NOT NULL</td>
<td>CHAR</td>
<td>1</td>
</tr>
<tr>
<td>COLNO(^a)</td>
<td>not applicable</td>
<td>NULL</td>
<td>SMALLINT</td>
<td>2</td>
</tr>
<tr>
<td>ORDERING(^a)</td>
<td>not applicable</td>
<td>NULL</td>
<td>CHAR</td>
<td>1</td>
</tr>
</tbody>
</table>

\(^a\) The table includes 64 columns of this type.
Index Component report list

This appendix lists all of the reports that are shipped with the Index Component of SQL Performance. A similar list is available in online Help (type HELP REPORTS on the Command line of any panel).

Table 39  IFCIDS used by SQL Performance reports

<table>
<thead>
<tr>
<th>Report</th>
<th>Description</th>
<th>IFCID 5</th>
<th>IFCID 8</th>
<th>IFCID 9</th>
<th>IFCID 316</th>
<th>IFCID 322</th>
<th>IFCID 323</th>
<th>IFCID 325</th>
</tr>
</thead>
<tbody>
<tr>
<td>IODDSTAT</td>
<td>Subsystem Getpage Volume</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IODIDPKG</td>
<td>Index Dependencies (Package)</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IODIDPND</td>
<td>Index Dependencies (Plan)</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IODIGETV</td>
<td>Index Getpage Volume</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IODISTAT</td>
<td>Index Statistics</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IODITABL</td>
<td>Table Information for an Index</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IODSTMTS</td>
<td>Statement List</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IODSTXDH</td>
<td>Statement Text</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>IODSTXDT</td>
<td>Table/Index Breakdown</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IODSTXTH</td>
<td>Statement Text</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>IODTAPPL</td>
<td>Application Group Getpage Volume</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IODTCRUD</td>
<td>Table CRUD Matrix</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>IODTGETV</td>
<td>Table Getpage Volume</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IODTSTAT</td>
<td>Table Statistics</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>IODTSTIN</td>
<td>Defined Indexes for a Table</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>
Using multiple System and SQL Performance for DB2 products

The System and SQL Performance for DB2 products from BMC Software are a family of tightly integrated performance monitoring and management tools for DB2. Most of the products share the same dialog architecture, enabling data sharing and exceptional ease of use across products. The following products can share the same started task and can be accessed through a common dialog:

- **APPTUNE for DB2** — A tool for tuning SQL statements and troubleshooting DB2 application performance.

- **Pool Advisor for DB2** — A tool for monitoring DB2 pools, identifying inefficient use, and suggesting and implementing modifications to gain maximum benefit from storage resources.

- **SQL Explorer for DB2** — A tool for analyzing SQL statements and database structures to optimize application performance.

- **MainView for DB2 – Data Collector** — A selectable component of the MainView for DB2 product that allows MainView users to share some functions of the Performance Activity products Data Collector.

  This component provides access to DB2 performance data that supplements the data already available via MainView for DB2.

- **System Performance for DB2** — A solution that combines the features and functions of MainView for DB2, OPERTUNE, and Pool Advisor with a supplemental set of comprehensive reports on all aspects of DB2.

- **SQL Performance for DB2** — A solution that combines the features and functions of the APPTUNE for DB2 and SQL Explorer for DB2 components with a comprehensive index reporting function that is available only with the solution.
When a single product is enabled, the main menu for that product is displayed. When multiple products or solutions are enabled, the main menu that is displayed reflects the active product mix. Figure 106 is an example of the main menu that is displayed when all System and SQL Performance for DB2 products are enabled.

**Figure 106  System and SQL Performance for DB2 main menu (DOMESELT)**

When multiple products share a product session, functions that are shared by all products are shown on the initial main menu. Select a product from the main menu to display the main menu for that product. Functions that are specific to each product are accessed through the product’s main menu.

All products share the same Data Collector and the same trace data sets. If you select a different Data Collector in any product, the Data Collector is switched in all products.

**NOTE**

Filters set in APPTUNE have no effect on Pool Advisor or System Performance reports because those products do not support filtering.
Glossary

The commands that are listed in this glossary apply to both SQL Performance for DB2 and to APPTUNE for DB2. The commands are described in detail in the *APPTUNE for DB2 User Guide*.

A

**ADDDB2 command**
A Data Collector command that temporarily adds a DB2 subsystem to the list of DB2s that can be monitored by a Data Collector.

**application group**
A logical collection of plans, programs, and users that represent the workload associated with a particular business function or individual.

**application profile**
A collection of one or more application groups.

**APPOFF command**
A Data Collector command used to stop APPTUNE data collection and free data reduction storage.

**APPON command**
A Data Collector command used to start data collection for APPTUNE after the Data Collector is started (when APPTUNE data collection is not started automatically) or after data collection has been stopped by an APPOFF command.

**APPROF statement/parameter**
A control statement (or parameter of the REPORT statement) used to specify the application group that will be used for group reporting when one of the APPTUNE group reports is specified or the APGRP qualifier is used.

**APPRESET command**
An APPTUNE Data Collector command used to write the data in the reduction table to the trace data set.

**APPSTAT command**
An APPTUNE Data Collector command used to display the current status of the APPTUNE intercept.
APPTUNE
A BMC Software product used to analyze DB2 application performance (a component of SQL Performance for DB2).

archive directory
A data set that contains a directory of archived data sets, used to select the data sets to use for batch reporting.

auto-refresh
A feature that causes the values displayed for DB2 instrumentation data to be updated automatically at an interval specified by the user.

B

batch reporting
The process used to produce printed reports from instantaneous or historical data. All reports can be processed in batch (APPTUNE and SQL Performance only).

batch reports
Reports designed in a format suitable for printing. These reports can also be displayed online.

BEEP command
A basic panel command that alternately disables and enables the terminal bell that rings whenever a message is displayed and when an exception condition is displayed.

C

CANCEL command
There are two versions of the CANCEL command:

- **User Session**: Causes the last panel displayed before the current panel to be redisplayed, discarding any data entry in the process.

- **Data Collector**: Cancels a product user session.

CAPS command
A basic panel command that alternately changes the text displayed on all panels to uppercase or mixed case.

CHANGES command
A basic panel command used to display the current Summary of Changes.

CLEAR command
A panel-specific command that erases the contents of the command text buffer on the Command Interface input panels.
CMD command
A basic panel command that displays the Command Interface Menu, from which you can choose one of the Command Interface input panels to issue commands to the Data Collector, MVS, DB2 and other BMC products. Synonymous with the CMDMENU command.

CMDBOT command
A basic panel command that causes the Command line to be displayed at the bottom of all product panels.

CMDDDB2 command
A basic panel command that displays the DB2 Command Interface input panel, from which you can issue commands to DB2.

CMDDDC command
A basic panel command that displays the Data Collector Command Interface input panel, from which you can issue commands to the Data Collector.

CMDMENU command
A basic panel command that displays the Command Interface Menu, from which you can choose one of the Command Interface input panels to issue commands to the Data Collector, MVS, DB2 and other BMC products. Synonymous with the CMD command.

CMDOPT command
A basic panel command that displays the OPERTUNE Command Interface input panel, from which you can issue commands to OPERTUNE.

CMDTOP command
A basic panel command that causes the Command line to be displayed at the top of all product panels.

COLOR command
A basic panel command that alternately enables and disables the color feature used to build product panels in color on terminals using native TSO without ISPF.

Command Interfaces
A set of panels used to issue and view the results of Data Collector, DB2, MVS, and OPERTUNE commands and to receive DB2 messages.

D
data class
A collection of DB2 or BMC trace records (IFCIDs) that can be specified in an output group. Only the data classes specified will be collected and stored in the trace data sets for that output group.
Data Collector
The component that coordinates requests for data from all product and solution users and retrieves data from DB2.

DATASOURCE statement
A control statement used to select the source of data for batch reporting (data set or Data Collector).

DB2LOAD statement/parameter
A control statement (or parameter of the REPORT statement) used to produce a data set (from the data gathered to generate a batch report) in a format that can be loaded into DB2 tables using the DB2 Load utility.

DBCS command
A basic panel command that alternately enables and disables DBCS support when you are using native TSO without ISPF.

DEBUG command
A Data Collector command that enables or disables various diagnostic functions useful in problem determination.

DELDB2 command
A Data Collector command that dynamically deletes a DB2 subsystem from the list of DB2s that can be monitored by a Data Collector.

DELPROD command
A Data Collector command that dynamically deactivates an active System and SQL Performance product.

DOMBLOD1
The utility that creates the appropriate CREATE TABLE DDL and LOAD utility control statements for the DB2LOAD data set written by the DOMBRPT1 utility.

DOMBPLOG
The utility that prints all entries from a report log data set.

DOMBRPT1
The utility that produces printed reports from a batch job.

DOMBSWIT
The utility that invokes the Data Collector SWITCH command from a batch job to switch trace recording to the next available VSAM trace data set.

DOMPLEX option set
A collection of attributes that define one or more Data Collectors and their components (for example, the DB2s that can be monitored and the trace data sets that are used).
DOWN command
A basic panel command that moves the display of scrollable lists forward in the direction of the bottom of the list. Synonymous with the FORWARD command.

DUMP command
A Data Collector command that enables or disables the diagnostic dump that occurs at abend retry, or takes a diagnostic dump of a specific job.

END command
A basic panel command that causes the last panel displayed before the current panel to be redisplayed, saving or committing any data entry in the process. Synonymous with the EXIT command.

ENQS command
A Data Collector command that displays enqueue conflicts or all enqueues held or waited for by a specific job or a specific DB2 subsystem.

EXPAND command
A command used to move from summary data to more detailed data within a report.

EXPAND statement/parameter
A control statement (or parameter of the REPORT statement) used to print batch reports in their expanded format.

Explain
A powerful function that creates a set of unique, historical baseline snapshots containing key DB2 catalog statistics for an application.

EXPORT command
A command used to copy the selected records for the current report to an external DASD data set.

EXPORT files
Sequential data sets used to save the data collected to generate a report. Can be used as input to online or batch reporting.

F

FIND command
A basic panel command that finds a specified string of text and moves the cursor to that text.

filter option set
A set of tailored options for the collection of data based on specific programs, plans, DB2 subsystems, and user IDs (APPTUNE and SQL Performance only).
FKA command
A basic panel command that changes the display of function keys, alternating between the primary keys, alternate keys, and no display. Synonymous with the KEYS and PFSHOW commands.

FKEYS command
A basic panel command that displays the User Function Key Values panel (used to change the default values for function keys). Synonymous with the PFKEYS and PFKS commands.

FORWARD command
A basic panel command that moves the display of scrollable lists forward in the direction of the bottom of the list. Synonymous with the DOWN command.

HELP command
A basic panel command that causes informational text to be displayed, the topic of which is determined by the parameter specified, the point at which the command is issued, or by the position of the cursor.

HILITE command
A basic panel command that enables highlighting support when you are using native TSO without ISPF.

historical reports
Reports used to view data gathered from recent and/or archived trace data.

HOME command
A basic panel command that moves the cursor to the first input field on the current panel.

I
IEDIT command
A command that exports command text to an ISPF edit session (when you are operating under ISPF).

IFCID
Instrumentation Facility Component identifier. The identifier assigned to a traceable DB2 event and to the associated trace record produced by DB2. This term is also used for records created by the System and SQL Performance products. IFCIDs generated by DB2 are preceded by “DB2” and records generated by the products are preceded by “BMC.”

IN-DB2 time
The time interval used to measure the execution time of an entire SQL statement. IN-DB2 time includes the time spent on associated housekeeping tasks. See also IN-SQL time.
IN-SQL time
   The time interval used by APPTUNE to measure the execution time of an SQL statement. IN-SQL measurement excludes the time spent on associated housekeeping tasks. See also IN-DB2 time.

instantaneous reports
   Reports used to view data obtained instantaneously from DB2.

INTERVAL statement/parameter
   A control statement (or a parameter of the REPORT statement) used to specify a time interval that applies to the data in batch reports.

K
KEYS command
   A basic panel command that changes the display of function keys, alternating between the primary keys, alternate keys, and no display. Synonymous with the FKA and PFSHOW commands.

L
LEFT command
   A basic panel command that shifts the display of data to the left when a wide-mode panel or report (132 columns) is displayed on a screen with a width of 80 columns.

LINESPP statement/parameter
   A control statement (or a parameter of the REPORT statement) used to specify the number of lines to print on each page.

LOCATE command
   A basic panel command that finds a specified list item and moves the row containing that item to the top of the display area.

M
MainView for DB2
   A BMC Software product used to monitor DB2 activity in real time and historically.

MODIFY command
   A Data Collector command that displays or modifies the product timeout limit and the APPTUNE collection interval.

N
NEXT command
   A command that retrieves and displays report data in segments when the data collected for a report is too large to fit in the report output buffer.
online reporting
The process used to view DB2 data on the screen. Current DB2 activity can be monitored while the system is operating, or historical data can be viewed that has been stored in the trace data sets.

OPERTUNE for DB2
A BMC Software product used to dynamically modify DB2 installation parameters. Pool Advisor and System Performance have an interface to OPERTUNE that allows users to issue commands to OPERTUNE either via the OPERTUNE Command Interface panel or as a result of an advisor request.

OPTIONS command
A basic panel command that displays the User Options Menu (offering a selection of panels used to modify the User Profile).

OUTLIM statement/parameter
A control statement (or parameter of the REPORT statement) used to specify the number of report groups to include in reports.

output group
The product component used to buffer trace records and to define and allocate the trace data sets to which records will be written from the output groups.

PANELID command
A basic panel command that alternately displays or suppresses display of the panel ID in the upper left corner of all product panels.

PFKEYS command
A basic panel command that displays the User Function Key Values panel (used to change the default values for function keys). Synonymous with the FKEYS and PFKS commands.

PFKS command
A basic panel command that displays the User Function Key Values panel (used to change the default values for function keys). Synonymous with the FKEYS and PFKEYS commands.

PFSHOW command
A basic panel command that changes the display of function keys, alternating between the primary keys, alternate keys, and no display. Synonymous with the KEYS and FKA commands.

Pool Advisor for DB2
A BMC Software product used to monitor and manage DB2 storage resources.
PRODUCTS command
A Data Collector command that displays a list of all the currently active BMC Software Performance products for DB2, which comprises the following products:

- Pool Advisor
- APPTUNE
- OPERTUNE
- System Performance for DB2
- SQL Performance for DB2
- MainView for DB2 - Data Collector

profile
See User Profile and application profile.

QUALIFIER statement/parameter
A control statement (or parameter of the REPORT statement) used to filter the data included in batch reports by identifier values.

QUIT command
A basic panel command that causes a quick, complete exit from the product in a manner equivalent to repeated executions of the CANCEL command.

report
A panel or printed output used to display DB2 data.

report log
A data set used to store images of screens and reports produced using the LOG command.

report logging
The process used to send a copy of a report or screen image to the user’s report log data set for subsequent viewing or printing.

report set
A comprehensive set of predefined reports that comes with all System and SQL Performance products.

REPORT statement
The control statement used to produce batch reports.

REPORT_DD statement/parameter
A control statement (or parameter of the REPORT statement) used to designate a default ddname to which report output should be directed.
RESET command
A panel-specific command that resets the default values for function keys to the values shipped with this product or resets the contents of the command text on the Command Interface panels to its initial value.

RESTART command
A command issued from only the product’s main menu that reprocesses the startup options in the User Profile as if an initial entry to the product was in progress.

RETRIEVE command
A basic panel command that redisplay the last command issued on the Command line (excluding navigational commands like EXIT).

RETURN command
A basic panel command that causes a series of EXIT commands to be issued until the product’s main menu is displayed.

RFIND command
A basic panel command that reissues the previous FIND command (including the direction).

RIGHT command
A basic panel command that shifts the display of data to the right when a wide-mode panel or report (132 columns) is displayed on a screen with a width of 80 columns.

SHOWCMDS command
A basic panel command used to display the Available Commands panel, which lists all commands that can be issued from the panel.

SHUTDOWN command
A Data Collector command used to stop the Data Collector subsystem in a normal manner.

SORT command
A basic panel command that rearranges the order of data on scrollable panels or the order of repeating groups of data in reports.

SORT/SORT2 statements/parameters
The control statements (or parameters of the REPORT statement) used to sort the data included in batch reports.

SPFOFF command
A basic panel command that disables the use of ISPF as the dialog display mechanism when it has been temporarily enabled using the SPFON command.
SPFON command
A basic panel command that enables the use of ISPF as the dialog display mechanism (SPFOFF is the default when you begin your session).

SQL
Structured Query Language. A language that can be used within programs and interactively to request information from a DB2 subsystem.

SQL Explorer for DB2
The SQL Explorer for DB2 product is an SQL analysis tool that enables you to solve performance problems that result from inefficient SQL statements (component of SQL Performance for DB2).

SQL Performance for DB2
A BMC Software solution that combines the features and functions of APPTUNE and SQL Explorer with additional index capabilities.

STATUS command
There are two versions of the STATUS command:

- **Report Manager:** When issued from a report, STATUS displays the Report Status panel. When issued from any other product panel, STATUS displays the Session Status panel.

- **Data Collector:** Displays status information about the Data Collector subsystem and its associated DB2 subsystems.

STATUSR command
A report-specific command that displays the Report Status panel containing a summary of information about the current report.

STATUSS command
A basic panel command that displays the Session Status panel containing a summary of information about the current user session.

STOP command
There are two versions of the STOP command:

- **Report Manager:** Cancels auto mode (under which values on the current panel are automatically updated and redisplayed at a specified interval).

- **Data Collector:** Stops the Data Collector subsystem in a normal manner.

STORAGE command
A Data Collector command that displays details of storage usage for a specific job or task on the system.
SWITCH command
 A Data Collector command that generates an archive from the currently active log file for the specified output group.

SYSPLEX command
 A Data Collector command that establishes or terminates sysplex communication between the local Data Collector and other members of its DOMPLEX or displays information about the status of the DOMPLEX and the DB2s it monitors.

System and SQL Performance products for DB2
 An integrated family of products that share common DB2 data collection facilities and a common interface. The System and SQL Performance products comprise the following products and solutions:

- APPTUNE
- MainView for DB2 - Data Collector
- Pool Advisor
- SQL Explorer
- SQL Performance for DB2
- System Performance for DB2

System Performance for DB2
 A BMC Software solution that combines the features and functions of MainView for DB2, Pool Advisor, and OPERTUNE for DB2 with additional reporting capabilities.

TERSE command
 A command used to suppress blank lines and nonessential data on the display screen in order to increase the area used to display report output data.

thread
 A DB2 mechanism that provides a path for an application to connect to (and perform work in) DB2 and that determines access to DB2 resources and services.

TOP command
 A command used after one or more NEXT commands to redisplay the first segment of a report.

UP command
 A basic panel command that moves the display of scrollable lists back in the direction of the top of the list. Synonymous with the BACKWARD command.

User Profile
 A collection of attributes that define a user’s access to product functions and reports.
**USERS command**
A Data Collector command that displays a list of users in session with a specified Data Collector.

**W**

**What-If Edit SQL Explain**
This process edits the SQL and a dynamic Explain is performed.

**What-If Index**
This process simulates the effects of adding or dropping an index or updating statistics for an index using cloned structures.

**Z**

**ZOOM command**
The process of moving from summary data on one report to detailed information about the same data on another report.
Index

A
access paths, explaining and comparing 195
action codes 150
Active Index With No Reads batch report 157
administration 33
APGRP qualifier 214
APPGROUP qualifier 214
application groups 153
application profiles 153
APPTUNE for DB2 component 22
architecture of SQL Performance 30
ASQEWAM1 152, 163

B
Batch Object Detail Analysis 157
batch reporting
  batch format 157
  overview 157
batch reports
  Active Index With No Reads 157
  Object Detail Analysis 157
  SQL Exceptions Analysis 157
  SQL Statement Analysis 157
  SQL Text 158
batch utilities, overview 158
BMC IFCIDs 243
BMC Software, contacting 2

C
client IDs 63
cloned objects 161, 170
collection options 34
commands
  EXPLAIN 155
  EXPORT 156
  HELP 48
  IM 187
  LOG 156
  PANELID 41
  SA 184
  SORT 153
  STATUS 156
  TERSE 44
comparing dynamic SQL
  from different periods of time 199
  from different versions of DB2 196
components of SQL Performance 22
conventions, documentation 16
COPYDIR data set 57
correlation IDs 63
customer support 3

c}

D
Data Collector
  overview 30
  status 44
data source information 164
DB2LOAD data set 208
DBRM names 63
DCB attributes, DMBRPT1 (produce printed reports) 208
DDL data set 185
DMDBMERG utility 55, 204
DMDBRTSU utility 215
documentation information 15, 16
DOMESELT panel 46, 246
DOMPLEX, group 31
DOMPLEX option set 30, 34
dynamic SQL
  comparing access paths, different periods of time 199
  comparing access paths, different versions of DB2 196

e}

E
electronic documentation 15
environment 30
epilogue time 155
examples
  browse DDL data set 185
  browse DDL data set member 188
display of cloned objects 177
dropping index in What-If Index 178
dynamic Explain on cloned objects 183
dynamic first key column statistics 175
Explain Results after index is dropped 179
Explain Results after new index is created 176
Explain Results showing cloned objects 187
examples (continued)
  Explain Results showing dropped index 179
  Explain Results showing new index 176
  Explain Results showing updated statistics 182
  list of members in DDL data set 188
  message when import process is complete 189
  navigating through reports 151
  navigation—SQL Workload Analysis Menu 151
  navigation—Statement List report 151
  PSSEXTRJ error report 197
  results of dynamic Explain on imported DDL 190
  standard report elements 43
  Update Index Access Path Statistics panel 181
  updating index statistics using What-If Index 180

  Explain Analysis report 155
  EXPLAIN command 155
  Explain processing 155
  Explain Report fields 230
  EXPORT command 156

F
  filter option set 34
  functions of SQL Performance 30

H
  header (reports) 44
  HELP command 48
  Help, online 15

I
  identifiers 151
  IFCIDs, list of 243
  IM command 187
  Index Advisor
    benefits 100
    overview 99
    process 101
    recommending an index 103, 108
    reports 114
    reviewing 110
  Index Component 25
  indexes
    identifying obsolete 51
    recommendation reports 114
    recommending 103, 108
    reviewing 110
  input parameters
    PSSCOMPJ 198, 201
    PSSEXTRJ 196, 197, 200
  IN-SQL elapsed time 153
  installation settings for Index Component 162
  interval (reports) 44
  INTERVAL statement 210
  IODADREO program 50
  IODDSTAT report 146
  IODESEL panel 162
  IODEWAL1 panel 165
  IODSTMTS report 151, 169
  IODSTXDH report 170, 171, 172
  IODTAPPL report 146
  IODTCRUD report 168
  IODGETV report 166, 168, 169
  IODTSTAT report 167
  IODSTIN report 167

L
  layout
    of panels 41
    of reports 43
  layout data set member 234
  list of, Index Component reports 243
  LOG command 156
  logging on 39
  logging reports 156

M
  main menu 40
  MainView for DB2—Data Collector 245
  measuring SQL statement execution 153

N
  navigating
    between reports 146, 151
    through the product 40
  Next Generation Logger (NGL), description 25

O
  object analysis reports 157
  Object Detail Analysis batch report 157
  object sets
    copying 71
    defining 65
    editing 68
    renaming 70
    selecting 68
    viewing 71
  online Help 15, 47
  online reports (list of) 243
  option sets, DOMPLEX 34
  options, collection 34
  overview of SQL Performance 21
package names 64
PANELID command 41
panels
  APPTUNE and Index Component Main Menu 162
  Explain Results 172, 184, 186, 187
  Import DDL 187
  Import DDL Member Selection List 189
  Save DDL 184
  Select Key Columns 174
  Specify Index Attributes 173
  SQL Workload Analysis Menu 163
  SQL Workload Initial Analysis Level 165
  System and SQL Performance for DB2 main menu 46, 246
  System and SQL Performance for DB2 Products logo 40
  Update First Key Column Statistics 175
  Update Index Access Path Statistics 181
parameters
  PSSCOMPJ 198, 201
  PSSEXTRJ 196, 197, 199
PASETPUP member 54
Performance Advisor
  benefits 50
  COPYDIR data set 57
  data collection, configuring 54
  DMDDBMERG data sets 208
  DMDDBMERG SYIN control statements 210
  DMDDBMERG utility 55, 204
  DMDDBRSTU data sets 217
  DMDDBRSTU utility 215
  IFCIDs 218
  indexes, obsolete 51
  integrating with DASD MANAGER 60
  INTERVAL statement 210
  loading data 55
  objects, identifying degraded 51
  overview 49
  PASETPUP member 54
  purging data 58
  QUALIFIER statement 213
  REORG Advisor 50
  REORG Advisor, using 24, 60
  reorganizing objects 50
  reporting on LOAD records 58
  sample members 220
  setting up 52
  statistics, loading 56
  tables 218
  tables, creating 53
  task overview 52
  utilities 203
plan name 64
Pool Advisor (using with SQL Performance) 245
product overview 21
product support 3
profiles, user 33
prologue time 154
PSSCOMPJ job 198, 201
PSSEXTRJ job 196, 197, 199
PSSLOAD job 197
PSSLOD1member 197
PSSHUNL1 member 197
PSSHUNL1D job 197
PSSMEML panel 189
PSSPW200 panel 172, 184, 186, 187
PSSPW300 panel 174
PSSPW00 panel 184
PSSPWDT0 panel 187
PSSPWED panel 175
PSSPWIXA panel 173
PSSPWIXS panel 181
PSSREPB layout data set 230, 234
publications, related 16
Q
QUALIFIER statement 213
R
R0001VFG data set 208
R0001VFL data set 208
RACF®@ATH secondary authorization exit 36
related documentation 15
related publications 16
REORG Advisor 50
report header 44
report identifiers 45
report layout 43
report logging 156
Report Manager 31
reports
  Application Group Getpage Volume 146
  Batch Object Detail Analysis 157
  DB2 Current Status 156
  Defined Indexes for a Table 167
  Explain 155
  interval 44
  Select Analysis Interval 164
  Statement List 151, 169
  Statement Text 170, 171, 172
  Subsystem Getpage Volume 146
  Table CRUD Matrix 168
  Table Getpage Volume 166, 168, 169
  Table Statistics 167
required settings 162
Runtime Component System (RTCS), description 26
S
SA command 184
saved DDL 185
saving DDL 183
scrolling options 42
security 36
Session Status 156
settings for Index Component 162
SORT command 153
sorting data on reports 153
SQL Exceptions Analysis batch report 157
SQL Explorer for DB2 component 23
SQL Performance
  architecture 30
  list of reports 243
  logo panel 40
  Main Menu 40
  overview 21
  report hierarchy 146
  using with other Performance products 245
SQL Statement Analysis batch report 157
SQL Text batch report 158
SQMINTVD 164
SRCTRACE data set 209
STATUS command 156
support, customer 3
sysplex environment 31
System and SQL Performance logo panel 40
System Performance for DB2 245

T
tasks
  adding an index using What-If Index 171
  dropping an index using What-If Index 177
  identifying What-If Index candidates 165
  importing What-If Index changes 186
  preparing to use What-If Index 162
  saving What-If Index DDL changes 183
  updating statistics using What-If Index 180
  using the What-If Index function 170
technical support 3
TERSE command 44

U
user IDs 64
User Profiles 33
using online reports 151
using SQL Performance with other products 246
Using the Index Component 161, 195
Using the What-If Index Feature 162

W
What-If Index feature 161, 165
Workload Access Path Compare component
  about 23, 61, 73, 99, 121
  capabilities 74
  compare process 77
  Comparing 87, 110
  Explaining 84, 108
  Extract sources 74
  HLQ.OBJECT.seqNumber 74
  HLQ.PATH.seqNumber 74
  HLQ.REPOS 74, 100
  HLQ.SQLTXT.seqNumber 74
  matching statements 77
  sample output 91, 114
  specifying JCL options 83, 110
  understanding the process 75, 101
Workload Index Advisor
  benefits 100
  process 101
  recommending an index 103, 108
  reports 114
  reviewing 110
Third-party product terms

The following terms apply to third-party products that are included with or in a BMC Software product as described in the BMC Software, Inc., License Agreement that is applicable to the BMC Software product.

Lucent Technologies License

Copyright © Lucent Technologies 1997

All Rights Reserved

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both the copyright notice and this permission notice and warranty disclaimer appear in supporting documentation, and that the name Lucent Technologies or any of its entities not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

LUCENT DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL LUCENT OR ANY OF ITS ENTITIES BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.