SQL Performance *for DB2* User Guide

Supporting

Version 11.1.00 of SQL Performance *for DB2*
Version 11.1.00 of SQL Explorer *for DB2*
Version 11.1.00 of APPTUNE *for DB2*

June 2013
Contacting BMC Software

Several methods are available for 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

<table>
<thead>
<tr>
<th>Address</th>
<th>Telephone</th>
<th>Fax</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC SOFTWARE INC</td>
<td>1 713 918 8800</td>
<td>1 713 918 8000</td>
</tr>
<tr>
<td>2101 CITYWEST BLVD</td>
<td>or</td>
<td></td>
</tr>
<tr>
<td>HOUSTON TX 77042-2827 USA</td>
<td>1 800 841 2031</td>
<td></td>
</tr>
</tbody>
</table>

Outside United States and Canada

<table>
<thead>
<tr>
<th>Telephone</th>
<th>Fax</th>
</tr>
</thead>
<tbody>
<tr>
<td>+01 713 918 8800</td>
<td>+01 713 918 8000</td>
</tr>
</tbody>
</table>

© Copyright 2005-2013 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.

Explorer is a trademark or registered trademark of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

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

SAP is a trademark or registered trademark of SAP AG in Germany and in several 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 UNPUBLISHED—RIGHTS RESERVED UNDER THE COPYRIGHT LAWS OF THE UNITED STATES.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

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 problems similar to yours and possible solutions
■ Order or download product documentation
■ Download products and maintenance
■ Report a problem or ask a question
■ Subscribe to receive proactive e-mail alerts
■ 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 1 800 537 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 problem
■ 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, contact Customer Support through one of the following methods:

- Send an e-mail message to customer_support@bmc.com. (In the Subject line, enter SupID:yourSupportContractID, such as SupID:12345.)
- In the United States and Canada, call 1 800 537 1813. Outside the United States and Canada, contact your local support center for assistance.
## Contents

### About this book
- Related publications ................................................................. 10
- Conventions ............................................................................. 10
- Summary of Changes ............................................................... 10

### Chapter 1 Overview of SQL Performance
- SQL Performance solution components ........................................ 15
  - APPTUNE for DB2 component ........................................ 16
  - SQL Explorer for DB2 component ..................................... 16
  - Performance Advisor component ....................................... 17
  - Workload Access Path Compare component ..................... 17
  - Workload Index Advisor component ................................ 18
  - Exception Advisor component ......................................... 18
  - REORG Advisor component ............................................ 19
  - Index component ............................................................ 19
  - Shared infrastructure components ................................... 20
- SQL Performance features ......................................................... 21
- SQL Performance for DB2 architecture ..................................... 24
  - Data Collector ..................................................................... 25
  - Report Manager ............................................................... 26
  - Sysplex support .................................................................. 26
  - Shared infrastructure architectural components ................ 27
- SQL Performance administration ............................................... 28
  - User Profiles ...................................................................... 29
  - DOMPLEX and filter option sets .......................................... 29
- SQL Performance security .......................................................... 32

### Chapter 2 Getting started with SQL Performance
- Logging on .............................................................................. 35
- Navigation ................................................................................. 36
- Panel layout ............................................................................... 37
- Online report layout ............................................................... 40
- System and SQL Performance for DB2 main menu ................ 43
- Online assistance ................................................................. 45

### Chapter 3 Managing performance with Performance Advisor
............................................................................................... 47
Why you need Performance Advisor ................................................................. 47
  Reorganizing objects .................................................................................... 48
  Eliminating obsolete indexes ........................................................................ 49
  Identifying degraded objects ......................................................................... 49
Task summary ..................................................................................................... 49
Setting up Performance Advisor ........................................................................ 50
  Creating tables ................................................................................................ 50
  Configuring automatic data collection ........................................................ 52
  Loading data into Performance Advisor PADB .............................................. 53
  Using COPYDIR .............................................................................................. 54
  Purging old data from Performance Advisor PADB ....................................... 55
REORG Advisor ................................................................................................ 57
  Integrating REORG Advisor with BMC DASD MANAGER PLUS ............... 58

Chapter 4  Working with object sets 59
Overview of object sets .................................................................................... 59
  Description of object set patterns ................................................................. 61
  Inclusion versus exclusion of objects from the workload ............................. 63
Defining an object set ......................................................................................... 64
  Selecting an object set to use ........................................................................ 67
  Viewing an object set .................................................................................... 67
  Editing an object set ..................................................................................... 68
  Renaming an object set ................................................................................ 70
  Copying an object set ................................................................................... 71
  Deleting an object set ................................................................................... 71

Chapter 5  Comparing access paths for workloads 73
Overview of workload comparisons .................................................................. 73
  Benefits ......................................................................................................... 74
  Workload Access Path Compare process ...................................................... 75
  Comparing results ......................................................................................... 78
Extracting and Explaining a workload ............................................................... 79
Explaining a workload a second time ............................................................... 84
Comparing two or more workloads ................................................................. 87
Working with workload comparison reports .................................................. 91
  Sample batch report for Workload Access Path Compare ............................. 91
  Specifying report options for workload comparisons .................................. 94
  Customizing the comparison report ............................................................. 95

Chapter 6  Recommending indexes for workloads 97
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 DB2 Universal Database Server for OS/390 and z/OS
- IBM 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 PF1 within any product. For more information, see Online assistance on page 45.
Related publications

From the BMC Support Central website, you can use the following methods to access related publications that support your product or solution:

- Link to the BMC Documentation Center (https://webapps.bmc.com/infocenter/index.jsp) to browse documentation sets.

- View BMC Quick Course Demos (short overviews of selected product concepts, tasks, or features), which are included in the BMC Documentation Center.

- Read individual product documents (books and notices) within the “A – Z Supported Product List.”

You can order hardcopy documentation from your BMC sales representative or from the support site. You can also subscribe to proactive alerts to receive e-mail messages when notices are issued.

**Tip**
You can access the BMC Support Central site at http://www.bmc.com/support.

Conventions

This document uses the following special conventions:

- **All syntax, operating system terms, and literal examples are presented in this typeface.**

- **Variable text in path names, system messages, or syntax is displayed in italic text:**
  
  `testsys/instance/fileName`

- This document uses a symbol to show menu sequences. For example, **Actions => Create Test** instructs you to choose the **Create Test** command from the **Actions** menu.

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 11.1.00—June 2013

Enhancements to version 11.1.00 of the SQL Performance for DB2 solution are listed in the following sections:

- FIND command in workload compare/index advisor—The FIND command has been added to the workload compare panels and reports. To locate a workload by searching for a text string, type FIND string on the Command line, and press Enter. Press PF5 to advance to the next instance of the specified text.

- NODATA control statement—The NODATA control statement can now override return codes. In designated cases, the NODATA control statement generates alternate values for the DMDBMERG utility's default return codes.

- DMDBMERG utility—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 input to DMDBMERG can be data retrieved directly from a Data Collector. To retrieve directly from a Data Collector, use the DATASOURCE(COLLECTOR(…)) control statement from archive files containing APPTUNE trace data

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 or tables. When LOADIN is present, the resulting LOAD control statements cause the generated load records to replace referenced table or tables.

- End of support for DB2 Version 8—Starting with this release, SQL Performance does not support IBM DB2 Version 8. Earlier releases continue to support Version 8.

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 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.
- 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.

- End of support for DB2 Version 7—Starting with this release, SQL Performance does not support IBM DB2 Version 7. Earlier releases continue to support Version 7.
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
- 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 21.

SQL Performance solution components

The SQL Performance solution includes the following components:

- APPTUNE for DB2
- SQL Explorer for DB2
- Performance Advisor component
- 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).

**Related Information**

- "Managing performance with Performance Advisor" on page 47

**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).
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).

Related Information

- “Comparing access paths for workloads” on page 73

Exception Advisor component

The Performance Advisor component 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).
REORG Advisor component

The REORG Advisor component is a REXX EXEC program, IODPAREO, 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, but do not show degraded performance since the time of the last REORG.

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).
Shared infrastructure components

SQL Performance also uses the infrastructure components.

DB2 Component Services (DBC)

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.

Next Generation Logger (NGL)

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.

Runtime Component System (RTCS)

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.

DB2 Product Configuration

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.
The SQL Performance for DB2 solution provides a variety of features and benefits for users.

Table 1 on page 21 summarizes these features.

<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>
<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>Feature</td>
<td>Benefit</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</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. The component 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>
<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>Feature</td>
<td>Benefit</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| 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 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  
  ■ BMC 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. |
<p>| 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. |
| 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. |</p>
<table>
<thead>
<tr>
<th>Feature</th>
<th>Benefit</th>
</tr>
</thead>
<tbody>
<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 can 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 on page 25 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 IBM z/OS 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 IBM DB2 or z/OS, 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 z/OS image in the sysplex can be defined in a DOMPLEX group. Each Data Collector can monitor all DB2s on its same z/OS image and all Data Collectors in the DOMPLEX can communicate with each other, making data from all of the DB2s across the DOMPLEX 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 27 illustrates sysplex communication in a SQL Performance DOMPLEX.

**Figure 2: Sysplex communication**

SQL Performance administrators have 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 architectural components

DB2 Product Configuration, NGL, and the Data Collector run within the DBC subsystem.
Figure 3 on page 28 illustrates the architecture of the shared components.

**Figure 3: Shared infrastructure component architecture**

For more information about managing these components, see the *System and SQL Performance for DB2 Administrator Guide*.

---

**Related Information**

- "Shared infrastructure components" on page 20

---

**SQL Performance administration**

SQL Performance uses profiles and option sets to control access to SQL Performance and related features in DB2.

Profiles and option sets 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.

The User Profile is also 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, data collection, summarization, and filtering are specific to DB2.

Each DB2 subsystem that can be monitored is defined in a DOMPLEX option set. The DOMPLEX option set also determines how the data collector is configured. You can modify these values to suit the needs of your site. The DOMPLEX option set allows you to specify the following operating characteristics:

■ Output groups including information on what data classes, logfile, and archive dataset names to use
■ Whether users are allowed to issue commands to DB2 and z/OS 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 are 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 that you 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 2 on page 30 lists the different collection options that you can specify through filters.

**Table 2: Collection options**

<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 are 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>Options</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</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, in order to reduce 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 data, and extrapolate the results</td>
</tr>
</tbody>
</table>
### Options

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exception thresholds and options</td>
</tr>
<tr>
<td>You can specify thresholds for the following values that cause exception records to be created:</td>
</tr>
<tr>
<td>■ Elapsed time</td>
</tr>
<tr>
<td>■ CPU time</td>
</tr>
<tr>
<td>■ Number of getpages</td>
</tr>
<tr>
<td>■ Number of synchronous I/Os</td>
</tr>
<tr>
<td>■ Deadlocks</td>
</tr>
<tr>
<td>■ Timeouts</td>
</tr>
<tr>
<td>■ Host variables</td>
</tr>
<tr>
<td>■ Exception WTOs</td>
</tr>
<tr>
<td>■ Exception-only rollups</td>
</tr>
<tr>
<td>■ Efficiency filters</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Negative SQL codes</td>
</tr>
<tr>
<td>You can choose whether to:</td>
</tr>
<tr>
<td>■ Generate exception records and messages</td>
</tr>
<tr>
<td>■ Issue WTO codes</td>
</tr>
<tr>
<td>■ Capture details for specific negative SQL codes</td>
</tr>
</tbody>
</table>

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**

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 z/OS 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 DSN3@ATH secondary authorization exit

Authorization for the following tasks is controlled through User Profile options:

- Monitoring DB2
- Accessing Data Collectors
- Issuing commands to the Data Collector, z/OS, and OPERTUNE

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

This chapter explains what you need to know in order to use the SQL Performance solution:

- “Logging on” on page 35
- “Navigation” on page 36
- “Panel layout” on page 37
- “Online report layout” on page 40
- “System and SQL Performance for DB2 main menu” on page 43
- “Online assistance” on page 45

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 36) is displayed briefly while the product initializes, followed by the System
and SQL Performance for DB2 main menu. This menu is displayed when multiple System and SQL Performance for DB2 products are installed.

Figure 4: System and SQL Performance for DB2 logo panel

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 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.
Panel layout

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 on page 37 shows the common attributes that are shared by SQL Performance panels.

Figure 5: SQL Performance panels

The following sections describe the common elements of SQL Performance panels.

Elements

Panel ID

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.

Environment identifier

The environment identifier is displayed to the right of the panel ID, separated from it by a slash (/). Possible values are

- I (when operating under ISPF)
- P (when operating under native TSO)

Panel description

Each SQL Performance panel displays a description in the center of the top line of the panel.
Time or line indicator

On nonscrollable panels, the time is displayed in the upper right corner in the format \textit{hh:mm:ss}, where:

\begin{itemize}
  \item \textit{hh} is hours in the range 00-23
  \item \textit{mm} is minutes in the range 00-59
  \item \textit{ss} is seconds in the range 00-59
\end{itemize}

On scrollable panels, the current line number and the total number of lines is displayed in the format \textit{LINE nnn OF nnn}.

If the panel is scrollable, the scroll amount (in the format \textit{Scroll ===> ____}) is also displayed at the end of the \texttt{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 remains in effect until you change it or end your session. The following values are valid:

\begin{itemize}
  \item \texttt{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.

  \item \texttt{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.

  \item \texttt{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 \texttt{PF8}. When scrolling up, place the cursor on the line to be displayed at the bottom of the scrollable area, and press \texttt{PF7}. If \texttt{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 \texttt{PAGE}.

  \item \texttt{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 \texttt{PAGE}.

  \item \texttt{n}nnn - Data is scrolled \texttt{n}nnn lines at a time, where \texttt{n}nnn is any number in the range 1 to 9999.
\end{itemize}
Command line

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.

Function keys

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:

- **PF1** - Help
- **PF2** - Split
- **PF3** - End
- **PF4** - Sort A (ascending)
- **PF5** - Sort D (descending)
- **PF6** - Showcmds
- **PF7** - Up
- **PF8** - Down
- **PF9** - Swap
- **PF10** - Left
- **PF11** - Right
- **PF12** - Cancel
- **PF13** - Home
- **PF14** - Keys
- **PF15** - End
- **PF16** - Terse
- **PF17** - Rfind
- **PF18** - Filter
- PF19 - Up
- PF20 - Down
- PF21 - Expand all
- PF22 - Left
- PF23 - Right
- PF24 - Retrieve

Message

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.

Online report layout

SQL Performance online reports have a number of common characteristics, including various standard elements that are displayed on all online reports.
Figure 6 on page 41 shows the common elements that are shared by SQL Performance online reports.

**Figure 6: Common online report elements**

![Diagram of report elements]

Table 3 on page 41 describes the common characteristics of SQL Performance online reports.

**Table 3: Report elements**

<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>Element</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</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</td>
<td>The interval of analysis. If the source is a Data Collector, the value \textit{mm/dd hh:mm} is displayed, indicating the start time of the interval, where:</td>
</tr>
<tr>
<td>Interval</td>
<td>■ (date) \textit{mm} is month in the range 01-12</td>
</tr>
<tr>
<td>start time</td>
<td>■ (date) \textit{dd} is day in the range 01-31</td>
</tr>
<tr>
<td>Interval</td>
<td>■ (time) \textit{hh} is hours in the range 00-23</td>
</tr>
<tr>
<td>stop time</td>
<td>■ (time) \textit{mm} is minutes in the range 00-59</td>
</tr>
<tr>
<td></td>
<td>\textbf{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.</td>
</tr>
<tr>
<td>Actions</td>
<td>Select action codes 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>
<tr>
<td>Element</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Report identifiers | Identifies the objects represented by the report data. As you navigate among reports, objects selected for reporting are displayed beside the appropriate report identifier. The combination of report identifiers displayed (below the Actions fields) depends on the report type selected and the data selected for reporting. SQL Performance online reports can display some or all of the following report identifiers:  
  ■ AppGrp - application group  
  ■ ConnID - connection ID  
  ■ CorrID - correlation ID  
  ■ Plan - plan  
  ■ Program - program  
  ■ Stmt No - SQL statement number  
  ■ Subsys - DB2 subsystem  
  ■ User - user ID  
  ■ ClntAp (client application)  
  ■ ClntID (client user ID)  
  ■ ClntWS (client workstation)  
  ■ WP No (work process number)  
  ■ WP ID (work process ID)  
  ■ Server (server)  |

**System and SQL Performance for DB2 main menu**

The System and SQL Performance for DB2 main menu provides access to all major System and SQL Performance product functions.
The System and SQL Performance for DB2 main menu (Figure 7 on page 44) is only displayed when you have installed both the SQL Performance and BMC System Performance for DB2 solutions.

**Figure 7: System and SQL Performance for DB2 main menu (DOMESELT)**

DOMESELT/I System and SQL Performance for DB2 18:17:00

Command ===> ________________________________________________________________

Current Data Collector : A62D    Status : ACTIVE

SELECT ONE OF THE FOLLOWING OPTIONS. THEN PRESS ENTER.

_ D. SYSTEM PERFORMANCE SOLUTION - DB2 SUBSYSTEM AND STORAGE POOL ANALYSIS
--- SQL PERFORMANCE SOLUTION ---
Q. APPTUNE AND INDEX COMPONENT DB2 APPLICATION AND INDEX ANALYSIS
S. SQL EXPLORER COMPONENT DB2 SQL ANALYSIS
A. PERFORMANCE ADVISORS ADVICE AND RECOMMENDATIONS

1. DOMPLEXES SELECT/CHANGE DOMPLEX CONNECTION
2. SESSION STATUS VIEW CURRENT SESSION RESOURCE USAGE
3. USER OPTIONS VIEW/MODIFY USER OPTIONS
4. LOG OPERATIONS VIEW/PRINT LOGGED SCREENS AND REPORTS
5. ADMINISTRATION MANAGE PRODUCT AND USER PROFILES

H. HELP
X. EXIT
Z. ABOUT THE SYSTEM AND SQL PERFORMANCE PRODUCTS

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.

- Use **Administration** to modify user profiles and option sets and to view zap maintenance.
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 PF1 (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

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.

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

Use Exit to terminate your session.

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 reflects the active product mix. Figure 7 on page 44 is an example of the main menu that is displayed when all System and SQL Performance products for DB2 are installed.
Online assistance

- 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 PF1 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

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

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

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 that the APPTUNE data shows (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 PLUS 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 on page 49 shows an example query.

**Figure 8: Example index query**

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

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, when you use 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 49 shows an example query that identifies degraded objects.

**Figure 9: Example object query**

```
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;
```

Task summary

You must perform several tasks to set up and use the Performance Advisor component.
Figure 10 on page 50 is a high-level overview of these tasks. For more information, see Performance Advisor reference on page 193.

Figure 10: Performance Advisor task overview

Setting up Performance Advisor

Complete the following tasks to prepare to use Performance Advisor:

- “Creating tables” on page 50
- “Configuring automatic data collection” on page 52
- “Loading data into Performance Advisor PADB” on page 53

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 4 on page 50 provides a list of the sample members that you can use to create each table.

Table 4: 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>SAMP member</td>
<td>Creates table(s) containing</td>
</tr>
<tr>
<td>-------------</td>
<td>------------------------------------------------------------------</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>IODDDLEV</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>IODDDLIIU</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>IODDDLR9</td>
<td>Real-time statistics accumulated daily, for DB2 Version 9 and later releases</td>
</tr>
<tr>
<td>IODDLRA</td>
<td>REORG candidates for the REORG Advisor feature</td>
</tr>
<tr>
<td>IODDLXIA</td>
<td>Rules for Exception Advisor</td>
</tr>
</tbody>
</table>

Note 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 have different procedures for aggregated and nonaggregated tables. For more information, see “Loading data into Performance Advisor PADB” on page 53 and “Purging old data from Performance Advisor PADB” on page 55.

**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 that 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 53.

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 contains 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 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 cover the period from 2 A.M. on the indicated day until 2 A.M. the following day; however, the INTVTIME column for this table shows 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 set the switch by
submitting a job that runs the DOMBSWIT utility.

---

**Loading data into Performance Advisor PADB**

This task uses the DMDBMERG utility.

**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.

For reference information about the utility, see “Performance Advisor reference” on
page 193.

**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 and parameter” on page 208.

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.

The examples in SAMP use \texttt{INTERVAL( START(-1) FOR(1) )}. This \texttt{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. The \texttt{INTERVAL} statement makes it unnecessary to code specific input data sets or a specific date-time range.

For more information, see “Using COPYDIR” on page 54.

7 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.

\textit{Tip}

Submit the jobs once daily after you have modified them for your needs.

\section*{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 newest entry replaces the oldest 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 fails. You can use the DOMARCB utility to remove uncataloged entries from the COPYDIR and reorganize the COPYDIR to improve DMDBMERG performance. For more information, see the System and SQL Performance for DB2 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 SAMP job PARTSUL 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:
   - Run PARTSUC1 once daily on each system. For data-sharing groups, run this on only one member of the group.
   - Run PARTSUC2 once daily at your centralized location, modifying SYSREC to capture all data that PARTSUC1 unloads.

**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.

**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 on page 56 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 table contents**

You can generate batch reports using DB2 LOAD utility records generated by DMDBMERG as input. See an example of this operation 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 that you can use.

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, QWHSSSID, and QWHADSGN columns to differentiate between DB2 instances.
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;
```

Example
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';
```

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

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

Example
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;
```

REORG Advisor

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

- The object’s level of disorganization
- The object’s level of performance degradation since the last REORG occurred

The advisor helps you avoid performing unnecessary reorganizations for tables and indexes that are disorganized but 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. The table’s columns reflect the calculations performed in analyzing the objects. The REORG_CANDIDATE column, containing values of Y or N, indicates REORG recommendations. You can query this table to get a list of recommendations, or you can use integration with DASD MANAGER PLUS 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 DSNSSAMP library member DSNTIJRX.

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

```
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 have BMC DASD MANAGER PLUS for DB2, you can integrate the REORG Advisor feature with the BMCTRIG feature.

Other topics in this user guide provide more information about integrating these features.

For more information, see the DASD MANAGER PLUS 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 describes how to work with object sets. Object sets identify the data upon which you want to extract SQL.

This chapter includes the following sections:

- “Overview of object sets” on page 59
- “Defining an object set” on page 64
- “Management of object set definitions” on page 67

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 the following items:

- Client IDs (CL)
- Correlation IDs (CR)
- Databases (DB)
- DBRM names (DM)
- Package names (PG)
- Plan names (PL)
- User IDs (US)
- Tablespace names (TS)
- Table names (TB)
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 is extracted (as shown in Figure 12 on page 60).

**Figure 12: Object sets and source parameters**

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 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

Use naming patterns to identify the objects you want to exclude or include in the object set.

Table 5 on page 61 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 5: Object types for object definitions

<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 for the client ID. This pattern is composed of the following values:</td>
<td>▶ APPTUNE trace data sets</td>
</tr>
<tr>
<td></td>
<td>■ Client ID: The name pattern for a client ID. This value can contain up to 16 characters.</td>
<td>▶ BMC Performance Database tables</td>
</tr>
<tr>
<td></td>
<td>■ Workstation: The name pattern for the workstation name to associate with the client ID. This value can contain up to 18 characters.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ Application: 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>Format: client.workstation.application</td>
<td></td>
</tr>
<tr>
<td>Correlation IDs (CR)</td>
<td>The name pattern for the correlation ID. This value can contain up to 12 mixed-case characters.</td>
<td>▶ APPTUNE trace data sets</td>
</tr>
<tr>
<td></td>
<td>Format: name</td>
<td>▶ BMC Performance Database tables</td>
</tr>
<tr>
<td>Database (DB)</td>
<td>The name pattern for the database. This value can contain up to 8 characters.</td>
<td>▶ APPTUNE trace data sets</td>
</tr>
<tr>
<td></td>
<td>Format: name</td>
<td>▶ BMC Performance Database tables</td>
</tr>
<tr>
<td></td>
<td></td>
<td>▶ DB2 catalog</td>
</tr>
<tr>
<td>Object type</td>
<td>Description</td>
<td>Applicable sources</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------</td>
<td>--------------------</td>
</tr>
</tbody>
</table>
| **DBRM names (DM)** | The identifier for a DBRM. This identifier is composed of the following values:  
  - **Plan name**: The name pattern for a plan name. This value can contain up to 8 uppercase characters. Leave the field blank for an implied wildcard.  
  - **DBRM name**: The name pattern for the DBRM name. This value can contain up to 8 uppercase characters.  
  
  **Format**: `planName.dbrmName` |  
  - APPTUNE trace data sets  
  - BMC Performance Database tables  
  - DB2 catalog  
  - Statement cache |
| **Package names (PG)** | The identifier for a package. This identifier is composed of the following values:  
  - **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.  
  
  **Format**: `collID.packageName.version` |  
  - APPTUNE trace data sets  
  - BMC Performance Database tables  
  - DB2 catalog  
  - Statement cache |
| **Plan name (PL)** | The name of the plan. This value can contain up to 8 uppercase characters.  
  **Format**: `name` |  
  - APPTUNE trace data sets  
  - BMC Performance Database tables  
  - DB2 catalog |
<table>
<thead>
<tr>
<th>Object type</th>
<th>Description</th>
<th>Applicable sources</th>
</tr>
</thead>
</table>
| Tablespace (TS) | The name pattern of the tablespace. This pattern is composed of the following values:  
  - **Database**: The name pattern for the database. This field can contain up to 8 characters.  
  - **Tablespace name**: This field can contain up to 8 characters.  
  **Format**: `database.tablespaceName` |  
  - APPTUNE trace data sets  
  - BMC Performance Database tables  
  - DB2 catalog |  
| Table (TB)      | The name pattern of the table. This pattern is composed of the following values:  
  - **Creator**: The name pattern for the table creator. This field can contain up to 128 characters.  
  - **Table name**: This field can contain up to 128 characters.  
  **Format**: `creator.tableName` |  
  - 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.  
  **Format**: `name` |  
  - APPTUNE trace data sets  
  - BMC Performance Database tables  
  - Statement cache |  

---

*a* 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*.

*b* You can use wildcard characters (`*`) as part of the pattern for each value.

---

**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.

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.

---

*Overview of object sets*
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 would 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.

---

## 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 59 to determine the naming patterns that you 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 on page 64) is displayed. The top of this panel includes options that allow you to select, view, or create workloads.

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

---

SQL Performance for DB2 User Guide
4 In the **Object Set Data Set** field, enter the name of the data set that contains 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 on page 65), enter the name for this set of naming patterns in the **Current Object Set Name** field.

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

```
PSSWC180 ------------------ Object Set Specification --------- Row 1 to 1 of 1
Command ===> ________________________________________________ Scroll ===> CSR
Current Object Set Name : DBOBJ1
Current Object Set Data Set : RDAVEM.RDAVEM.MYDS
Remarks : Payroll Application_____________________
Action column options : I = Insert  R = Repeat  D = Delete  Z = Zoom
? = Insert from popup
Include/Exclude options : Plus sign = Include  Minus sign = Exclude
Incl/ Obj Name or
Act Excl  Type Name Pattern
--- ----- ---- ---------------------------------------------------------------
  +     __ _______________________________________________________________
```

7 In the **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 “Description of object set patterns” on page 61.

■ 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 66) is displayed.

**Figure 15: Enter Name Specification panel (PSSWC185)**

```
PSSWC185 ------ Enter Name Pattern Specification  0----------
Command ===> Enter Name Pattern Specification
Include/Exclude : 1  1 - Include
2 - Exclude
Pattern Type : 7  1 - Plan
2 - Package
3 - DBRM
4 - Clientid
5 - Userid
6 - Corrid
7 - Database
8 - Tablespace
9 - Table
```

Press ENTER to accept input and continue.
Enter CANCEL to disregard input and return.

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 for which you are creating a name pattern and press **Enter** to display the appropriate **objectType** Specification panel.

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

- To 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.
Management of 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

This task describes how to select an object set.

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

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

   The Extract SQL for Workload panel is displayed.

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

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

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

6. On the Object Set List panel, type S next to the object that you want to use and press Enter.

   The Extract SQL for Workload panel is displayed.

Viewing an object set

This task describes how to view an object set.

1. On the Performance Advisors panel, select 1 (Workload Access Path Compare) or 2 (Workload Index Advisor) and press Enter.
2 On the Workload Access Path Compare Menu or the 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 Name field, type %.

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, type 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, type 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 PF3 until you return to the Extract SQL for Workload panel.

**Editing an object set**

This task describes how to edit an existing object set.

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

2 On the Workload Access Path Compare Menu or the Index Advisor Menu, select 1 (Extract and Explain SQL) 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 64), 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 select the name of the object set from a list, type % and press Enter. On the Object Set List panel, enter E next to the object set that you want to edit and press Enter.
Tip
To locate an object set by searching for a text string, type FIND string on the Command line, and press Enter. Press PF5 to move the cursor to the next instance of the specified text.

Figure 16: Object Set List panel (PSSWC175)

<table>
<thead>
<tr>
<th>Command</th>
<th>Scroll</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSSWC175</td>
<td>CSR</td>
</tr>
<tr>
<td>Current Object Set Name : %</td>
<td></td>
</tr>
<tr>
<td>Current Object Set Data Set : RDAVEM.MYDS</td>
<td></td>
</tr>
<tr>
<td>Action column options:</td>
<td></td>
</tr>
<tr>
<td>I = Insert/New C = Copy E = Edit R = Rename D = Delete S = Select V = view</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Remarks</td>
</tr>
<tr>
<td>E DBOBJ1</td>
<td>Payroll Application</td>
</tr>
<tr>
<td>MYOJBSET</td>
<td>Test Data</td>
</tr>
</tbody>
</table>

6 On the Object Set Specification panel (Figure 14 on page 65), edit the object set as needed, according to the instructions in Table 6 on page 69:

Table 6: Editing the Object Set Specification panel

<table>
<thead>
<tr>
<th>To do this</th>
<th>Perform these steps</th>
</tr>
</thead>
</table>
| Change an existing object by typing over a line | Change the existing values in the following fields by typing over them:  
- Incl/Excl  
- Obj Type  
- Name or Name Pattern |
| 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 “Description of object set patterns” on page 61.)  
4 In the Name or Name Pattern field, enter the name pattern of the objects that you want to include or exclude and press Enter. |
<table>
<thead>
<tr>
<th>To do this</th>
<th>Perform these steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Insert a new object by using pop-up menus</td>
<td>1  In the <strong>Act</strong> field, type ?.</td>
</tr>
<tr>
<td></td>
<td>2  In the <strong>Enter Name Pattern Specification panel</strong>, type 1 (<strong>Include</strong>) or 2 (<strong>Exclude</strong>) to indicate whether you want to include or exclude objects that match that pattern from the workload.</td>
</tr>
<tr>
<td></td>
<td>3  In the <strong>Pattern Type</strong> field, enter the number corresponding to the object type of the object for which you are creating a name pattern and press <strong>Enter</strong>.</td>
</tr>
<tr>
<td></td>
<td>4  On the <strong>Enter objectType Specification panel</strong>, enter the values that make up the name pattern. (For information, see “Description of object set patterns” on page 61.)</td>
</tr>
<tr>
<td></td>
<td>5  Press <strong>Enter</strong>.</td>
</tr>
<tr>
<td>Delete an object</td>
<td>In the <strong>Act</strong> field next to the object specification that you want to delete, type <strong>D</strong> and press <strong>Enter</strong>.</td>
</tr>
<tr>
<td>Repeat an object</td>
<td>In the <strong>Act</strong> field next to the object specification that you want to repeat, type <strong>R</strong> and press Enter.</td>
</tr>
<tr>
<td></td>
<td>You can modify the <strong>Incl/Excl</strong>, <strong>Object Type</strong>, and <strong>Name or Name Pattern</strong> fields by typing over them.</td>
</tr>
</tbody>
</table>

### Renaming an object set

This task describes how to rename an object set.

1. On the Performance Advisors 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 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, enter **R** next to the object set you want to rename, and press **Enter**.
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

This task describes how to copy an existing object set.

1. On the Performance Advisors 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 Name field, type %.

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

6. On the Object Set List panel, type C next to the object set that you want to copy and press Enter.

7. In the Copy Object Set panel, type 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 an object set with the same name that already exists 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 the copy operation.

### Deleting an object set

This task describes how to delete an object set.
1 On the Performance Advisors panel, select 1 (Workload Access Path Compare) or 2 (Workload Index Advisor) and press Enter.

2 On the Workload Access Path Compare Menu or the 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 Name field, type %.

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, enter D next to the object set that you want to delete and press Enter.
Comparing access paths for workloads

You can use the Workload Access Path Compare component to analyze SQL statements and generate 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. This chapter explains how to do so.

This chapter includes the following sections:

■ “Overview of workload comparisons” on page 73
■ “Comparing results” on page 78
■ “Extracting and Explaining a workload” on page 79
■ “Explaining a workload a second time” on page 84
■ “Comparing two or more workloads” on page 87
■ “Working with workload comparison reports” on page 91

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 can
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
■ Statement cache
■ 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
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 you to run your application on the second system. Thus, you can predict changes or problems before moving an application to another subsystem.

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

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 hlq.OBJECT.seqNumber and 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

Workload Access Path Compare process

The Workload Access Path Compare process encompasses the following basic tasks:

1. Defining object sets to identify objects in the workload (as described in “Working with object sets” on page 59).
Overview of workload comparisons

2 Extracting and Explaining a workload (as described in “Extracting and Explaining a workload” on page 79).

3 Explaining the same workload on a different subsystem, with a different set of environmental factors, or at a different point in time (as described in “Explaining a workload a second time” on page 84).

4 Comparing results and report any differences (as described in “Comparing results” on page 78).
Figure 17 on page 77 illustrates the process.

Figure 17: Workload Access Path Compare process (two different subsystems)

For more information, view the Quick Course SQL Performance for DB2 - Workload Compare. You must have a BMC Support ID to view the Quick Course.
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. For more information, see “Sample batch report for Workload Access Path Compare” on page 91.
Extracting and Explaining a workload

This task uses the Workload Access Path Compare component to define a workload and then Extracts and Explains 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 “Working with object sets” on page 59). 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 239.

- 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 80).

**Figure 18: Workload Access Path Compare Main Menu (PSSWC100)**

<table>
<thead>
<tr>
<th>Command</th>
<th>Select one of the following options and then press ENTER to continue.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Extract and Explain SQL - Define workload</td>
</tr>
<tr>
<td>2</td>
<td>Explain SQL - Explain SQL from Extract step</td>
</tr>
<tr>
<td>3</td>
<td>Compare access paths - Compare Explain results &amp; generate report</td>
</tr>
</tbody>
</table>

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

**c** Select 1 (Extract and Explain SQL).

**d** Specify a workload HLQ 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 are appended and the data sets are created if they do not already exist.

**e** Press Enter to display the Extract SQL for Workload panel (Figure 19 on page 80).

**Figure 19: Extract SQL for Workload (PSSWC105)**

<table>
<thead>
<tr>
<th>Command</th>
<th>Select to edit an Object Set Definition (blank to add new, % to show list)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Object Set Name : MYOBJECTSET</td>
</tr>
<tr>
<td></td>
<td>Object Set Data Set : RDAVEM.TEST.OBJSETDB</td>
</tr>
</tbody>
</table>

Select one of the following Workload sources and press Enter to continue

- DB2 catalog
- APPTUNE trace data
- DBRM library
- BMC Performance Database
- SQL Text data set
- Statement CACHE

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 HLQ 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. 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 described in Table 7 on page 81).

### Table 7: 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 33 on page 240</td>
</tr>
</tbody>
</table>
Extracting and Explaining a workload

<table>
<thead>
<tr>
<th>Source type</th>
<th>Panels to complete</th>
<th>Related info</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPTUNE trace data</td>
<td>■ SQL Workload Source Extract from APPTUNE Trace Data</td>
<td>Table 34 on page 240</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>Note: The SQL Workload Source - Archive Trace Data panel is displayed only if you specified Y in the List APPTUNE data set field on the previous panel. On this panel, select the trace data sets that you want to use and then press PF3 to continue. 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. 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. They are not saved in the APPTUNE archive directory.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DBRM library</td>
<td>SQL Workload Source Extract from DBRM library</td>
<td>Table 35 on page 242</td>
</tr>
<tr>
<td>BMC Performance</td>
<td>■ Extract SQL for Performance Advisor Database</td>
<td>Table 36 on page 242</td>
</tr>
<tr>
<td>Database</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 37 on page 245</td>
</tr>
<tr>
<td>Statement CACHE</td>
<td>SQL Workload Source Extract from DB2 Statement CACHE</td>
<td>Table 38 on page 246</td>
</tr>
</tbody>
</table>

b  Press Enter to display the SQL Workload Explain panel (Figure 20 on page 82).

**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 must use the same plan table owner (such as BMC). Note 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 uses 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, 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 on page 83).

**Figure 21: Batch Job panel (PSSPB000)**

```
PSSPB000 ------------------------ Batch Job ------------------------
Command "="=>

Specify a partitioned (include member) or sequential JCL data set.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>JCL Data Set . . .</td>
<td>'RDAVEM.SQLXPLR.CNTL(SQLX####)'</td>
</tr>
<tr>
<td>Current Counter</td>
<td>24</td>
</tr>
<tr>
<td>Set JCL Options</td>
<td>N</td>
</tr>
<tr>
<td>Build Job . . .</td>
<td>Y</td>
</tr>
<tr>
<td>Edit Data Set . .</td>
<td>Y</td>
</tr>
<tr>
<td>Submit . . .</td>
<td>N</td>
</tr>
</tbody>
</table>
```

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 8 on page 84.

Table 8: Batch Job options

<table>
<thead>
<tr>
<th>Field</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>JCL Data Set</td>
<td>Specify the data set to which Workload Access Path Compare writes 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. <strong>Note:</strong> If any of the data sets do not exist, the Allocate Data Set panel is displayed, enabling you to allocate a new data set.</td>
</tr>
<tr>
<td>Current Counter</td>
<td>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.</td>
</tr>
<tr>
<td>Set JCL Options</td>
<td><em>(optional)</em> Specify Y to review or change options for creating JCL. For information about setting JCL options, see the SQL Explorer for DB2 User Guide.</td>
</tr>
<tr>
<td>Build Job</td>
<td>Specify whether to generate the JCL into the data set that you specified in the JCL Data Set field. <strong>Note:</strong> After you build and edit the JCL the first time, an asterisk (*) is shown in both the Build Job and Edit Data Set fields. The asterisk ensures that you do not accidentally regenerate and overwrite existing JCL.</td>
</tr>
<tr>
<td>Edit Data Set</td>
<td>Specify whether to edit the data set online. If you specify Y, the JCL is displayed in an ISPF edit session.</td>
</tr>
<tr>
<td>Submit</td>
<td>Specify whether to submit the JCL for execution. BMC recommends that you specify N the first time the Batch Job panel is displayed. After the product generates the JCL, make any necessary changes to your data set. When the Batch Job panel is displayed again, specify Y in the Submit field to execute the JCL.</td>
</tr>
</tbody>
</table>

b Press Enter to complete the actions that you specified.

## 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.

3. Specify that you want to Explain SQL that has already been extracted:
   a. Select 2 (Explain SQL).
   b. Verify that the workload HLQ contains the value that was used when the SQL was extracted.
   c. Press Enter to display the SQL Workload Compare List for Explain panel (Figure 22 on page 85).

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

Figure 22: SQL Workload Compare List for Explain (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>Explain SQL</td>
<td>Explain Source</td>
</tr>
<tr>
<td>----------</td>
<td>------------</td>
<td>----------------</td>
</tr>
<tr>
<td>Ps五个</td>
<td>CATALOG DBAC DEBF RDAJLG.062A.STMTACCT.CLIENTC9.COMBO -ALL</td>
<td></td>
</tr>
<tr>
<td>Ps五个</td>
<td>CATALOG DEDK DEDR DB2 CAT SOURCE FROM DEDR EXPLAIN DEDK</td>
<td></td>
</tr>
<tr>
<td>Ps五个</td>
<td>CATALOG DEDK DEBF DB2 CAT SOURCE FROM DEBF EXPLAIN DEDK</td>
<td></td>
</tr>
<tr>
<td>Ps五个</td>
<td>TRACE DEDK DEDK APP TRACE SOURCE DEDK EXPLAIN DEDK</td>
<td></td>
</tr>
<tr>
<td>Ps五个</td>
<td>CATALOG DEDR DEDR PG SOURCE DEDR EXPLAIN DEDR</td>
<td></td>
</tr>
<tr>
<td>Ps五个</td>
<td>CATALOG DEBF DEBF COLLID and PG SOURCE DEBF EXPLAIN DEBF</td>
<td></td>
</tr>
</tbody>
</table>

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 are displayed.
   c. Press PF3 to return to the workload list, and press Enter to display the SQL Workload Explain panel.
To locate a workload by searching for a text string, type `FIND string` on the Command line, and press Enter. Press PF5 to advance to the next instance of the specified text.

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 valid only 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 must use the same plan table owner (such as BMC). You 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 you do not have this authority, the product uses 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, 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 where the Explain operation is performed.

This subsystem does not have to be the same DB2 where 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.
6 Use the Batch Job panel to generate, edit, and submit JCL to perform the Explain operation.

For more information about the options on the Batch Job panel, see Table 8 on page 84.

Comparing two or more workloads

In this task, you compare two or more Explain results and generate a report.

Before you begin

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 that the workload HLQ 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 on page 88).
This panel lists all workloads that were previously explained and that exist for the specified workload HLQ.

**Figure 23: SQL Workload Compare (PSSWC130)**

<table>
<thead>
<tr>
<th>Command</th>
<th>Scroll</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSR</td>
<td></td>
</tr>
</tbody>
</table>

**Processing mode**: O (O = Online, B = Batch)

Baseline (0) workload and one or more comparison workloads (1-9)

**Actions**: V D 0-9

**Explain SQL**

<table>
<thead>
<tr>
<th>type</th>
<th>source</th>
<th>DB2</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>XD</td>
<td>CATALOG</td>
<td>DEDR</td>
<td>DEDR V8V9DICI.V8V9DIFF.%</td>
</tr>
<tr>
<td>XD</td>
<td>CATALOG</td>
<td>DEBF</td>
<td>DEBF V8V9DICI.V8V9DIFF.%</td>
</tr>
</tbody>
</table>

---

4. Specify report options and select the workload files:

   a. In the **Processing mode** field, type **O** to run the comparison online or **B** to run the comparison in batch.

   **Tip**

   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 94*.

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

   **Tip**

   To locate a workload by searching for a text string, type **FIND string** on the **Command** line, and press **Enter**. Press **PF5** to move the cursor to the next appearance of the specified text string.

   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 is made, but you can still view the baseline in the report.

   For information about how comparisons are made, see “Comparing results” on page 78.
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 8 on page 84.

  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 (Figure 24 on page 89) is displayed.

5 If you generated an online report, review the report on the SQL Workload Compare Report panel. To locate specific workloads, see the tip to **Step 4.c on page 88**.

Figure 24: SQL Workload Compare Report (PSSWC140)

```
<table>
<thead>
<tr>
<th>Actions: S T H</th>
<th>Plan</th>
<th>Program</th>
<th>Queryno</th>
<th>Cost Diff</th>
<th>Weighted</th>
<th>Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Name</td>
<td>Name</td>
<td></td>
<td>Cost Diff</td>
<td></td>
<td>In</td>
</tr>
<tr>
<td>DCIINSTL</td>
<td>AINTEPL</td>
<td>139, 139</td>
<td>29325</td>
<td>29325</td>
<td>STATS INDEX</td>
<td></td>
</tr>
<tr>
<td>DCIINSTL</td>
<td>AINTEPL</td>
<td>642, 642</td>
<td>240</td>
<td>240</td>
<td>STATS INDEX</td>
<td></td>
</tr>
<tr>
<td>DCIINSTL</td>
<td>AINTEPL</td>
<td>183, 183</td>
<td>213</td>
<td>213</td>
<td>PATH STATS INDEX</td>
<td></td>
</tr>
<tr>
<td>DCIINSTL</td>
<td>AINEBMCL</td>
<td>848, 848</td>
<td>15</td>
<td>15</td>
<td>STATS INDEX</td>
<td></td>
</tr>
<tr>
<td>DCIINSTL</td>
<td>AINTEPL</td>
<td>517, 517</td>
<td>13</td>
<td>13</td>
<td>STATS</td>
<td></td>
</tr>
<tr>
<td>DCIINSTL</td>
<td>AINESYNOD</td>
<td>829, 829</td>
<td>13</td>
<td>13</td>
<td>STATS</td>
<td></td>
</tr>
<tr>
<td>DCIINSTL</td>
<td>AINESYNOD</td>
<td>823, 823</td>
<td>8</td>
<td>8</td>
<td>STATS</td>
<td></td>
</tr>
<tr>
<td>DCIINSTL</td>
<td>AINTEPL</td>
<td>717, 717</td>
<td>1</td>
<td>1</td>
<td>STATS INDEX</td>
<td></td>
</tr>
</tbody>
</table>
```

Table 9 on page 90 describes how to use the report.
### Table 9: Reviewing an online report

<table>
<thead>
<tr>
<th>Action</th>
<th>Steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Show detailed information for a workload</td>
<td>Type <strong>S</strong> next to the statement for which you want to show detailed information and press <strong>Enter</strong>. The SQL Workload Compare Detail Report panel (Figure 25 on page 90) is displayed.</td>
</tr>
<tr>
<td>Hide detailed information for a workload</td>
<td>Type <strong>H</strong> next to the entry for which you want to hide details and press <strong>Enter</strong>.</td>
</tr>
<tr>
<td>Show the SQL text for a statement</td>
<td>Type <strong>T</strong> next to the statement for which you want to show SQL text and press <strong>Enter</strong>.</td>
</tr>
<tr>
<td>Show information for all entries</td>
<td>Type <strong>S ALL</strong> on the <strong>Command</strong> line and press <strong>Enter</strong>.</td>
</tr>
<tr>
<td>Hide information for all entries</td>
<td>Type <strong>H ALL</strong> on the <strong>Command</strong> line and press <strong>Enter</strong>.</td>
</tr>
</tbody>
</table>

**Figure 25: SQL Workload Compare Detail Report (PSSWC142)**

<table>
<thead>
<tr>
<th>FILE</th>
<th>COMMANDS</th>
<th>HELP</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSSWC142</td>
<td>SQL Workload Compare Detail Report</td>
<td>Scroll ====&gt; CSR</td>
</tr>
<tr>
<td>Command ===&gt;</td>
<td></td>
<td>More: &gt;</td>
</tr>
<tr>
<td>Actions:</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>S</strong></td>
<td>H</td>
<td></td>
</tr>
<tr>
<td>EXPL SRC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LABL SSID SSID NAME COLLID VERSION STMTNO COST</td>
<td></td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>----</td>
<td>-------</td>
</tr>
<tr>
<td>XD01 DEBF DEBF AINTEPL</td>
<td>183</td>
<td>25</td>
</tr>
<tr>
<td>XD00 DEDR DEDR AINTEPL</td>
<td>183</td>
<td>4</td>
</tr>
<tr>
<td>QB PL M TB AC MT</td>
<td>I</td>
<td>NEW</td>
</tr>
<tr>
<td>LABL BL NO E NO TY CL TABLE INDEX</td>
<td>O</td>
<td>UJOG</td>
</tr>
</tbody>
</table>

This panel shows detailed information for the workload selected.

**Tip:** You can use **S** or **H** in the action field to show or hide detailed information for a specific entry.

For information about the fields in this panel, see “Workload Access Path Compare and Index Advisor report fields” on page 247.

To locate specific workloads, see the tip in Step 4.c on page 88.

---

**Note**

Fields might 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 247.
Working with workload comparison reports

This topic 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 94
- “Customizing the comparison report” on page 95

Sample batch report for Workload Access Path Compare

The following figure 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 252.
Working with workload comparison reports

SQL Performance for DB2 User Guide
Chapter 5  Comparing access paths for workloads

Working with workload comparison reports

---

**Table Example**

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Frequency</th>
<th>Value</th>
<th>T</th>
<th>Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>XDDO COLUMN_2_SML</td>
<td>0.232318</td>
<td>8000</td>
<td>F</td>
<td>1</td>
</tr>
<tr>
<td>XDDO COLUMN_2_SML</td>
<td>0.033399</td>
<td>A328</td>
<td>F</td>
<td>1</td>
</tr>
<tr>
<td>XDDO COLUMN_2_SML</td>
<td>0.01873</td>
<td>AEE0</td>
<td>F</td>
<td>1</td>
</tr>
<tr>
<td>XDDO COLUMN_2_SML</td>
<td>0.017462</td>
<td>04D0</td>
<td>F</td>
<td>1</td>
</tr>
<tr>
<td>XDDO COLUMN_2_SML</td>
<td>0.012769</td>
<td>B070</td>
<td>F</td>
<td>1</td>
</tr>
<tr>
<td>XDDO COLUMN_2_SML</td>
<td>0.012769</td>
<td>B070</td>
<td>F</td>
<td>1</td>
</tr>
<tr>
<td>XDDO COLUMN_2_SML</td>
<td>0.012769</td>
<td>B070</td>
<td>F</td>
<td>1</td>
</tr>
<tr>
<td>XDDO COLUMN_2_SML</td>
<td>0.012769</td>
<td>B070</td>
<td>F</td>
<td>1</td>
</tr>
<tr>
<td>XDDO COLUMN_2_SML</td>
<td>0.012769</td>
<td>B070</td>
<td>F</td>
<td>1</td>
</tr>
<tr>
<td>XDDO COLUMN_3_CHA</td>
<td>0.001965</td>
<td>40C1C1C2D5F0F0F0F0F4F8F3</td>
<td>F</td>
<td>1</td>
</tr>
<tr>
<td>XDDO COLUMN_3_CHA</td>
<td>0.001965</td>
<td>40C1C1C2D5F0F0F0F0F4F7F4</td>
<td>F</td>
<td>1</td>
</tr>
<tr>
<td>XDDO COLUMN_3_CHA</td>
<td>0.001965</td>
<td>40C1C1C2D5F0F0F0F0F4F7F1</td>
<td>F</td>
<td>1</td>
</tr>
<tr>
<td>XDDO COLUMN_3_CHA</td>
<td>0.001965</td>
<td>40C1C1C2D5F0F0F0F0F4F6F8</td>
<td>F</td>
<td>1</td>
</tr>
<tr>
<td>XDDO COLUMN_3_CHA</td>
<td>0.001965</td>
<td>40C1C1C2D5F0F0F0F0F4F6F5</td>
<td>F</td>
<td>1</td>
</tr>
<tr>
<td>XDDO COLUMN_3_CHA</td>
<td>0.008841</td>
<td>B520</td>
<td>F</td>
<td>1</td>
</tr>
<tr>
<td>XDDO COLUMN_3_CHA</td>
<td>0.009332</td>
<td>90B3</td>
<td>F</td>
<td>1</td>
</tr>
<tr>
<td>XDDO COLUMN_3_CHA</td>
<td>0.009823</td>
<td>8ED8</td>
<td>F</td>
<td>1</td>
</tr>
<tr>
<td>XDDO COLUMN_3_CHA</td>
<td>0.010314</td>
<td>A260</td>
<td>F</td>
<td>1</td>
</tr>
</tbody>
</table>

---

**Summary**

- **Total Statements:** 3
- **Total Tables:** 3

---

**Note:** The content above is a hypothetical example of how a workload comparison report might be structured and read. Actual reports may differ in format and detail.
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 (Figure 27 on page 94), type any character in the Report options field on the SQL Workload Compare panel.

   Figure 27: SQL Workload Compare Report Options panel (PSSWC137)

   PSSWC137 ------------ SQL Workload Compare Report Options ---------------------
   Command ===> Specify options and press PF3 to return to SQL Workload Compare.
   Layout data set : AFDQA.V111TEST.SAMP(PSSREPB)
   Edit Layout data set : N (Y/N)
   --------------------- Batch Compare Options -------------------------------
   Generate report title : Y (Y/N)
   Title : SQL WORKLOAD COMPARE REPORT
   Summary report : Y (Y/N)
   Sort order : 1  1. Cost (desc)
               2. Weighted cost (cost * executions)
               3. Plan/program, cost (desc)
   Detail Report :
       Show SQL : Y (Y/N)
       Show obj : Y (Y/N)

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.sampleLib(PSSREPB) member. The name of this library varies 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 252.

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.
N produces a batch report with no title.

Y adds a user-specified title to the report. If you select this option, enter a title of up to 65 alphanumeric characters in the Title field.

b 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 4.c on page 95).
- N produces a batch report with no summary.

c If you specified Y in Step 4.b on page 95, 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.

d 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.sampleLib(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 252. For information about the fields that make up the report, see “Workload Access Path Compare and Index Advisor report fields” on page 247.
Recommending indexes for workloads

This chapter describes how to use the Workload Index Advisor component to create the right indexes to minimize the cost of running a set of SQL statements.

This chapter includes the following sections:

- “Overview of the Workload Index Advisor” on page 97
- “Extracting SQL and recommending an index” on page 100
- “Recommending an index for a workload a second time” on page 105
- “Reviewing index recommendations” on page 108
- “Working with index recommendation reports” on page 111

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 can 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
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 indexes 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 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 “Working with object sets” on page 59).

2. “Extracting SQL and recommending an index” on page 100.

3. “Recommending an index for a workload a second time” on page 105.

4. “Reviewing index recommendations” on page 108.
Figure 28 on page 100 illustrates this process.

**Figure 28: Index Advisor process**

For more information, view the Quick Course *SQL Performance for DB2 - Using Recommindex*. You must have a BMC Support ID to view the Quick Course.

### 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 “Working with object sets” on page 59). 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 239.

- 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 101).

   **Figure 29: Index Advisor Menu (PSSWC100)**

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

   1. Extract SQL and Recommend index
   2. Recommend
   3. Review Recommendations

   Specify workload high level qualifier:

   HLQ: PSS.V101TEST.WLCIX (Max 26 characters)
c Select **1 (Extract SQL and Recommend)**.

d Specify a workload HQL 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 are appended and the data sets are created if they do not already exist.

e Press **Enter** to display the Extract SQL for Workload panel.

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 “Working with object sets” on page 59.

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 type 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.

The panels displayed depend on the source type you selected. You can use the panels to specify 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 10 on page 103**). Related information appears at the end of this topic.
### Table 10: Source Workload Extract panels by source

<table>
<thead>
<tr>
<th>Source type</th>
<th>Panels to complete</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2 catalog</td>
<td>SQL Workload Source Extract from DB2 catalog</td>
</tr>
<tr>
<td>APPTUNE trace data</td>
<td>SQL Workload Source Extract from APPTUNE Trace Data</td>
</tr>
<tr>
<td></td>
<td>SQL Workload Source - Archive Trace Data</td>
</tr>
<tr>
<td></td>
<td>SQL Workload Source - Add Archive Trace Data</td>
</tr>
<tr>
<td><strong>Note:</strong> The SQL Workload Source - Archive Trace Data panel is displayed only if you specified <strong>Y</strong> 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. To add a data set that is not displayed in the list, type <strong>A</strong> in any action field or on the <strong>Command</strong> line and enter the data set name in the SQL Workload Source - Add Archive Trace Data panel. 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>
</tr>
<tr>
<td>BMC Performance Database</td>
<td>Extract SQL for Performance Advisor Database</td>
</tr>
<tr>
<td></td>
<td>Extract parameters for Performance Advisor Database</td>
</tr>
<tr>
<td>SQL text data set</td>
<td>SQL Workload Extract from SQL Text Data Set</td>
</tr>
<tr>
<td>Statement CACHE</td>
<td>SQL Workload Source Extract from DB2 Statement CACHE</td>
</tr>
</tbody>
</table>

**b** After you have finished entering your source information, press **Enter** to display the Recommindex panel (Figure 30 on page 103).

**Figure 30: Recommindex panel (PSSWC120)**

```
PSSWC125:--------------------------------- Recommindex --------------------------------- Command ===>
| Index Advisor Report Options : _ |
| Specify SQL workload qualifiers : |
| Explain type : XD 1. XD = Explain dynamic |
| Plan Table Owner : BMC (USERID,authid) |
| Explain SSID : DEBF |
| Import DDL : 'PSS.V111TEST.DEMO.JCL(DEMOIX2)' |
| DDL Option : 3 1. Recommindex 2. DDL 3. Both |
| Remarks : IXA W/ IMPORTED DDL |
```
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 114 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 must 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 uses 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, 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 is performed.

e. In the **Import DDL** field, enter the name of the data set that contains the DDL statements for indexes that you want to change, delete, or drop for the workload during your analysis.

   You can generate these DDL statements with the What-If Index processing
f In the DDL Option field, specify:

- 1 If you want to only generate the recommended index without using DDL
- 2 If you want to use the DDL without recommending indexes
- 3 If you want to generate the recommended indexes and apply the DDL to your analysis

Option 1 is the default.

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

You can enter up to 40 alphanumeric characters.

h Press Enter to display the Batch Job panel.

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. Related information appears in one of the tables in the topic on Extracting and Explaining a workload.

b Press Enter to complete the actions that you specified.

Related Information

- “Workload Access Path Compare and Index Advisor report fields” on page 247
- “Extracting and Explaining a workload” on page 79

Recommending an index for a workload a second time

Perform this task to 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.

3. Choose to recommend an index:
   a. Select 2 (Recommindex).
   b. Verify that the workload HLQ is the same as that used when performing the procedure “Extracting SQL and recommending an index” on page 100.
   c. Press Enter to display the SQL Workload List for Recommindex panel (Figure 31 on page 106).

This panel lists all workloads that:
- Have been previously extracted by either the workload comparison or the index recommendation process
- 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</td>
<td>Explain Source</td>
<td>DB2</td>
</tr>
<tr>
<td>-----</td>
<td>--------</td>
<td>------</td>
</tr>
<tr>
<td>XD</td>
<td>CATALOG</td>
<td>DEER</td>
</tr>
<tr>
<td>XD</td>
<td>CATALOG</td>
<td>DEER</td>
</tr>
<tr>
<td>XD</td>
<td>CATALOG</td>
<td>DEER</td>
</tr>
<tr>
<td>XD</td>
<td>CATALOG</td>
<td>DEER</td>
</tr>
<tr>
<td>XD</td>
<td>CATALOG</td>
<td>DEER</td>
</tr>
<tr>
<td></td>
<td>DEBF</td>
<td>DEBF</td>
</tr>
<tr>
<td>*** Bottom of data ***</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Tip

To locate a workload by searching for a text string, type FIND string on the Command line, and press Enter. Press PF5 to move the cursor to the next appearance of the specified text string.

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, to view a workload type V in the adjacent action field and press Enter. The workload qualifiers that were used when the workload was extracted are displayed. To return to the workload list, press PF3.

5 On 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 114 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 valid only 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 must use the same plan table owner (such as BMC). You 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 you do not have this authority, the product uses 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, 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 is 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.

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 8 on page 84.

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. The optimizer used this index while Explaining at least one statement in the workload.

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

- No savings—Index Advisor recommends 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 are associated with creating this index.
In the Index Recommendation Report (Figure 32 on page 109), 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 247.

**Figure 32: Index Recommendation Report**

<table>
<thead>
<tr>
<th>TABLE NAME</th>
<th>PAG COM CARDF</th>
<th>NPAGESF</th>
<th>S R PAGE FRE POOL</th>
<th>SEG</th>
<th>C S I2 PART NAC</th>
<th>ORD_LN_ITEM</th>
<th>INDEX NAME</th>
<th>SAVINGS</th>
<th>#STMTS</th>
<th>RECOMMEND</th>
<th>R G D NLEAF</th>
<th>NLVL</th>
<th>POOL</th>
<th>KEY</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC00001</td>
<td>411908</td>
<td>6 CREATE D N N</td>
<td>1</td>
<td>1 Bp0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC00012</td>
<td>260514</td>
<td>3 CREATE D N N</td>
<td>55</td>
<td>2 Bp0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC00008</td>
<td>136</td>
<td>1 CREATE D N N</td>
<td>157385</td>
<td>3 Bp0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC00011</td>
<td>13</td>
<td>1 CREATE D N N</td>
<td>164228</td>
<td>3 Bp0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC00010</td>
<td>27</td>
<td>27 CREATE D N N</td>
<td>134302</td>
<td>3 Bp0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1X A01055</td>
<td>0</td>
<td>165 KEEP P Y Y</td>
<td>187916</td>
<td>3 Bp5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1x B01055</td>
<td>0</td>
<td>3 KEEP D N N</td>
<td>86748</td>
<td>4 Bp5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1X C01055</td>
<td>0</td>
<td>18 KEEP D N N</td>
<td>269842</td>
<td>4 Bp5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1X D01055</td>
<td>0</td>
<td>57 KEEP D N N</td>
<td>270223</td>
<td>4 Bp5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1X E01055</td>
<td>0</td>
<td>14 KEEP D N N</td>
<td>343912</td>
<td>4 Bp49</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC00004</td>
<td>0</td>
<td>5 CREATE D N N</td>
<td>119108</td>
<td>3 Bp0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC00006</td>
<td>0</td>
<td>0 NOT SEEN D N N</td>
<td>134302</td>
<td>3 Bp0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC00007</td>
<td>0</td>
<td>0 NOT SEEN D N N</td>
<td>2</td>
<td>1 Bp0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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 HLQ 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 110).
This panel lists all workloads in the specified repository that have generated indexes.

**Figure 33: Index Advisor Workloads (PSSWC245)**

<table>
<thead>
<tr>
<th>Command</th>
<th>Scroll</th>
<th>CSR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify a workload</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Actions:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S V D</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Explain SQL</td>
<td>Explain Source</td>
<td>Remarks</td>
</tr>
<tr>
<td>XD</td>
<td>SQLFILE</td>
<td>DEDR</td>
</tr>
<tr>
<td>XD</td>
<td>CATALOG</td>
<td>DEDK</td>
</tr>
<tr>
<td>XD</td>
<td>SQLFILE</td>
<td>DEDR</td>
</tr>
<tr>
<td>XD</td>
<td>SQLFILE</td>
<td>DEDR</td>
</tr>
</tbody>
</table>

**Tip**
To locate a workload by searching for a text string, type `FIND string` on the `Command` line, and press `Enter`. Press `PF5` to move the cursor to the next appearance of the specified text string.

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 typing `V` and pressing `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 on page 110), specify the action that you want to perform:

**Figure 34: Recommindex: Index Recommendation Report panel (PSSWC140)**

<table>
<thead>
<tr>
<th>FILE</th>
<th>COMMANDS</th>
<th>HELP</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSSWC140</td>
<td>RECOMMINDEX : INDEX RECOMMENDATION REPORT</td>
<td>Scroll ===› CSR More: &gt;</td>
</tr>
<tr>
<td>Actions:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S T H</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TABLE</td>
<td>NAME</td>
<td>PCT</td>
</tr>
<tr>
<td>NAME</td>
<td>NAME</td>
<td>PAGES</td>
</tr>
<tr>
<td>V8V9DFTB</td>
<td>V8V9DFTB</td>
<td>99</td>
</tr>
<tr>
<td>BMC00001</td>
<td>BMC00001</td>
<td>19</td>
</tr>
<tr>
<td>BMC00002</td>
<td>BMC00002</td>
<td>0</td>
</tr>
<tr>
<td>V8V9DFTB_INDEX3</td>
<td>V8V9DFTB_INDEX3</td>
<td>0</td>
</tr>
</tbody>
</table>

---

Reviewing index recommendations

110 SQL Performance for DB2 User Guide
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**.

Figure 35 on page 111 shows an example of an expanded statement.

**Figure 35: Example of an expanded statement**

<table>
<thead>
<tr>
<th>TABLE</th>
<th>PCT</th>
<th>PCT</th>
<th>CARDF</th>
<th>NPAGESF</th>
<th>E</th>
<th>L</th>
<th>FREE</th>
<th>PCT BUFFER</th>
<th>S</th>
<th>SI</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAME</td>
<td>PAGES</td>
<td>COMP</td>
<td>CARDF</td>
<td>NPAGESF</td>
<td>S</td>
<td>R</td>
<td>PAGE</td>
<td>FREE</td>
<td>POOL</td>
<td>C</td>
</tr>
<tr>
<td>-------</td>
<td>-------</td>
<td>------</td>
<td>-------</td>
<td>---------</td>
<td>---</td>
<td>---</td>
<td>------</td>
<td>-------</td>
<td>------</td>
<td>---</td>
</tr>
<tr>
<td>V8V9DFTB</td>
<td>99</td>
<td>0</td>
<td>2036</td>
<td>255</td>
<td>E</td>
<td>A</td>
<td>0</td>
<td>0</td>
<td>BP0</td>
<td></td>
</tr>
</tbody>
</table>

**INDEX NAME**

<table>
<thead>
<tr>
<th>SAVINGS</th>
<th>#STMTS</th>
<th>RECOMMEND</th>
<th>R</th>
<th>G</th>
<th>D</th>
<th>NLEAF</th>
<th>NLVL</th>
<th>POOL</th>
<th>BUFFER</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC00001</td>
<td>19</td>
<td>CREATE</td>
<td>D</td>
<td>N</td>
<td>N</td>
<td>11</td>
<td>2</td>
<td>BP0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>KEY</th>
<th>CD</th>
<th>CO</th>
<th>INDEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>COLUMN</td>
<td>NO</td>
<td>SQ</td>
<td>O</td>
</tr>
<tr>
<td>-------</td>
<td>----</td>
<td>----</td>
<td>---</td>
</tr>
<tr>
<td>COLUMN_15</td>
<td>13</td>
<td>1</td>
<td>A</td>
</tr>
</tbody>
</table>

**EXPL SRC**

<table>
<thead>
<tr>
<th>SSID</th>
<th>SSID</th>
<th>NAME</th>
<th>COLLID</th>
<th>VERSION</th>
<th>STMTNO</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEDK</td>
<td>DEDK</td>
<td>V8V9DIFF</td>
<td>V8V9DIC</td>
<td>2.01</td>
<td>305</td>
</tr>
</tbody>
</table>

```sql
DECLARE CUR-V8V9DIFF-TWO CURSOR FOR
SELECT DISTINCT COLUMN_15
FROM V8V9DFTB
WHERE COLUMN_15 > '2008-08-11-16.14.50.435571'
FOR FETCH ONLY
```

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.

To locate a workload by searching for a text string, see the note to Step 3.c on page 109.

**Note**

Fields might 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 247.

---

**Working with index recommendation reports**

This topic describes how to work with index recommendation reports that are generated. It includes information about
Sample batch report for Index Advisor

The following figure 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 252.

---

**Figure 36: Sample of Index Advisor report**

```sql
1--SQL EXPLORER 11.1.0 PARMS- RELEASED JUNE 2013 COMPILED APR 25 2013
(C) COPYRIGHT 1996 - 2013 BMC SOFTWARE, INC.
TITLE='INDEX RECOMMENDATION REPORT'
INDEX=YES
SSID=DEDK
PLAN=DIAG101A
EXECUTE=X
PTOWNR=BMC

`INDEX RECOMMENDATION REPORT` 1

| TABLE | NAME | PAGES | COMP | CARDF | NPAGESF | S | R | PAGE | FREE | POOL | SIZE | PARTS | NACTIVEF | STATSTIME |
|-------|------|-------|------|-------|----------|--|---|------|------|------|------|--------|-----------|
|       | NAME | NUM | HIGH2KEY | LOW2KEY | COLCARDF |      |    |      |      |      |      |        |           |

---

112 SQL Performance for DB2 User Guide
Working with index recommendation reports

Chapter 6  Recommending indexes for workloads    113

---

```
INDEX                                        U C C                BUFFER       FIRST      FULL   CLUSTER
NAME                SAVINGS #STMTS RECOMMEND R G D     NLEAF NLVL POOL       KEYCARD   KEYCARD     RATIO
------------------ -------- ------ --------- - - - --------- ---- -------- --------- --------- ---------
BMC00001                 19      1 CREATE    D N N        11    2 BP0           2036      2036  0.000000
KEY           CO CO   INDEX
COLUMN        NO SQ O NAME
------------ --- -- - ------------------
COLUMN_15     13  1 A BMC00001
EXPL SRC                                                             STMT    EXEC WEIGHTED SRC
SSID SSID NAME     COLLID       VERSION                    STMTNO    COST   COUNT     COST TYPE
1'INDEX RECOMMENDATION REPORT'                                                                                                 2
DEDK DEDK V8V9DIFF V8V9DICI     2.01                          305      57       1       57 C
EXPLAIN
TIME
2011-01-20-16.46.37.110000
DECLARE  CUR-V8V9DIFF-TWO CURSOR FOR
SELECT   DISTINCT COLUMN_15
FROM     V8V9DFTB
WHERE    COLUMN_15 > '2008-08-11-16.14.50.435571'
FOR      FETCH ONLY
INDEX                                        U C C                BUFFER       FIRST      FULL   CLUSTER
NAME                SAVINGS #STMTS RECOMMEND R G D     NLEAF NLVL POOL       KEYCARD   KEYCARD     RATIO
------------------ -------- ------ --------- - - - --------- ---- -------- --------- --------- ---------
BMC00002                  0      1 NO SAVING D N N         4    2 BP0            592       592  0.000000
KEY           CO CO   INDEX
COLUMN        NO SQ O NAME
------------ --- -- - ------------------
COLUMN_3_CHAR   3  1 A BMC00002
EXPL SRC                                                             STMT    EXEC WEIGHTED SRC
SSID SSID NAME     COLLID       VERSION                    STMTNO    COST   COUNT     COST TYPE
1'INDEX RECOMMENDATION REPORT'                                                                                                 3
DEDK DEDK V8V9DIFF V8V9DICI     2.01                          293       1       1        1 C
EXPLAIN
TIME
2011-01-20-16.46.37.110000
DECLARE  CUR-V8V9DIFF-ONE CURSOR FOR
SELECT   COLUMN_3_CHAR_12
FROM     V8V9DFTB
WHERE    COLUMN_3_CHAR_12 = 'ABC'
FOR      FETCH ONLY
INDEX                                        U C C                BUFFER       FIRST      FULL   CLUSTER
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
KEY           CO CO   INDEX
COLUMN        NO SQ O NAME
------------ --- -- - ------------------
COLUMN_2_SML   2  2 A V8V9DFTB_INDEX3
COLUMN_3_CHAR   3  3 A V8V9DFTB_INDEX3
COLUMN_4       4  4 A V8V9DFTB_INDEX3
COLUMN_5       5  5 A V8V9DFTB_INDEX3
KEY1_INTEGER   1  6 A V8V9DFTB_INDEX3
COLUMN_10      8  7 A V8V9DFTB_INDEX3
COLUMN_11      9  8 A V8V9DFTB_INDEX3
COLUMN_12     10  9 A V8V9DFTB_INDEX3
COLUMN_13     11 10 A V8V9DFTB_INDEX3
COLUMN_14     12 11 A V8V9DFTB_INDEX3
COLUMN_15     13 12 A V8V9DFTB_INDEX3
COLUMN_16     14 13 A V8V9DFTB_INDEX3
COLUMN_17     15 14 A V8V9DFTB_INDEX3
COLUMN_18     16 15 A V8V9DFTB_INDEX3
COLUMN_19     17 16 A V8V9DFTB_INDEX3
COLUMN_20     18 17 A V8V9DFTB_INDEX3
COLUMN_21     19 18 A V8V9DFTB_INDEX3
EXPLAIN
SSIP SRC                                                             STMT    EXEC WEIGHTED SRC
SSID SSID NAME     COLLID       VERSION                    STMTNO    COST   COUNT     COST TYPE
1'INDEX RECOMMENDATION REPORT'                                                                                                 3
DEDK DEDK V8V9DIFF V8V9DICI     2.01                          312       2       1        2 C
EXPLAIN
TIME
2011-01-20-16.46.37.110000
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 Recommindex panel, 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 37: Index Advisor Report Options (PSSWC275)](image)

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.sampleLib(PSSREPB) member. The name of this library varies 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 252.

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 38 on page 115) is displayed.

**Figure 38: Layout Data Set panel (PSSWCLDS)**

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

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 the **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 Recommindex 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 252.

For information about the fields that make up the report, see “Workload Access Path Compare and Index Advisor report fields” on page 247.
Exploring advice for Exceptions

This chapter presents the following topics:

- “Overview of the Exception Advisor” on page 117
- “Running the Exception Advisor batch job” on page 119
- “Viewing the Exception Advisor reports online” on page 129
- “Exception Advisor output” on page 132
- “History of statement performance” on page 133
- “Comparison graphs” on page 134
- “SQL text and host variables” on page 135
- “Advice” on page 136

Overview of the Exception Advisor

The Performance Advisor component of SQL Performance for DB2 maintains a DB2 Performance Advisor Database (PADB).

Using the PADB, the Exception Advisor identifies the root causes of previously triggered elapsed time and CPU time exception executions and makes recommendations.

The 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. The Exception Advisor bases its recommendations on observed conditions.
For more information, view the Quick Course *SQL Performance for DB2 - Using Exception Advisor.*
You must have a BMC Support ID to view the Quick Course.

**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 does not look at predicted performance (such as Explain data) but at actual performance history. Because the Exception Advisor uses existing data, you incur no further cost for data collection.

With the 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 “Managing performance with Performance Advisor” on page 47.

3. Run the Exception Advisor job.
   For more information, see “Running the Exception Advisor batch job” on page 119.

4. Use online Exception Advisor or batch reports in tuning efforts.
   For more information, see “Viewing the Exception Advisor reports online” on page 129.

Running the Exception Advisor batch job

Use this procedure to run the Exception Advisor batch job.

**Before you begin**

- Ensure that 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 that you have loaded the collected exception data into the PADB.
  For more information about creating and maintaining a PADB in DB2, see “Managing performance with Performance Advisor” on page 47.

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 120.

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 121
   - “Exception Advisor parameters” on page 125

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:

```
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.

//PAEXC JOB (ACCT),'PERFORMANCE ADVISOR', <= MODIFY
// MSGCLASS=X, <= CHECK
// CLASS=A <= CHECK
//
//***************************************************************/
// INSTRUCTIONS
//
// *     ?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)
//SYSIN 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.startdate="CURRENT_DATE - 1 DAY"

***********************************************************************/
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.stopdate="CURRENT_DATE - 1 DAY"
* 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. Imbed 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

Additional parameters:

**exception.filter=**

* 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.

**exception.level=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). For 1, reports are generated as advisor output.

**exception.printreports=1**

* exception.tabcreator: default table owner/creator if subsequent table specifications do not include the owner/creator.

**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.

**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.

**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.

**exception.hvtab=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. *
* IF OMITTED: MONTHLY_STMT_STATISTICS *
* IF 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. *
* IF OMITTED: WEEKLY_STMT_STATISTICS *
* IF 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. *
* IF OMITTED: DAILY_STMT_STATISTICS *
* IF 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. *
* IF OMITTED: No interval reporting *
* IF NULL: No interval reporting *
* ***********************************************************************
* exception.interval=

Chapter 7  Exploring advice for Exceptions  123
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
IF OMITTED: 1
IF NULL: 1
exception.nondml=1

exception.starthist: expressed in the same form as startime and stoptime, this determines how far back history will be reported.
IF OMITTED: No limit
IF NULL: No limit
exception.starthist=

exception.updateresults: whether to update the XADVISOR tables with the results of this run for subsequent online viewing.
0 = do not update tables with results
1 = update tables with results
IF 1 is specified, exception.level must be set to DETAIL.
IF OMITTED: 0
IF NULL: 1
exception.updateresults=1

exception.purgeresults: if updateresults is in effect, this specifies the age at which existing rows in the XADVISOR results tables will be purged.
IF OMITTED: "CURRENT_DATE - 7 DAYS"
IF NULL: No rows deleted
exception.purgeresults="CURRENT_DATE - 7 DAYS"

exception.runtab: 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.
IF OMITTED: XADVISOR_RUNS
IF NULL: XADVISOR_RUNS
exception.runtab=XADVISOR_RUNS

exception.ruletab: table that holds advice related to defined rules (from //RULES DD), if updateresults is 1.
If specified without owner, exception.tabcreator is the owner.
IF OMITTED: XADVISOR_RULES
IF NULL: XADVISOR_RULES
exception.ruletab=XADVISOR_RULES

exception.evnttab: table that holds basic information about exceptions. If updateresults is 1.
If specified without owner, exception.tabcreator is the owner.
IF OMITTED: XADVISOR_EVENTS
IF NULL: XADVISOR_EVENTS
Input parameters control various aspects reported on by the Exception Advisor. Table 11 on page 126 describes these input parameters. 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 11: Exception Advisor parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Omitted value</th>
<th>Null value</th>
</tr>
</thead>
</table>
| exception.baseline | Table that contains baseline statement statistics  
Exception Advisor uses this value to compare performance characteristics of an exceptional statement execution with statistics for the same statement in a baseline. | BASELINE_STMT_STATISTICS    | No baseline reporting                |
| exception.daily | Table that contains daily statement statistics  
Exception Advisor uses this value to compare performance characteristics of an exceptional statement execution with statistics for the same statement in a daily table. | DAILY_STMT_STATISTICS       | No daily reporting                   |
| exception.evnttab | If updateresults is 1, table that holds basic information about exceptions | XADVISOR_EVENTS             | XADVISOR_EVENTS                     |
| exception.filter | Use to limit the statements to be evaluated  
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.  
**Example:** exception.filter="WHERE PROGRAM='PROGRAM1' + AND PLAN='PLAN1'"  
You can use any columns in the STATS_EXCEPTION table for filtering. | No filtering                | No filtering                       |
<p>| exception.hisetab | If updateresults is 1, table that holds statement history entries a | XADVISOR_HISTORY_ENTRIES    | XADVISOR_HISTORY_ENTRIES            |
| exception.histtab | If updateresults is 1, table that holds header information for statement history a | XADVISOR_HISTORY_TITLES     | XADVISOR_HISTORY_TITLES             |
| exception.hvtab | Table that contains host variable value information a, b | STMT_EXCEPTIONS_HV          | No host variable reporting          |</p>
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Omitted value</th>
<th>Null value</th>
</tr>
</thead>
</table>
| exception.interval | Table that contains interval statement statistics  
Exception Advisor uses this value to compare performance characteristics of an exceptional statement execution with statistics for the same statement in an interval table. | No interval reporting         | No interval reporting       |
| exception.level | Whether to evaluate multiple exceptions individually or together as one:  
- SUMMARY evaluates all exceptions of same type from the same statement as a single exception.  
- DETAIL evaluates each exception individually. | DETAIL                        | DETAIL                      |
| exception.monthly | Table that contains monthly statement statistics  
Exception Advisor uses this value to compare performance characteristics of an exceptional statement execution with statistics for the same statement in a monthly table. | MONTHLY_STMT_STATISTICS        | No monthly reporting        |
| exception.nondml | Whether to process statements other than SELECT, OPEN, FETCH, CLOSE, INSERT, UPDATE, and 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 | 1                             | 1                            |
<p>| exception.obtab | Table that contains exception object information $^a$, $^b$ | STMT_EXCEPTIONS_OB            | No object reporting         |</p>
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Omitted value</th>
<th>Null value</th>
</tr>
</thead>
<tbody>
<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 are not 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>
<tr>
<td>exception.purgeresults</td>
<td>If updateresults is in effect, 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)</td>
<td>XADVISOR_RULES</td>
<td>XADVISOR_RULES</td>
</tr>
<tr>
<td>exception.runtab</td>
<td>Table that holds information pertaining to this and other executions of the advisor, such as number and time range of exceptions reported.</td>
<td>XADVISOR_RUNS</td>
<td>XADVISOR_RUNS</td>
</tr>
<tr>
<td>exception.startdate</td>
<td>The first date for which exceptions are evaluated</td>
<td>CURRENT_DATE - 1 DAY</td>
<td>Unlimited start</td>
</tr>
<tr>
<td></td>
<td>Use format yyyy-mm-dd or supported SQL expressions such as CURRENT_DATE - 1 DAY.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Blank specifies no limit.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>exception.starthist</td>
<td>The date at which to start reporting historical data</td>
<td>No limit</td>
<td>No limit</td>
</tr>
<tr>
<td></td>
<td>Use the same format as the exception start date.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>exception.stopdate</td>
<td>The last date for which exceptions are evaluated</td>
<td>CURRENT_DATE - 1 DAY</td>
<td>Date of most recent record in statement exception table</td>
</tr>
<tr>
<td></td>
<td>Use format yyyy-mm-dd or supported SQL expressions such as CURRENT_DATE - 1 DAY.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Blank specifies Date of most recent record in statement exception table.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>exception.tab</td>
<td>Table that contains statement exception information</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>This table must exist and be populated with data from the DMDBMERG utility.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>exception.tabcreator</td>
<td>Default table owner or creator if subsequent table specifications do not</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>include the owner or creator</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>exception.texttab</td>
<td>Table that contains SQL text</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></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</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>subsequent online viewing</td>
<td></td>
<td></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</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>thresholds</td>
<td></td>
<td></td>
</tr>
<tr>
<td>exception.weekly</td>
<td>Table that contains weekly statement statistics</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Exception Advisor uses this value to compare performance</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>characteristics of an exceptional</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>statement execution with statistics for the same statement in a weekly</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>table.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Parameter**
  - **Description**
  - **Omitted value**
  - **Null value**

- **exception.tab**
  - Table that contains statement exception information
  - This table must exist and be populated with data from the DMDBMERG utility.
  - STMT_EXCEPTIONS
  - STMT_EXCEPTIONS

- **exception.tabcreator**
  - Default table owner or creator if subsequent table specifications do not include the owner or creator
  - BMCSFTWR
  - BMCSFTWR

- **exception.texttab**
  - Table that contains SQL text
  - STMT_TEXT
  - No text reporting

- **exception.updateresults**
  - Whether to update the ADVISOR tables with the results of this run for subsequent online viewing
  - 0 = Do not update tables with results
  - 1 = Update tables with results
  - If 1 is specified, exception.level must be set to DETAIL.
  - 0
  - 1

- **exception.valuetab**
  - Updateresults is 1, table that holds exception values that exceeded thresholds
  - XADVISOR_VALUES
  - XADVISOR_VALUES

- **exception.weekly**
  - Table that contains weekly statement statistics
  - Exception Advisor uses this value to compare performance characteristics of an exceptional statement execution with statistics for the same statement in a weekly table.
  - WEEKLY_STMT_STATISTICS
  - No weekly reporting

---

**Viewing the Exception Advisor reports online**

Use this procedure to view the Exception Advisor online reports.

---

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

**b** This table is optional; if present it must be populated with data from the DMDBMERG utility.
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 119.

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 39 on page 130).

   **Figure 39: Exception Advisor panel (IODEWCMB/I)**

<table>
<thead>
<tr>
<th>IODEWCMB/I</th>
<th>Exception Advisor</th>
<th>11:48:24</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command ===&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Current Data Collector: BA1D</td>
<td>Status: ACTIVE</td>
<td>Data Collection: READY</td>
</tr>
<tr>
<td>Change settings below to locate Advisor Run Table. Then press Enter.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DB2 subsystem where Performance Advisor Database resides: DEFF (blank for list)</td>
<td></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>
<td></td>
</tr>
<tr>
<td>BMCAFD_XARUN</td>
<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.

   The **Current Data Collector** field shows the subsystem ID of the currently selected Data Collector. If no Data Collector is selected, this field is blank.

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

   - ACTIVE
   - DOWN
STOPPING

INVALID

If no Current Data Collector subsystem is 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 z/OS 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.

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

b  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 where the Exception Advisor recorded its results. This is the exception.runtab value from your Exception Advisor batch job.

c  Press **Enter** to view the EXCEPTION ADVISOR-EXCEPTIONS panel (Figure 40 on page 131).

**Figure 40: EXCEPTION ADVISOR-EXCEPTIONS panel**

```plaintext
BMCSftwr.PAXARPT2    --  EXCEPTION ADVISOR-EXCEPTIONS  --       02/06 11:49:18
For +: R-Rules   S-SQL Text V-Hostvars O-Objects 1-More
For :: A-Advice H-History

Exception Advisor runtime: 2010-09-08-21.03.36.211202 level: Detail

<table>
<thead>
<tr>
<th>Timestamp</th>
<th>SSID</th>
<th>Plan</th>
<th>Program</th>
<th>Stmt</th>
<th>Type</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>2010-08-29-21.00.49.328718 DHZ6 AFDPGM @PGM050 269 OPEN CPU TIME</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( CPU TIME</td>
<td>0.647173</td>
<td>Threshold: &gt; 1.000000)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( SYNCIO WAIT PERCENTAGE</td>
<td>43.751999</td>
<td>Threshold: &gt; 30.000000)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( GETPAGE COUNT</td>
<td>7656.000000</td>
<td>Threshold: &gt; 0.000000)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( GETPAGES PER SYNCIO</td>
<td>1.004000</td>
<td>Threshold: &lt; 2.000000)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( GLOBAL CONTENTION</td>
<td>0.000000</td>
<td>Threshold: &gt; 30.000000)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( CHILD L-LOCK WAIT</td>
<td>0.000000</td>
<td>Threshold: &gt; 30.000000)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( PAGE P-LOCK WAIT</td>
<td>0.000000</td>
<td>Threshold: &gt; 30.000000)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

+ 2010-08-29-21.00.49.328718 DHZ6 AFDPGM @PGM050 269 OPEN ELAPSED TM |
| ( ELAPSED TM           | 8.777678 | Threshold: > 1.000000) |
| ( SYNCIO WAIT PERCENTAGE | 43.751999 | Threshold: > 30.000000) |
| ( GETPAGE COUNT        | 7656.000000 | Threshold: > 0.000000) |
| ( GETPAGES PER SYNCIO  | 1.004000 | Threshold: < 2.000000) |
| ( GLOBAL CONTENTION    | 0.000000 | Threshold: > 30.000000) |
| ( CHILD L-LOCK WAIT    | 0.000000 | Threshold: > 30.000000) |
| ( OTHER L-LOCK WAIT    | 0.000000 | Threshold: > 30.000000) |
| ( PAGE P-LOCK WAIT     | 0.000000 | Threshold: > 30.000000) |
```
3 On the EXCEPTION ADVISOR-EXCEPTIONS panel, you can view more information about the exceptions.

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

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

On exception detail lines that start with a . (period), you can execute the following commands by overwriting the . (period) with the appropriate 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
History of statement performance

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

<table>
<thead>
<tr>
<th>Elapsed time</th>
<th>Average History level</th>
<th>Statement Type</th>
<th>Calls</th>
<th>Selects</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;&gt;</td>
<td>0.000021 MONTHLY 2009-06</td>
<td>SELECT</td>
<td>4000</td>
<td>4000</td>
</tr>
<tr>
<td>&gt;&gt;&gt;&gt;&gt;&gt;</td>
<td>1.853088 MONTHLY 2009-10</td>
<td>SELECT</td>
<td>25892</td>
<td>25892</td>
</tr>
<tr>
<td>&gt;&gt;&gt;&gt;&gt;&gt;&gt;</td>
<td>1.783560 WEEKLY 2009-10-11</td>
<td>SELECT</td>
<td>10466</td>
<td>10466</td>
</tr>
<tr>
<td>&gt;&gt;&gt;&gt;&gt;&gt;&gt;</td>
<td>2.011105 WEEKLY 2009-10-18</td>
<td>SELECT</td>
<td>13845</td>
<td>13845</td>
</tr>
<tr>
<td>&gt;&gt;&gt;&gt;&gt;&gt;&gt;</td>
<td>1.741576 WEEKLY 2009-10-25</td>
<td>SELECT</td>
<td>1581</td>
<td>1581</td>
</tr>
<tr>
<td>&gt;&gt;&gt;&gt;&gt;&gt;&gt;</td>
<td>1.349979 WEEKLY 2009-11-01</td>
<td>SELECT</td>
<td>6394</td>
<td>6394</td>
</tr>
<tr>
<td>&gt;&gt;&gt;&gt;&gt;&gt;&gt;</td>
<td>1.349979 MONTHLY 2009-11</td>
<td>SELECT</td>
<td>6394</td>
<td>6394</td>
</tr>
<tr>
<td>&gt;&gt;&gt;&gt;&gt;&gt;&gt;</td>
<td>2.009972 MONTHLY 2009-12</td>
<td>SELECT</td>
<td>6394</td>
<td>6394</td>
</tr>
<tr>
<td>&gt;&gt;&gt;&gt;&gt;&gt;&gt;</td>
<td>0.304685 WEEKLY 2009-12-13</td>
<td>SELECT</td>
<td>5548</td>
<td>5548</td>
</tr>
<tr>
<td>&gt;&gt;&gt;&gt;&gt;&gt;&gt;</td>
<td>0.351906 WEEKLY 2009-12-20</td>
<td>SELECT</td>
<td>2817</td>
<td>2817</td>
</tr>
<tr>
<td>&gt;&gt;&gt;&gt;&gt;&gt;&gt;</td>
<td>2.009972 WEEKLY 2009-12-27</td>
<td>SELECT</td>
<td>214</td>
<td>214</td>
</tr>
<tr>
<td>&gt;&gt;&gt;&gt;&gt;&gt;&gt;&gt;&gt;&gt;&gt;&gt;&gt;&gt;</td>
<td>6.893685 XCPT:01-20-00.59.38.988630</td>
<td>SELECT</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

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
  - 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

DSNDHQ (SYSM/DHQ6 8.1) Member DHQ6 of group DSNDHQ
Plan SPRTST00 DBRM SPRTST00 STATIC DELETE Section 6
Statement 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

DSNDHZ (SYSM/DHZ6 9.1) Member DHZ6 of group DSNDHZ
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

**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:
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.

**Historical comparison of CPU time with SYNCIO WAIT PERCENTAGE**

<table>
<thead>
<tr>
<th>CPU time</th>
<th>SYNCIO WAIT PERCENTAGE Value</th>
<th>History level</th>
<th>Statement Type</th>
<th>Calls</th>
<th>Opens</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.484</td>
<td>MONTHLY 2009-07</td>
<td>CURSOR</td>
<td>56</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>2.924</td>
<td>WEEKLY 2009-11-29</td>
<td>CURSOR</td>
<td>54</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>2.924</td>
<td>MONTHLY 2009-12</td>
<td>CURSOR</td>
<td>54</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>1.226</td>
<td>MONTHLY 2010-01</td>
<td>CURSOR</td>
<td>60</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>1.226</td>
<td>WEEKLY 2010-01-17</td>
<td>CURSOR</td>
<td>60</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>10.109</td>
<td>XCPT: 10 occurrences</td>
<td>OPEN</td>
<td>10</td>
<td>10</td>
<td></td>
</tr>
</tbody>
</table>

**Historical comparison of CPU time with GETPAGE COUNT**

<table>
<thead>
<tr>
<th>CPU time</th>
<th>GETPAGE COUNT</th>
<th>Average</th>
<th>History level</th>
<th>Statement Type</th>
<th>Calls</th>
<th>Opens</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;</td>
<td></td>
<td>3894.53571</td>
<td>MONTHLY 2009-07</td>
<td>CURSOR</td>
<td>56</td>
<td>8</td>
</tr>
<tr>
<td>&gt;</td>
<td></td>
<td>7730.83333</td>
<td>WEEKLY 2009-11-29</td>
<td>CURSOR</td>
<td>54</td>
<td>9</td>
</tr>
<tr>
<td>&gt;</td>
<td></td>
<td>7730.83333</td>
<td>MONTHLY 2009-12</td>
<td>CURSOR</td>
<td>54</td>
<td>9</td>
</tr>
<tr>
<td>&gt;</td>
<td></td>
<td>7730.73333</td>
<td>MONTHLY 2010-01</td>
<td>CURSOR</td>
<td>60</td>
<td>10</td>
</tr>
<tr>
<td>&gt;</td>
<td></td>
<td>7730.73333</td>
<td>WEEKLY 2010-01-17</td>
<td>CURSOR</td>
<td>60</td>
<td>10</td>
</tr>
<tr>
<td>&gt;</td>
<td></td>
<td>46355.00000</td>
<td>XCPT: 10 occurrences</td>
<td>OPEN</td>
<td>10</td>
<td>10</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.

**Historical comparison of CPU time with LOCK WAIT PERCENTAGE**

<table>
<thead>
<tr>
<th>CPU time</th>
<th>LOCK WAIT PERCENTAGE Value</th>
<th>History level</th>
<th>Statement Type</th>
<th>Calls</th>
<th>Deletes</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;</td>
<td>0.000</td>
<td>MONTHLY 2009-09</td>
<td>DELETE</td>
<td>9</td>
<td>3</td>
</tr>
<tr>
<td>&gt;</td>
<td>0.000</td>
<td>WEEKLY 2009-09-27</td>
<td>DELETE</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>&gt;</td>
<td>0.005</td>
<td>MONTHLY 2009-10</td>
<td>DELETE</td>
<td>63</td>
<td>21</td>
</tr>
<tr>
<td>&gt;</td>
<td>0.000</td>
<td>WEEKLY 2009-10-04</td>
<td>DELETE</td>
<td>12</td>
<td>4</td>
</tr>
<tr>
<td>&gt;</td>
<td>0.000</td>
<td>BASELINE 2009-10-10-1300</td>
<td>DELETE</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>&gt;</td>
<td>0.000</td>
<td>WEEKLY 2009-10-11</td>
<td>DELETE</td>
<td>18</td>
<td>6</td>
</tr>
<tr>
<td>&gt;</td>
<td>0.000</td>
<td>WEEKLY 2009-10-18</td>
<td>DELETE</td>
<td>21</td>
<td>7</td>
</tr>
<tr>
<td>&gt;</td>
<td>0.008</td>
<td>WEEKLY 2009-10-25</td>
<td>DELETE</td>
<td>21</td>
<td>7</td>
</tr>
<tr>
<td>&gt;</td>
<td>0.007</td>
<td>WEEKLY 2009-11-01</td>
<td>DELETE</td>
<td>20</td>
<td>6</td>
</tr>
<tr>
<td>&gt;</td>
<td>0.002</td>
<td>MONTHLY 2009-11-12</td>
<td>DELETE</td>
<td>59</td>
<td>19</td>
</tr>
<tr>
<td>&gt;</td>
<td>0.004</td>
<td>WEEKLY 2009-11-08</td>
<td>DELETE</td>
<td>12</td>
<td>4</td>
</tr>
<tr>
<td>&gt;</td>
<td>0.008</td>
<td>WEEKLY 2009-11-15</td>
<td>DELETE</td>
<td>18</td>
<td>6</td>
</tr>
<tr>
<td>&gt;</td>
<td>0.000</td>
<td>WEEKLY 2009-11-22</td>
<td>DELETE</td>
<td>9</td>
<td>3</td>
</tr>
<tr>
<td>&gt;</td>
<td>0.002</td>
<td>WEEKLY 2009-11-29</td>
<td>DELETE</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>&gt;</td>
<td>0.001</td>
<td>MONTHLY 2009-12</td>
<td>DELETE</td>
<td>9</td>
<td>3</td>
</tr>
<tr>
<td>&gt;</td>
<td>0.000</td>
<td>WEEKLY 2009-12-06</td>
<td>DELETE</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>&gt;</td>
<td>0.000</td>
<td>MONTHLY 2010-01</td>
<td>DELETE</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>&gt;</td>
<td>0.000</td>
<td>XCPT:01-21-08.40.28.9272</td>
<td>DELETE</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

---

**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 text:
UPDATE AFDQQA QATBLO1
SET DATETIME_EFF = CURRENT_TIMESTAMP
WHERE COL_A = 13
```
The following example shows a detail report with host variables, SQL text, and objects:

<table>
<thead>
<tr>
<th>Host variables:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td># 1 (INTEGER)</td>
<td>38</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Accessed objects:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Object name    (Type)</td>
<td>BP Name</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------</td>
</tr>
<tr>
<td>AFDQA.QATBL01 (T)</td>
<td>BP0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SQL text:</th>
<th></th>
</tr>
</thead>
</table>

Advice contained in Rules member IODRUEXC 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:** SyncIOWait 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.

**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 explains how to use the Index Component to generate reports based on the selection criteria specified.

This chapter includes the following sections:

- “Index Component report set” on page 137
- “Using panels” on page 142
- “Using online reports” on page 143
- “Viewing reports” on page 144
- “Batch reporting” on page 148

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.

Report hierarchies

Figure 41 on page 138 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:
Figure 41: Index Component reports
Application Group Getpage Volume reports

Figure 42 on page 139 shows the navigation from the Application Group Getpage Volume report.

Figure 42: IODTAPPL reports
Table Getpage Volume reports

Figure 43 on page 140 shows the navigation from the Table Getpage Volume report.

Figure 43: IODTGETV reports
Index Getpage Volume reports

Figure 44 on page 141 shows the navigation from the Index Getpage Volume report.

Figure 44: 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 and 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 when the Data Collector is active. You can 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 during which reporting takes place. That time period consists of one or more intervals from the data in the current trace data sets or in the source data set.

For more information about reporting intervals, see the online Help. To access 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 45 on page 143, 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 PF1 (Help). For detailed information about using online Help, see “Online assistance” on page 45.

Figure 45: Example of navigation--SQL Workload Analysis Menu (ASQEWM1)

Using online reports

As you navigate 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 46 on page 143, the identifiers (in bold print) show the statement list for table RDADMB.DMBTBDYN1, which is running on DB2 subsystem DECX.

Figure 46: Example of navigation--Statement List report
Viewing reports

The reporting process begins at the APPTUNE and Index Component Main Menu.

From the APPTUNE and Index Component Main Menu (Figure 47 on page 144), invoke the SQL Workload Analysis Menu (Figure 48 on page 144) to specify data collection criteria.

**Figure 47: APPTUNE and Index Component Main Menu (IODESEL C)**

```
DOMEPNL3 I       APPTUNE and Index Component Main Menu            14:36:12
Command ====> ________________________________________________________________
BMC24018 *WARNING* Unidentified parameter ignored
Current Data Collector : NO1A    Status : ACTIVE   Data Collection : READY
Select one of the following options. Then press Enter.
  0. Statement Cache      - Analyze dynamic statement cache statistics
  1. SQL Workload         - Analyze current and historical SQL workloads
  2. DB2 Status           - View current DB2 status by subsystem
  3. Explain Interface    - Explain an SQL statement
  4. Application Profiles - Create and maintain application groups
  5. Command Interface    - Issue commands, view responses
  6. Archive Directory    - View/manage the directory of trace archives
  7. About APPTUNE
```

**Figure 48: SQL Workload Analysis Menu (ASQEWAM1)**

```
DOMCPNL3 SQL WORKLOAD ANALYSIS MENU 00:21:17
REPORT CRITERIA:
SOURCE OF DATA . . . : SUBSYSTEM
FOR DB2 SSIDS . . . . : *
DURATION . . . . . : NO TIME LIMIT
INITIAL REPORT . . . : INDEX TABLE GETPAGE VOLUME
APPLICATION PROFILE . : DEFAULT           OWNER :
ZOOM FILTER . . . . . : N
PERFORMANCE FILTER . : NONE
SELECT ONE OF THE FOLLOWING OPTIONS.
  1. WORKLOAD ANALYSIS - BEGIN WORKLOAD REPORTING
  2. INITIAL REPORT    - SELECT THE INITIAL REPORT TO BE VIEWED
  3. REPORT TYPE       - SELECT THE REPORT TYPE (DATA OR GRAPH)
  4. APPLICATION PROFILE - SELECT THE PROFILE FOR GROUP REPORTING
  5. TIME INTERVAL     - SPECIFY THE TIME FRAME FOR REPORTING
  6. DATA SOURCE, DB2(S) - SELECT DATA SOURCE AND DB2 SUBSYSTEMS
  7. ZOOM, PERFORMANCE FILTER - SPECIFY OPTIONS FOR FASTER REPORT ACCESS
```

Application groups and application profiles

SQL Performance enables you to define application groups and application profiles.

Application groups contain the plans, programs, users, DB2s, and correlation IDs that you select, and application profiles contain the application groups you define.
With this information you can select the aspects of a DB2 workload that 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.

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. Only the time spent executing an SQL statement is measured. Measurements based on IN-DB2 time are less focused and overestimate the actual cost of SQL statement execution.

*Figure 49 on page 145* 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 49 on page 145* depicts the three distinct phases of SQL statement execution: prologue time, IN-SQL elapsed time, and epilogue time.

**Figure 49: SQL statement execution measurement**

*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.

You can issue an EXPLAIN command from any SQL statement reported on the SQL Statement Text report or by using the Explain Object Specification panel (PSSPA115).
Explain output is displayed in the Explain Results panel (Figure 50 on page 147) 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 50: Explain Results panel (PSSPW200)

FILE COMMANDS OPTIONS HELP
--------------------------------------------- Scroll ===> CSR
PSSPW200 Explain Results for SQLTEXT
Command ===>                                                  More: >
More: >
Actions: S H R RS RW RI XD XS XP W T C D U IM SA
LABEL STMTNO COST*RATE SQL-STATEMENT
C XD01       117 1012.256592 SELECT * FROM PXB.EMP WHERE LASTNAME = 'BAKER'
COST*RATE QB PL MIX QTYPE METH ACC MTCH IX TBNAME IXNAME NU JO G
XD01 1012.2565 1 1 0 SELECT 0 R 0 N EMP N N N N
*************************************************************************** Bottom of Data****************************************************************************

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, view the Quick Course SQL Performance for DB2 - Using the Explain Interface.

You must have a BMC Support ID to view the Quick Course.

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 lets you 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 Table/Index Breakdown Report (IODSTXDT)**

The Batch Table/Index Breakdown report displays the full text of the selected SQL statement and an overview of GETPAGE and I/O activity for the tables (and their indexes) that have been accessed by the corresponding dynamic SQL statement (when SQL text summarization is enabled). This report displays one line of statistics for tables and one line for indexes and individual lines for each table and index. Only tables and indexes with GETPAGE activity are reported.

**Batch Application Group GETPAGE Volume Report (IODTAPPL)**

The Batch Application Group GETPAGE Volume report displays an overview of GETPAGE and I/O activity by application group. The groups that are reported are the groups that are defined by a specified application profile. The report displays one line of statistics for each application group in the selected application profile. Only groups with GETPAGE activity are reported. Objects encountered in the data that do not match any specification (explicit or implicit) in any group are reported in a group called DEFAULT_GROUP.

**Batch Table GETPAGE Volume Report (IODTGETV)**

The Batch Table GETPAGE Volume report provides an overview of GETPAGE and I/O activity for all tables (and their indexes) on the sysplex, ordered by the volume of GETPAGEs associated with the tables. This report lists one line of statistics for each table. Only tables with GETPAGE activity are reported.
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 chapter relating to batch utilities in the *System and SQL Performance for DB2 Administrator Guide*.

Batch report qualifiers

You can use batch qualifiers with each of the reports.

Table 12 on page 150 shows these qualifiers.

<table>
<thead>
<tr>
<th>Report</th>
<th>IODDSTAT</th>
<th>IODIGETV</th>
<th>IODITABL</th>
<th>IODSTMITS</th>
<th>IODSTXTDT</th>
<th>IODTAPPL</th>
<th>IODTGETV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report description</td>
<td>Subsystem Getpage Volume</td>
<td>Index Getpage Volume</td>
<td>Table Information for an Index</td>
<td>Statement List</td>
<td>Table/Index Breakdown</td>
<td>Application Group Getpage Volume</td>
<td>Table Getpage Volume</td>
</tr>
<tr>
<td>DB2</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Application Group</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operator ID</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Conn ID / WP number</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Corr ID / WP ID</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Plan name</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>DBRM / PG / Server</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Report</td>
<td>IODDSTAT</td>
<td>IODIGETV</td>
<td>IODITABL</td>
<td>IODSTMTS</td>
<td>IODSTXDT</td>
<td>IODTAPPL</td>
<td>IODTGETV</td>
</tr>
<tr>
<td>---------------------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
</tr>
<tr>
<td>Buffer pool ID</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Database name</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Page set</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Table name</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Table creator</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
Using the Index Component

This chapter explains how to use the Index Component of SQL Performance to extend the capability of APPTUNE object analysis by collecting and reporting on column usage data for SQL statements.

This chapter includes the following sections:

- “Overview of the Index Component” on page 153
- “Preparing to use What-If Index” on page 154
- “Identifying What-If Index candidates” on page 157
- “Using the What-If Index function” on page 161

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
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 “Index Component reports” on page 137.

**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 Administrator 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 51 on page 154).

**Figure 51: APPTUNE and Index Component Main Menu (DOMEPNL3)**

```
DOMEPNL3 I           APPTUNE and Index Component Main Menu            14:36:12
Command ===> ________________________________________________________________
Current Data Collector : NO1A    Status : ACTIVE   Data Collection : READY
Select one of the following options. Then press Enter.
_  0. Statement Cache      - Analyze dynamic statement cache statistics
  1. SQL Workload         - Analyze current and historical SQL workloads
  2. DB2 Status           - View current DB2 status by subsystem
  3. Explain Interface    - Explain an SQL statement
  4. Application Profiles - Create and maintain application groups
  5. Command Interface    - Issue commands, view responses
  D. Archive Directory    - View/manage the directory of trace archives
  Z. About APPTUNE
```
2 Select SQL Workload, and press Enter.

The SQL Workload Analysis Menu is displayed (Figure 52 on page 155). 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 52: SQL Workload Analysis Menu**

<table>
<thead>
<tr>
<th>Command</th>
<th>SQL WORKLOAD ANALYSIS MENU 00:21:17</th>
</tr>
</thead>
<tbody>
<tr>
<td>REPORT CRITERIA:</td>
<td></td>
</tr>
<tr>
<td>SOURCE OF DATA</td>
<td>SUBSYSTEM</td>
</tr>
<tr>
<td>FOR DB2 SSIDS</td>
<td>*</td>
</tr>
<tr>
<td>START TIME</td>
<td>01/23/2013 21:22:59</td>
</tr>
<tr>
<td>DURATION</td>
<td>NO TIME LIMIT</td>
</tr>
<tr>
<td>INITIAL REPORT</td>
<td>INDEX TABLE GETPAGE VOLUME</td>
</tr>
<tr>
<td>APPLICATION PROFILE</td>
<td>DEFAULT</td>
</tr>
<tr>
<td>OWNER</td>
<td>ZOOM FILTER</td>
</tr>
<tr>
<td>PERFORMANCE FILTER</td>
<td>NONE</td>
</tr>
</tbody>
</table>

Select one of the following options.

6 1. WORKLOAD ANALYSIS - BEGIN WORKLOAD REPORTING
2. INITIAL REPORT - SELECT THE INITIAL REPORT TO BE VIEWED
3. REPORT TYPE - SELECT THE REPORT TYPE (DATA OR GRAPH)
4. APPLICATION PROFILE - SELECT THE PROFILE FOR GROUP REPORTING
5. TIME INTERVAL - SPECIFY THE TIME FRAME FOR REPORTING
6. DATA SOURCE, DB2(S) - SELECT DATA SOURCE AND DB2 SUBSYSTEMS
7. ZOOM, PERFORMANCE FILTER - SPECIFY OPTIONS FOR FASTER REPORT ACCESS

3 Select Data source, DB2(s) and press Enter to display the Data Source panel (Figure 53 on page 155).

**Figure 53: Data Source panel (DOMEPNL3)**

<table>
<thead>
<tr>
<th>Command</th>
<th>Data Source 14:42:09</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type the data source for reporting below. Then Exit.</td>
<td></td>
</tr>
<tr>
<td>Data source . . . . . S (D =Data set)</td>
<td></td>
</tr>
<tr>
<td>Data set . . . . . . . AFDQA.Q01A.ACCT.ARCHIVE.GDG.G0272V00</td>
<td></td>
</tr>
<tr>
<td>DB2 SSID list . . . . . *___ ___ ( * for all)</td>
<td></td>
</tr>
</tbody>
</table>

4 On the Data Source panel, specify data source information for reporting, and press PF3.

**WARNING**

Index analysis is sensitive to certain objects and their statistics, and to the SQL executed against those objects. For this reason 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 52 on page 155), select 
Time interval and press Enter.

The Select Analysis Interval report (Figure 54 on page 156) is displayed.

**Figure 54: Select Analysis Interval report (DOMEPNL4)**

```
DMEPNL4 I View a Report Command ===> ________________________ Scroll ===> CSR
BMCSftwrs.SQMINTVD -- SELECT ANALYSIS INTERVAL -- 03/07 14:43:20
Select a single interval or a range of intervals. Type "S" beside a single
interval or beside the first and last intervals in the range, and press
Enter. Exit when finished.
Actions: F-Filtering Criteria

+------------------ Interval ----------------+        Number     Active
Begin               End                 Duration  DB2   Entries    Filter
------------------  ------------------  --------  ----  -------   --------
+  03/07/11  00:00:00  03/07/11  14:43:19  14:43:20  DEC9     3421   AMD1
+  03/07/11  00:00:00  03/07/11  14:43:19  14:43:19  DEDR    25231   AMD1
+  03/06/11  00:00:00  03/06/11  23:59:59  23:59:59  DEC9     3425   AMD1
+  03/06/11  00:00:00  03/06/11  23:59:59  23:59:59  DEDR     9535   AMD1
+  03/05/11  12:07:59  03/05/11  23:59:59  11:52:00  DEC9     3497   AMD1
+  03/05/11  12:07:59  03/05/11  23:59:59  11:52:00  DEDR     9245   AMD1
+  03/04/11  13:56:00  03/04/11  23:59:59  10:03:59  DEC9     3417   AMD1
+  03/04/11  13:55:00  03/04/11  23:59:59  10:04:59  DEDR    14739   AMD1
+  03/04/11  00:00:00  03/04/11  13:55:59  13:55:59  DEC9     3319   AMD1
+  03/04/11  00:00:00  03/04/11  13:54:59  13:54:59  DEDR    32912   AMD1
```

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 PF3 when finished.

7 At the SQL Workload Analysis menu, select Initial report, and press Enter.

The SQL Workload Initial Analysis Level panel is displayed (Figure 55 on page 156).

**Figure 55: SQL Workload Initial Analysis Level panel (DOMEPNL3)**

```
DMCPNL3 SQL WORKLOAD INITIAL ANALYSIS LEVEL 00:28:24
COMMAND ===> __________________________
CURRENT INITIAL REPORT : INDEX TABLE GETPAGE VOLUME

SELECT ONE OF THE FOLLOWING INITIAL REPORTS TO DISPLAY, THEN PRESS ENTER.
22 APPTUNE REPORTS:
  1. DB2 SUBSYSTEM ID
  2. PROGRAM/DBRM
  3. PLAN
  4. USER/OPERATOR ID
  5. APPLICATION GROUP
  6. CONNECTION ID
  7. SQL STATEMENT
  8. SQL ERROR CODE
  9. CORRID
 10. OBJECTS
 11. CLIENT APPLICATION NAME
 12. CLIENT WORKSTATION NAME
 13. CLIENT USER ID
 14. INTERVAL
 15. SAP
 16. LOGICAL DB2 (DS GROUP OR SSID)
 17. REQUESTING LOCATION
 18. IMPLICIT QUALIFIER
 21. SUBSYSTEM GETPAGE VOLUME
 22. TABLE GETPAGE VOLUME
 23. INDEX GETPAGE VOLUME
 24. APPLICATION GROUP GETPAGE VOLUME
```
8 In the APPTUNE Reports field, select Table Getpage Volume, and press Enter.

9 At the SQL Workload Analysis menu, select Workload analysis, and press Enter.

The Table Getpage Volume Report is displayed. From this panel, you can zoom to other reports for more detailed table statistics, index statistics, a CRUD matrix for a table, or a list of SQL statements for a table.

For more information, see “Identifying What-If Index candidates” on page 157.

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.

To identify What-If Index candidates

1 Use the information in “Preparing to use What-If Index” on page 154 to define options for reporting.

Examine the Table Getpage Volume report (Figure 56 on page 158) 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.

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 RDADB.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%.

### Table Getpage Volume report (IODTGETV)

<table>
<thead>
<tr>
<th>Creator</th>
<th>Name</th>
<th>Number</th>
<th>Index%</th>
<th>Sync I/O AvgTime</th>
<th>Async I/O AvgTime</th>
<th>Total Pages</th>
<th>Change Stmts</th>
<th>Stmt %</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>RDADB DMBTBDYN1</td>
<td>168032</td>
<td>0.0%</td>
<td>2.7 0.00000</td>
<td></td>
<td>61410</td>
<td>49128</td>
<td>80.0%</td>
</tr>
<tr>
<td>+</td>
<td>SYSIBM SYSTABLES</td>
<td>24647</td>
<td>50.0%</td>
<td>2.0 0.00000</td>
<td></td>
<td>12299</td>
<td>12288</td>
<td>99.9%</td>
</tr>
<tr>
<td>+</td>
<td>SYSIBM SYSTABLESPACE</td>
<td>12292</td>
<td>66.7%</td>
<td>1.0 0.00000</td>
<td></td>
<td>12290</td>
<td>12288</td>
<td>100%</td>
</tr>
<tr>
<td>+</td>
<td>SYSIBM SYSCOLUMNS</td>
<td>605</td>
<td>11.7%</td>
<td>60.5 0.00000</td>
<td></td>
<td>10</td>
<td>2</td>
<td>40.0%</td>
</tr>
<tr>
<td>+</td>
<td>BMCDAA52 V52_SQLX_RULES</td>
<td>76</td>
<td>0.0%</td>
<td>38.0 0.00052</td>
<td>2</td>
<td>29</td>
<td>2</td>
<td>0.0%</td>
</tr>
<tr>
<td>+</td>
<td>RDADB PLAN_TABLE</td>
<td>20</td>
<td>0.0%</td>
<td>5.0 0.00028</td>
<td>2</td>
<td>28</td>
<td>2</td>
<td>2.0%</td>
</tr>
<tr>
<td>+</td>
<td>RDADB DSN_DETCOST_TABLE</td>
<td>16</td>
<td>0.0%</td>
<td>4.0 0.00069</td>
<td>2</td>
<td>60</td>
<td>4</td>
<td>2.0%</td>
</tr>
<tr>
<td>+</td>
<td>RDADB DSN_PREDICAT_TABLE</td>
<td>16</td>
<td>0.0%</td>
<td>4.0 0.00269</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>2.0%</td>
</tr>
<tr>
<td>+</td>
<td>RDADB DSN_STATEMENT_TABLE</td>
<td>12</td>
<td>0.0%</td>
<td>3.0 0.00051</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>2.0%</td>
</tr>
<tr>
<td>+</td>
<td>RODMB STRUCTURE_TABLE</td>
<td>7</td>
<td>0.0%</td>
<td>7.0 0.00014</td>
<td>1</td>
<td>59</td>
<td>1</td>
<td>0.0%</td>
</tr>
<tr>
<td>+</td>
<td>SYSIBM SYSTABLEPART</td>
<td>2</td>
<td>66.7%</td>
<td>1.0 0.00000</td>
<td></td>
<td>2</td>
<td>2</td>
<td>0.0%</td>
</tr>
</tbody>
</table>

2 To view table statistics, type T over the plus sign (+) beside the table, and press Enter.

The Table Statistics report is displayed (Figure 57 on page 158). This report shows the physical characteristics of the table, along with RUNSTAT statistics that you can use to analyze the index.

### Table Statistics report (ASQEQRPN)

<table>
<thead>
<tr>
<th>Logical DB Name</th>
<th>Card</th>
<th>Table Owner</th>
<th>Table Name</th>
<th>Pctpages</th>
<th>Last Stats</th>
<th>Record Length</th>
<th>Colcount</th>
<th>ColCard</th>
<th>Pkno</th>
<th>Indexes</th>
</tr>
</thead>
<tbody>
<tr>
<td>DECX</td>
<td>-1.000</td>
<td>RDADB</td>
<td>DMBTBDYN1</td>
<td>-1</td>
<td>0001-01-01-00.00.00.000000</td>
<td>44</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3 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 58 on page 159). This report helps you to determine whether indexes exist for a table. In this example, the table has no defined indexes.

### Figure 58: Defined Indexes for a Table report (IODTSTIN)

<table>
<thead>
<tr>
<th>Logical DB2 Name: DECX</th>
<th>Card</th>
<th>-1.000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table Owner : RDADMB</td>
<td>Number Pages:</td>
<td>-1</td>
</tr>
<tr>
<td>Table Name : DMBTBDYN</td>
<td>Pctpages :</td>
<td>-1.00</td>
</tr>
<tr>
<td>Database : DMBDBDYN</td>
<td>Colcount :</td>
<td>0</td>
</tr>
<tr>
<td>Tablespace : DMBTSDYN</td>
<td>Record Length :</td>
<td>44</td>
</tr>
<tr>
<td>Last Stats : 0001-01-01-00.00.00.000000</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Figure 59: Table Getpage Volume report (IODTGETV)

<table>
<thead>
<tr>
<th>Creator</th>
<th>Name</th>
<th>Number Index%</th>
<th>Stmt AvgTime</th>
<th>Sync I/O</th>
<th>Async I/O</th>
<th>Total Pages</th>
<th>Change Stats</th>
<th>Change Stmt</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>RDADMB</td>
<td>168032</td>
<td>0.0%</td>
<td>2.7</td>
<td>0.00000</td>
<td>61410</td>
<td>49128</td>
<td>80.0%</td>
</tr>
<tr>
<td>+</td>
<td>SYSIBM</td>
<td>24647</td>
<td>50.0%</td>
<td>2.0</td>
<td>0.00000</td>
<td>12299</td>
<td>12288</td>
<td>99.9%</td>
</tr>
<tr>
<td>+</td>
<td>SYSIBM</td>
<td>12292</td>
<td>66.7%</td>
<td>1.0</td>
<td>0.00000</td>
<td>12290</td>
<td>12288</td>
<td>100%</td>
</tr>
<tr>
<td>+</td>
<td>SYSIBM</td>
<td>805</td>
<td>11.7%</td>
<td>7.0</td>
<td>0.00143</td>
<td>7</td>
<td>0</td>
<td>0.0%</td>
</tr>
<tr>
<td>+</td>
<td>BMCDAA52</td>
<td>1464</td>
<td>2.0%</td>
<td>2.0</td>
<td>0.00000</td>
<td>4</td>
<td>0</td>
<td>0.0%</td>
</tr>
<tr>
<td>+</td>
<td>RDADMB</td>
<td>20</td>
<td>0.0%</td>
<td>5.0</td>
<td>0.00284</td>
<td>3</td>
<td>0</td>
<td>0.0%</td>
</tr>
<tr>
<td>+</td>
<td>RDADMB</td>
<td>16</td>
<td>0.0%</td>
<td>4.0</td>
<td>0.000690</td>
<td>2</td>
<td>0</td>
<td>0.0%</td>
</tr>
<tr>
<td>+</td>
<td>RDADMB</td>
<td>16</td>
<td>0.0%</td>
<td>4.0</td>
<td>0.000269</td>
<td>2</td>
<td>0</td>
<td>0.0%</td>
</tr>
<tr>
<td>+</td>
<td>RDADMB</td>
<td>12</td>
<td>0.0%</td>
<td>3.0</td>
<td>0.00051</td>
<td>1</td>
<td>0</td>
<td>0.0%</td>
</tr>
<tr>
<td>+</td>
<td>SYSIBM</td>
<td>1464</td>
<td>2.0%</td>
<td>2.0</td>
<td>0.001385</td>
<td>4</td>
<td>2</td>
<td>0.0%</td>
</tr>
<tr>
<td>+</td>
<td>SYSDATABASE</td>
<td>7</td>
<td>0.0%</td>
<td>7.0</td>
<td>0.000143</td>
<td>2</td>
<td>59</td>
<td>1</td>
</tr>
<tr>
<td>+</td>
<td>RDADMB</td>
<td>2</td>
<td>0.0%</td>
<td>1.0</td>
<td>0.000000</td>
<td>0</td>
<td>0</td>
<td>0.0%</td>
</tr>
<tr>
<td>+</td>
<td>RDADMB</td>
<td>2</td>
<td>0.0%</td>
<td>1.0</td>
<td>0.000000</td>
<td>0</td>
<td>0</td>
<td>0.0%</td>
</tr>
</tbody>
</table>

4 Press PF3 until you return to the Table Getpage Volume report (Figure 59 on page 159).

5 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 60 on page 160). 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 60: Table CRUD Matrix report (IODTCRUD)**

<table>
<thead>
<tr>
<th>Logical DB2 Name: DECX</th>
<th>Table Creator: RDADMB</th>
<th>Table Name: DMBTBDYN1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Column Name</td>
<td>Indexable</td>
<td>Indexable</td>
</tr>
<tr>
<td>COL01</td>
<td>61410</td>
<td>0</td>
</tr>
<tr>
<td>COL02</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>COL03</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>COL04</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>COL05</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>COL06</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>COL07</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>COL08</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

6 Press PF3 to return to the Table Getpage Volume report (Figure 61 on page 160).

**Figure 61: Table Getpage Volume report (IODTGETV)**

<table>
<thead>
<tr>
<th>Creator Name</th>
<th>Table Name</th>
<th>Number</th>
<th>Index% /Stmt AvgTime</th>
<th>Number</th>
<th>AvgTime</th>
<th>Number</th>
<th>Pages</th>
<th>Stmts</th>
<th>Stmt %</th>
</tr>
</thead>
<tbody>
<tr>
<td>RDADMB</td>
<td>DMBTBDYN1</td>
<td>168032</td>
<td>0.0%</td>
<td>2.7</td>
<td>0.00000</td>
<td>1</td>
<td>0.00861</td>
<td>61410</td>
<td>49128</td>
</tr>
<tr>
<td>SYSIBM</td>
<td>SYSTABLES</td>
<td>24647</td>
<td>50.0%</td>
<td>2.0</td>
<td>0.00000</td>
<td>0</td>
<td>0</td>
<td>12299</td>
<td>12288</td>
</tr>
<tr>
<td>SYSIBM</td>
<td>SYSTABLESPACE</td>
<td>12292</td>
<td>66.7%</td>
<td>1.0</td>
<td>0.00000</td>
<td>0</td>
<td>0</td>
<td>12290</td>
<td>12288</td>
</tr>
<tr>
<td>SYSIBM</td>
<td>SYSCOLUMNS</td>
<td>605</td>
<td>11.7%</td>
<td>60.5</td>
<td>0.00000</td>
<td>0</td>
<td>0</td>
<td>12290</td>
<td>12288</td>
</tr>
<tr>
<td>BMCDAA52</td>
<td>V52_SQLX_RULES</td>
<td>76</td>
<td>0.0%</td>
<td>38.0</td>
<td>0.00000</td>
<td>2</td>
<td>0.01385</td>
<td>29</td>
<td>0.0%</td>
</tr>
<tr>
<td>RDADMB</td>
<td>PLAN_TABLE</td>
<td>20</td>
<td>0.0%</td>
<td>5.0</td>
<td>0.00000</td>
<td>0</td>
<td>0</td>
<td>59</td>
<td>0.0%</td>
</tr>
<tr>
<td>RDADMB</td>
<td>DSN_DET_COST_TABLE</td>
<td>16</td>
<td>0.0%</td>
<td>4.0</td>
<td>0.00036</td>
<td>2</td>
<td>0.00136</td>
<td>2</td>
<td>0.0%</td>
</tr>
<tr>
<td>RDADMB</td>
<td>DSN_PREDICAT_TABLE</td>
<td>16</td>
<td>0.0%</td>
<td>4.0</td>
<td>0.000269</td>
<td>0</td>
<td>0.0136</td>
<td>2</td>
<td>0.0%</td>
</tr>
<tr>
<td>RDADMB</td>
<td>DSN_STATEMENT_TABLE</td>
<td>12</td>
<td>0.0%</td>
<td>3.0</td>
<td>0.0005</td>
<td>0</td>
<td>0.00060</td>
<td>4</td>
<td>0.0%</td>
</tr>
<tr>
<td>SYSIBM</td>
<td>SYSDATABASE</td>
<td>7</td>
<td>0.0%</td>
<td>7.0</td>
<td>0.000143</td>
<td>1</td>
<td>0.00087</td>
<td>2</td>
<td>0.0%</td>
</tr>
<tr>
<td>RDADMB</td>
<td>DSN_STATEMENT_TABLE</td>
<td>2</td>
<td>0.0%</td>
<td>1.0</td>
<td>0.000000</td>
<td>0</td>
<td>0.00000</td>
<td>2</td>
<td>0.0%</td>
</tr>
<tr>
<td>SYSIBM</td>
<td>SYSTABLEPART</td>
<td>2</td>
<td>0.0%</td>
<td>1.0</td>
<td>0.000000</td>
<td>0</td>
<td>0.00000</td>
<td>2</td>
<td>0.0%</td>
</tr>
</tbody>
</table>

7 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 62 on page 161). This report shows all statements that use the specified table.
8 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 63 on page 161).

9 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” on page 161.

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.
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 162
- “Dropping an index by using What-If Index” on page 168
- “Updating statistics on an index by using What-If Index” on page 170
- “Saving What-If Index DDL changes” on page 173
- “Importing What-If Index changes to another statement for a table” on page 175
- “Adding an expression to a What-If Index” on page 179

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.

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

- Follow the instructions in “Preparing to use What-If Index” on page 154 to specify options in preparation for your analysis.
- Follow the instructions in “Identifying What-If Index candidates” on page 157 to locate a table that could benefit from the addition of an index.
To add an index

1 On the Statement Text report (Figure 64 on page 163), type X (for Explain Stmt) over the plus sign (+) beside the SQL statement text to be Explained, and press Enter.

Figure 64: Statement Text report (IODSTXDH)

The Explain Results panel (Figure 65 on page 163) is displayed.

Figure 65: 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 66 on page 164). Notice that the objects are created in database PSSnnn (where nnnn is a generated random number). The objects are created when you perform an Explain.
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.

**Figure 66: Explain Results panel with cloned objects**

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 67 on page 164). Specifying a What-If Index (W) on the table displays default values for the new index as a starting point. Specifying a W on an existing index in the clone uses that index definition as a starting point for your new index.

**Figure 67: Specify Index Attributes panel (PSSPWIXA)**

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 68 on page 165).
Tip

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.

Figure 68: Select Key Columns panel (PSSPW300)

For each column of the key, enter the key column sequence number and the order (Ascending, Descending or Random).

SEQ ORDER COLNO COLUMN NAME     CARD TYPE LEN HI2KEY L
A  1 EMPNO                       -1 CHAR  6  4040404040404040  4
A  2 FIRSTNME                    -1 VARCHAR 12 4040404040404040  4
A  3 MIDINIT                     -1 CHAR  1  4040404040404040  4
A  4 LASTNAME                    -1 VARCHAR 15 4040404040404040  4
A  5 WORKDEPT                    -1 CHAR  3  4040404040404040  4
A  6 PHONENO                     -1 CHAR  4  4040404040404040  4
A  7 HIREDATE                    -1 DATE  4  4040404040404040  4
A  8 JOB                         -1 CHAR  8  4040404040404040  4
A  9 EDLEVEL                     -1 SMALLINT 2  4040404040404040  4
A 10 SEX                         -1 CHAR  1  4040404040404040  4

5 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 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 179.

6 When you finish entering key column information, press Enter to display the Update Index Access Path Statistics panel (Figure 69 on page 166).
Some of the values on this panel are gleaned from existing statistics.

**Figure 69: Update Index Access Path Statistics panel (PSSPWIXS)**

Some of the values on this panel are gleaned from existing statistics.

**Figure 69: 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>
</tbody>
</table>

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 70 on page 166).

**Figure 70: Update First Key Column Statistics panel (PSSPWED)**

Some of the values on this panel are gleaned from existing statistics.

**Figure 70: Update First Key Column Statistics panel (PSSPWED)**

<table>
<thead>
<tr>
<th>PSSPWED</th>
<th>UPDATE_FIRST_KEY_COLUMN_STATISTICS</th>
<th>Columns 00001 00031</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command</td>
<td>====&gt;</td>
<td>Scroll ====&gt; CSR</td>
</tr>
</tbody>
</table>

FREQUENCY VALUE - VARCHAR 15 -

****** ************************************************ Top of Data ************************************************
000001 0.500000000000 CCCCCCCCCCCCCC

****** ************************************************ Bottom of Data ************************************************

c  Edit the default values as needed (Figure 71 on page 166).

**Figure 71: Edited first key column statistics**

Some of the values on this panel are gleaned from existing statistics.

**Figure 71: Edited first key column statistics**

<table>
<thead>
<tr>
<th>PSSPWED</th>
<th>UPDATE_FIRST_KEY_COLUMN_STATISTICS</th>
<th>Columns 00001 00031</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command</td>
<td>====&gt;</td>
<td>Scroll ====&gt; CSR</td>
</tr>
</tbody>
</table>

FREQUENCY VALUE - VARCHAR 15 -

****** ************************************************ Top of Data ************************************************
000001 0.500000000000 ALLISTER
000002 0.300000000000 BAKER
000003 0.200000000000 ELDON

****** ************************************************ Bottom of Data ************************************************

d  Press **PF3** until the Explain Results panel is displayed (Figure 72 on page 167).

Notice that the new index is displayed in the list. At this point, the index has not yet been created; it is created 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 performing incremental changes followed by Explains lets you 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 compares each successive What-If Index Explain to the previous Explain.

Figure 72: Explain Results panel showing new index

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 73 on page 167). The WI01 label represents the Explain on the statement for the cloned objects. The asterisk and highlighting make it easier to see the differences between the Explains.

Figure 73: Explain Results panel after new index is created

9 To show the cloned objects, type S in the space beside the clone and press Enter.

The cloned objects are displayed (Figure 74 on page 168).
10 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 XD in the space beside the clone, and press Enter.

The results of your changes are displayed in the Explain Results panel. You can then 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 154
- “Identifying What-If Index candidates” on page 157
- “Adding an index by using What-If Index” on page 162

The examples in this section are continued from the task “Adding an index by using What-If Index” on page 162.
To drop an index

1. On the Explain Results panel (Figure 75 on page 169), type D beside the index that you want to drop, and press Enter.

Figure 75: Specifying an index to be dropped in What-If Index

The index to be dropped from the clone is removed from the display (Figure 76 on page 169). At this point, the index has not yet been dropped even though it is no longer displayed on the panel; it is 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 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 compares each successive What-If Index Explain to the previous Explain.
2 To perform a dynamic Explain, type XD in the space beside the clone, and press Enter.

The results of the Explain are displayed (Figure 77 on page 170). The WI02 label represents the Explain on the cloned objects with new index myindex and having dropped index XEMP2.

Figure 77: Explain Results panel after index is dropped

3 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 154
- “Identifying What-If Index candidates” on page 157
- “Adding an index by using What-If Index” on page 162

The examples in this section are continued from the previous task, “Dropping an index by using What-If Index” on page 168.
To update statistics

1. At the Explain Results panel (Figure 78 on page 171), type `U` in the space beside the index and press Enter.

**Figure 78: Specifying statistics to be updated for an index using What-If Index**

The Update Index Access Path Statistics panel is displayed (Figure 79 on page 171).

**Figure 79: Update Index Access Path Statistics panel (PSSPWIXS)**

2. Update the statistical values as needed (Figure 78 on page 171).

**Note**

To update statistics for the first key column, select the **Update first key column statistics** option. You can modify this information on the Update First Key Column Statistics panel that is displayed.

**Figure 80: Update Index Access Path Statistics panel with updated statistics**

---

Chapter 9 Using the Index Component 171
First Key Card   100
Full Key Card    100
Cluster Ratio    90              (0-100)
Leaf Pages . . . 20
Number of Levels 2

_ Update first key column statistics

The updated statistics are displayed in Figure 80 on page 171

3 Press PF3 to display the Explain Results panel (Figure 81 on page 172).

Figure 81: Explain Results panel showing updated statistics

FILE COMMANDS OPTIONS HELP
-------------------------------------------------------------------------------
PSSPW200 Explain Results for SQLTEXT
Command ===>                                                  Scroll ===> CSR
More:       >
Actions: S H R RS RW RI XD XS XP W P T C U IM SA
XD CLONE  Objects in database PSS60364
STMTNO SQL-STATEMENT
117 SELECT * FROM PXB.EMP WHERE LASTNAME = 'BAKER'
CREATOR TBNAME          CARD    NPAGES COMP  NACTIVE PARENT CHILD BPOOL
PXB     EMP               -1        -1   -1       -1      0     0 BP0
CREATOR IXNAME          1STKEY     FULLKEY NLEAF NLEVELS CLUSTERRAT
PXB     myindex            100         100    20       2 0.90000000
LBL   STMTNO     COST*RATE SQL-STATEMENT
WI02     117    344.638184 SELECT * FROM PXB.EMP WHERE LASTNAME = 'BAKER'
WI01*    117    344.638184 SELECT * FROM PXB.EMP WHERE LASTNAME = 'BAKER'
XD01     117   1012.256592 SELECT * FROM PXB.EMP WHERE LASTNAME = 'BAKER'
COST*RATE QB PL MIX QTYPE  METH ACC MTCH IX TBNAME   IXNAME   NU J O G
WI02  344.63818  1  1   0 SELECT    0 I      1 N  EMP      myindex  N  N N N
WI01* 344.63818  1  1   0 SELECT    0 I      1 N  EMP      myindex  N  N N N
XD01  1012.2565  1  1   0 SELECT    0 R      0 N  EMP               N  N N N
*******************************************************************************

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 82 on page 173). 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 162)
- Index *XEMP2* dropped (see “Dropping an index by using What-If Index” on page 168)
Statistics updated on index myindex (see “Updating statistics on an index by using What-If Index” on page 170)

Figure 82: Results of dynamic Explain on cloned objects after statistical update

---

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 during the session.

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 to another statement that references the objects during a What-If Index operation, 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 saving your DDL at incremental points enables you to revert to an earlier version.

**Before you begin**

Perform the following procedures to create an index by using cloned objects:

- “Preparing to use What-If Index” on page 154
- “Identifying What-If Index candidates” on page 157
To save What-If Index DDL changes

1. On the Explain Results panel (Figure 83 on page 174), type SA in the space beside the clone for which you want to save the DDL, and press Enter.

   **Figure 83: Explain Results panel (PSSPW200)**

   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 84 on page 174).

   **Figure 84: Save DDL panel (PSSPWD00)**

   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 85 on page 175).
In DDL that is saved from this process, DROP INDEX statements are prefixed with an X. The X must be deleted before you can execute the statement.

---

**Figure 85: Browse saved DDL data set**

```
ISRBROBA RDADAC.SQLX610.CNTL(DDL0047)   Line D000000 DDL Complete
Command ====>                            Scroll ====> CSR
****************************** Top of Data ******************************
-- DDL GENERATED ON Mon Mar 7 15:04:08 2011
XDROP INDEX PXB.XEMP2 ;
COMMIT ;
CREATE INDEX PXB."myindex"
ON     PXB.EMP(LASTNAME ASC) ;
COMMIT ;
UPDATE  SYSIBM.SYSINDEXES
SET     FIRSTKEYCARDF = 100, FULLKEYCARDF = 100, CLUSTERRATIOF =
        0.900000000000, NLEVELS = 2, NLEAF = 20, STATSTIME =
        CURRENT_TIMESTAMP
WHERE   CREATOR = 'PXB'
        AND   NAME = 'myindex' ;
COMMIT ;
INSERT INTO    SYSIBM.SYSCOLDIST
VALUES  (0, CURRENT_TIMESTAMP, 'N', 'PXB', 'EMP', 'LASTNAME',
        X'C1D3D3C9E2E3C5D940404040404040', 'F', 0, '', 1,
        0.500000000000) ;
INSERT INTO    SYSIBM.SYSCOLDIST
VALUES  (0, CURRENT_TIMESTAMP, 'N', 'PXB', 'EMP', 'LASTNAME',
        X'C2C1D2C5D940404040404040404040', 'F', 0, '', 1,
        0.300000000000) ;
INSERT INTO    SYSIBM.SYSCOLDIST
VALUES  (0, CURRENT_TIMESTAMP, 'N', 'PXB', 'EMP', 'LASTNAME',
        X'C5D3C4D6D540404040404040404040', 'F', 0, '', 1,
        0.200000000000) ;
COMMIT ;
****************************** Bottom of Data ******************************
```

---

**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:
To import What-If Index changes

1. On the Explain Results panel (Figure 86 on page 176), type C in the space beside a statement, and press Enter to clone the objects for that statement.

```
FILE COMMANDS OPTIONS HELP
PSSPW200      Explain Results for SQLTEXT
Command ===>                                                  Scroll ===> CSR
More:       >
Actions: S H R RS RW RI XD XS XP W P T C D U IM SA
LBL STMTNO COST*RATE SQL-STATEMENT
C XD01  0 1012.638184 SELECT EMPNO FROM PXB.EMP WHERE JOB LIKE 'MGR%'
C XD01  1012.6381  1 1   0 SELECT  O R 0 N EMP
IIM CLONE  Objects in database PSS18130
STMTNO SQL-STATEMENT
0 SELECT EMPNO FROM PXB.EMP WHERE JOB LIKE 'MGR%
CREATOR TBNAME CARD NPAGES COMP NACTIVE PARENT CHILD BP00
PXB EMP -1 -1 -1 -1 0 0 BP0
CREATOR IXNAME 1STKEY FULLKEY NLEVELS CLUSTERRAT
PXB XEMP2 -1 -1 -1 -1 0.00000000
LBL STMTNO COST*RATE SQL-STATEMENT
C XD01  0 1012.638184 SELECT EMPNO FROM PXB.EMP WHERE JOB LIKE 'MGR%
C XD01  1012.6381  1 1   0 SELECT  O R 0 N EMP
IIM CLONE  Objects in database PSS18130
STMTNO SQL-STATEMENT
0 SELECT EMPNO FROM PXB.EMP WHERE JOB LIKE 'MGR%
CREATOR TBNAME CARD NPAGES COMP NACTIVE PARENT CHILD BP00
PXB EMP -1 -1 -1 -1 0 0 BP0
CREATOR IXNAME 1STKEY FULLKEY NLEVELS CLUSTERRAT
PXB XEMP2 -1 -1 -1 -1 0.00000000
```

The cloned objects are displayed (Figure 87 on page 176).

```
FILE COMMANDS OPTIONS HELP
PSSPW200      Explain Results for SQLTEXT
Command ===>                                                  Scroll ===> CSR
More:       >
Actions: S H R RS RW RI XD XS XP W P T C D U IM SA
IM CLONE  Objects in database PSS18130
STMTNO SQL-STATEMENT
0 SELECT EMPNO FROM PXB.EMP WHERE JOB LIKE 'MGR'
CREATOR TBNAME CARD NPAGES COMP NACTIVE PARENT CHILD BP00
PXB EMP -1 -1 -1 -1 0 0 BP0
CREATOR IXNAME 1STKEY FULLKEY NLEVELS CLUSTERRAT
PXB XEMP2 -1 -1 -1 -1 0.00000000
```

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 88 on page 177). 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.
You do not need to alter the "X" DROP INDEX statement to use it with the IM command. The index is not included in the cloned objects.

Figure 88: Import DDL panel (PSSPWDI0)

PSSPWDIO -------------------------------- Import DDL ---------------------------
Command ===>

Specify a partitioned (include member) or sequential data set for DDL.
If partitioned, member name may be wildcarded to display list of members.
DDL Data Set . . . 'RDADAC.SQL610.CNTL(DDL*)'
Browse Data Set Y (Y/N - Browse DDL data set)
Import DDL . . . Y (Y/N - Import DDL from data set)

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 89 on page 177).

Figure 89: List of members in DDL data set

<table>
<thead>
<tr>
<th>ISRBROM</th>
<th>RDADAC.SQL610.CNTL</th>
<th>Row 00001 of 00015</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command</td>
<td>===&gt;</td>
<td>Scroll ===&gt; CSR</td>
</tr>
<tr>
<td>Name</td>
<td>Prompt</td>
<td>Size</td>
</tr>
<tr>
<td>DDL0008</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DDL0009</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DDL0017</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DDL0018</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DDL0019</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DDL0020</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DDL0021</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DDL0022</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DDL0023</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DDL0024</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DDL0025</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DDL0026</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DDL0027</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DDL0028</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S DDL0047</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>End</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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 90 on page 177).

Figure 90: Browse DDL data set member

<table>
<thead>
<tr>
<th>ISRBROBA</th>
<th>RDADAC.SQL610.CNTL(DDL0047)</th>
<th>Line 00000000 Col 001 080</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command</td>
<td>===&gt;</td>
<td>Scroll ===&gt; CSR</td>
</tr>
<tr>
<td>----------</td>
<td>-------------------------------------</td>
<td>--------------------------------------</td>
</tr>
<tr>
<td>************ Top of Data ************</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-- DDL GENERATED ON Mon Mar 7 15:04:08 2011</td>
<td></td>
<td></td>
</tr>
<tr>
<td>XDROP INDEX PXB.XEMP2 ;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
5 Verify whether this is the member you want to import, then press PF3.

The list of data set members is displayed again.

6 Repeat Step 4 on page 177 and Step 5 on page 178 until you locate the member you want to import, then press PF3.

The Import DDL Member Selection List panel is displayed (Figure 91 on page 178).

**Figure 91: Import DDL Member Selection List panel (PSSPMEML)**

```
PSSPMEML --- Import DDL Member Selection List -------------- Row 7 to 15 of 15
COMMAND ===> SCROLL ===> CSR

Specify 'S' to import or 'B' to browse DDL member(s) and then press ENTER.

MEMBER
- --------
DOL0021
DOL0022
DOL0023
DOL0024
DOL0025
DOL0026
DOL0027
DOL0028
S  DOL0047

********************************************************************** Bottom of data **********************************************************************
```

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 92 on page 178).

**Figure 92: Message displayed when import process is complete**

```
PSSPMEML --- Import DDL Member Selection List -------------- IMPORT complete
COMMAND ===> SCROLL ===> CSR

Specify 'S' to import or 'B' to browse DDL member(s) and then press ENTER.

MEMBER
- --------
S  DOL0047

********************************************************************** Bottom of data **********************************************************************
```
8 Press PF3 until the Explain Results panel is displayed (Figure 93 on page 179).

Notice that index myindex is now displayed in the list with the statistics from the imported DDL.

**Figure 93: Explain Results panel (PSSPW200)**

```
FILE COMMANDS OPTIONS HELP
-------------------------------------------------------------------------------
PSSPW200 Explain Results for SQLTEXT
Command ===>                                                  Scroll ===> CSR
More:       >
Actions: S H R RS RW RI XD XS XP W P T C D U IM SA
XD CLONE Objects in database PSS18130
STMTNO SQL-STATEMENT
  0 SELECT EMPNO FROM PXB.EMP WHERE JOB LIKE 'MGR%'
CREATOR TBNAME CARD NPAGES COMP NACTIVE PARENT CHILD BPOO
PXB EMP -1 -1 -1 -1 0 0 BPO
CREATOR IXNAME 1STKEY FULLKEY NLEAF NLEVELS CLUSTERRAT
PXB myindex 100 100 2 0.90000000
LBL STMTNO COST*RATE SQL-STATEMENT
WD01 0 1012.638184 SELECT EMPNO FROM PXB.EMP WHERE JOB LIKE 'MGR%'
COST*RATE QB PL MIX QTYPE METH ACC MTCH IX TBNAME IXNAME NU JO G
WD01 1012.6381 1 1 0 SELECT 0 R 0 N EMP
******************************************************************************** Bottom of Data ********************************************************************************
```

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 94 on page 179), the imported changes had no effect on this statement.

**Figure 94: Results of dynamic Explain on imported DDL**

```
FILE COMMANDS OPTIONS HELP
-------------------------------------------------------------------------------
PSSPW200 Explain Results for SQLTEXT
Command ===>                                                  Scroll ===> CSR
More:       >
Actions: S H R RS RW RI XD XS XP W P T C D U IM SA
CLONE Objects in database PSS18130
LXL STMTNO COST*RATE SQL-STATEMENT
WD01 0 1012.638184 SELECT EMPNO FROM PXB.EMP WHERE JOB LIKE 'MGR%'
WD01 0 1012.638184 SELECT EMPNO FROM PXB.EMP WHERE JOB LIKE 'MGR%'
COST*RATE QB PL MIX QTYPE METH ACC MTCH IX TBNAME IXNAME NU JO G
WD01 1012.6381 1 1 0 SELECT 0 R 0 N EMP
WD01 1012.6381 1 1 0 SELECT 0 R 0 N EMP
******************************************************************************** Bottom of Data ********************************************************************************
```

---

**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 95 on page 180) is displayed.

   **Figure 95: Explain Edit panel**
   
   ```sql
   EDIT       EXPLAIN                                               Columns
   00001 00080 Command ===>                                                       Scroll
   ===> CSR
   ****** *************************************************************** Top of
   Data************************************************************************
   ==MSG> -Warning- The UNDO command is not available until you change
   ==MSG> your edit profile using the command RECOVERY
   ON.
   000001 SELECT
   EMPNO
   000002 FROM
   PSS.EMP
   000003 WHERE UPPER(LASTNAME, 'EN_US') =
   'JOE'
   000004 AND UPPER(FIRSTNAME, 'EN_US') = 'JOHN'
   ```

2. Enter the SQL statements that you want to use and press **PF3**.

   The Explain or Execute Parameters panel (Figure 96 on page 180) is displayed.

   **Figure 96: Explain or Execute Parameters panel (PSSPA117)**
   
   ```sql
   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
   ```

3. At the Explain or Execute Parameters panel, select **Explain** and press **Enter**.
The Explain Results panel (Figure 97 on page 181) is displayed.

**Figure 97: 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 98 on page 181) shows the clone.

**Figure 98: 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.

Specify attributes for a new index to be added to the table, select **Specify key columns**, and press **Enter**.

The Select Key Columns panel is displayed. If the table already had some indexed expressions, those expressions are displayed on the panel.

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 100 on page 182), enter the expression upon which you want to create an index.
c Press PF3 to return to the Select Key Columns panel.

The Select Key Columns (Figure 100 on page 182) displays the expression that you created.

**Figure 100: Select Key Columns showing an expression**

```
<table>
<thead>
<tr>
<th>SEQ</th>
<th>ORDER</th>
<th>EXPRESSION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

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 PF3 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 is 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.
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 101 on page 183) shows the detailed information.

Figure 101: Explain Results panel showing an index on expression
This chapter explains how to use the PSSEXTRC function of SQL Performance to analyze dynamic SQL from APPTUNE archive trace data sets.

This chapter presents the following topics:

■ “Overview of the Explain and Compare functions” on page 185
■ “Comparing dynamic SQL from different versions of DB2” on page 186
■ “Comparing dynamic SQL from different periods of time” on page 189

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 “Comparing access paths for workloads” on page 73.
Comparing dynamic SQL from different versions of DB2

In this task, you Explain and Compare access paths for dynamic SQL from a DB2 Version 9 subsystem against the access path of that same SQL on a DB2 Version 10 subsystem.

1 Collect dynamic SQL data on a DB2 Version 9 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 9 subsystems and, if possible, turn off data collection for all other DB2 subsystems.

2 Execute the dynamic SQL on the DB2 Version 9 subsystem.

3 Create an archived trace data set:
   a Issue the SWITCH command for the output group containing the APSTMT class (or specify ALL output groups).
   b Examine the resulting archive task output to determine the name of the archive that was created (for use in the next step).

For detailed instructions, see the APPTUNE for DB2 User Guide.

4 Run PSSEXTRJ on the DB2 Version 9 subsystem, using the archived trace data set that you created in Step 3 on page 186.

PSSEXTRJ performs a dynamic Explain on all dynamic SQL that is captured from the DB2 Version 9 subsystem. The job requires you to enter the input parameters shown in Table 14 on page 188.

<table>
<thead>
<tr>
<th>Input parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSIDEX</td>
<td>Subsystem where the dynamic SQL is executed</td>
</tr>
<tr>
<td>SSID</td>
<td>Subsystem where Explain is performed</td>
</tr>
</tbody>
</table>

Table 13: PSSEXTRJ input parameters for DB2 Version 9 subsystem
<table>
<thead>
<tr>
<th>Input parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CREATOR</td>
<td>Value that you create to identify the dynamic SQL to be Explained</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> You must specify the same value for CREATOR in Step 4 on page 186, Step 7 on page 187, and Step 8 on page 188. This value can be 1 to 8 characters in length.</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 PROGRAM</td>
</tr>
</tbody>
</table>

If PSSEXTRJ finds no dynamic SQL that matches the given DB2 subsystem, the job issues an error report (Figure 102 on page 187). Verify the value of the SSIDEX parameter and the contents of your archived trace data set and resubmit the job.

**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"
```

5. Run PSSHUNLD to unload the requested Explain records, which were created in PSSEXTRJ from the DB2 Version 9 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 on page 186.

---

**Note**
The value that you enter for OBJOWNR must be identical to the value that you enter for CREATOR in the PSSEXTRJ and PSSCOMPJ jobs.

6. Load the DB2 Version 9 history records into the appropriate DB2 Version 10 history tables.

This step is required because Compare can read from only one set of history tables.

a. Update member PSSHLOD1 to load the DB2 Version 9 history records into the appropriate Version 10 history tables.

b. Submit PSSHLOAD to load the records into the Version 10 tables.

7. Run PSSEXTRJ on the DB2 Version 10 subsystem, using the same archived trace data set that you used when you ran the job on the DB2 Version 9 subsystem in Step 4 on page 186.
PSSEXTRJ performs a dynamic Explain on all dynamic SQL that was captured from the DB2 Version 9 subsystem. The job requires you to enter the input parameters shown in Table 14 on page 188.

Table 14: PSSEXTRJ input parameters for DB2 Version 10 subsystem

<table>
<thead>
<tr>
<th>Input parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSIDEX</td>
<td>Subsystem where dynamic SQL is executed</td>
</tr>
<tr>
<td>SSID</td>
<td>Subsystem where Explain is 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. This value can be 1 to 8 characters in length.</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 PROGRAM</td>
</tr>
</tbody>
</table>

<sup>a</sup> You must specify the same value for CREATOR in Step 4 on page 186, Step 7 on page 187, and Step 8 on page 188.

8. Run PSSCOMPJ on the DB2 Version 10 subsystem to Compare the loaded DB2 Version 9 history records to the DB2 Version 10 history records.

The job requires you to enter the input parameters shown in Table 15 on page 188.

Table 15: PSSCOMPJ input parameters for DB2 Version 10 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 10 subsystem) to be compared to an older history. This value is typically set to 0.</td>
</tr>
<tr>
<td>COMPOLD</td>
<td>Version of an Explain history (in this case, the Explain from the DB2 Version 9 subsystem) to be compared to a newer history. This value is typically set to -1.</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. This value can be 1 to 8 characters in length.</td>
</tr>
<tr>
<td>OBJECT&lt;sup&gt;b&lt;/sup&gt;</td>
<td>Type of object to be Compared: PROGRAM containing the dynamic SQL</td>
</tr>
<tr>
<td>NAME</td>
<td>Object name of first PROGRAM. 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 PROGRAMs that you would need to process.</td>
</tr>
</tbody>
</table>
### Input parameter | Description
--- | ---
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.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
</table>
a | You must specify the same value for CREATOR in Step 4 on page 186, Step 7 on page 187, and Step 8 on page 188.  
b | 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.

### Comparing dynamic SQL from different periods of time

In this task, you 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 9 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 is 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 on page 189.
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 16 on page 190.

### Table 16: PSSEXTRJ input parameters

<table>
<thead>
<tr>
<th>Input parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSIDEX</td>
<td>Subsystem where dynamic SQL is run</td>
</tr>
<tr>
<td>SSID</td>
<td>Subsystem where Explain is performed</td>
</tr>
<tr>
<td>CREATOR</td>
<td>Value that you create to identify the dynamic SQL to be Explained. This value can be 1 to 8 characters in length.</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 PROGRAM</td>
</tr>
</tbody>
</table>

*a* You must specify the same value for CREATOR for PSSEXTRJ in Step 4 on page 189, Step 6 on page 190, and for PSSCOMPJ in Step 7 on page 191.

If PSSEXTRJ finds no dynamic SQL that matches the given DB2 subsystem, the job issues an error report (Figure 103 on page 190). Verify the value of the SSIDEX parameter and the contents of your archived trace data set and resubmit the job.

**Figure 103: 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"
```

5. 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.

6. Run PSSEXTRJ on the specified subsystem, using the same archived trace data set that was used when you ran the job in Step 4 on page 189.

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 17 on page 190.

### Table 17: PSSEXTRJ input parameters

<table>
<thead>
<tr>
<th>Input parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSIDEX</td>
<td>Subsystem where dynamic SQL is executed</td>
</tr>
</tbody>
</table>
7 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 18 on page 191.

Table 18: PSSCOMPJ input parameters

<table>
<thead>
<tr>
<th>Input parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSID</td>
<td>Subsystem where Explain is 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. This value can be 1 to 8 characters in length.</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 PROGRAM</td>
</tr>
</tbody>
</table>

<sup>a</sup> You must specify the same value for CREATOR for PSSEXTRJ in Step 4 on page 189, Step 6 on page 190, and for PSSCOMPJ in Step 7 on page 191.
<table>
<thead>
<tr>
<th>Input parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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. |

- You must specify the same value for CREATOR for PSSEXTRJ in Step 4 on page 189, Step 6 on page 190, and for PSSCOMPJ in Step 7 on page 191.

- 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 “Managing performance with Performance Advisor” on page 47. This appendix includes the following sections:

- “Performance Advisor utilities” on page 193
- “Performance Advisor tables” on page 234
- “Sample members” on page 236

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.

DMDBMERG utility

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.
Performance Advisor utilities

The input to DMDBMERG can be data that is:
■ Retrieved directly from a Data Collector

To retrieve directly from a Data Collector, use the
DATASOURCE(COLLECTOR(….)) control statement.
■ From archive files containing APPTUNE trace data
— To request that DMDBMERG retrieve the data from appropriate archives, use

the //DOMBARC DD statement to designate an archive directory (also known
as COPYDIR) that provides indexing. The indexing permits allocation of
correct archives based on the tables and time intervals requested.

— To request that DMDBMERG retrieve the data from specific archives, use

the //TRACEIN DD statement.

If you specify any of these sources without specifying //LOADIN DD, the generated
rows are appended to the table or tables being updated.
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 LOADPLUS for DB2 as the next step to
maintain Performance Advisor tables.
Figure 104 on page 194 shows sample JCL and instructions to run the DMDBMERG
utility.
Figure 104: DMDBMERG JCL
//DMDBMERG
JOB (ACCT),'GENERATE PA TABLES', <== MODIFY
//
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 SOLUTION
//* GUIDE.
//*
//* NOTE - THE DATE EXAMPLES ARE IN USA FORMAT. MODIFY MM/DD/YY
//*
TO MATCH THE FORMAT SPECIFIED IN THE GLOBAL OPTIONS PANEL.
//*
//*----> CHANGE: ?BMC-HLQ?
= DATA COLLECTOR HIGH-LEVEL QUALIFIER

194 SQL Performance for DB2 User Guide


/*
/*TABLES EXEC PGM=DMDBMERG
/*
/*STEPLIB DD DISP=SHR,DSN=?BMC-HLQ?.BBLINK
/*
/*SYSPRINT DD SYSOUT=** **MESSAGES**
/*
/*DOMCUST DD DISP=SHR,DSN=?BMC-HLQ?.BBCSTM **CUSTOM DATASET
/DO
/*MPROF DD DISP=SHR,DSN=?BMC-HLQ?.PROFILE **PROFILE DATASET
/DO
/*MHELP DD DISP=SHR,DSN=?BMC-HLQ?.BBHELP **HELP DATASET
/DO
/*MAUTH DD DISP=SHR,DSN=?BMC-HLQ?.SECURITY **AUTHORITY DATASET
/DO
/*BARC DD DISP=SHR,DSN=?BMC-HLQ?.COPYDIR **ARCHIVE DIRECTORY
/DO
/*TRACEIN DD DISP=SHR,DSN=?TRACEDSN? **OPTIONAL SPECIFICATION **
/*
/* ** OF INPUT ARCHIVE TRACE **
/*
/* ** DATA. BY DEFAULT THE **
/*
/* ** DATA IS DYNAMICALLY **
/*
/* ** LOCATED, USING DOMBARC. **
/*
/*
/*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, **DUMMY, TABLE OUTPUT **
/*SPACE=(CYL,(5,5),RLSE), **WILL NOT BE PRODUCED **
/*DCB=(RECFM=VBS,LRECL=32760,BLKSIZE=8192)
/*
/* THE DB2LOAD AND LOADCTL STATEMENTS BELOW SPECIFY "GENERIC" **
/*DESTINATIONS FOR LOAD DATA AND LOAD CONTROL OUTPUT. **
/*YOU CAN ALSO SPECIFY DEDICATED DD STATEMENTS CORRESPONDING TO **
/*INDIVIDUAL TABLES, IN ORDER TO SEGREGATE THE DATA BY TABLE. **
/*THIS CAN BE HELPFUL FOR THE SUBSEQUENT LOAD UTILITY STEPS, WHERE **
/*EACH LOAD UTILITY CAN REFERENCE A DEDICATED DD STATEMENT IN ORDER **
/*TO RECEIVE ONLY THE DATA APPLICABLE TO THE TABLESPACE BEING LOADED. **
/*
/* TABLE DB2LOAD LOADCTL
/*-------------------------- --------------------------
/*OBJ_STATISTICS LDOBSTAT CTOBSTAT
/*STMT_STATISTICS LDSTSTAT CTSTSTAT
/*SUMMARY LDSTSUMI CTSTSUMI
/*STMT_STATISTICS_OB LDSTSTOB CTSTSTOB
/*WKLSTATISTICS LDWKSTAT CTWKSTAT
/*DAILY_OBJ_STATS LDOBSTAD CTOBSTAD
/*DAILY_STMT_STATISTICS LDSTSTAD CTSTSTAD
/*DAILY_STMT_SUMMARY LDSTSUMO CTSTSUMD
/*DAILY_WKLDSTATISTICS LDWKSTAD CTWKSTAD
/*WEEKLY_OBJ_STATISTICS LDOBSTAW CTOBSTAW
/*WEEKLY_STMT_STATISTICS LDSTSTAW CTSTSTAW
/*WEEKLY_STMT_SUMMARY LDSTSUMW CTSTSUMW
/*MONTHLY_OBJ_STATISTICS LDSTSTAM CTSTSTAM
/*MONTHLY_STMT_STATISTICS LDSTSUMM CTSTSUMM
/*MONTHLY_WKLDSTATISTICS LDWKSTAM CTWKSTAM
*/
Performance Advisor utilities

/* DATABASE OBJECT STATISTICS */
/* BASELINE_STMT_SUMMARY */
/* SUMMARY STATEMENTS */
/* BASELINE_WKLD_STATISTICS */
/*_stmt_summary */
/* stmt_texts */
/* stmt_errors */
/* stmt_exceptions */
/* stmt_exception HV */
/* stmt_exception OB */

// 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), ** LOADING UTILITY **
//            DCB=(RECFM=VB,LRECL=32756,BLKSIZE=32760) **
// LOADCTL DD DISP=(NEW,PASS), ** DB2 LOAD UTILITY CONTROL **
//            UNIT=WORK, DSN=&&CNTL, ** STATEMENTS **
//            SPACE=(CYL,(5,5),RLSE),
//            DCB=(RECFM=FB,LRECL=3120)

/* SORT WORK FILES - ADJUST SIZE RELATIVE TO THE AMOUNT OF INPUT DATA */
/* SORTWK01 DD DISP=(NEW,PASS),UNIT=WORK,SPACE=(CYL,(10,10),RLSE)
SORWK02 DD DISP=(NEW,PASS),UNIT=WORK,SPACE=(CYL,(10,10),RLSE)
SORWK03 DD DISP=(NEW,PASS),UNIT=WORK,SPACE=(CYL,(10,10),RLSE)
SORWK04 DD DISP=(NEW,PASS),UNIT=WORK,SPACE=(CYL,(10,10),RLSE)

/* SYSOUT DD SYSOUT=* ** SORT MESSAGES **

/* 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 */
/* DATASOURCE(COLLECTOR(XXXX)) - SPECIFIES THAT DATA BE RETRIEVED */
/* DIRECTLY FROM DBC SUBSYSTEM XXXX */
/* (OVERIDES TRACEIN, DOMBARC) */
/* APPROF(PROFILE.OWNER) - APPLICATION PROFILE USED WHEN */
/* GROUP FILTERING IS SPECIFIED */
/* QUALIFIER( - GLOBAL (DEFAULT) QUALIFIERS FOR RESTRICTING */
/* DB2(XXXX) - " BY DB2 SSID NAMED XXXX */
/* ,DSGROUP( ) - " BY DS GROUP */
/* ,PLAN( ) - " BY PLAN */

SQL Performance for DB2 User Guide
Performance Advisor utilities

/*
   .COLL(    ) - " BY COLLECTION ID
   .PROGRAM(    ) - " BY DBRM/PACKAGE ID
   .CONN(    ) - " BY CONNECT-IDS
   .OPER( ) - " BY OPER ID
   .CORR( ) - " BY CORRELATION ID
   .APGRP(  ) - " BY APPLICATION PROFILE-APPTUNE ONLY
   .CALLENTYPE( ) - " BY CALL TYPE (STATIC, DYNAMIC)
   .STMTTYPE( ) - " BY STMT TYPE (OPEN, SELECT, ETC.)
   .BP(   ) - " BY BUFFER POOL
   .DBNAME(   ) - " BY DATABASE NAME
   .PAGESET(  ) - " BY PAGESET NAME
   .TBCREATOR( ) - " BY ASSOCIATED TABLE CREATOR
   .TBNAME(    ) - " BY ASSOCIATED TABLE NAME

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

CCSID(NNNN) - SPECIFIES THE EBCDIC CHARACTER SET TO BE
   USED FOR CONVERSION FROM/TO UNICODE;
   DEFAULT IS 0037

DECIMAL(X) - SPECIFIES U FOR U.S. FORMAT, PERIOD DECIMAL
   POINTS IN OUTPUT, E FOR EUROPEAN FORMAT,
   COMMA DECIMAL POINTS (OVERRIDES SETTING IN
   PROFILE)

NODATA(N) - SPECIFIES AN INTEGER 0-8 TO BE USED AS
   RETURN CODE WHEN NO DATA IS SELECTED

FILSZ(N) - SPECIFIES AN INTEGER 0-99999999 AS AN
   ESTIMATE OF INPUT RECORDS FOR SORT PROGRAM

AVGRLEN(N) - SPECIFIES AN INTEGER 0-99999 AS AN ESTIMATE
   OF AVERAGE RECORD SIZE FOR SORT PROGRAM

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_SUMMARY STATEMENT SUMMARY BY INTVL
   STMT_STATISTICS_OB OBJ STATS PER STATEMENT BY INTVL
   WKLD_STATISTICS WORKLOAD STATISTICS BY INTERVAL
   DAILY_OBJSTATISTICS OBJECT STATS PER DAY BY INTVL
   DAILY_STMT_STATISTICS STATEMENT STATISTICS BY DAY
   DAILY_STMT_SUMMARY STATEMENT SUMMARY BY DAY
   DAILY_WKLD_STATISTICS WORKLOAD STATISTICS BY DAY
   WEEKLY_OBJSTATISTICS OBJECT STATS PER WEEK
   WEEKLY_STMT_STATISTICS STATEMENT STATISTICS BY WEEK
   WEEKLY_STMT_SUMMARY STATEMENT SUMMARY BY WEEK
   WEEKLY_WKLD_STATISTICS WORKLOAD STATISTICS BY WEEK
   MONTHLY_OBJSTATISTICS OBJECT STATS PER MONTH
   MONTHLY_STMT_STATISTICS STATEMENT STATISTICS BY MONTH
   MONTHLY_STMT_SUMMARY STATEMENT SUMMRY BY MONTH
   MONTHLY_WKLD_STATISTICS WORKLOAD STATISTICS BY MONTH
   BASELINE_OBJSTATISTICS OBJECT STATISTICS BASELINE
Table 19 on page 198 describes the data sets that are processed by DMDBMERG.

Table 19: DMDBMERG data sets

<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 is written to this data set, which can be passed to a load utility in another step to load the data.</td>
<td>RECFM=VB, LRECL = 32752, BLKSIZE = 32756</td>
</tr>
<tr>
<td>DOMAUTH</td>
<td>The VSAM data set containing the user profile security values</td>
<td>Not applicable</td>
</tr>
<tr>
<td>DOMBARC</td>
<td>The VSAM data set that stores the names of the archived trace data sets for use by the archive directory</td>
<td>Not applicable</td>
</tr>
<tr>
<td>DOMCUST</td>
<td>The sequential data set containing the report customization elements</td>
<td>Not applicable</td>
</tr>
<tr>
<td>DOMHELP</td>
<td>The VSAM data set that contains help members</td>
<td>Not applicable</td>
</tr>
<tr>
<td>DOMPROF</td>
<td>The VSAM data set containing the user profiles</td>
<td>Not applicable</td>
</tr>
<tr>
<td>R0001VFG</td>
<td>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>(optional)</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.</td>
<td></td>
</tr>
<tr>
<td>R0001VFL</td>
<td>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>(optional)</td>
<td>This value can be adjusted according to the number of selected records.</td>
<td></td>
</tr>
<tr>
<td>DDNAME</td>
<td>Description</td>
<td>DCB Attributes</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>SORTWKnn (optional)</td>
<td>Interim data sets for the SORT utility working storage&lt;br&gt;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 this DD statement’s specification.</td>
<td>Not applicable</td>
</tr>
</tbody>
</table>
| SRCTRACE (optional) | Data set containing a listing of all records found in the input archive files<br>A single line is generated for each valid record, including an indication of whether the record is used to generate table data. | ■ RECFM = FBA  
■ LRECL = 121  
■ BLKSIZE = any multiple of LRECL |
| STEPLIB           | Data Collector load library                                                  | Not applicable |
| SYSIN             | DMDBMERG control statements                                                 | Not applicable |
| SYSOUT            | SORT messages                                                               | Not applicable |
| SYSPRINT          | DMDBMERG messages and statistics                                            | ■ RECFM = FBA  
■ LRECL = 121  
■ BLKSIZE = any multiple of LRECL |
| 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<br>The input data sets can be any of the following data sets: archived data sets or EXPORT data sets.<br>You must either include the TRACEIN data set or specify the INTERVAL control statement.<br>By default, DOMBARC dynamically locates the data. | Not applicable |
## TRACEWRK

Interim data set for holding the records that were selected and sorted for table data. 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.

<table>
<thead>
<tr>
<th>DCB Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>RECFM = VBS</td>
</tr>
<tr>
<td>LRECL = 32760</td>
</tr>
<tr>
<td>BLKSIZE = 8192</td>
</tr>
</tbody>
</table>

You have the option to use dedicated DD statements with this data set. This increases efficiency. For more information see “Optional dedicated DD statements for DB2LOAD and LOADCTL” on page 200.

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:

- Multiply 32 KB (maximum size of each record) with the number of records to calculate a total size of the input records.

- Adjust this figure to the track size and approximate sizes of the initial allocation.

### Optional dedicated DD statements for DB2LOAD and LOADCTL

You have the option to use dedicated DD statements with the DB2LOAD and LOADCTL DD statements as described. These additional DD statements are dedicated to a particular table's load data and control statements. Using these dedicated DD statements, you can increase the efficiency of subsequent LOAD steps by running a separate step to load each table. If you use dedicated DD statements for all tables being generated, set DB2LOAD and LOADCTL statements to DUMMY.

The additional DD statements supported appear in Table 20 on page 200:

### Table 20: Optional DD statements for the DMDBMERG utility

<table>
<thead>
<tr>
<th>Table name</th>
<th>Dedicated DB2LOAD</th>
<th>Dedicated LOADCTL</th>
</tr>
</thead>
<tbody>
<tr>
<td>STMT_STATISTICS</td>
<td>LDSTSTAT</td>
<td>CTSTSTAT</td>
</tr>
<tr>
<td>OBJ_STATISTICS</td>
<td>LDOBSTAT</td>
<td>CTOBSTAT</td>
</tr>
<tr>
<td>WKLD_STATISTICS</td>
<td>LDWKSTAT</td>
<td>CTWKSTAT</td>
</tr>
<tr>
<td>STMT_STATISTICS_OB</td>
<td>LDSTSTOB</td>
<td>CTSTSTOB</td>
</tr>
<tr>
<td>DAILY_STMT_STATISTICS</td>
<td>LDSTSTAD</td>
<td>CTSTSTAD</td>
</tr>
<tr>
<td>DAILY_OBJ_STATISTICS</td>
<td>LDOBSTAD</td>
<td>CTOBSTAD</td>
</tr>
</tbody>
</table>
## Table name

<table>
<thead>
<tr>
<th>Table name</th>
<th>Dedicated DB2LOAD</th>
<th>Dedicated LOADCTL</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAILY_WKLD_STATISTICS</td>
<td>LDWKSTAD</td>
<td>CTWKSTAD</td>
</tr>
<tr>
<td>WEEKLY_STMT_STATISTICS</td>
<td>LDSTSTAW</td>
<td>CTSTSTAW</td>
</tr>
<tr>
<td>WEEKLY_OBJ_STATISTICS</td>
<td>LDOBSTAW</td>
<td>CTOBSTAW</td>
</tr>
<tr>
<td>WEEKLY_WKLD_STATISTICS</td>
<td>LDWKSTAW</td>
<td>CTWKSTAW</td>
</tr>
<tr>
<td>MONTHLY_STMT_STATISTICS</td>
<td>LDSTSTAM</td>
<td>CTSTSTAM</td>
</tr>
<tr>
<td>MONTHLY_OBJ_STATISTICS</td>
<td>LDOBSTAM</td>
<td>CTOBSTAM</td>
</tr>
<tr>
<td>MONTHLY_WKLD_STATISTICS</td>
<td>LDWKSTAM</td>
<td>CTWKSTAM</td>
</tr>
<tr>
<td>BASELINE_STMT_STATISTICS</td>
<td>LDSTSTAB</td>
<td>CTSTSTAB</td>
</tr>
<tr>
<td>BASELINE_OBJ_STATISTICS</td>
<td>LDOBSTAB</td>
<td>CTOBSTAB</td>
</tr>
<tr>
<td>BASELINE_WKLD_STATISTICS</td>
<td>LDWKSTAB</td>
<td>CTWKSTAB</td>
</tr>
<tr>
<td>STMT_ERRORS</td>
<td>LDSTERRS</td>
<td>CTSTERRS</td>
</tr>
<tr>
<td>STMT_EXCEPTIONS</td>
<td>LDSTEXCP</td>
<td>CTSTEXCP</td>
</tr>
<tr>
<td>STMT_EXCEPTIONS_HV</td>
<td>LDSTEXHV</td>
<td>CTSTEXHV</td>
</tr>
<tr>
<td>STMT_EXCEPTIONS_OB</td>
<td>LDSTEXOB</td>
<td>CTSTEXOB</td>
</tr>
<tr>
<td>STMT_TEXT</td>
<td>LDSTTEXT</td>
<td>CTSTTEXT</td>
</tr>
</tbody>
</table>

### Control statements

The control statements listed in Table 21 on page 201 can be used with the DMDBMERG utility.

#### Table 21: DMDBMERG SYSIN control statements

<table>
<thead>
<tr>
<th>Control Statement</th>
<th>Use this Control Statement to</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPROF (profile.owner)</td>
<td>Specify the application profile for group reporting.</td>
</tr>
<tr>
<td>AVGRLEN</td>
<td>Specify an integer from 0–99999999 to estimate the records' expected average length for a SORT program.</td>
</tr>
<tr>
<td>CCSID (nnnn)</td>
<td>Specify the EBCDIC character set to be used for conversion to or from Unicode.</td>
</tr>
<tr>
<td>DECIMAL (x)</td>
<td>Specify the decimal format in output, overriding profile settings:</td>
</tr>
<tr>
<td></td>
<td>■ U for U.S. format (period decimal point)</td>
</tr>
<tr>
<td></td>
<td>■ E for European format (comma decimal point)</td>
</tr>
</tbody>
</table>
Control Statement | Use this Control Statement to
--- | ---
DATASOURCE | Specify the source of data for batch reporting. DATASOURCE can be specified only once in a job stream.
If DATASOURCE is specified, it must appear before any REPORT or QUALIFIER statement. You must also specify either COLLECTOR or ARCHIVE_DD. You cannot specify both.
If DATASOURCE is omitted, the TRACEIN data set is used as the source. If there is no TRACEIN DD in the JCL, the Archive Directory is searched for archived trace data sets based on the interval selection that you specified as parameters for the batch reports. The TRACEIN DD is dynamically allocated with the data sets that are retrieved from the Archive directory.

FILSZ \((n)\) | Specify an integer from \(0–99999\) to estimate the number of records input into the utility via archives and the LOADIN DD statement for a SORT program.\(^a\)

FIRST_DAY \((\text{Sunday})\) | Specify the first day of the week for weekly tables.

INTERVAL | Specify a time interval for the data to be included in tables. For more information, see “INTERVAL statement” on page 205.
You must either include the TRACEIN data set or specify the INTERVAL control statement.

NODATA \((n)\) | Specify the return code that the DMDBMERG utility generates in various situations where no data is generated.
For information about default return codes when no data is generated, see “Default return code tables” on page 204.

PURGE_DAYS \((nn)\) | Purge data older than the specified number of days.

QUALIFIER \((\text{type})\) | Use qualifier values to restrict the data that is selected for tables. For more information about valid qualifiers, see Table 22 on page 203.

TABLE | Produce tables.

TBCREATOR \((\text{name})\) | Specify the table owner for generated load utility control statements.

\(^a\) Use these control statements if you can accurately estimate (based on prior runs, for example) the average length and number of records. The external SORT program uses your estimates to optimize the SORT’s efficiency. These control statements provide control over the external SORT program’s work file allocation.
These control statements are optional, but FILSZ must be specified prior to AVGRLEN if AVGRLEN is specified.

---

**Qualifiers Table**

Table 22 on page 203 displays a list of tables and the qualifiers that can be specified on the QUALIFIER statements.
Table 22: Qualifiers Table

<table>
<thead>
<tr>
<th>Table(s)</th>
<th>Valid qualifiers</th>
</tr>
</thead>
<tbody>
<tr>
<td>OBJ_STATISTICS</td>
<td>BP</td>
</tr>
<tr>
<td>DAILY_OBJ_STATISTICS</td>
<td>DB2</td>
</tr>
<tr>
<td>WEEKLY_OBJ_STATISTICS</td>
<td>DSGROUP</td>
</tr>
<tr>
<td>MONTHLY_OBJ_STATISTICS</td>
<td>DBNAME</td>
</tr>
<tr>
<td>BASELINE_OBJ_STATISTICS</td>
<td>PAGESET</td>
</tr>
<tr>
<td>TBCREATOR</td>
<td>TBNNAME</td>
</tr>
<tr>
<td>STMT_STATISTICS</td>
<td>BP</td>
</tr>
<tr>
<td>DAILY_STMT_STATISTICS,</td>
<td>DB2</td>
</tr>
<tr>
<td>WEEKLY_STMT_STATISTICS</td>
<td>DSGROUP</td>
</tr>
<tr>
<td>MONTHLY_STMT_STATISTICS</td>
<td>DBNAME</td>
</tr>
<tr>
<td>BASELINE_STMT_STATISTICS</td>
<td>PAGESET</td>
</tr>
<tr>
<td>STMT_SUMMARY</td>
<td>TBNNAME</td>
</tr>
<tr>
<td>DAILY_STMT_SUMMARY</td>
<td>APGRP</td>
</tr>
<tr>
<td>WEEKLY_STMT_SUMMARY</td>
<td>CALLTYPE</td>
</tr>
<tr>
<td>MONTHLY_STMT_SUMMARY</td>
<td>COLLECTION</td>
</tr>
<tr>
<td>BASELINE_STMT_SUMMARY</td>
<td>CONNECTION</td>
</tr>
<tr>
<td>STMT_ERRORS</td>
<td>CORRNAME</td>
</tr>
<tr>
<td>STMT_EXCEPTIONS</td>
<td>DB2</td>
</tr>
<tr>
<td>STMT_EXCEPTIONS_HV</td>
<td>DSGROUP</td>
</tr>
<tr>
<td>STMT_EXCEPTIONS_OB</td>
<td>OPERATOR</td>
</tr>
<tr>
<td>STMTTYPE</td>
<td>PLAN</td>
</tr>
<tr>
<td>TEXTHASH</td>
<td>PROGRAM</td>
</tr>
<tr>
<td>STMT_ERRORS</td>
<td>STMTTYPE</td>
</tr>
<tr>
<td>STMT_EXCEPTIONS_HV</td>
<td>TEXTHASH</td>
</tr>
</tbody>
</table>
The tables in this section display the return code that the DMDBMERG utility generates under the conditions specified when no data is selected for input or generated for output.

The NODATA control statement can override those values designated with one asterisk (*). In the designated cases, the NODATA control statement generates alternate values for the DMDBMERG utility's default return codes. The syntax for the NODATA control statement is `NODATA (n)`, where `n` is 0–8 inclusive.

Table 23 on page 205 lists the default return codes that are generated when the LOADIN DD statement is coded.

### Default return code tables

<table>
<thead>
<tr>
<th>Table(s)</th>
<th>Valid qualifiers</th>
</tr>
</thead>
<tbody>
<tr>
<td>STMT_STATISTICS_OB</td>
<td>APGRP, BP, CALLTYPE, COLLECTION, CONNECTION, CORRNAME, DB2, DSGROUP, DBNAME, OPERATOR, PAGESET, PLAN, PROGRAM, STMTTYPE, TBCREATOR, TBNAME, TEXTHASH</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>WKLD_STATISTICS</th>
<th>APGRP, COLLECTION, CONNECTION, CORRNAME, DB2, DSGROUP, OPERATOR, PLAN, PROGRAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAILY_WKLD_STATISTICS</td>
<td></td>
</tr>
<tr>
<td>WEEKLY_WKLD_STATISTICS</td>
<td></td>
</tr>
<tr>
<td>MONTHLY_WKLD_STATISTICS</td>
<td></td>
</tr>
<tr>
<td>BASELINE_WKLD_STATISTICS</td>
<td></td>
</tr>
</tbody>
</table>
Table 23: Conditions with LOADIN DD statement coded

<table>
<thead>
<tr>
<th>Condition</th>
<th>Message</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error in allocation of selected archive</td>
<td>BMC24711</td>
<td>8</td>
</tr>
<tr>
<td>No archives selected</td>
<td>BMC24701</td>
<td>8*</td>
</tr>
<tr>
<td>No data selected from archives for any table</td>
<td>NO RECORDS FOUND</td>
<td>8*</td>
</tr>
<tr>
<td>Data generated for at least one, but not all, DB2LOAD or LDxxxxxx DD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>statements</td>
<td>BMC24309</td>
<td></td>
</tr>
<tr>
<td>Note: This message occurs and subsequent processing is</td>
<td></td>
<td></td>
</tr>
<tr>
<td>terminated unless the NODATA control statement specifies a value less</td>
<td></td>
<td></td>
</tr>
<tr>
<td>than 8 for n.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 24 on page 205 lists the default return codes that are generated when no LOADIN DD statement is coded.

Table 24: Conditions with no LOADIN DD statement coded

<table>
<thead>
<tr>
<th>Condition</th>
<th>Message</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error in allocation of selected archive</td>
<td>BMC24711</td>
<td>8</td>
</tr>
<tr>
<td>No archives selected</td>
<td>BMC24701</td>
<td>4*</td>
</tr>
<tr>
<td>No data selected from archives for any table</td>
<td>NO RECORDS FOUND</td>
<td>4*</td>
</tr>
<tr>
<td>Data generated for at least one, but not all, DB2LOAD or LDxxxxxx DD</td>
<td>None</td>
<td>0</td>
</tr>
<tr>
<td>statements</td>
<td></td>
<td></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 25 on page 206 describes the keywords and options for each keyword.
Table 25: INTERVAL statement and parameter keywords

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Value</th>
<th>Description of value</th>
</tr>
</thead>
</table>
| START (optional) | Specifies the beginning date and time for the data to be included. This keyword must be specified if FOR or END is specified. The START date and time must be prior to the date and time that the job is run. If no START keyword is specified the start time defaults to the timestamp of the first record in the TRACEIN data set. If there are multiple DB2s being reported that do not all share the same collection interval, the start time for each DB2 can be different. **Example:** There are two DB2s being reported:  
  - DB2A has a collection interval of 24 hours, starting at midnight.  
  - DB2B has a collection interval of 8 hours (intervals starting at midnight, 8:00 A.M. and 4:00 P.M.).  
  The report is activated at 10:00 A.M. The interval for DB2A starting at midnight is active, so midnight is the start time for DB2A. The interval for DB2B starting at 8:00 A.M. is active, so 8:00 A.M. is the start time for DB2B. **Note:** Records are cut at the SQL statistics collection interval that is specified in 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:** S | date | Beginning date of the interval. When specified without a time, the time defaults to midnight (12:00 A.M.) on the specified date (midnight = beginning of day). |
|                |                                                                             | -nn   | Number of days prior to the current date to be used for the beginning date. Valid values include any number from -0 through -60. For example, if -5 is specified on 03/06/2005 (USA format), the start date is 03/01/2005. Use -0 to indicate the current date. When specified without a time, the time defaults to midnight (12:00 A.M.) on the specified date (midnight = beginning of day). |
|                |                                                                             | time  | Beginning time for the data to be included. Valid values are any time in the range 00:00:00-23:59:59 or 00:00-23:59. 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. |
| date,time      |                                                                             |       | 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. |
| -nn,time       |                                                                             |       | When both a number of days and a time are specified, the date that is represented by -nn is calculated, and the selected data begins with the first record having a timestamp that is equal to or greater than the calculated date and specified time. |
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 must 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 data 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 When specified without a time, the time defaults to midnight (12:00 A.M.) on the specified date (midnight = beginning of day).</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-nn</td>
<td>Number of days prior to the current date to be used for the ending date Valid values are any number in the range -0 through -60. For example, if -5 is specified on 03/06/2005 (USA format), the end date is 03/01/2005. Use -0 to indicate the current date. When specified without a time, the time defaults to midnight (12:00 A.M.) on the specified date (midnight = beginning of day).</td>
<td></td>
<td></td>
</tr>
<tr>
<td>time</td>
<td>Ending time for the data to be included Valid values are any time in the range 00:00:00-23:59:59 or 00:00-23:59. When specified without a date, the end date defaults to the start date. For this reason, an end time without a date is valid only if the specified time is later in the day than the start time. For example, if the start date and time are 01/01/2005 8:00:00 (USA format) and you specify an end time without a date, the default date is 01/01/2005 and the specified time must be later than 8:00:00.</td>
<td></td>
<td></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>-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>
<td></td>
<td></td>
</tr>
<tr>
<td>Keyword</td>
<td>Description</td>
<td>Value</td>
<td>Description of value</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------</td>
<td>---------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>FOR</td>
<td>Specifies the duration of the interval (as an alternative to specifying an</td>
<td>nnn</td>
<td>Duration of the interval stated as a number of days</td>
</tr>
<tr>
<td>(optional)</td>
<td>END time)</td>
<td></td>
<td>Valid values are any number in the range 1-364.</td>
</tr>
<tr>
<td></td>
<td>END and FOR are two different ways to specify the ending time. If both</td>
<td>time</td>
<td>Duration of the interval stated hours, minutes, and seconds.</td>
</tr>
<tr>
<td></td>
<td>keywords are specified, END is ignored.</td>
<td></td>
<td>Valid values are any time in the range 00:00-23:59 or 00:00-23:59.</td>
</tr>
<tr>
<td></td>
<td>If no END or FOR keyword is specified, the end time defaults to the</td>
<td>nnn,time</td>
<td>Duration of the interval stated as a number of days and hours</td>
</tr>
<tr>
<td></td>
<td>timestamp of the last record in the data set.</td>
<td></td>
<td>For example, a specification of 30,6:00 means an interval of 30 days and 6 hours.</td>
</tr>
<tr>
<td></td>
<td>If neither a START nor an END or FOR is specified the entire contents of</td>
<td></td>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td></td>
<td>the TRACEIN data set are selected.</td>
<td></td>
<td>INTERVAL(START(02/22/2007,6:00:00) - FOR(6:00:00))</td>
</tr>
<tr>
<td></td>
<td>Valid abbreviation: F</td>
<td></td>
<td>This INTERVAL statement selects data generated on 02/22/2007 (USA format) between</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>6:00 a.m. and 12:00 noon.</td>
</tr>
</tbody>
</table>

**QUALIFIER statement and parameter**

The QUALIFIER statement uses qualifier values to filter the data for inclusion in reports that are produced by subsequent REPORT statements in the same job stream.

If default qualifiers are specified in the report, the QUALIFIER statement overrides those default qualifiers. If the qualifiers are locked in any of the requested reports, a message is printed in the Report Disposition Summary, and the default qualifiers are used for those reports.

**Note**

A batch report specifying DATASOURCE(COLLECTOR()) can filter on more qualifiers than a batch report using archive data. Changes to the collector have enabled additional data-reduction capabilities that allow it to process the data with more granularity than can be done with archive data.

When used as a parameter of the REPORT statement, the included or excluded qualifiers relate only to the reports that are specified in that REPORT statement. The QUALIFIER parameter overrides default qualifiers and any previously specified QUALIFIER statement.
Valid abbreviations for QUALIFIER are QUAL and Q. Figure 105 on page 209 shows the syntax of the QUALIFIER statement and parameter.

**Figure 105: Syntax of the QUALIFIER statement and parameter**

![Syntax Diagram]

Table 26 on page 209, Table 27 on page 210, and Table 28 on page 212 provide details and permissible values of the keywords for the QUALIFIER statement and parameter.

The INCLUDE and EXCLUDE keywords filter the records that are selected to satisfy report requests. You can specify either INCLUDE or EXCLUDE for each qualifier type, but you cannot specify both of them. INCLUDE is the default selection.

**Table 26: INCLUDE and EXCLUDE**

<table>
<thead>
<tr>
<th>Value</th>
<th>Description of value</th>
</tr>
</thead>
<tbody>
<tr>
<td>INCLUDE</td>
<td>Includes in the reports only records with the qualifier values that are specified in this statement</td>
</tr>
<tr>
<td></td>
<td>Valid abbreviation: I</td>
</tr>
<tr>
<td>EXCLUDE</td>
<td>Excludes from the reports records containing the qualifier values that are specified in this statement</td>
</tr>
<tr>
<td></td>
<td>The EXCLUDE keyword cannot be specified with DB2.</td>
</tr>
<tr>
<td></td>
<td>Valid abbreviation: E</td>
</tr>
</tbody>
</table>

The *type* values define the type of qualifiers to be included or excluded. At least one qualifier type must be specified.

**Note**

For a list of appropriate qualifiers for each report, see Table 28 on page 212.

At least one value is required for each specified qualifier type. Multiple values must be separated by commas (,). Wildcards are permitted. For more information about wildcards, see “Using Wildcards” on page 231.

If you are producing a report with SAP data, specify CONN(*connectionID*) in your QUALIFIER statement or parameter to report work process numbers, specify
CORR(*correlationID*) to report work process IDs, and specify G(*DBRM/package*) to report servers.

**Table 27: QUALIFIER statement values**

<table>
<thead>
<tr>
<th>Value</th>
<th>Description of value</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPGROUP</td>
<td>Application group</td>
</tr>
<tr>
<td>BP</td>
<td>Buffer pool ID</td>
</tr>
<tr>
<td>CALLTYPE</td>
<td>Type of SQL statement that was executed</td>
</tr>
<tr>
<td>CONNECTION</td>
<td>Connection ID</td>
</tr>
<tr>
<td>CORRELATION</td>
<td>Correlation ID</td>
</tr>
<tr>
<td>DB2</td>
<td>DB2 SSID</td>
</tr>
<tr>
<td>DBNAME</td>
<td>Database name</td>
</tr>
<tr>
<td>DSGROUP</td>
<td>Data sharing group</td>
</tr>
<tr>
<td>OPERATOR</td>
<td>Original operator ID</td>
</tr>
<tr>
<td>PAGESET</td>
<td>Page set name</td>
</tr>
<tr>
<td>PLAN</td>
<td>Plan name</td>
</tr>
<tr>
<td>PROGRAM</td>
<td>Database request module or package name</td>
</tr>
</tbody>
</table>

**Note:** DBRM and PACKAGE are still supported

Valid abbreviations:
- APGRP, AG (for APPGROUP)
- B (for BP)
- STATIC or DYNAMIC (for CALLTYPE)
- CONN, N (for CONNECTION)
- CORR (for CORRELATION)
- U (for DB2)
- DBN (for DBNAME)
- DSG (for DSGROUP)
- OPER, OPID, O (for OPERATOR)
- PGS (for PAGESET)
- P (for PLAN)
- PROG (for PROGRAM)
<table>
<thead>
<tr>
<th>Value</th>
<th>Description of value</th>
</tr>
</thead>
<tbody>
<tr>
<td>STMTTYPE</td>
<td>Statement type, corresponding to statement types reported in APPTUNE (such as FETCH, OPEN, CLOSE), with underscores instead of blanks (for example, CALL_STATEMENT) 22 characters or less, wildcards not supported No abbreviation</td>
</tr>
<tr>
<td>TBCREATOR</td>
<td>Table creator</td>
</tr>
<tr>
<td></td>
<td>Valid abbreviation: TBC</td>
</tr>
<tr>
<td>TBNAME</td>
<td>Table name</td>
</tr>
<tr>
<td></td>
<td>The length of this field can be up to 128 characters. To continue the field on the next line, add a hyphen (-) to the last character of the line, and continue the name field on column 1 of the next line. For example: TBNAME(LONGTABLENAMEFORDOMPLEXANDSHORTCOLSANDEVERYTHING)</td>
</tr>
<tr>
<td></td>
<td>Valid abbreviation: TBN</td>
</tr>
<tr>
<td>TEXTHASH</td>
<td>Text hash or key, as reported in APPTUNE reports SQMBSTMT or SQMBTEXT</td>
</tr>
<tr>
<td></td>
<td>You can use a text key from one of these reports as a qualifier to select data for the corresponding statements in SQMBSTMT, SQMCACTX. 20 hexadecimal characters, wildcards not supported Valid abbreviations: TXTH or TH</td>
</tr>
</tbody>
</table>

a If you are producing a report with SAP data:

- Specify CONN(connectionID) in your QUALIFIER statement or parameter to report work process numbers.
- Specify CORR(correlationID) to report work process IDs.
- Specify G(DBRM/package) to report servers.

Although you are not prevented from specifying any qualifier with any report, not all qualifiers are valid for all reports. Be careful to choose qualifiers that are appropriate to the data that you are collecting.

Table 28 on page 212 lists the reports that can be run in batch and indicates which qualifiers are valid with each report. The reports that are designed specifically for batch processing are listed first.
<table>
<thead>
<tr>
<th>Report</th>
<th>Description</th>
<th>Valid qualifiers</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQMBOBJD</td>
<td>Batch Object Detail Analysis</td>
<td>APPGROUP, BP, CALLTYPE, CONNECTION, DBNAME, DB2, DSGROUP, OPERATOR, PAGESSET, PLAN, PROGRAM, STMTTYPE, TBCREATOR, TBNAME, TEXTHASH</td>
</tr>
<tr>
<td>SQMBSTMT</td>
<td>Batch SQL Statement Analysis</td>
<td>APPGROUP, CALLTYPE, CONNECTION, DB2, DSGROUP, OPERATOR, PLAN, PROGRAM, STMTTYPE, TEXTHASH</td>
</tr>
<tr>
<td>SQMBTEXT</td>
<td>Batch SQL Text</td>
<td>APPGROUP, CALLTYPE, CONNECTION, DB2, DSGROUP, OPERATOR, PLAN, PROGRAM, STMTTYPE, TEXTHASH</td>
</tr>
<tr>
<td>Report</td>
<td>Description</td>
<td>Valid qualifiers</td>
</tr>
<tr>
<td>------------</td>
<td>---------------------------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>SQMBWKLD</td>
<td>Statement Performance by interval</td>
<td>APPGROUP CALLTYPE CONNECTION DB2 DSGROUP OPERATOR PLAN PROGRAM STMTTYPE TEXTHASH</td>
</tr>
<tr>
<td>SQMBXCEP</td>
<td>Batch SQL Exceptions Analysis</td>
<td>APPGROUP CALLTYPE CONNECTION DB2 DSGROUP OPERATOR PLAN PROGRAM STMTTYPE TEXTHASH</td>
</tr>
<tr>
<td>SQMCACEX</td>
<td>Exception Analysis</td>
<td>APPGROUP CALLTYPE CONNECTION DB2 DSGROUP OPERATOR PLAN PROGRAM STMTTYPE TEXTHASH</td>
</tr>
<tr>
<td>Report</td>
<td>Description</td>
<td>Valid qualifiers</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------------------</td>
<td>------------------------------------------</td>
</tr>
<tr>
<td>SQMCACTA</td>
<td>Client Application Analysis (DATA)</td>
<td>APPGROUP CALLTYPE CONNECTION DB2 DSGROUP OPERATOR PLAN PROGRAM STMTTYPE a TEXTHASH a</td>
</tr>
<tr>
<td>SQMCACTB</td>
<td>Workstation Analysis (DATA)</td>
<td>APPGROUP CALLTYPE CONNECTION DB2 DSGROUP OPERATOR PLAN PROGRAM STMTTYPE a TEXTHASH a</td>
</tr>
<tr>
<td>SQMCACTC</td>
<td>Connection ID Analysis (DATA)</td>
<td>APPGROUP CALLTYPE CONNECTION DB2 DSGROUP OPERATOR PLAN PROGRAM STMTTYPE a TEXTHASH a</td>
</tr>
<tr>
<td>Report</td>
<td>Description</td>
<td>Valid qualifiers</td>
</tr>
<tr>
<td>-----------</td>
<td>------------------------------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>SQMCACTD</td>
<td>Requesting Location Analysis</td>
<td>APPGROUP CALLTYPE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CONNECTION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DB2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSGROUP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OPERATOR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PLAN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PROGRAM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>STMTTYPE (^a)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TEXTHASH (^a)</td>
</tr>
<tr>
<td>SQMCACTG</td>
<td>Group Analysis (DATA)</td>
<td>APPGROUP CALLTYPE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CONNECTION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DB2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSGROUP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OPERATOR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PLAN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PROGRAM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>STMTTYPE (^a)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TEXTHASH (^a)</td>
</tr>
<tr>
<td>SQMCACTI</td>
<td>Correlation ID Analysis (DATA)</td>
<td>APPGROUP CALLTYPE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CONNECTION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DB2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSGROUP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OPERATOR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PLAN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PROGRAM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>STMTTYPE (^a)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TEXTHASH (^a)</td>
</tr>
<tr>
<td>Report</td>
<td>Description</td>
<td>Valid qualifiers</td>
</tr>
<tr>
<td>----------</td>
<td>----------------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>SQMCACTL</td>
<td>Logical DB2 Analysis</td>
<td>APPGROUP a&lt;br&gt;CALLTYPE a&lt;br&gt;CONNECTION a a&lt;br&gt;DB2 a&lt;br&gt;DSGROUP a&lt;br&gt;OPERATOR a a&lt;br&gt;PLAN a a&lt;br&gt;PROGRAM a a&lt;br&gt;STMTTYPE a a&lt;br&gt;TEXTHASH a</td>
</tr>
<tr>
<td>SQMCACTO</td>
<td>Object Analysis</td>
<td>BP a&lt;br&gt;DB2 a&lt;br&gt;DSGROUP a&lt;br&gt;DBNAME a&lt;br&gt;PAGESET a&lt;br&gt;TBCREATOR a&lt;br&gt;TBNNAME a</td>
</tr>
<tr>
<td>SQMCACTP</td>
<td>Plan Analysis (DATA)</td>
<td>CALLTYPE a&lt;br&gt;CONNECTION a a&lt;br&gt;DB2 a&lt;br&gt;DSGROUP a&lt;br&gt;OPERATOR a a&lt;br&gt;PLAN a a&lt;br&gt;PROGRAM a a&lt;br&gt;STMTTYPE a a&lt;br&gt;TEXTHASH a</td>
</tr>
<tr>
<td>SQMCACTQ</td>
<td>Implicit Qualifier Analysis</td>
<td>APPGROUP a&lt;br&gt;CALLTYPE a&lt;br&gt;CONNECTION a&lt;br&gt;DB2 a&lt;br&gt;DSGROUP a&lt;br&gt;OPERATOR a a&lt;br&gt;PLAN a a&lt;br&gt;PROGRAM a a&lt;br&gt;STMTTYPE a a&lt;br&gt;TEXTHASH a</td>
</tr>
<tr>
<td>Report</td>
<td>Description</td>
<td>Valid qualifiers</td>
</tr>
<tr>
<td>----------</td>
<td>------------------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>SQMCACTR</td>
<td>Program Analysis (DATA)</td>
<td>CALLTYPE, CONNECTION, DB2, DSGROUP, OPERATOR, PLAN, PROGRAM, STMTTYPE, TEXTHASH</td>
</tr>
<tr>
<td>SQMCACTS</td>
<td>Subsystem Analysis (DATA)</td>
<td>CALLTYPE, CONNECTION, DB2, DSGROUP, OPERATOR, PLAN, PROGRAM, STMTTYPE, TEXTHASH</td>
</tr>
<tr>
<td>SQMCACTU</td>
<td>User Analysis (DATA)</td>
<td>APPGROUP, CALLTYPE, CONNECTION, DB2, DSGROUP, OPERATOR, PLAN, PROGRAM, STMTTYPE, TEXTHASH</td>
</tr>
<tr>
<td>SQMCACTV</td>
<td>Interval Analysis</td>
<td>APPGROUP, CALLTYPE, CONNECTION, DB2, DSGROUP, OPERATOR, PLAN, PROGRAM, STMTTYPE, TEXTHASH</td>
</tr>
<tr>
<td>Report</td>
<td>Description</td>
<td>Valid qualifiers</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------------------------------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>SQMCACTW</td>
<td>Workstation ID Analysis (DATA)</td>
<td>APPGROUP CALLTYPE CONNECTION DB2 DSGROUP OPERATOR PLAN PROGRAM STMTTYPE (^a) TEXTHASH (^a)</td>
</tr>
<tr>
<td>SQMCACTX</td>
<td>SQL Statement Analysis (DATA)</td>
<td>APPGROUP CALLTYPE CONNECTION DB2 DSGROUP OPERATOR PLAN PROGRAM STMTTYPE TEXTHASH</td>
</tr>
<tr>
<td>SQMCAGTA</td>
<td>Client Application Analysis (GRAPH)</td>
<td>APPGROUP CALLTYPE CONNECTION DB2 DSGROUP OPERATOR PLAN PROGRAM STMTTYPE (^a) TEXTHASH (^a)</td>
</tr>
<tr>
<td>Report</td>
<td>Description</td>
<td>Valid qualifiers</td>
</tr>
<tr>
<td>------------</td>
<td>------------------------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>SQMCAGTB</td>
<td>Workstation Analysis (GRAPH)</td>
<td>APPGROUP CALLTYPE CONNECTION DB2 DSGROUP OPERATOR PLAN PROGRAM STMTTYPE a TEXTHASH a</td>
</tr>
<tr>
<td>SQMCAGTC</td>
<td>Connection ID Analysis (GRAPH)</td>
<td>APPGROUP CALLTYPE CONNECTION DB2 DSGROUP OPERATOR PLAN PROGRAM STMTTYPE a TEXTHASH a</td>
</tr>
<tr>
<td>SQMCAGTG</td>
<td>Group Analysis (GRAPH)</td>
<td>APPGROUP CALLTYPE CONNECTION DB2 DSGROUP OPERATOR PLAN PROGRAM STMTTYPE a TEXTHASH a</td>
</tr>
<tr>
<td>Report</td>
<td>Description</td>
<td>Valid qualifiers</td>
</tr>
<tr>
<td>------------</td>
<td>--------------------------------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>SQMCAGTI</td>
<td>Correlation ID Analysis (GRAPH)</td>
<td>APPGROUP, CALLTYPE,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CONNECTION,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DB2, DSGROUP, OPERATOR,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PLAN, PROGRAM, STMTTYPE,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>a, TEXTHASH</td>
</tr>
<tr>
<td>SQMCAGTP</td>
<td>Plan Analysis (GRAPH)</td>
<td>CALLTYPE, CONNECTION,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>a, DB2, DSGROUP, OPERATOR,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>a, PLAN, PROGRAM,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>a, STMTTYPE,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>a, TEXTHASH</td>
</tr>
<tr>
<td>SQMCAGTR</td>
<td>Program Analysis (GRAPH)</td>
<td>CALLTYPE, CONNECTION,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>a, DB2, DSGROUP, OPERATOR,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>a, PLAN, PROGRAM,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>a, STMTTYPE,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>a, TEXTHASH</td>
</tr>
<tr>
<td>SQMCAGTS</td>
<td>Subsystem Analysis (GRAPH)</td>
<td>CALLTYPE, CONNECTION,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>a, DB2, DSGROUP, OPERATOR,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>a, PLAN, PROGRAM,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>a, STMTTYPE,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>a, TEXTHASH</td>
</tr>
<tr>
<td>Report</td>
<td>Description</td>
<td>Valid qualifiers</td>
</tr>
<tr>
<td>----------</td>
<td>------------------------------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>SQMCAGTU</td>
<td>User Analysis (GRAPH)</td>
<td>APPGROUP CALLTYPE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CONNECTION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DB2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSGROUP OPERATOR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PLAN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PROGRAM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>STMTTYPE (^a)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TEXTHASH (^a)</td>
</tr>
<tr>
<td>SQMCAGTW</td>
<td>Workstation ID Analysis (GRAPH)</td>
<td>APPGROUP CALLTYPE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CONNECTION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DB2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSGROUP OPERATOR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PLAN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PROGRAM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>STMTTYPE (^a)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TEXTHASH (^a)</td>
</tr>
<tr>
<td>SQMCAGTX</td>
<td>SQL Statement Analysis (GRAPH)</td>
<td>APPGROUP CALLTYPE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CONNECTION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DB2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSGROUP OPERATOR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PLAN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PROGRAM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>STMTTYPE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TEXTHASH</td>
</tr>
<tr>
<td>SQMCCPU</td>
<td>Costliest SQL Statements</td>
<td>DB2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSGROUP</td>
</tr>
</tbody>
</table>

\(^a\) Indicates valid values for TEXTHASH and STMTTYPE depend on the context and might change depending on the specific report or analysis.
<table>
<thead>
<tr>
<th>Report</th>
<th>Description</th>
<th>Valid qualifiers</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQMCERRC</td>
<td>SQL Error Analysis</td>
<td>APPGROUP CALLTYPE CONNECTION DB2 DSGROUP OPERATOR PLAN PROGRAM STMTTYPE TEXTHASH</td>
</tr>
<tr>
<td>SQMCOBCD</td>
<td>Object Access Matrix</td>
<td>DB2^b DSGROUP PLAN</td>
</tr>
<tr>
<td>SQMCOBJW</td>
<td>SQL Workload and Object Analysis</td>
<td>DB2^b DSGROUP PLAN</td>
</tr>
<tr>
<td>SQMCTIME</td>
<td>Highest Elapsed SQL Statements</td>
<td>DB2 DSGROUP PLAN</td>
</tr>
<tr>
<td>SQMGDETL</td>
<td>Detail Analysis</td>
<td>APPGROUP CONNECTION DB2 DSGROUP OPERATOR PLAN PROGRAM TEXTHASH</td>
</tr>
<tr>
<td>SQMINTF1</td>
<td>Filter Collection Options</td>
<td>BP DB2 DSGROUP DBNAME PAGESET TBCREATOR TBNAME</td>
</tr>
<tr>
<td>Report</td>
<td>Description</td>
<td>Valid qualifiers</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------</td>
<td>--------------------------------------</td>
</tr>
<tr>
<td>SQMINTF2</td>
<td>Filter Exceptions Options</td>
<td>BP, DB2, DSGROUP, DBNAME, PAGESET, TBCREATOR, TBNAME</td>
</tr>
<tr>
<td>SQMINTVD</td>
<td>Select Analysis Interval</td>
<td>BP, DB2, DSGROUP, DBNAME, PAGESET, TBCREATOR, TBNAME</td>
</tr>
<tr>
<td>SQMSOBJB</td>
<td>Buffer Pool Analysis</td>
<td>BP, DB2, DSGROUP, DBNAME, PAGESET, TBCREATOR, TBNAME</td>
</tr>
<tr>
<td>SQMSOBJD</td>
<td>Database Analysis</td>
<td>BP, DB2, DSGROUP, DBNAME, PAGESET, TBCREATOR, TBNAME</td>
</tr>
<tr>
<td>SQMSOBJP</td>
<td>Page Set Analysis</td>
<td>BP, DB2, DSGROUP, DBNAME, PAGESET, TBCREATOR, TBNAME</td>
</tr>
<tr>
<td>Report</td>
<td>Description</td>
<td>Valid qualifiers</td>
</tr>
<tr>
<td>------------------</td>
<td>----------------------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>SQMSOBJT</td>
<td>Table Access Summary</td>
<td>BP, DB2, DSGROUP, DBNAME, PAGESET, TBCREATOR, TBNAME</td>
</tr>
<tr>
<td>SQMSOBTC</td>
<td>Object and Connection ID Analysis</td>
<td>BP, DB2, DSGROUP, DBNAME, PAGESET, TBCREATOR, TBNAME</td>
</tr>
<tr>
<td>SQMSOBTI</td>
<td>Object and Correlation ID Analysis</td>
<td>BP, DB2, DSGROUP, DBNAME, PAGESET, TBCREATOR, TBNAME</td>
</tr>
<tr>
<td>SQMSOBTM</td>
<td>Table Access Matrix</td>
<td>BP, DB2, DSGROUP, DBNAME, PAGESET, TBCREATOR, TBNAME</td>
</tr>
<tr>
<td>Report</td>
<td>Description</td>
<td>Valid qualifiers</td>
</tr>
<tr>
<td>--------------</td>
<td>---------------------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>SQMSOBTP</td>
<td>Object and Plan Analysis</td>
<td>APPGROUP, BP, CALLTYPE,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CONNECTION, DBNAME, DB2,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSGROUP, OPERATOR, PAGESET</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PLAN, PROGRAM, STMTTYPE,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TBCREATOR, TBNAME, TEXTHASH</td>
</tr>
<tr>
<td>SQMSOBTR</td>
<td>Object and Program Analysis</td>
<td>APPGROUP, BP, CALLTYPE,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CONNECTION, DBNAME, DB2,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSGROUP, OPERATOR, PAGESET</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PLAN, PROGRAM, STMTTYPE,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TBCREATOR, TBNAME, TEXTHASH</td>
</tr>
<tr>
<td>Report</td>
<td>Description</td>
<td>Valid qualifiers</td>
</tr>
<tr>
<td>-----------</td>
<td>------------------------------------------</td>
<td>----------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SQMSOBTS</td>
<td>Object and SQL Statement Analysis</td>
<td>APPGROUP BP CALLTYPE CONNECTION DBNAME DB2 DSGROUP OPERATOR PAGESET PLAN PROGRAM STMTTYPE TBCREATOR TBNAME TEXTHASH</td>
</tr>
<tr>
<td>SQMSOBTT</td>
<td>Table and Index Analysis</td>
<td>BP DB2 DSGROUP DBNAME PAGESET TBCREATOR TBNAME</td>
</tr>
<tr>
<td>SQMSOBTU</td>
<td>Object and User Analysis</td>
<td>APPGROUP BP CALLTYPE CONNECTION DBNAME DB2 DSGROUP OPERATOR PAGESET PLAN PROGRAM STMTTYPE TBCREATOR TBNAME TEXTHASH</td>
</tr>
<tr>
<td>Report</td>
<td>Description</td>
<td>Valid qualifiers</td>
</tr>
<tr>
<td>--------------</td>
<td>----------------------------------</td>
<td>------------------------------------------------------</td>
</tr>
<tr>
<td>SQMSOBTX</td>
<td>Object and SQL Statement Detail</td>
<td>APPGROUP, CALLTYPE, CONNECTION, DB2, DSGROUP, OPERATOR, PLAN, PROGRAM, STMTYPE, TEXTHASH</td>
</tr>
<tr>
<td>SQMSOBXC</td>
<td>Insert Access Analysis</td>
<td>APPGROUP, BP, CONNECTION, DBNAME, DB2, DSGROUP, OPERATOR, PAGESET, PLAN, PROGRAM, TBCREATOR, TBNAME, TEXTHASH</td>
</tr>
<tr>
<td>SQMSOBXD</td>
<td>Delete Statement Analysis</td>
<td>APPGROUP, BP, CONNECTION, DBNAME, DB2, DSGROUP, OPERATOR, PAGESET, PLAN, PROGRAM, TBCREATOR, TBNAME, TEXTHASH</td>
</tr>
<tr>
<td>Report</td>
<td>Description</td>
<td>Valid qualifiers</td>
</tr>
<tr>
<td>------------</td>
<td>--------------------------</td>
<td>----------------------------</td>
</tr>
<tr>
<td>SQMSOBXR</td>
<td>Read Access Analysis</td>
<td>APPGROUP, BP, CONNECTION,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DBNAME, DB2, DSGROUP,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OPERATOR, PAGESET, PLAN,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PROGRAM, TBCREATOR, TBNAME,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TEXTHASH</td>
</tr>
<tr>
<td>SQMSOBXU</td>
<td>Update Access Analysis</td>
<td>APPGROUP, BP, CONNECTION,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DBNAME, DB2, DSGROUP,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OPERATOR, PAGESET, PLAN,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PROGRAM, TBCREATOR, TBNAME,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TEXTHASH</td>
</tr>
<tr>
<td>SQSCACEX</td>
<td>SAP Exception Analysis</td>
<td>APPGROUP, CALLTYPE,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CONNECTION, DB2, DSGROUP,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OPERATOR, PLAN, PROGRAM,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>STMTTYPE, TEXTHASH</td>
</tr>
<tr>
<td>Report</td>
<td>Description</td>
<td>Valid qualifiers</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------------------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>SQSCACTC</td>
<td>SAP WP Number Analysis</td>
<td>APPGROUP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CONNECTION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DB2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSGROUP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OPERATOR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PLAN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PROGRAM</td>
</tr>
<tr>
<td>SQSCACTG</td>
<td>SAP Application Group Analysis (DATA)</td>
<td>APPGROUP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CONNECTION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DB2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSGROUP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OPERATOR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PLAN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PROGRAM</td>
</tr>
<tr>
<td>SQSCACTI</td>
<td>SAP WP ID Analysis</td>
<td>APPGROUP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CONNECTION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DB2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSGROUP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OPERATOR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PLAN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PROGRAM</td>
</tr>
<tr>
<td>SQSCACTP</td>
<td>SAP Plan Analysis (DATA)</td>
<td>DB2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSGROUP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PLAN</td>
</tr>
<tr>
<td>SQSCACTR</td>
<td>SAP Server Analysis</td>
<td>APPGROUP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CONNECTION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DB2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSGROUP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OPERATOR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PLAN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PROGRAM</td>
</tr>
<tr>
<td>SQSCACTS</td>
<td>SAP Subsystem Analysis (DATA)</td>
<td>DB2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSGROUP</td>
</tr>
<tr>
<td>Report</td>
<td>Description</td>
<td>Valid qualifiers</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>SQSCACTU</td>
<td>SAP User Analysis (DATA)</td>
<td>APPGROUP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CONNECTION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DB2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSGROUP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OPERATOR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PLAN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PROGRAM</td>
</tr>
<tr>
<td>SQSCACTX</td>
<td>SAP SQL Statement Analysis</td>
<td>APPGROUP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CALLTYPE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CONNECTION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DB2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSGROUP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OPERATOR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PLAN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PROGRAM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>STMTTYPE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TEXTHASH</td>
</tr>
<tr>
<td>SQSCERRC</td>
<td>SAP SQL Error Analysis</td>
<td>APPGROUP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CALLTYPE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CONNECTION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DB2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSGROUP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OPERATOR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PLAN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PROGRAM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>STMTTYPE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TEXTHASH</td>
</tr>
<tr>
<td>SQSCOBCD</td>
<td>SAP Object Access Matrix</td>
<td>DB2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSGROUP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PLAN</td>
</tr>
<tr>
<td>SQSCOBJW</td>
<td>SAP SQL Workload and Object</td>
<td>DB2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSGROUP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PLAN</td>
</tr>
</tbody>
</table>
### Report Description

<table>
<thead>
<tr>
<th>Report</th>
<th>Description</th>
<th>Valid qualifiers</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQSGDETL</td>
<td>SAP Detail Analysis</td>
<td>APPGROUP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CONNECTION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DB2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DSGROUP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OPERATOR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PLAN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PROGRAM</td>
</tr>
</tbody>
</table>

**a** Supported for batch if DATASOURCE(COLLECTOR(…)) is specified

**b** If DB2 is specified, you must specify at least one other qualifier

### Example

In the following example that uses QUALIFIER, this statement sets a global default qualifier that includes data for all plans beginning with PR. This QUALIFIER statement applies only to SQMBSTMT because the REPORT statement for SQMBSTMT has a QUALIFIER parameter that overrides the global QUALIFIER.

```
QUALIFIER(PLAN(PR*)) -
REPORT(NAME(SQMBSTMT) -
  QUALIFIER(DB2(DB2A) APGRP(FINANCE)) -
REPORT(NAME(SQMCACTR))
```

### Using Wildcards

Only an asterisk (*) and exclamation mark (!) are valid wildcards.

An asterisk, when used alone, matches all values for the identifier. All characters following an asterisk are ignored.

**Note**

If you use a lone asterisk (*) to include all values of an identifier, the effect is the same as making no specification for that identifier. All values of that identifier are included. If you use a lone asterisk to exclude all values of an identifier, no records are reported.

An asterisk, when used at the end of a character string matches all identifiers beginning with that character string.

An asterisk, where components of a qualifier are separated by a period (.), can be used within each component.
Example
Specifying **HWW4PP for an authid finds no matches.

The exclamation point is used to replace a single character at any position in a string. You cannot use this wildcard in the text hash or data sharing group qualifier.

Example
Specifying PLAN!A for Plan Name matches all plans with PLAN in the first four character positions and A in the sixth character position (for example, PLAN1A, PLAN2A, PLAN3A).

DMDBRTSU utility

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. DMDBRTSU generates statements and data to maintain BMCINDEXSPACESTATS and BMCTABLESPACE tables (for DB2 Version 9 and later).

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 LOADPLUS 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 on a daily basis, so the utility runs 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 106 on page 232 shows sample JCL and instructions to run the DMDBRTSU utility.

Figure 106: DMDBRTSU JCL

```
//DMDBRTSU   JOB   (ACCT), 'BMC RTS TABLES'.              === MODIFY
//               MSGCLASS=X.                                === CHECK
//               CLASS=A                                    === CHECK
```
Table 29 on page 233 describes the data sets that are processed by DMDBRTSU.

Table 29: DMDBRTSU data sets

<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>
</tbody>
</table>
### DDNAME | Description | DCB Attributes
--- | --- | ---
TABLDEF | Not used | Set to DUMMY
UNLDCTL | Holds utility statements for unloading real-time stats from DB2 tables | RECFM=FB LRECL=80 BLKSIZE=any multiple of LRECL
LOADCTL | Holds utility statements for loading real-time stats into BMC historical tables | RECFM=FB LRECL=80 BLKSIZE=any multiple of LRECL
SYSIN | Model utility statements, subject to DB2 release, containing tokens for date, DB2 environment | Not applicable

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 might 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

The following 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</th>
</tr>
</thead>
<tbody>
<tr>
<td>OBJ_STATISTICS</td>
<td>Statistics per object per defined collection interval</td>
<td>X</td>
</tr>
<tr>
<td>DAILY_OBJ_STATISTICS</td>
<td>Statistics per object per day</td>
<td>X</td>
</tr>
<tr>
<td>WEEKLY_OBJ_STATISTICS</td>
<td>Statistics per object per week</td>
<td>X</td>
</tr>
<tr>
<td>MONTHLY_OBJ_STATISTICS</td>
<td>Statistics per object per month</td>
<td>X</td>
</tr>
<tr>
<td>BASELINE_OBJ_STATISTICS</td>
<td>Statistics per object representing acceptable performance</td>
<td>X</td>
</tr>
<tr>
<td>STMT_STATISTICS</td>
<td>Statistics per statement per defined collection interval</td>
<td>X</td>
</tr>
<tr>
<td>DAILY_STMT_STATISTICS</td>
<td>Statistics per statement per day</td>
<td>X</td>
</tr>
<tr>
<td>WEEKLY_STMT_STATISTICS</td>
<td>Statistics per statement per week</td>
<td>X</td>
</tr>
<tr>
<td>MONTHLY_STMT_STATISTICS</td>
<td>Statistics per statement per month</td>
<td>X</td>
</tr>
<tr>
<td>BASELINE_STMT_STATISTICS</td>
<td>Statistics per statement representing acceptable performance</td>
<td>X</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>X</td>
</tr>
<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>X</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>X</td>
</tr>
<tr>
<td>Table</td>
<td>Description</td>
<td>IFCID</td>
</tr>
<tr>
<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>X</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>X</td>
</tr>
<tr>
<td>STMT_STATISTICS_OB</td>
<td>Statistics per object per statement per defined collection interval</td>
<td>X</td>
</tr>
<tr>
<td>STMT_TEXT</td>
<td>SQL statement text</td>
<td>X</td>
</tr>
<tr>
<td>STMT_ERRORS</td>
<td>SQL executions finishing with SQLCODE &lt; 0</td>
<td>X</td>
</tr>
<tr>
<td>STMT_EXCEPTIONS</td>
<td>SQL executions exceeding defined thresholds</td>
<td>X</td>
</tr>
<tr>
<td>STMT_EXCEPTIONS_HV</td>
<td>Host variable values associated with statement exceptions</td>
<td>X</td>
</tr>
<tr>
<td>STMT_EXCEPTIONS_OB</td>
<td>Object statistics associated with statement exceptions</td>
<td>X</td>
</tr>
<tr>
<td>INDEX_USAGE</td>
<td>Indexes with dates last accessed for read, change, update, delete</td>
<td>X</td>
</tr>
<tr>
<td>BMCTABLESPACESTATS</td>
<td>Table space real-time stats per day</td>
<td></td>
</tr>
<tr>
<td>BMCINDEXSPACESTATS</td>
<td>Index space real-time stats per day</td>
<td></td>
</tr>
</tbody>
</table>

* 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 in Table 31 on page 237.

Table 31: Performance Advisor samples

<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>IODDDDLBA</td>
<td>Creates table with APPTUNE data aggregated for baseline comparisons</td>
</tr>
<tr>
<td>IODDDDLDA</td>
<td>Creates table with APPTUNE data accumulated daily</td>
</tr>
<tr>
<td>IODDDDLDB</td>
<td>Creates a database</td>
</tr>
<tr>
<td>IODDDDLDR</td>
<td>Drops a database</td>
</tr>
<tr>
<td>IODDDDLLEV</td>
<td>Creates table with APPTUNE data representing exception and error events</td>
</tr>
<tr>
<td>IODDDDLIN</td>
<td>Creates table with APPTUNE data by workload interval</td>
</tr>
<tr>
<td>IODDDDLIU</td>
<td>Creates table with aggregated index usage information collected by APPTUNE</td>
</tr>
<tr>
<td>IODDDDLTE</td>
<td>Creates table with aggregated SQL text collected by APPTUNE</td>
</tr>
<tr>
<td>IODDDLRT</td>
<td>Creates table with real-time statistics accumulated daily, used by all releases of DB2</td>
</tr>
<tr>
<td>IODDDLRE</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>IODLODCA</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>IODLODIN</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>IODLODMO</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><strong>SAMP member</strong></td>
<td><strong>Description</strong></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 components.

It contains the following sections:

- “Description of source type information” on page 239
- “Working with reports” on page 246
- “Return codes for Workload Access Path Compare and Index Advisor” on page 256
- “Working tables used by Index Advisor” on page 257

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 extracted for the Workload Access Path Compare component or Workload Index Advisor.

Table 32 on page 239 lists the applicable worksheet required by each source.

Table 32: Applicable worksheets

<table>
<thead>
<tr>
<th>Source</th>
<th>Worksheet</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2 catalog</td>
<td>Table 33 on page 240</td>
</tr>
<tr>
<td>APPTUNE trace data sets</td>
<td>Table 33 on page 240</td>
</tr>
<tr>
<td>DBRM libraries</td>
<td>Table 35 on page 242</td>
</tr>
</tbody>
</table>
Table 33 on page 240 describes the information that you enter when identifying DB2 catalogs as the source of the extracted SQL.

Table 33: 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 the subsystem ID of the DB2 system from which you want to extract SQL text.</td>
<td>The DB2 catalog on this subsystem is 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 34 on page 240 describes the fields that you enter when identifying APPTUNE trace data sets as the source of the extracted SQL.

Table 34: 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 the DB2 subsystem on which the SQL text was executed.</td>
<td>The Workload Compare and Index Advisor feature 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>Specify SSID for Extract</td>
<td>Specify whether to ignore literal values for numbers and strings in dynamic SQL when determining matches.</td>
<td>Enter <strong>Y (Yes)</strong> to ignore literal values in the text of a dynamic SQL statement. All dynamic SQL statements that are otherwise the same are treated as the same SQL statement. Enter <strong>N (No)</strong> to honor literal values. All dynamic SQL statements that are otherwise the same but have different literal values are treated as different SQL statements. Workload Access Path Compare and Index Advisor can ignore literal values even if you did not choose this option when APPTUNE collected the data in the archived trace data set.</td>
</tr>
<tr>
<td>Field name</td>
<td>What you enter</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-------------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| List OPTION trace data sets| Specify whether to choose an archive trace data set from a list.              | - 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.  
  - 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. |
| 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  
  — 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. | 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. |

**SQL Workload Source - Add Archive Trace Data (PSSWC118)**
Field name | What you enter | Description
--- | --- | ---
DSN | Specify the name of an APPTUNE archived trace data set to be used for the extraction. | ■ 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 35 on page 242 describes the fields that you enter when identifying DBRM libraries as the source of the extracted SQL.

Table 35: 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>Specify SSID for Explain</td>
<td>Specify the DB2 subsystem on which the SQL text from the DBRM library is to be explained. Objects do not have to be bound on this subsystem, but the tables must exist.</td>
</tr>
<tr>
<td></td>
<td>Table qualifier</td>
<td>Specify a table qualifier. This table qualifier is used to qualify any unqualified SQL text in the Explain process.</td>
</tr>
</tbody>
</table>
| | Specify PDS with member name or wildcard member | Specify the data set and member name of the DBRM library. You can use wildcards in the member name. Examples:  
MYUSERID.TEST.DBRMLIB(TEST)  
MYUSERID.PROD.DBRMLIB(APROD*)  
MYUSERID.PROD.DBRMLIB(*)  
The wildcard character can be anywhere within the name (leading, trailing, or in the middle). |

Table 36 on page 242 describes the information that you enter when identifying a Performance Advisor Database (PADB) as the source of the extracted SQL.

Table 36: 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>PADB Table Creator a</td>
<td>Specify the name of the owner or creator that was used when the Performance Advisor Database was created. If the creator name has changed from the default value, edit the field to contain the updated value.</td>
</tr>
<tr>
<td>Field name</td>
<td>What you enter</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>-------------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>PADB STMT Text Table Name</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>
<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>
</tbody>
</table>
### Description of source type information

<table>
<thead>
<tr>
<th>Field name</th>
<th>What you enter</th>
<th>Description</th>
</tr>
</thead>
</table>
| DB2 SSID where the data was collected           | Specify the subsystem ID of the DB2 subsystem from which statement text and statistics were originally collected. | ▪ This value is used as a search parameter for the data in the Performance Advisor Database.  
▪ You can use a wildcard character (* or %) as part of the name (for example, DBN*). |

#### Extract parameters for Performance Advisor Database (PSSWC165)

| 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 uses the workload interval time closest to the specified date. The specified time is 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 defines 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 is DAYS. If you selected the WEEKLY statistics table, the duration label is WEEKS.

Note: If you omit both the LAST N and Approximate Start Date field from the batch parameters, the default value is LAST 1 and the time period depends on the table selected (as described above).

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 37 on page 245 describes the fields that you enter when identifying a SQL text data set as the source of the extracted SQL.

Table 37: 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>SQL Workload Source Extract from SQL Text Data Set (PSSWC116)</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>
<tr>
<td>Specify SSID for Explain</td>
<td>Specify SSID for Explain</td>
<td>The objects do not have to be bound on this subsystem.</td>
</tr>
</tbody>
</table>
| Table qualifier                   | Specify the table qualifier                               | 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 uses 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)). |
### Field name | What you enter | Description
---|---|---
**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 38 on page 246 describes the information that you enter when identifying the DB2 statement cache as the source of the extracted SQL.

### Table 38: 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>Specify SSID for Extract</strong></td>
<td>Specify the subsystem ID of the DB2 system from which you want to extract SQL text.</td>
<td>The Statement cache on this subsystem is 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 is created if the DSN_STATEMENT_CACHE_TABLE does not already exist. The runner of the batch job must 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 recomminindex reports. It includes information about
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.

The list, in Table 39 on page 247 is in alphabetic order by field label name. The first column shows whether the field is used in Workload Access Path Compare (WLC), Index Advisor (IA), or both. Other columns display the DB2 column name, the DB2 table or other source, and the section of the report where it appears.

The hlq.llqSAMP(PSSREPB) layout data set member controls the field labels. 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 252.

Table 39: Workload Access Path Compare and Index Advisor report field labels (from PSSREPB layout data set)

<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>SYSTABLEPART</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>TABS</td>
</tr>
<tr>
<td>WLC</td>
<td>CARDF</td>
<td>CARDF</td>
<td>SYSTABLES</td>
<td>DIST</td>
</tr>
<tr>
<td>Both</td>
<td>CLUSTER RATIO</td>
<td>CLUSTERRATIOF</td>
<td>SYSINDEXES</td>
<td>INDX</td>
</tr>
<tr>
<td>Apply</td>
<td>Report field label</td>
<td>DB2 column name</td>
<td>DB2 table or other source</td>
<td>Report section</td>
</tr>
<tr>
<td>-------</td>
<td>-------------------</td>
<td>-----------------</td>
<td>---------------------------</td>
<td>----------------</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>COLCARDDF</td>
<td>COLCARDDF</td>
<td>SYSCOLUMNS</td>
<td>COLS</td>
</tr>
<tr>
<td>Both</td>
<td>COLLID</td>
<td>CLID</td>
<td>SYSPACKAGE, APPTUNE trace data</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></td>
<td></td>
<td>SORTC_JOIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SORTC_ORDERBY</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SORTC_GROUPBY</td>
<td></td>
<td></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_SCHEMA</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, APPTUNE trace data</td>
<td>STMT</td>
</tr>
<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>FIRSTKEYCARDDF</td>
<td>SYSINDEXES</td>
<td>INDEX</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>Apply</td>
<td>Report field label</td>
<td>DB2 column name</td>
<td>DB2 table or other source</td>
<td>Report section</td>
</tr>
<tr>
<td>-------</td>
<td>--------------------</td>
<td>-----------------</td>
<td>---------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>Both</td>
<td>FULL KEYCARD</td>
<td>FULLKEYCARDDF</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>SYSSCOLDIST</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>SYSSCOLDIST</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,</td>
<td>STMT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>APPTUNE trace data</td>
<td></td>
</tr>
<tr>
<td>Apply</td>
<td>Report field label</td>
<td>DB2 column name</td>
<td>DB2 table or other source</td>
<td>Report section</td>
</tr>
<tr>
<td>-------</td>
<td>--------------------</td>
<td>----------------------------------</td>
<td>---------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>WLC</td>
<td>NEW UJOG</td>
<td>SORTN_UNIQUE, SORTN_JOIN, SORTN_ORDERBY, SORTN_GROUPBY</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>
<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 HINT</td>
<td>OTHER</td>
<td>SQL text</td>
<td>COLS</td>
</tr>
<tr>
<td>WLC</td>
<td>P A</td>
<td>PRIMARY_ACCESSSTYPE</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>Apply</td>
<td>Report field label</td>
<td>DB2 column name</td>
<td>DB2 table or other source</td>
<td>Report section</td>
</tr>
<tr>
<td>-------</td>
<td>-------------------</td>
<td>----------------</td>
<td>--------------------------</td>
<td>---------------</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>INDEX</td>
</tr>
<tr>
<td>Both</td>
<td>STATSTIME</td>
<td>STATSTIME</td>
<td>SYSCOLUMNS</td>
<td>COLS</td>
</tr>
<tr>
<td>WLC</td>
<td>STATSTIME</td>
<td>STATSTIME</td>
<td>SYSCOLUMNS</td>
<td>DIST</td>
</tr>
<tr>
<td>WLC</td>
<td>STATSTIME</td>
<td>STATSTIME</td>
<td>SYSCOLUMNS</td>
<td>TABS</td>
</tr>
<tr>
<td>IA</td>
<td>STMT COST</td>
<td>TOTAL_COST</td>
<td>DSN_STATEMENT_TABLE</td>
<td>STMT</td>
</tr>
<tr>
<td>Both</td>
<td>STMTNO</td>
<td>STMTNO</td>
<td>DSN_STATEMNT_TABLE</td>
<td>STMT</td>
</tr>
<tr>
<td>WLC</td>
<td>T</td>
<td>TYPE</td>
<td>SYSCOLUMNS</td>
<td>DIST</td>
</tr>
<tr>
<td>WLC</td>
<td>TABLE</td>
<td>TNAME</td>
<td>PLAN_TABLE</td>
<td>PATH</td>
</tr>
<tr>
<td>Both</td>
<td>TABLE CREATOR</td>
<td>CREATOR</td>
<td>SYSTABLES</td>
<td>TABL</td>
</tr>
<tr>
<td>WLC</td>
<td>TABLE CREATOR</td>
<td>CREATOR</td>
<td>SYSTABLES</td>
<td>PATH</td>
</tr>
<tr>
<td>Both</td>
<td>TABLE NAME</td>
<td>NAME</td>
<td>SYSTABLES</td>
<td>TABL</td>
</tr>
<tr>
<td>WLC</td>
<td>TB NO</td>
<td>TABNO</td>
<td>PLAN_TABLE</td>
<td>PATH</td>
</tr>
<tr>
<td>WLC</td>
<td>TB TY</td>
<td>TABLE_TYPE</td>
<td>PLAN_TABLE</td>
<td>PATH</td>
</tr>
<tr>
<td>Both</td>
<td>TSNAME</td>
<td>TSNAME</td>
<td>SYSTABLES</td>
<td>TABL</td>
</tr>
<tr>
<td>Both</td>
<td>U R</td>
<td>UNIQUERULE</td>
<td>SYSINDEXES</td>
<td>INDEX</td>
</tr>
<tr>
<td>IA</td>
<td>UPDAT</td>
<td>UPDATE</td>
<td>SQL text</td>
<td>COLS</td>
</tr>
<tr>
<td>Both</td>
<td>VERSION</td>
<td>VERSION</td>
<td>SYSPLAN, SYSPACKAGE, APPTUNE trace data</td>
<td>STMT</td>
</tr>
<tr>
<td>WLC</td>
<td>W O</td>
<td>WHEN_OPTIMIZE</td>
<td>PLAN_TABLE</td>
<td>PATH</td>
</tr>
<tr>
<td>Both</td>
<td>WEIGHTED COST</td>
<td>WEIGHTED</td>
<td>none</td>
<td>STMT</td>
</tr>
</tbody>
</table>
Changing the layout of the compare report

A layout data set member controls the appearance of the online and batch reports.

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 the copied member. You can save one or more layout members.

A sample of the default member is shown in Figure 107 on page 253.

Note
The name of hlq.llqSAMP varies 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 values that appear in Table 40 on page 253:
Table 40: Layout data set values that can be edited

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
</table>
| WIDTH          | Specify the width of a column. If the display width of a field is too short to show all of the values in the field, 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 (*) fill the entire field.
  ■ For characters, the product truncates the character text from the right. |
| SHOW           | Specify whether to display a column value on the report.
  ■ S - SHOW the column
  ■ H - HIDE the column                                                                 |
| COLUMN HEADING | Specify the heading of the report column. If the value that you enter in the column heading is longer than the width of the column, the column width expands to the width of the heading.
  To display a two word heading on two lines, type the two words separated by a space. For example, "EXPL SSID" appears on the report as EXPL SSID.
  To display a two word heading on the same line, separate the words with an underscore. For example, "EXPL_SSID" appears on the report as 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 247.

Figure 107: Default layout data set member (PSSREPB)

```sql
* SQL PERFORMANCE
* 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.

**STMT**

* **THIS IS INFORMATION FOR AN SQL STATEMENT.**

<table>
<thead>
<tr>
<th>WIDTH</th>
<th>SHOW</th>
<th>COL_NAME</th>
<th>COLUMN HEADING</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 S</td>
<td></td>
<td>EXPSSID</td>
<td>EXPL SSID</td>
</tr>
<tr>
<td>4 S</td>
<td></td>
<td>SSID</td>
<td>SRC SSID</td>
</tr>
<tr>
<td>8 S</td>
<td></td>
<td>PROG</td>
<td>NAME</td>
</tr>
<tr>
<td>12 S</td>
<td></td>
<td>CLID</td>
<td>COLLID</td>
</tr>
<tr>
<td>26 S</td>
<td></td>
<td>VERSION</td>
<td>VERSION</td>
</tr>
<tr>
<td>6 S</td>
<td></td>
<td>STMTNO</td>
<td>STMTNO</td>
</tr>
<tr>
<td>7 S</td>
<td></td>
<td>TOTAL_COST</td>
<td>STMT COST</td>
</tr>
<tr>
<td>7 S</td>
<td></td>
<td>COUNT</td>
<td>EXEC COUNT</td>
</tr>
<tr>
<td>8 S</td>
<td></td>
<td>WEIGHTED</td>
<td>WEIGHTED COST</td>
</tr>
<tr>
<td>8 S</td>
<td></td>
<td>DIFF</td>
<td>WEIGHTED DIFF</td>
</tr>
<tr>
<td>24 S</td>
<td></td>
<td>DIFFIN</td>
<td>DIFF IN</td>
</tr>
<tr>
<td>1 S</td>
<td></td>
<td>CTYP</td>
<td>SRC TYPE</td>
</tr>
<tr>
<td>26 H</td>
<td></td>
<td>TIMESTAMP</td>
<td>EXTRACT TIME</td>
</tr>
<tr>
<td>26 S</td>
<td></td>
<td>BIND_TIME</td>
<td>EXPLAIN TIME</td>
</tr>
</tbody>
</table>

**PATH**

* **THIS IS INFORMATION FOR AN ACCESS PATH STEP.**

<table>
<thead>
<tr>
<th>WIDTH</th>
<th>SHOW</th>
<th>COL_NAME</th>
<th>COLUMN HEADING</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 S</td>
<td></td>
<td>QBLOCKNO</td>
<td>QB BL</td>
</tr>
<tr>
<td>2 S</td>
<td></td>
<td>PLANNO</td>
<td>PL NO</td>
</tr>
<tr>
<td>1 S</td>
<td></td>
<td>METHOD</td>
<td>M E</td>
</tr>
<tr>
<td>2 S</td>
<td></td>
<td>TABNO</td>
<td>TB NO</td>
</tr>
<tr>
<td>2 S</td>
<td></td>
<td>ACCESTYPE</td>
<td>AC TY</td>
</tr>
<tr>
<td>2 S</td>
<td></td>
<td>MATCHCOLS</td>
<td>MT CL</td>
</tr>
<tr>
<td>12 S</td>
<td></td>
<td>TNAME</td>
<td>TABLE</td>
</tr>
<tr>
<td>12 S</td>
<td></td>
<td>ACCESSNAME</td>
<td>INDEX</td>
</tr>
<tr>
<td>2 S</td>
<td></td>
<td>INDEXONLY</td>
<td>I O</td>
</tr>
<tr>
<td>4 S</td>
<td></td>
<td>SORTN</td>
<td>NEW UJOG</td>
</tr>
<tr>
<td>4 S</td>
<td></td>
<td>SORTC</td>
<td>COMP UJOG</td>
</tr>
<tr>
<td>1 S</td>
<td></td>
<td>PREFETCH</td>
<td>P F</td>
</tr>
<tr>
<td>1 S</td>
<td></td>
<td>COLUMN_FN_EVAL</td>
<td>C F</td>
</tr>
<tr>
<td>2 S</td>
<td></td>
<td>MIXOPSEQ</td>
<td>MX SQ</td>
</tr>
<tr>
<td>3 S</td>
<td></td>
<td>TSLOCKMODE</td>
<td>LCK MOD</td>
</tr>
<tr>
<td>1 S</td>
<td></td>
<td>MERGE_JOIN_COLS</td>
<td>J C</td>
</tr>
<tr>
<td>1 S</td>
<td></td>
<td>PAGE_RANGE</td>
<td>P R</td>
</tr>
<tr>
<td>1 S</td>
<td></td>
<td>JOIN_TYPE</td>
<td>J T</td>
</tr>
<tr>
<td>6 S</td>
<td></td>
<td>OBLOCK_TYPE</td>
<td>OBLOCK TYPE</td>
</tr>
<tr>
<td>8 S</td>
<td></td>
<td>CREATOR</td>
<td>TABLE CREATOR</td>
</tr>
<tr>
<td>8 S</td>
<td></td>
<td>ACCESSCREATOR</td>
<td>INDEX CREATOR</td>
</tr>
<tr>
<td>2 S</td>
<td></td>
<td>TABLE_TYPE</td>
<td>TB TY</td>
</tr>
<tr>
<td>2 S</td>
<td></td>
<td>ACCESS_DEGREE</td>
<td>AC DG</td>
</tr>
<tr>
<td>2 S</td>
<td></td>
<td>ACCESS_PGROUP_ID</td>
<td>AC PG</td>
</tr>
<tr>
<td>1 S</td>
<td></td>
<td>JOIN_DEGREE</td>
<td>J D</td>
</tr>
<tr>
<td>2 S</td>
<td></td>
<td>JOIN_PGROUP_ID</td>
<td>JN PG</td>
</tr>
<tr>
<td>2 S</td>
<td></td>
<td>SORTC_PGROUP_ID</td>
<td>SC PG</td>
</tr>
<tr>
<td>2 S</td>
<td></td>
<td>SORTN_PGROUP_ID</td>
<td>SN PG</td>
</tr>
<tr>
<td>1 S</td>
<td></td>
<td>PARALLELISM_MODE</td>
<td>P M</td>
</tr>
<tr>
<td>4 S</td>
<td></td>
<td>CORRELATION_NAME</td>
<td>CORR NAME</td>
</tr>
<tr>
<td>6 S</td>
<td></td>
<td>GROUP_MEMBER</td>
<td>GROUP MEMBER</td>
</tr>
<tr>
<td>1 S</td>
<td></td>
<td>WHEN_OPTIMIZE</td>
<td>W O</td>
</tr>
<tr>
<td>26 H</td>
<td></td>
<td>BIND_TIME</td>
<td>EXPLAIN TIME</td>
</tr>
<tr>
<td>4 S</td>
<td></td>
<td>OPTHINT</td>
<td>OPT HINT</td>
</tr>
<tr>
<td>4 S</td>
<td></td>
<td>HINT_USED</td>
<td>HINT USED</td>
</tr>
<tr>
<td>1 S</td>
<td></td>
<td>PRIMARY_ACCESSSTYPE</td>
<td>P A</td>
</tr>
<tr>
<td>2 S</td>
<td></td>
<td>PARENT_OBLOCKNO</td>
<td>PA QB</td>
</tr>
</tbody>
</table>
**TABLE**
* This is information for a DB2 table.

<table>
<thead>
<tr>
<th>WIDTH</th>
<th>SHOW</th>
<th>COL_NAME</th>
<th>COLUMN HEADING</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 H</td>
<td></td>
<td>CREATOR</td>
<td>TABLE CREATOR</td>
</tr>
<tr>
<td>18 S</td>
<td></td>
<td>NAME</td>
<td>TABLE NAME</td>
</tr>
<tr>
<td>8 H</td>
<td></td>
<td>DBNAME</td>
<td>DBNAME</td>
</tr>
<tr>
<td>8 H</td>
<td></td>
<td>TSNME</td>
<td>TSNME</td>
</tr>
<tr>
<td>8 H</td>
<td></td>
<td>EDPROC</td>
<td>EDIT PROC</td>
</tr>
<tr>
<td>3 S</td>
<td></td>
<td>PCTPAGES</td>
<td>PCT PAGES</td>
</tr>
<tr>
<td>3 S</td>
<td></td>
<td>PCTROWCOMP</td>
<td>PCT COMP</td>
</tr>
<tr>
<td>9 S</td>
<td></td>
<td>CARDF</td>
<td>CARDF</td>
</tr>
<tr>
<td>9 S</td>
<td></td>
<td>NPAGESF</td>
<td>NPAGESF</td>
</tr>
<tr>
<td>1 S</td>
<td></td>
<td>ENCODING_SCHEME</td>
<td>E S</td>
</tr>
<tr>
<td>1 S</td>
<td></td>
<td>LOCKRULE</td>
<td>L R</td>
</tr>
<tr>
<td>4 S</td>
<td></td>
<td>FREEPAGE</td>
<td>FREE PAGE</td>
</tr>
<tr>
<td>3 S</td>
<td></td>
<td>PCTFREE</td>
<td>PCT FREE</td>
</tr>
<tr>
<td>8 S</td>
<td></td>
<td>BPOOL</td>
<td>BUFFER POOL</td>
</tr>
<tr>
<td>1 S</td>
<td></td>
<td>COMPRESS</td>
<td>C</td>
</tr>
<tr>
<td>3 S</td>
<td></td>
<td>SEGSIZE</td>
<td>SEG SIZE</td>
</tr>
<tr>
<td>4 S</td>
<td></td>
<td>PARTITIONS</td>
<td>PARTS</td>
</tr>
<tr>
<td>9 S</td>
<td></td>
<td>NACTIVEF</td>
<td>NACTIVEF</td>
</tr>
<tr>
<td>26 S</td>
<td></td>
<td>STATSTIME</td>
<td>STATSTIME</td>
</tr>
<tr>
<td>5 S</td>
<td></td>
<td>PCTUPD</td>
<td>% UPDATE</td>
</tr>
</tbody>
</table>

**COLUMNS**
* This is information for a column in a table.

<table>
<thead>
<tr>
<th>WIDTH</th>
<th>SHOW</th>
<th>COL_NAME</th>
<th>COLUMN HEADING</th>
</tr>
</thead>
<tbody>
<tr>
<td>18 S</td>
<td></td>
<td>NAME</td>
<td>COLUMN NAME</td>
</tr>
<tr>
<td>3 S</td>
<td></td>
<td>COLNO</td>
<td>COL NUM</td>
</tr>
<tr>
<td>5 S</td>
<td></td>
<td>FETCH</td>
<td>FETCH</td>
</tr>
<tr>
<td>5 S</td>
<td></td>
<td>UPDATE</td>
<td>UPDAT</td>
</tr>
<tr>
<td>5 S</td>
<td></td>
<td>GROUPBY</td>
<td>GRPB</td>
</tr>
<tr>
<td>5 S</td>
<td></td>
<td>ORDERBY1</td>
<td>SORT1</td>
</tr>
<tr>
<td>5 S</td>
<td></td>
<td>ORDERBY2</td>
<td>SORT2</td>
</tr>
<tr>
<td>5 S</td>
<td></td>
<td>ORDERBY3</td>
<td>SORT3</td>
</tr>
<tr>
<td>5 S</td>
<td></td>
<td>EQUAL</td>
<td>EQUAL</td>
</tr>
<tr>
<td>5 S</td>
<td></td>
<td>RANGE</td>
<td>RANGE</td>
</tr>
<tr>
<td>5 S</td>
<td></td>
<td>OTHER</td>
<td>OTHER</td>
</tr>
<tr>
<td>5 S</td>
<td></td>
<td>STAGE1</td>
<td>STAG1</td>
</tr>
<tr>
<td>5 S</td>
<td></td>
<td>INDEXABLE</td>
<td>IXABL</td>
</tr>
<tr>
<td>8 S</td>
<td></td>
<td>COLTYPE</td>
<td>COLUMN TYPE</td>
</tr>
<tr>
<td>5 S</td>
<td></td>
<td>LENGTH</td>
<td>COL LEN</td>
</tr>
<tr>
<td>3 S</td>
<td></td>
<td>SCALE</td>
<td>SCALE</td>
</tr>
<tr>
<td>1 S</td>
<td></td>
<td>NULLS</td>
<td>N</td>
</tr>
<tr>
<td>18 S</td>
<td></td>
<td>NAME</td>
<td>COLUMN NAME</td>
</tr>
<tr>
<td>3 S</td>
<td></td>
<td>COLNO</td>
<td>COL NUM</td>
</tr>
<tr>
<td>26 H</td>
<td></td>
<td>STATSTIME</td>
<td>STATSTIME</td>
</tr>
<tr>
<td>16 S</td>
<td></td>
<td>HIGH2KEY</td>
<td>HIGH2KEY</td>
</tr>
<tr>
<td>16 S</td>
<td></td>
<td>LOW2KEY</td>
<td>LOW2KEY</td>
</tr>
<tr>
<td>9 S</td>
<td></td>
<td>COLCARDF</td>
<td>COLCARDF</td>
</tr>
</tbody>
</table>

**INDEX**
* This is information for an index on a table.

<table>
<thead>
<tr>
<th>WIDTH</th>
<th>SHOW</th>
<th>COL_NAME</th>
<th>COLUMN HEADING</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 H</td>
<td></td>
<td>CREATOR</td>
<td>INDEX CREATOR</td>
</tr>
<tr>
<td>18 S</td>
<td></td>
<td>NAME</td>
<td>INDEX NAME</td>
</tr>
<tr>
<td>8 S</td>
<td></td>
<td>SAVINGS</td>
<td>SAVINGS</td>
</tr>
<tr>
<td>6 S</td>
<td></td>
<td>NUMSTMTS</td>
<td>#STMTS</td>
</tr>
<tr>
<td>9 S</td>
<td></td>
<td>RECOMMEND</td>
<td>RECOMMEND</td>
</tr>
<tr>
<td>1 S</td>
<td></td>
<td>UNIQUERULE</td>
<td>U R</td>
</tr>
<tr>
<td>1 S</td>
<td></td>
<td>CLUSTERING</td>
<td>C G</td>
</tr>
<tr>
<td>1 S</td>
<td></td>
<td>CLUSTERED</td>
<td>C D</td>
</tr>
<tr>
<td>9 S</td>
<td></td>
<td>NLEAF</td>
<td>NLEAF</td>
</tr>
<tr>
<td>4 S</td>
<td></td>
<td>NLEVELS</td>
<td>NLEVEL</td>
</tr>
<tr>
<td>8 S</td>
<td></td>
<td>BPOOL</td>
<td>BUFFER POOL</td>
</tr>
<tr>
<td>9 S</td>
<td></td>
<td>FIRSTKEYCARD</td>
<td>FIRST KEYCARD</td>
</tr>
<tr>
<td>9 S</td>
<td></td>
<td>FULLKEYCARD</td>
<td>FULL KEYCARD</td>
</tr>
<tr>
<td>9 S</td>
<td></td>
<td>CLUSTERRATIO</td>
<td>CLUSTER RATIO</td>
</tr>
<tr>
<td>26 H</td>
<td></td>
<td>STATSTIME</td>
<td>STATSTIME</td>
</tr>
</tbody>
</table>
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 an Workload Access Path Compare job.

Table 41 on page 257 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 41: 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 report.</td>
<td>0</td>
</tr>
<tr>
<td>This return code occurs under the following conditions:</td>
<td></td>
</tr>
<tr>
<td>- Nothing matched, so nothing could be compared.</td>
<td>1</td>
</tr>
<tr>
<td>- Nothing printed in the Detail report (when the workload contains archived trace data).</td>
<td></td>
</tr>
<tr>
<td>This return code occurs under the following conditions:</td>
<td>2</td>
</tr>
<tr>
<td>- A change in access path, indexes, stats, or text occurred, but the cost 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 is adverse.</td>
<td>4</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 42 on page 257 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 42: 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 code.</td>
<td>8</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 43 on page 258 shows the DSN_SORT_TABLE.

Table 43: 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 44 on page 258 shows the DSN_SORTKEY_TABLE.

Table 44: 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>Column name</td>
<td>From DB2 catalog table</td>
<td>NULL?</td>
<td>Type</td>
<td>Length</td>
</tr>
<tr>
<td>-------------</td>
<td>------------------------</td>
<td>---------------</td>
<td>------------</td>
<td>--------</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 45 on page 259 shows the DSN_VIRTUAL_INDEXES table.

Table 45: 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>Column name</td>
<td>From DB2 catalog table</td>
<td>NULL?</td>
<td>Type</td>
<td>Length</td>
</tr>
<tr>
<td>-------------</td>
<td>------------------------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
</tr>
<tr>
<td>PADDDED</td>
<td>not applicable</td>
<td>NOT NULL</td>
<td>CHAR</td>
<td>1</td>
</tr>
<tr>
<td>COLNO n&lt;sup&gt;a&lt;/sup&gt;</td>
<td>not applicable</td>
<td>NULL</td>
<td>SMALLINT</td>
<td>2</td>
</tr>
<tr>
<td>ORDERING n&lt;sup&gt;a&lt;/sup&gt;</td>
<td>not applicable</td>
<td>NULL</td>
<td>CHAR</td>
<td>1</td>
</tr>
</tbody>
</table>

<sup>a</sup> 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 46: IFCIDS used by SQL Performance reports

<table>
<thead>
<tr>
<th>Report</th>
<th>Description</th>
<th>IFCID</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>IODDSTAT</td>
<td>Subsystem Getpage Volume</td>
<td></td>
</tr>
<tr>
<td>IODIDPKG</td>
<td>Index Dependencies (Package)</td>
<td></td>
</tr>
<tr>
<td>IODIDPND</td>
<td>Index Dependencies (Plan)</td>
<td></td>
</tr>
<tr>
<td>IODIGETV</td>
<td>Index Getpage Volume</td>
<td></td>
</tr>
<tr>
<td>IODISTAT</td>
<td>Index Statistics</td>
<td></td>
</tr>
<tr>
<td>IODITABL</td>
<td>Table Information for an Index</td>
<td></td>
</tr>
<tr>
<td>IODSTMTS</td>
<td>Statement List</td>
<td></td>
</tr>
<tr>
<td>IODSTXDH</td>
<td>Statement Text</td>
<td></td>
</tr>
<tr>
<td>IODSTXDT</td>
<td>Table/Index Breakdown</td>
<td></td>
</tr>
<tr>
<td>IODSTXTH</td>
<td>Statement Text</td>
<td></td>
</tr>
<tr>
<td>IODTAPPL</td>
<td>Application Group Getpage Volume</td>
<td></td>
</tr>
<tr>
<td>IODTCRUD</td>
<td>Table CRUD Matrix</td>
<td></td>
</tr>
<tr>
<td>IODTGETV</td>
<td>Table Getpage Volume</td>
<td></td>
</tr>
<tr>
<td>IODTSTAT</td>
<td>Table Statistics</td>
<td></td>
</tr>
<tr>
<td>IODTSTIN</td>
<td>Defined Indexes for a Table</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.

- **BMC System Performance for DB2** — A solution that combines the features and functions of MainView for DB2, OPERTUNE for DB2, and Pool Advisor for DB2 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 108 on page 264 is an example of the main...
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.
Index

(AODSTMTS report 149

A

access paths, explaining and comparing 185
action codes 141
Active Index With No Reads batch report 148
administration 28
APGRP qualifier 208
APPGROUP qualifier 208
application groups 144
application profiles 144
APPTUNE (for DB2)
component 16
ASQEWAM1 144, 154

B

Batch Object Detail Analysis 148
batch reporting
batch format 148
overview 147
batch reports
Active Index With No Reads 148
Object Detail Analysis 148
SQL Exceptions Analysis 148
SQL Statement Analysis 148
SQL Text 148
batch utilities, overview 150
BMC IFCIDs 261

C

client IDs 61
cloned objects 153, 161
commands

EXPLAIN 146
EXPORT 147
HELP 45
IM 176
LOG 147
PANELID 37
SA 173
SORT 144
STATUS 148
TERSE 42
comparing dynamic SQL
from different periods of time 189
from different versions of DB2 186–188
components of SQL Performance 15
conventions, documentation 10
COPYDIR data set 54
correlation IDs 61

D

Data Collector
overview 25
status 42
data source information 155
database 61
DB2LOAD data set 198
DBRM 208
DBRM names 62
DCB attributes, DOMBRPT1 (produce printed reports) 198–200
DDL data set 175
DMDBMERG utility 53, 193
DMDBRTSU utility 232
documentation information 10
DOMESELT 44, 45, 264
DOMESELT panel 44, 263
DOMPLEX option set 25
DOMPLEX option sets (overview) 29
DSN3@ATH secondary authorization exit 33
environment 24
epilogue time 146
examples
  browse DDL data set 175
  browse DDL data set member 177
  display of cloned objects 168
  dropping index in What-If Index 168
  dynamic Explain on cloned objects 172
  edited first key column statistics 166
  Explain Results after index is dropped 170
  Explain Results after new index is created 167
  Explain Results showing cloned objects 176
  Explain Results showing dropped index 169
  Explain Results showing new index 167
  Explain Results showing updated statistics 172
  list of members in DDL data set 177
  message when import process is complete 178
  navigating through reports 143
  navigation\xd1 SQL Workload Analysis Menu 142
  navigation\xd1 Statement List report 143
  PSSEXTRJ error report 187
  results of dynamic Explain on imported DDL 179
  standard report elements 40
  Update Index Access Path Statistics panel 170
  updating index statistics using What-If Index 170
  Explain Analysis report 146
  EXPLAIN command 146
  Explain processing 146
  Explain Report fields 247
  EXPORT command 147

functions of SQL Performance 24

header (reports) 40

identifiers 143

IFCID, list of 261
IM command 176
IN-SQL elapsed time 145
Index Advisor
  benefits 98
  overview 97
  recommending an index 101
  reports 111
  reviewing 108
Index Component 19
indexes
  identifying obsolete 49
  recommendation reports 111
  recommending 101
  reviewing 108
input parameters
  PSSCOMPJ 186–188, 191
  PSSEXTRJ 186, 189
installation settings for Index Component 153
interval (reports) 42
INTERVAL statement 205
IODADREO program 48
IODESEL panel 144
IODSTMTS report 143, 160
IODSTXDH report 149, 161, 162
IODTCRUD report 160
IODTGETV report 159
IODTSTAT report 158
IODTSTIN report 159

layout
  of panels 37
  of reports 40
layout data set member 252
list of, Index Component reports 261
LOG command 147, 148
logging on 35
logging reports 147, 148

main menu 36
MainView (for DB2) -- Data Collector 263
measuring SQL statement execution 145
navigating between reports 137, 143 through the product 36

object analysis reports 148
Object Detail Analysis batch report 148
object set
  renaming 70
  copying 71
  editing 68
  selecting 67
  viewing 67
object sets
  defining 64
online Help 45
online reports (list of) 261
option sets
  DOMPLEX 29
options
  collection 30

PACKAGE 208
PANELID command 37
panels
  APPTUNE and Index Component Main Menu 154
  Explain Results 163, 174, 176
  Import DDL 176
  Import DDL Member Selection List 178
  Save DDL 174
  Select Key Columns 165
  Specify Index Attributes 164
  SQL Workload Analysis Menu 155
  SQL Workload Initial Analysis Level 156
  System and SQL Performance main menu 43, 263
  System and SQL Performance Products logo 35
  Update First Key Column Statistics 166
  Update Index Access Path Statistics 170
PASETUP member 50
Performance Advisor

benefits 47
data collection, configuring 52
DMDBMERG data sets 198
DMDBMERG SYSIN control statements 199
DMDBMERG utility 193
DMDBRTSU data sets 232
DMDBRTSU utility 232
IFCIDs 234
integrating with DASD MANAGER 58
INTERVAL statement 205
loading data 53
objects, identifying degraded 49
overview 47
PASETUP member 50
purging data 55
REORG Advisor, using 19, 57
reorganizing objects 48
reporting on LOAD records 56
sample members 236
setting up 50
statistics, loading 53
tables 234
tables, creating 50
task overview 49
utilities 193
plan name 62
Pool Advisor (using with SQL Performance) 263
prologue time 145
PSSCOMPJ job 186–188, 191
PSSEXTRJ job 186, 187, 189
PSSHLOAD job 187
PSSHLOD1member 187
PSSHUNL1 member 187
PSSHUNLD job 187
PSSMEML panel 178
PSSPW200 panel 163, 174, 176
PSSPW300 panel 165
PSSPWD00 panel 174
PSSPWI0 panel 176
PSSPWED panel 166
PSSPWIXA panel 164
PSSPWIXS panel 170
PSSREPB layout data set 247, 252
publications, related 10

R
R0001VFG data set 198
R0001VFL data set 198
records used by reports 261
related publications 10
report header 40
report identifiers 43
report logging 147, 148
Report Manager 26
reports
  Batch Object Detail Analysis 148
  DB2 Current Status 146
  Explain 146
  Defined Indexes for a Table 159
  interval 42
  Select Analysis Interval 156
  Statement List 143, 161
  Statement Text 161, 162
  Table CRUD Matrix 160
  Table Getpage Volume 158, 159
  Table Statistics 158
required settings 153

S
SA command 173
saved DDL 173
saving DDL 173
scrolling options 38
security 32
Session Status 148
settings for Index Component 153
SORT command 144
sorting data on reports 144
SQL Exceptions Analysis batch report 148
SQL Explorer (for DB2)
  component 16
SQL Performance
  architecture 24
  list of reports 261
  logo panel 35
  overview 15
  report hierarchy 137
  using with other Performance products 263
SQL Statement Analysis batch report 148
SQL Text batch report 148
SQMINITVD 156
SRCTRACE data set 199
STATUS command 148
sysplex environment 26
System and SQL Performance logo panel 35
System Performance (for DB2) 263

T

table 63
tablespace 63
tasks
  adding an index using What-If Index 162
  dropping an index using What-If Index 168
  identifying What-If Index candidates 157
  importing What-If Index changes 175
  preparing to use What-If Index 154
  saving What-If Index DDL changes 173
  updating statistics using What-If Index 170
  using the What-If Index function 161
TERSE command 42

U

user IDs 63
User Profiles 29
using online reports 143
using SQL Performance with other products 263
Using the Index Component 153, 185
Using the What-If Index Feature 154

W

What-If Index feature 153, 157
Workload Access Path Compare component
  about 17, 117
  capabilities 74
  compare process 78
  Comparing 87, 108
  Explaining 84, 105
  Extract sources 73
  hlq.OBJECT.seqNumber 74
  hlq.PATH.seqNumber 74
  hlq.REPOS 74, 97
  hlq.SQLTXT.seqNumber 74
  matching statements 78
  sample output 91, 112
  specifying JCL options 79, 105
  understanding the process 75, 99
Workload Index Advisor
  benefits 98
  recommending an index 101
  reports 111
  reviewing 108
Workload Index Advisor reviewing 110