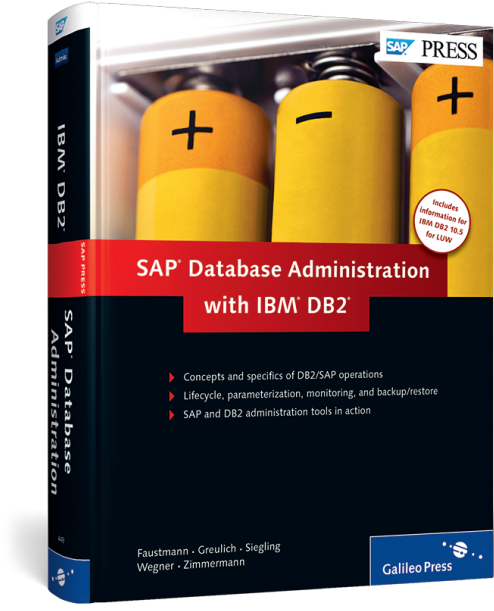
André Faustmann, Michael Greulich, André Siegling,
Benjamin Wegner, and Ronny Zimmerman

SAP® Database Administration with IBM® DB2®



Galileo Press

Bonn • Boston

Contents at a Glance

Contents

If you're new to SAP system administration with IBM DB2 for Linux, UNIX, and Windows databases, or you already have some experience and now want to gain a general overview of other features or functions, you've found the right book!

1 Introduction

IBM is a well-known information technology company that offers a wide range of products from computing technology, which covers the needs of entire data-center infrastructures, to equipment for small businesses and point-of-sales/services. From this wide range of IBM products, our focus in this book is on the database software that is also supported by SAP and its products.

DB2 databases overview

This book demonstrates and explains the powerful capabilities of the IBM DB2 for LUW (Linux, UNIX, and Windows) database (in the course of this book, we'll commonly refer to this database as DB2 for LUW). Over the past few years, some multifunctional tools have emerged that are a result of a very close cooperation between SAP and IBM.

When we talk about IBM's database, we need to be specific concerning which of the different products we are referring to.

There are three main streams within IBM's DB2 product portfolio (see Figure 1.1):

- IBM DB2 for Linux, UNIX, and Windows (LUW)
- IBM DB2 for z/OS
- IBM DB2 for i

Let's take a closer look at each of these.

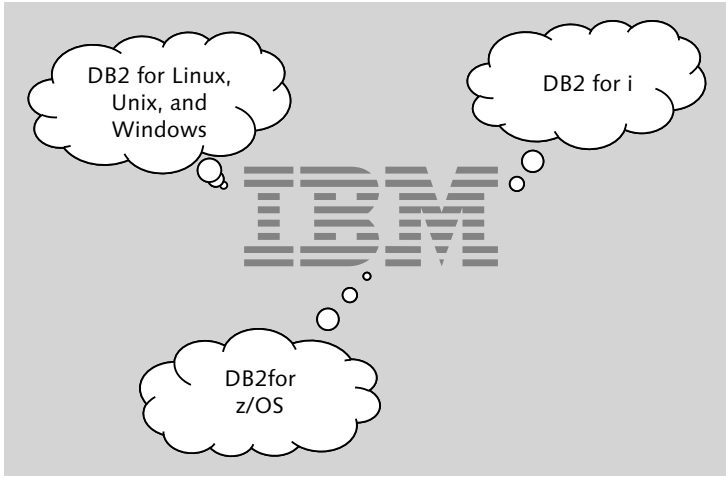


Figure 1.1 Different DB2 Databases Provided by IBM

IBM DB2 for LUW

The *IBM DB2 for Linux, UNIX, and Windows* (LUW) database is the database we are explaining in this book. Over the different releases in the past, the naming has changed a bit, so depending on the source you may see different nomenclatures. The following terms are taken from the SAP Product Availability Matrix (PAM) where the supported databases for different products are listed:

- DB2/UDB 8
- DB2 for LUW V9.1
- DB2 for LUW V9.7
- DB2 for LUW 10.1[1]

IBM DB2 for z/OS

The second database in our list of IBM's products, DB2 for z/OS, isn't just a database for another operating system. This product has its own product management and development stream. It's designed for IBM z mainframes and has a strong integration into the operating system. The handling of this database is also different compared to the DB2 for LUW database. The old name for this product was DB2/390, which was based

1 The "V" for version, as in V9.1 and V9.7, is usually not included for the latest software release of the database. So if a newer release (e.g., 11.0) is available, DB2 for LUW 10.1 will then be named DB2 for LUW V10.1.

on the OS390 operating system. In some rare cases, you might find this name as well.

IBM DB2 for i

Again, this database software is another product, not just another version for a different operating system. In the past, it was called DB2/400 or DB2 for i5/OS. This is a relational database management system (RDBMS) that works together with IBM i on IBM Power Systems (including AS/400, iSeries, and System i) and has a strong integration into the operating system.

DB4 and DB6

Now that you know something about the different designations of IBM's database products, there are two other names that may be used in the SAP context: DB4 (named for OS 400) and DB6 (RS6000 hardware is considered to the inspiration for DB6). If you see these terms, note that DB4 refers to the DB2 for i, and DB6 refers to the DB2 for LUW.

Informix

Finally, we also have to mention *Informix*. When IBM took over the Informix company in 2001, it also integrated the database of the same name into its product portfolio. Informix is still developed as its own product, but usually as a dedicated software product, so you won't mix it up with the other DB2 database products.

1.1 Who This Book Is For

Everyone interested in database technologies and administration in the wide field of SAP system landscapes should benefit from reading this book. If you are new to SAP system administration, this book shall help you understand the concepts related to SAP systems and the IBM DB2 database. You'll also learn how to perform basic system administration tasks and procedures.

This book also addresses system administrators who are already familiar with SAP system administration, but now find themselves with a new database vendor and the need for guidance regarding the IBM DB2-specific issues they will face.

1.2 Focus of This Book

This book will help you understand the basic concepts of the software we just discussed. If you are new to IBM DB2 for LUW, you'll get a good start with this book.

When you are administering SAP landscapes, you'll find a wide range of different tasks in your daily work life. Many will be planned, but you might also face unplanned failures, downtimes, and errors. In most cases, you should find ideas on how to handle the issues in your current situation within this book. The most important procedures will be explained in detail, whereas other issues will be addressed with an overview. In those cases, we'll point to documentation that covers everything in detail, such as in PDF documents, SAP Notes, or other documentation provided by SAP or IBM.

Make sure to check the SAP or IBM documentation when making far-reaching decisions, because contents in the documentation and recommendations may change over time. This book can help to find the relevant information and the sources to find the up-to-date information for your specific situation.

1.3 Contents of This Book

Chapter 2 After this introduction, we start Chapter 2 by spending some words on the SAP solutions we consider to be relevant for this book. First, we'll discuss the three-layer architecture that forms an SAP system. We'll draw a picture that shows how SAP systems can be distributed over different hosts, and then we'll focus on the database layer. You'll get an idea here of how IBM's products can meet the requirements of special situations regarding the database layer. We'll also look into the details of an SAP instance, including how the processes are organized and how they connect to a database.

Chapter 3 In Chapter 3, we discuss the IBM DB2 for LUW database. You'll learn about the database's architecture in detail. We start with the general concepts of database systems and explain the relational data model as the basis for this database as well. The introduction to the topic closes

with an overview of SQL as the language for accessing data stored within a database.

Within the architecture, we describe the kernel processes, log management, required file systems, buffer and memory management, organization of tablespaces, and log and trace files as an important source to gather information in certain situations.

Finally, we'll take a closer look at special and powerful features such as database partitioning, high availability, and the pureScale cluster.

Chapter 4

The lifecycle of SAP systems goes hand in hand with its databases, as described in Chapter 4. The chapter starts by giving an overview of the installation of SAP solutions. After installing, it's often necessary to apply patches as bug- and security fixes or to get new features. We'll touch on this topic as well. Then we show you how to copy SAP instances if needed. You'll also learn about the upgrade process and how SAP systems can be deleted if necessary.

Chapter 5

One of the biggest parts of this book is the chapter about the administration of the IBM DB2 for LUW database. We explain how to stop and start the database. Then we go over the DB2 command line processor (CLP). We also focus on the DB layout and the changes that might be necessary. You'll also learn about administration using the SAP tools and the powerful Transaction DBACOCKPIT. After explaining how to do these administration tasks, we then provide specific examples.

Chapter 6

The insurance for every administrator is the system backup. You hope that you seldom or never need it, but you should definitely have backups at least for your productive data. Chapter 6 shows how you can back up IBM DB2 for LUW environments. Then we show how to recover data and databases in case of failures, data loss, or unintended deletion of database contents. We discuss the tools that the database comes with, and even if you use centralized backup software and hardware, these database tools may be used in background. As an example, we'll show the integration of backups for IBM DB2 for LUW databases in the HP Data Protector backup software.

Chapter 7

Similar to the backup software, you'll use central monitoring software in your system landscape. One implementation that is very powerful,

especially for SAP system landscapes and environments, is SAP Solution Manager. In Chapter 7, we show you how to integrate the SAP systems in SAP Solution Manager monitoring, and which parameters need to be monitored.

Chapter 8 In this chapter, we focus on a powerful feature of IBM DB2 for LUW: database partitioning. This feature provides a lot of powerful possibilities to improve the performance for large databases, especially SAP NetWeaver Business Warehouse (BW).

Chapter 9 Finally, in Chapter 9 we provide some examples of what you might do if you face failures while you are responsible for an SAP system based on an IBM DB2 for LUW database. First, we explain how to get help or find information in general, followed by some examples that will inspire you if you face such situations.

We hope that the following chapters will provide the knowledge and insights you need as you work with system administration in SAP systems with IBM DB2 for LUW.

The clear best practice recommendation of IBM is, if possible, to set each parameter, which is dynamically administrable by STMM, to `AUTOMATIC`. SAP follows this recommendation precisely while installing DB2 for LUW, with an exception of $\texttt{INSTANCE\_MEMORY}_{\texttt{DBM}}$ because it gets a constant memory value.

```
***  stmmCostBenefitRecord ***
Type: PCKCACHESZ
PageSize: 4096
Saved Misses: 108
Benefit:
  -> Simulation size: 9047
  -> Total seconds saved: 4 (+ 204238061 ns)
  -> Normalized seconds/page: 2.3213E-06
Cost:
  -> Simulation size: 9047
  -> Total seconds saved: 4 (+ 204238061 ns)
  -> Normalized seconds/page: 2.3213E-06
Current Size: 30000
Minimum Size: 2624
Potential Increase Amount: 12032
Potential Increase Amount From OS: 15000
Potential Decrease Amount: 5984
Pages Available For OS: 5984
Interval Time: 200.194
     ■ ■ ■
***  stmmCostBenefitRecord ***
Type: PCKCACHESZ
PageSize: 4096
Original Size: 30000
Desired New Size: 45000
Actual New Size: 45000
Minimum Size: 2624
Potential Increase Amount: 0
Potential Increase Amount From OS: 0
Potential Decrease Amount: 5984
Pages Available For OS: 5984
Interval Time: 200.194
```

Figure 3.17 Log Entries of STMM in stmm.<number>.log

3.4 Tablespaces

Next to process architecture and memory management, data storage constitutes the third pillar of a DBMS. Each database requires a storage structure to store its data. In the environment of relational databases, this data is called a *tablespace*. In DB2 for LUW databases as well, all data such as tables, indexes, and so on are stored in tablespaces.

3.4.1 Basis of DB2 for LUW Tablespaces

Tablespaces constitute a logical storage structure on the database level. The database reproduces its storage structure on physical files or RAW devices. In the DB2 for LUW environment, this storage location is called a container, regardless of whether it's a file, a folder, or a raw device. Figure 3.18 shows an example.

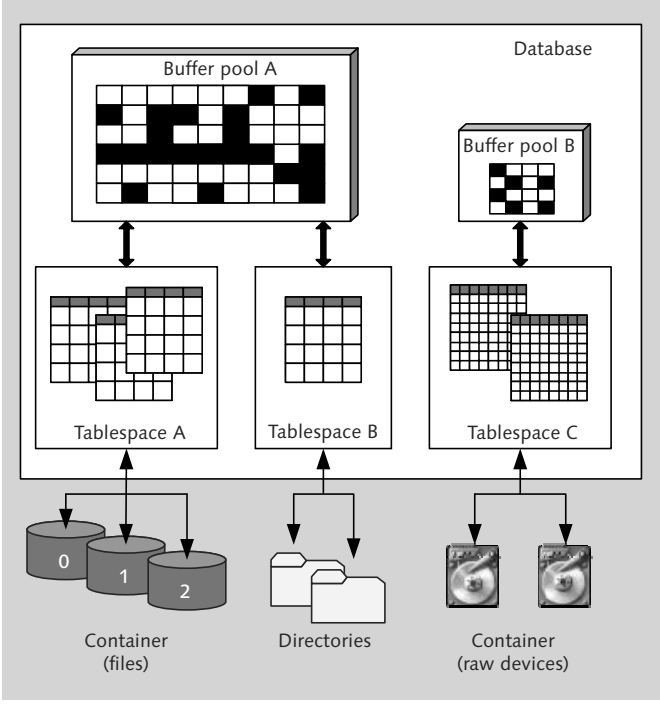


Figure 3.18 DB2 for LUW Tablespaces and Containers

Tablespace categories

The DB2 for LUW database essentially distinguishes four tablespace categories:

- **Large tablespace**
  This is the main category of DB2 for LUW tablespaces in an SAP environment. Like the regular tablespace, this tablespace category stores tables and indexes. It was developed to overcome the boundaries of the "old" regular tablespace, can hold significantly larger tables, and supports more than 255 columns on one data page.

- **Regular tablespace**
  This is the normal category of DB2 for LUW tablespaces. It stores all permanent data; that is, tables and indexes. You can even store large objects (LOB) data in this tablespace. The catalog tablespace SYSCATSPACE is a special regular tablespace. It's obligatory for each DB2 for LUW database and contains the system catalog tables.
- **System temporary tablespace**
  In this tablespace, temporary and database-internal data is stored, which is generated during different operations, such as sorting, creating an index, or reorganizing tables in a tablespace.
- **User temporary tablespace**
  This tablespace stores temporary declared tables that the user creates in the database.

During the creation of a DB2 for LUW database, at least one catalog tablespace (as regular tablespace), one large tablespace, and one system temporary tablespace are created for ensuring database functionality.

Types of tablespace management

The tablespace management, executed by you or the database administrator, is important for the database's functioning. DB2 for LUW has three concepts for tablespace management:

- **System Managed Storage tablespace (SMS)**
  The operating system takes over container management here. Containers are folders in which objects are stored. Each database object (for example, a table), has at least one physical file you access by usual system calls to the operating system. The properties of a SMS tablespace include the following:
  - During the creation of database objects, a file is created for each object and stored in a folder, which is the container of this object. If there are several containers, the data pages of the object `RoundRobin` are distributed over these containers.
  - Large tablespaces can't be managed via SMS.
  - A table and the corresponding indexes must be located in one tablespace.
  - If necessary, the object files in the containers grow until the file system is full. If there are several containers or directories, the tablespace is full after the first container or directory is filled.

- After a tablespace has been created, changes such as adding a container are no longer possible.
- I/O is executed through the file cache of the operating system.

- **Database Managed Storage tablespace (DMS)**
  The database takes over container management itself here. Containers are files or raw devices (device files). Following are the properties of a DMS tablespace:
  - During creation, containers (files) are specified by a container size. They can grow by use of the `AUTORESIZE` option, until the optional `MAXSIZE` is reached.
  - A table and the corresponding indexes can be located in different tablespaces.
  - After tablespace creation, it's possible to add, increase, decrease, or delete containers.
  - Performance is about 5–15 % better than for SMS tablespaces because DMS is using preallocated space.
  - Higher management efforts are required than with SMS tablespaces, but there is better control over the data storage.
- **Automatic Storage**
  Compared to SMS and DMS, this is actually not a tablespace management type but rather uses both management types. It offers an automated tablespace management on a higher level. For more information, see Section 3.4.1.

Note

Starting with versions 10.1 and 10.1 FP1 of DB2 for LUW, SMS, and DMS tablespaces are declared outdated for permanent data. Therefore, these two types should only be used for tablespaces with temporary data, and all permanent data in the future should be stored in automatic storage tablespaces.

Tablespace configuration

If you administrate the DB2 for LUW database in an SAP environment, you generally use the DBA Cockpit (see Chapter 5). Nevertheless, we'll now show you how to create a tablespace by using an example to illustrate various important tablespace parameters. The statement in Listing 3.1 creates a tablespace in an SAP environment.

```
db2 "CREATE LARGE TABLESPACE EXAMPLETBLS
IN DATABASE PARTITION GROUP SAPNODEGRP_PRD
PAGESIZE 16384
MANAGED BY AUTOMATIC STORAGE
USING STOGROUP IBMSTOGROUP
INITIALSIZE 32 M
INCREASESIZE 32 M
MAXSIZE 1 G
EXTENTSIZE 2
PREFETCHSIZE AUTOMATIC
BUFFERPOOL IBMDEFAULTBP
NO FILE SYSTEM CACHING
DROPPED TABLE RECOVERY OFF"
```

Listing 3.1 Create a Tablespace in SAP

The resulting tablespace is stored in the storage group `IBMSTOGROUP` (see Section 3.4.1). Tablespace configuration is executed with the following parameters:

- `PAGESIZE`

  The parameter `Pagesize` specifies the size of the data pages in the tablespace. A page is the smallest usable data block in a tablespace; that is, a growing table is always extended by one page as the contents grow. The standard value `PAGESIZE` is defined during the creation of a database through the database parameter `pagesize`; otherwise, it's 4KB. The page size of a tablespace limits the maximum size, according to tablespace category and type. Table 3.3 gives you an overview.

Page Size	DMS Regular Tablespace	DMS Large Tablespace or Automatic Storage	SMS Regular Tablespace	SMS Tablespace in Automatic Storage
4 KB	64 GB	8 TB	64 GB	8 TB
8 KB	128 GB	16 TB	128 GB	16 TB
16 KB	256 GB	32 TB	256 GB	32 TB
32 KB	512 GB	64 TB	512 GB	64 TB

Table 3.3 Limits by Tablespace Page Size

- `IN DATABASE PARTITION GROUP`
  You have to specify a database partition group here; otherwise, the `IBMDEFAULTGROUP` would be taken by default (or `IBMTEMPGROUP` for the system temporary tablespaces).
- `MANAGED BY AUTOMATIC STORAGE USING STOGROUP`
  As a target for the container, a storage group is used here (see Section 3.4.2). This is possible since DB2 for LUW 10.1. The usage of a target path or raw device, which was used before 10.1, is also possible.

Note: Alternative

The following alternative can also be used:
`MANAGED BY DATABASE USING <container-specifications> –`
For DMS tablespaces, you have to use containers (with location and size) or raw devices as a target.

- `INCREASESIZE, INITIALSIZE, MAXSIZE`
  Containers are created with the size `INITIALSIZE`; they grow automatically according to the step size `INCREASESIZE`, until `MAXSIZE`. If these parameters aren't defined, the standard values are `INCREASESIZE, INITIALSIZE = specified by DBM`, and `MAXSIZE=NONE`. (The not shown parameter `AUTORESIZE` is `YES` by default, which only makes sense in case of an `AUTOMATIC STORAGE` tablespace.) It's important to know that these parameters belong to the entirety of all containers. For example, if you have four containers, and the `INCREASESIZE` is 32MB, then every container will grow up with 8MB in one increasing step.
- `EXTENTSIZE`
  This has two meanings:
  - If the tablespace has more than one container, `EXTENTSIZE` determines the number of pages that are written in a container during the round-robin process, before the next container is used.
  - `EXTENTSIZE` specifies in which portions (in pages) space is allocated for tables or indexes within the containers of a tablespace. When these pages are full, the next area of pages is reserved with the appropriate number of pages.

  These two meanings ensure that a table is spread equally over all containers of a tablespace as illustrated by Figure 3.19.

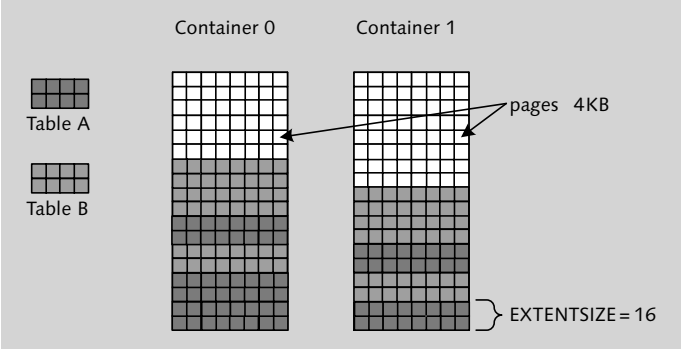


Figure 3.19 EXTENTSIZE in a Tablespace

The standard value for `EXTENTSIZE` is determined by the parameter `dft_extent_sz` for the DB2 for LUW database; the standard value in an SAP environment is `2`. The default value is only used if no `EXTENTSIZE` value is specified in the `CREATE TABLESPACE` statement.

- `PREFETCHSIZE`
  This specifies how many pages are loaded from the tablespace into the buffer pool during the prefetching function. The standard value for the DB2 for LUW instance is determined by the parameter `dft_prefetch_sz` and equals `AUTOMATIC`.
- `BUFFERPOOL`
  This defines the buffer pool the tablespace has to use; the page size of the buffer pool has to be consistent with the page size of the tablespace. The standard value indicates the default buffer pool `IBMDEFAULTBP`.
- `NO FILE SYSTEM CACHING`
  This deactivates the local file system cache for I/O operations. The default value depends on the operating system, the file system, and the data type (temporary, LOB, etc.). The DB2 for LUW documentation contains an extensive table giving the values for the different operation and file systems, and indicates how and when the different I/O types (direct IO, concurrent I/O) can be inserted. In general, you can say that the I/O performance takes advantage of using this option.
- `DROPPED TABLE RECOVERY ON`
  This activates the function for restoring deleted tables with the

ROLLFORWARD DATABASE command. The default value in an SAP environment is OFF.

Show tablespace information

As administrator, it's important for you to know how to access information on tablespaces in the DB2 for LUW database. The following command enables you to check administrative details of all tablespaces:

```
db2 list tablespaces show detail
```

Figure 3.20 shows an extract of this command output. You see information on the tablespace R15#STABD from SAP system R15.

```
Tablespace ID                        = 8
Name                                 = R15#STABD
Type                                 = Database managed space
Contents                             = All permanent data. Large table space.
State                                = 0x0000
  Detailed explanation:
    Normal
Total pages                          = 1327104
Useable pages                        = 1327102
Used pages                           = 1325460
Free pages                           = 1642
High water mark (pages)              = 1325460
Page size (bytes)                    = 16384
Extent size (pages)                  = 2
Prefetch size (pages)                = 2
Number of containers                 = 1
Minimum recovery time                = 2012-12-27-08.15.19.000000
```

Figure 3.20 Tablespace Information

Figure 3.20 shows you that the tablespace R15#STABD belongs to the DMS type and to the category Large Tablespace. You can also see the statistics of the pages, which indicate how large the tablespace is and how many pages are available. Because AUTORESIZE is activated by default in SAP environments, the number of free pages should not be large during normal tablespace growth. The number of free pages doesn't increase unless tablespace data is deleted; for this reason, reorganization possibly can be reasonable (see Section 3.4.3). The High water mark indicates the highest filling level of the tablespace, as well as EXTENTSIZE, PREFETCHSIZE, and the number of containers belonging to the tablespace.

Tablespace status

The status of the tablespace is yet another key piece of information in Figure 3.20. A hexadecimal coding indicates the tablespace status. Most status values are related to tablespace operations executing external DB2

for LUW commands; for example, backup, loading, and so on. Table 3.4 briefly overviews the main statuses of a DB2 for LUW tablespace.

Status Code	Significance
0x0	Normal: The tablespace is active and usable.
0x20	Upcoming backup: Tablespace backup must be performed before write access is possible.
0x40	Ongoing recovery: Tablespace recovery is being performed.
0x80	Upcoming recovery: After completing a restore operation, a tablespace recovery has to be performed before it can be used again.
0x100	Upcoming restore: After creating the containers for a tablespace that has to be restored, the tablespace is waiting for restore.
0x400	Ongoing reorganization: A table of the tablespace is currently being reorganized.
0x800	Ongoing backup: The tablespace is currently being saved.
0x2000	Ongoing restore: The tablespace is currently being restored.
0x4000	Offline: One container is damaged. The tablespace can't be used.
0x10000000	DMS rebalancing function: Between the containers, a rebalancing is executed (see Section 3.4.1).

Table 3.4 Important Status Values for DB2 for LUW Tablespaces

Altogether, there are 25 different status values for tablespaces. It's important to know that tablespaces can have several status values at the same time, under certain circumstances.

Parameters that influences tablespaces

Besides the options to configure tablespaces during their creation, a number of other parameters influence the usage of tablespaces. The three main parameters are listed here:

- `NUM_IOSERVERS` (database parameter)
  This parameter defines how many `db2pfchr` (prefetchers) are started for each database, for loading data (pages) from the tablespace to the

buffer pool in advance. The current default value is set on AUTOMATIC; that is, it complies with maximum parallelism (see the next parameter) but is at least 3.

- DB2_PARALLEL_IO (DB2 profile registry variable)
  If this parameter isn't set, the DB2 for LUW database assumes that there is a physical disk behind each container; that is, for a tablespace with two containers, I/O parallelism is set to 2. Beyond this parameter, a complex behavior is hiding; for example, you can also specify different degrees of parallelism for different tablespaces depending on their storage layout. We can't go more into detail here. In practice, especially in SAP environments, it's proven that the most simple and effective way to achieve good I/O performance is to parallelize the I/O over the number of containers for a tablespace. SAP default is to not set DB2_PARALLEL_IO.
- DB2_SET_MAX_CONTAINER_SIZE (DB2 profile registry variable)
  If this parameter is set, it defines a limit to the container size in an automatic storage tablespace. If a container file reaches this limit, the automatic storage will create a new container file.

Internal organization in tablespaces

We won't discuss the internal organization of data storage in tablespaces and containers in detail at this point because it would exceed the scope of this book. Nevertheless, for your work with tablespaces, you need to know some terminology. In the following, we particularly discuss the internal organization of the DMS tablespace and its containers with stripes, stripe sets, and ranges. Figure 3.21 illustrates one tablespace with four (very small) containers. Container 0 and Container 1 each receive six extents (size of an extent = EXTENTSIZE), while Container 2, 3, and 4 each receive three extents.

A *stripe* is an order of extents that is used across containers. The Stripes 0, 1, and 2 contain the Containers 0, 1, and 2, whereas the Stripes 3, 4, and 5 only contain the Containers 0 and 1, and so on. A *range* consists exactly of those stripes, which hold the same containers in a coherent way; thus, Range 0 contains the Stripes 0-2 (Extent 0-8), Range 1 contains the Stripes 3-5 (Extent 9-14), and so on. The stripe sets represent the third unit. A *stripe set* is a group of containers of a tablespace, which is separated from other containers of that tablespace; that is, they don't form

a stripe together. Figure 3.21 illustrates this principle. Within a stripe set, the extents are distributed via round-robin over the involved containers.

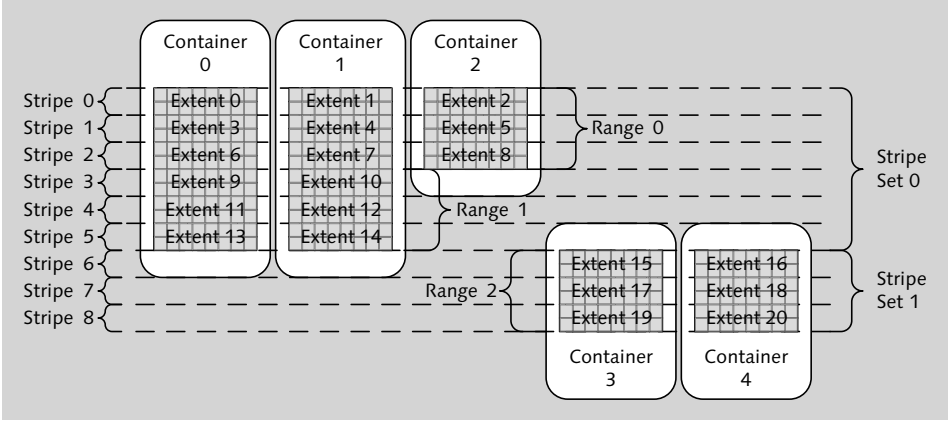


Figure 3.21 Internal Organization of a Tablespace

The arrangement of the tablespace elements just named results in the *tablespace map*. The tablespace map for the example in Figure 3.21 is as follows (excluding `EXTENTSIZE=16`) in Table 3.5.

Tablespace map

Range No.	Stripe Set	Stripe Offset	Max. Extent	Max. Pages	Start Stripe	End Stripe	Container
0	0	0	8	143	0	2	3 (0, 1, 2)
1	0	0	14	239	3	5	2 (0, 1)
2	1	6	20	336	6	8	2 (3, 4)

Table 3.5 Tablespace Map

It's important to understand the basic structure of a DMS tablespace, particularly for increasing or decreasing procedures (see Section 3.4.3).

As a DB2 user, you can use the following command to access information on the tablespace map and containers:

```
db2 get snapshot for tablespaces on <DB Name>
```

The output of this query gives you a lot of information on a tablespace and its container. Figure 3.22 shows a small example.

```
Tablespace name                            = PR2#USER1D
  Tablespace ID                            = 34
  Tablespace Type                          = Database managed space
  Tablespace Content Type                  = All permanent data. Large table space.
  Maximum tablespace size (bytes)          = NONE
  Increase size (bytes)                    = AUTOMATIC
  Time of last successful resize           =
  Last resize attempt failed               = No
  Rebalancer Mode                          = No Rebalancing
  Storage paths have been dropped          = No
  Minimum Recovery Time                    =
  Number of quiescers                      = 0
  Number of containers                     = 1
  Container Name                           = /db2/PR2/sapdata1/db2pr2/NODE0000/PR2/T0000034/C0000000.LRG
      Container ID                         = 0
      Container Type                       = File (extent sized tag)
      Total Pages in Container             = 2048
      Usable Pages in Container            = 2046
      Stripe Set                           = 0
      Container is accessible              = Yes
      File system ID                       = 2050
      File system used space (bytes)       = 72656637952
      File system total space (bytes)      = 97052438528
  Table space map:
   Range  Stripe Stripe  Max         Max  Start  End    Adj.   Containers
   Number Set    Offset  Extent      Page Stripe Stripe
   [   0] [   0]      0    1022      2045      0   1022   0    1 (0)
```

Figure 3.22 Details of a Tablespace (Extract)

With this query, you find information on automatic resizing (INCREASE SIZE, TIME OF LAST SUCCESSFUL RESIZE, etc.), for example, or on REBALANCER MODE. Furthermore, the individual containers and tablespaces are displayed, so you can recognize to which STRIPE SET this container belongs. In an SAP environment, you can find a graphical presentation of the tablespace map in the WebDynpro version of the DBA Cockpit (see Chapter 5).

Now that you have a good foundational understanding of what tablespaces are, in the following sections, we'll delve into some of the details of the topics already discussed.

3.4.2 Automatic Storage

As indicated before, Automatic Storage was launched with DB2 for LUW 9. Since then, this concept has been improved with each new release; and since DB2 for LUW 10.1 it even supports the latest database storage concepts such as multi-temperature storage. Because Automatic Storage is activated by default since DB2 for LUW release 9.1, you need to know its basic concepts and main processes.

Simplify by Automatic Storage

Automatic Storage puts tablespace management on a higher level, and thereby dramatically simplifies administration. Thus, available storage capacity is managed on the database level and not for each individual tablespace anymore. Figure 3.23 shows this from a logical perspective.

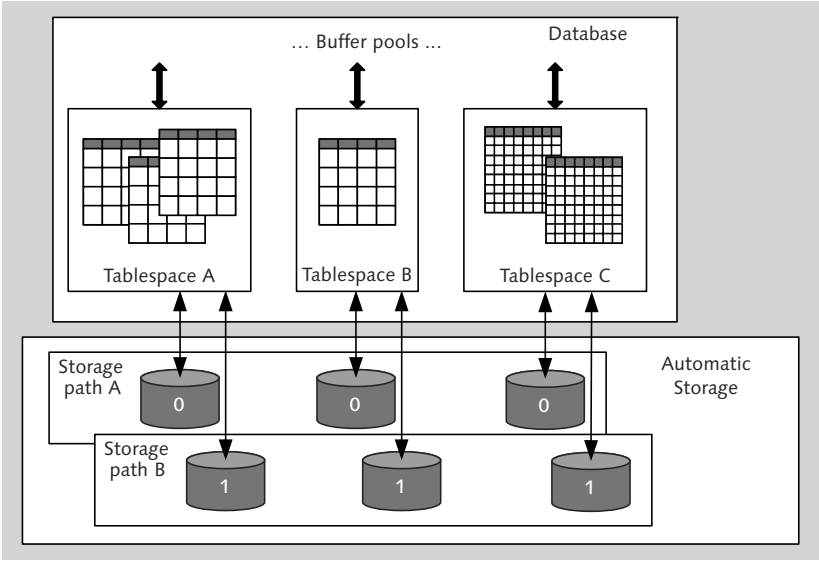


Figure 3.23 Automatic Storage

The groundwork for the usage of Automatic Storage is already laid out during the creation of a database. The following extract of a command shows the creation of an SAP database for the system PR2:

```
create database PR2
automatic storage yes on
      /db2/PR2/sapdata1, /db2/PR2/sapdata2
dbpath on /db2/PR2 …
```

The option `automatic storage yes` activates Automatic Storage, while the clause `on /db2/PR2/sapdata1, /db2/PR2/sapdata2` specifies the storage path. The second option `dbpath on` determines the highest point of the hierarchical directory structure of a database. The database directory structure holds all information related to this database, such as database configuration files, history files, and so on (for more information, see Section 3.5). Hence, the option `dbpath` enables you to physically separate

the actual data files (that is, containers), from database management information.

Obligatory tablespaces

After creating the database with Automatic Storage and a storage path, the three obligatory tablespaces are created automatically (catalog tablespace, system temporary tablespace, user tablespace). All other tablespaces, created by the administrator or installation scripts, consequently don't contain information on storage locations. All other options (`EXTENTSIZE`, etc.) for tablespaces can be used as discussed previously. Automatic Storage independently creates the containers in the storage path. If several storage paths are available, the tablespace is distributed as equally as possible.

Scheme of container nomination

The creation and nomination of containers follows a defined scheme: *<storage path>/<instance name>/NODE####/<database name>/T#######/C#######.<ending>*. The schemes elements are described here:

- **Storage path**
  Defined during the creation of a database (see the previous command extract below Figure 3.23).
- **Instance name**
  The name of the DB2 for LUW instance to which the database belongs.
- **NODE####**
  The unique database partition number.
- **Database name**
  The name of the database to which this container belongs.
- **T#######**
  The unique tablespace ID within the database.
- **C#######**
  The unique container ID within the tablespace.
- **Ending**
  A code of three letters reflecting the tablespace type:
  - `CAT`: Container of system catalog tablespace.
  - `TMP`: Container of a system temporary tablespace.
  - `UTM`: Container of a tablespace for temporary user tables.
  - `USR`: Container of a regular tablespace.
  - `LRG`: Container of a large tablespace.

In this example, the path to a tablespace container `PR2#DDICD (ID = 24)` would be as follows:

`/db2/PR2/sapdata1/db2pr2/NODE0000/PR2/T0000024/C0000000.LRG`

As described previously, the simplified administration constitutes the major advantage of Automatic Storage. This is most evident when storage space is added to a tablespace.

Note: Storage Paths and Parallelization

Storage paths and parallelization

In a small or mid-sized environment, the database is often installed on one physical LUN of a storage system. Normally this logical unit number (LUN) is presented with a single mount point (or drive) to the operating system.

Does it make sense to put more than one storage path on this single LUN? Yes, absolutely! Initially, Automatic Storage puts one container for a tablespace on every storage path. Therefore, you get a parallelization for the I/O of this tablespace by the number of storage paths (see parameter `DB2_PARALLEL_IO` earlier). This is the reason SAP wants to start with four storage paths during the installation.

(In the following section, we use some examples with DB2 for LUW for SAP with fewer than four storage paths. This is only for a simpler illustration and not a recommendation.)

3.4.3 Tablespace Organization and Extension

Tablespace extension before Automatic Storage

At this point we want to make a little excursion in the DB2 for LUW world before the time of Automatic Storage, just for a better understanding of its advantages. Auto-resizing of tablespaces have existed since DB2 for LUW 8.2.2, before the launch of Automatic Storage. As we already indicated, the options (`AUTORESIZE`, `INCREASESIZE`, `INITIALSIZE`, `MAXSIZE`) determine how, and how far, a tablespace and its containers can grow. It's essential that DB2 for LUW extends only those containers that were used in the last range. Auto-resizing stops after one of the containers of the last range can't be extended, or `MAXSIZE` has been reached. Figure 3.24 shows an example with one tablespace having two containers of a different size.

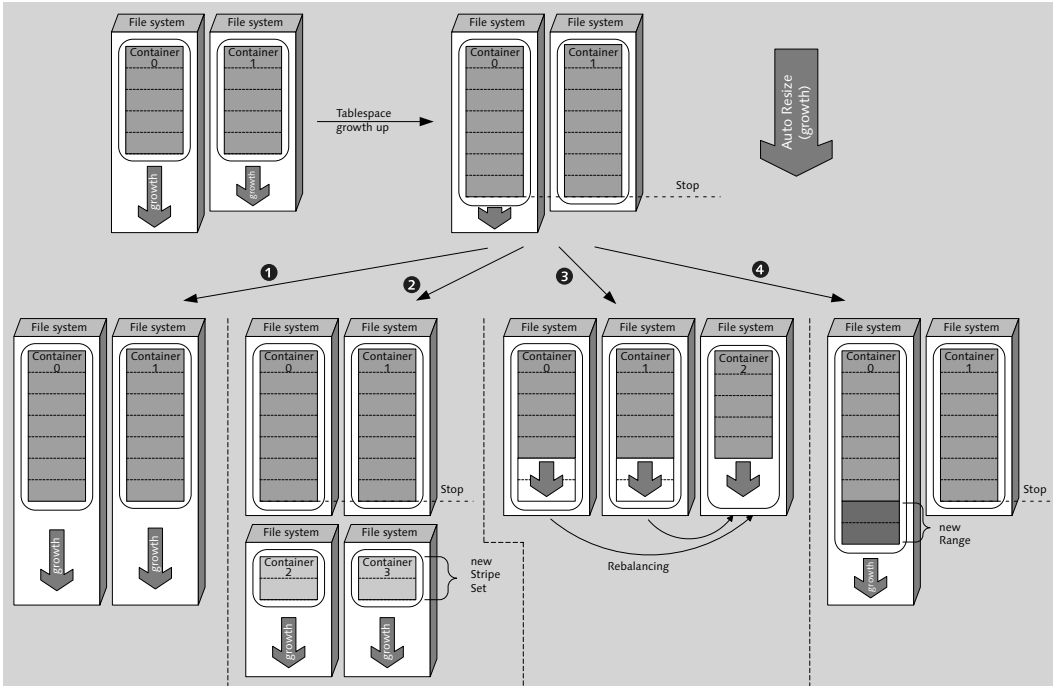


Figure 3.24 Extension of a DMS Tablespace

The illustration shows that administrators have four alternatives for extending tablespaces:

1. By extending the file system of the full container (or both containers) both containers can grow again. This is the easiest solution.
2. The tablespace size increases due to the creation of new containers. The administrator adds containers with the statement `ALTER TABLESPACE <TablespaceName> BEGIN NEW STRIPE SET <Container>` and, thereby, starts a new stripe set in the tablespace. The old containers remain unchanged.
3. The third alternative is to add another container to an existing stripe set of the tablespace, using the statement `ALTER TABLESPACE <TablespaceName> ADD TO STRIPE SET ...`. If the new container is large enough to include all stripe sets, it's positioned in such a way that it starts in the first stripe of the set as well (as indicated in Figure 3.24). If the new container isn't large enough, it's positioned so that

it ends with the last stripe of the stripe set. The result of adding containers to an existing stripe set is a rebalancing process (described in more detail later in this section).

4. The last and not recommendable alternative is to create a new range indirectly by extending only the containers with enough room to grow.

Tablespace extension with Automatic Storage

If a tablespace is managed by Automatic Storage, and the storage capacity for the containers is used up, the administrator only has two alternatives left. As he works on a higher level with storage paths due to Automatic Storage, he has no direct access to containers. Thus, the alternatives 2 through 4 in the preceding list can't be used here, or in other words, these alternatives are automated. Figure 3.25 shows a database with two storage paths.

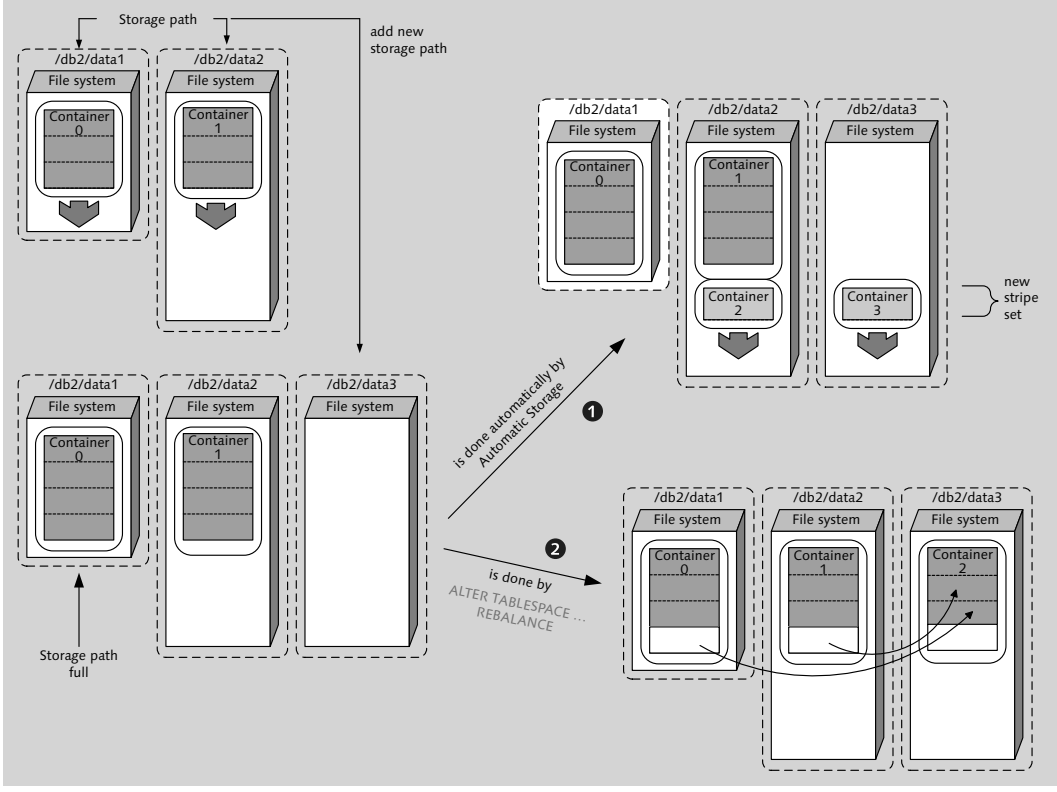


Figure 3.25 New Storage Path in Automatic Storage: Behavior of a DMS Tablespace

The easiest alternative is still to increase the capacity of a storage path. As a result, the Automatic Storage mechanism of the DB2 for LUW database recognizes this, and the containers can grow up. This is the preferred recommendation. If you have any chance to increase the capacity of the storage paths, do it.

However, sometimes you have to add a new storage path to the database. The way Automatic Storage handles and uses this new path depends of your decisions. You have two alternatives:

- You add the storage path and do nothing more. Automatic Storage then acts similar to alternative 2 described earlier. The new storage path isn't used yet, but is only applied after the previous containers are full. If this is the case, a new container is created in the new storage path for each tablespace that must be extended. In doing this, a new stripe set is always created, without any rebalancing operations.
- After adding the new storage path, you execute a rebalancing for each tablespace by use of this statement: `ALTER TABLESPACE <Tablespace-Name> REBALANCE`. As a result, another container is added to the tablespace and the extents are distributed anew. This second option is clearly recommended by IBM.

Parallelism for performance

Warning!

If you add a single storage path, you could lose I/O performance. For example, if you have a DB2 for LUW database with four storage paths and one container per path for tablespace X, then you automatically have a parallelism of 4 (when `DB2_PARALLEL_IO` is not set). If the storage paths are full, and you add a new path, one new container is created on this path. If the old paths are full, the new stripe set is only using the new container, and the parallelism for the tablespace decreases to 1. So, if all four old paths are full, the right decision is to add four new paths to keep the parallelism for the tablespace.

Delete storage paths

Besides adding storages paths, you can also delete paths. For deletion, you first mark a storage path with the statement `ALTER DATABASE DROP STORAGE ON <StoragePath>`. Then, the administrator has to empty the DMS tablespace containers by use of the rebalancing statement. Automatic Storage-administrated SMS tablespaces for temporary tables marked for deletion in the storage path must be deleted. When the storage path is free, Automatic Storage removes it from database configuration. Then, the

SMS tablespaces for temporary tables have to be recreated. If you don't want to delete SMS tablespaces, you have to restart the database because SMS tablespaces are redefined during start after a storage path is deleted.

The rebalancing process for tablespaces of the DB2 for LUW database has already been mentioned several times. Rebalancing reorganizes the extents of a tablespace in the containers. The addition of a container, executed by Automatic Storage or manually, always aims to distribute the extents of a tablespace equally on all its containers—if this is supported by container sizes. The deletion of a container aims to clear this container and to distribute the extents on the remaining containers as equally as possible.

Rebalancing

If you use Automatic Storage, the administrator can start rebalancing manually for some special maintenance tasks. During the normal system operations (and if you have equal sized storage paths), the Automatic Storage functionality acts as if no rebalancing is necessary. With an explicit start of a rebalancing run, it's possible to balance disparities, which may come from missing space or suboptimal tablespace extensions operations. This may appear as a step backward, compared to automatic rebalancing for a DMS tablespace without Automatic Storage, but it isn't. The problem of rebalancing is its massive I/O load. For this reason, it's a major advantage for the administrator to be able to control the start of rebalancing. From DB2 for LUW 10.1 on, you're even able to stop rebalancing operations in times with intensive loading and to restart them later with the command `ALTER TABLESPACE <TablespaceName> REBALANCE SUSPEND | RESUME`.

If a tablespace is in the rebalancing process, you can view this with the statement `db2 get snapshot for tablespaces on <DB Name>` (see the previous section). For more detailed information on the rebalancing status of a tablespace, use the table function `MON_GET_REBALANCE_STATUS`, for example, with the following statement:

```
db2 "select varchar(tbsp_name, 20) as tbsp_name,
rebalancer_mode, rebalancer_status,
rebalancer_extents_remaining,
rebalancer_extents_processed,
rebalancer_start_time  from
table(mon_get_rebalance_status(NULL,-2)) AS T"
```

This statement enables you to see when rebalancing was started, how many extents were already realized, and how many are still open.

Tips & Tricks

Now that you understand the basics of the DB2 for LUW tablespaces, following are some rules of thumb about the administration of these tablespaces.

Rules of thumb for DMS tablespaces only:

- Try to have equal-sized containers in one stripe set of a DMS tablespace. Unequal container sizes may lead to I/O hotspots.
- Manual increasing containers other than those of the highest stripe set poses the risk of rebalancing.
- If you create new containers in an existing stripe set that are smaller than the existing containers, the new containers will be "hanging from the ceiling" instead of "standing on the floor." That is, they use the highest range of this stripe set.
- Multiple stripe sets in a tablespace will be removed after a redirected restore. That is, in a standard DMS tablespace, all previously existing containers (of possibly different sizes) will finally form one stripe set. A database restore without the `REDIRECT` clause keeps the stripe sets.

Rules of thumb for Automatic Storage tablespaces only:

- Use storage paths of equal size to maintain the possibility the containers of a tablespace being of mutual size by use of rebalancing, if necessary.

Rules of thumb for DMS and Automatic Storage tablespaces:

- Auto growth mechanisms (in Automatic Storage or autoresize tablespaces) only affect those containers using the highest range (in the highest stripe set).
- With Automatic Storage or autoresize tablespaces, you should focus your attention on the free space available in the underlying file systems.
- With respect to the performance of the `BACKUP DATABASE` command, try to do the following:
  - Distribute your database's objects among several tablespaces (no single tablespace installation).
  - Have several containers per DMS or Automatic Storage tablespaces instead of one big container.
  - Avoid dominant tablespaces (i.e., disproportionally larger than the average tablespace size).

3.4.4 Storage Groups and Multi-Temperature Storage

Storage groups

With DB2 for LUW 10.1, storage management was enhanced by a new concept: the *storage groups*. A storage group is a group of established storage paths, in which containers from Automatic Storage tablespaces can be stored. The aim of storage groups is to bundle storage paths with the same properties, with regard to their quality. I/O performance is the decisive quality criterion for storage paths used for multi-temperature storage.

Multi-temperature storage

The application of multi-temperature storage, also called multi-tiered storage, relies on prioritizing data. Prioritizing follows these criteria: access frequency, maximum access time, and volatility. According to this approach, data priority is today regarded as a three-stage model:

1. **Hot data**
   This data type has the highest priority because it has the highest access frequency, and its access time must be as short as possible. Hot data generally takes up only a small amount of overall data. This data is stored in very fast and therefore use very expensive data carriers, for example, internal Fusion-io cards or solid-state drive (SSD) arrays.
2. **Warm data**
   This data type has average priority because it's regularly accessed. The access time should be short; therefore, such data is also stored in fast storage systems, for example, flip-chip (FC) arrays.
3. **Cold data**
   This data is rarely accessed, so that longer access times are tolerable. Accordingly, this data has lower priority. Cold data is by far the largest data amount in a system. It can be stored in slower and lower-priced storage systems, for example, hard drive arrays with SAS or serial ATA (SATA) discs.

However, which data has which priority depends on the company and can be very different. Therefore, the installation of multi-storage should be preceded by a thorough analysis.

The new storage groups in DB2 for LUW 10.1 support data distribution on multi-temperature storage systems, if necessary, in combination with partitioned tables. Figure 3.26 shows an example.

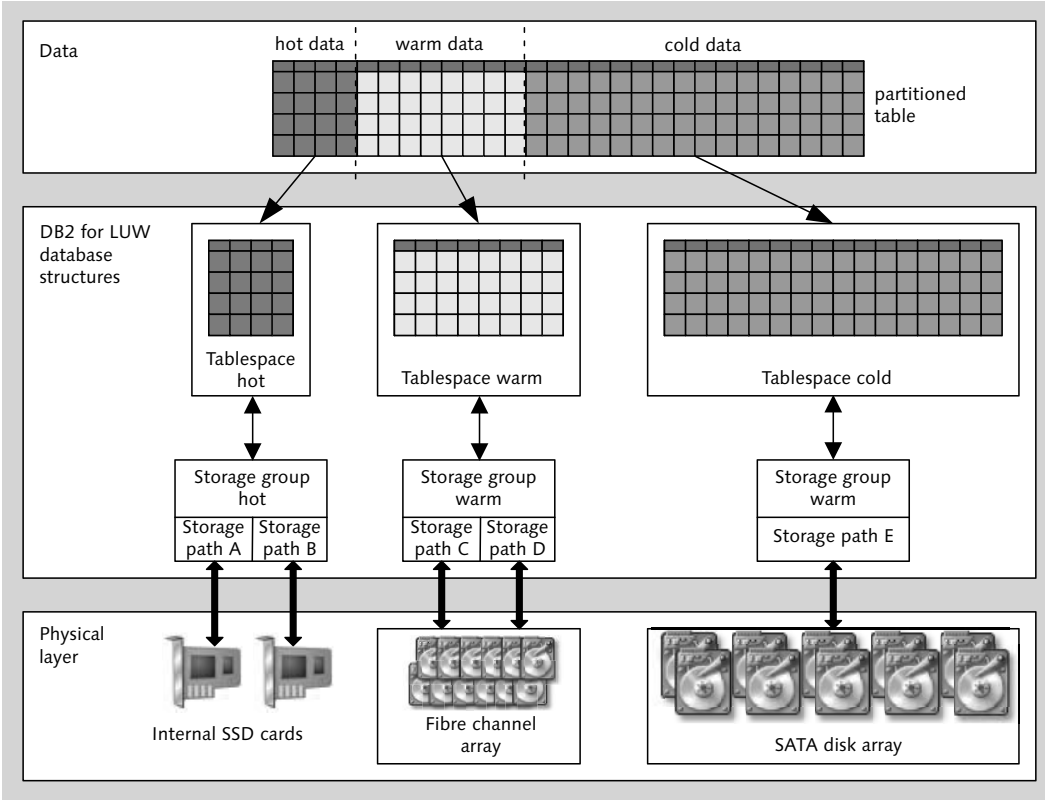


Figure 3.26 Multi-Temperature Storage with Storage Groups

Starting from DB2 for LUW 10.1, the concept of storage groups is applied after you activate Automatic Storage during database creation. Following this, the standard storage group `IBMSTOGROUP` is created automatically, and all storage paths are initially grouped here. Other storage groups can be created with the statement `CREATE STOGROUP`. The statement `ALTER STOGROUP` enables you to manage all storage groups.

Information about storage groups

The two main alternatives for accessing information on storage groups of a database are the following:

- To find information on storage groups, for example, storage group ID, read transfer rate, or controller overhead, query `SYSCAT.STOGROUPS` catalog views, with the statement `db2 "select * from SYSCAT.STOGROUPS"`.

- To access information on the storage paths of a storage group, its usage status, and capacities, query the table function `ADMIN_GET_STORAGE_PATHS`, for example, with the statement:

```
db2 "SELECT VARCHAR(STORAGE_GROUP_NAME, 15) AS STOGROUP,
     VARCHAR(DB_STORAGE_PATH, 20) AS STORAGE_PATH,
     VARCHAR(DB_STORAGE_PATH_STATE, 12) AS PATH_STATE,
     FS_TOTAL_SIZE, FS_USED_SIZE, STO_PATH_FREE_SIZE FROM
     TABLE(ADMIN_GET_STORAGE_PATHS('',-1)) AS T"
```

Figure 3.27 shows the output of this statement.

```
vm23:db2pr2 74> db2 "SELECT VARCHAR(STORAGE_GROUP_NAME, 15) AS STOGROUP, VARCHAR(DB_STORAGE_PATH, 20) AS STORAGE_PATH,
 VARCHAR(DB_STORAGE_PATH_STATE, 12) AS PATH_STATE, FS_TOTAL_SIZE, FS_USED_SIZE, STO_PATH_FREE_SIZE FROM TABLE(ADMIN_GE
T_STORAGE_PATHS('',-1)) AS T"

STOGROUP        STORAGE_PATH         PATH_STATE   FS_TOTAL_SIZE       FS_USED_SIZE        STO_PATH_FREE_SIZE
--------------- -------------------- ------------ ------------------- ------------------- -------------------
IBMSTOGROUP     /db2/PR2/sapdata1    IN_USE              97052438528          44098441216          52886888448
IBMSTOGROUP     /db2/PR2/sapdata2    NOT_IN_USE           4226125824            290234368           3935891456

  2 record(s) selected.
```

Figure 3.27 Information on a Storage Group

If you use DB2 for LUW database 10.1 in an SAP system, you can use the features of storage groups and multi-temperature storage. For analyzing and partitioning of SAP tables, you can use the IBM ABAP tool DB6 Partitioning Administrator (see SAP Note 1686102).

3.4.5 Reclaimable Storage

Release of unused storage

Starting from DB2 for LUW 9.7, an additional tablespace attribute is available: the online release of allocated but unused storage space to the file system. To offer this option, the Row-ID (RID) and the physical address of a data page were decoupled in the internal storage management of DMS tablespaces. This also enables the administrator to release the unused areas of a tablespace, lying below the high water mark (HWM) of the tablespace.

Figure 3.28 illustrates this procedure, followed by a description.

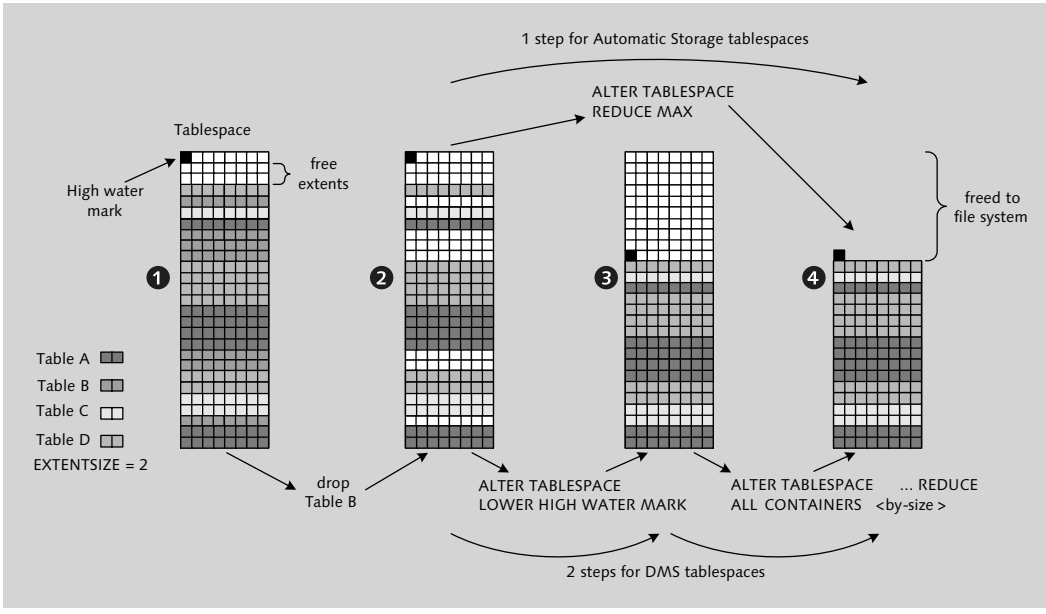


Figure 3.28 Reclaim Storage

High water mark (HWM)

1. The initial situation is a tablespace with four objects. The high water mark (HWM) defines the highest page ever used in the tablespace. In this example, there is a free area of extents lying directly below the HWM, which has already been released through a deletion process. Before DB2 for LUW 9.7, it was possible to rerelease this free area below HWM, without having to perform extensive reorganization work.
2. If a table is deleted, different extents become free within the tablespace. Here, the reclaim storage feature comes into play.

Differences between DMS and Automatic Storage

3. If you use a DMS tablespace without Automatic Storage, you first have to lower the HWM with the statement `ALTER TABLESPACE <TBSP_Name> LOWER HIGH WATER MARK`. This statement moves extents to close the gaps and to shift the HWM on the lowest possible position (HWM = used pages). Extent movement processes can take a long time, depending on the size, the number of gaps, and the load. Also, this process is very I/O intensive; therefore, it can be suspended with the command `ALTER TABLESPACE <TBSP_Name> LOWER HIGH WATER MARK STOP`. The

following statement enables you to monitor extent movement processes:

```
db2 "SELECT varchar(tbsp_name,15) as tbsp_name,
     last_extent, num_extents_moved, num_extents_left,
     total_move_time from table
     (mon_get_extent_movement_status('<TBSP_NAME>',-1))
     AS T"
```

Figure 3.29 show an example output of this statement.

```
vm32:db2t32 129> db2 "SELECT varchar(tbsp_name,15) as tbsp_name, last_extent, num_extents_
moved, num_extents_left, total_move_time from table (mon_get_extent_movement_status('T32#F
ACTD',-1)) AS T"

TBSP_NAME       LAST_EXTENT NUM_EXTENTS_MOVED NUM_EXTENTS_LEFT TOTAL_MOVE_TIME
--------------- ----------- ----------------- ---------------- --------------------
T32#FACTD              2800               438             1052                 8872

  1 record(s) selected.
```

Figure 3.29 Monitoring Extent Movements

- `LAST_EXTENT`: Last extent moved during this operation.
- `NUM_EXTENTS_MOVED`: Number of extents moved during these operations until now.
- `NUM_EXTENTS_LEFT`: Number of extents left to be moved during this operation.
- `TOTAL_MOVE_TIME`: Previous runtime of the operation in milliseconds (msec).

4. Finally, the extents released from above HWM are returned to the file system by downsizing the corresponding containers. Use the following statement to trigger this process: `ALTER TABLESPACE <TBSP_Name> REDUCE ...`. In case of an Automatic Storage tablespace, it's sufficient to use this statement with the option MAX; the preceding step (lower HWM) is executed automatically. A DMS tablespace requires both statements. The statement `ALTER TABLESPACE <TBSP_Name> REDUCE …` has the following options:
   - `MAX`: Removes all free extents and provides the file system with maximum storage capacity (works only with Automatic Storage tablespaces).

- `<empty>`: When no option is indicated, all containers are downsized by the free extents from above HWM and those already allocated by the file system. In the initial situation from Figure 3.28, this would only apply to the three rows above HWM. The extents aren't moved!
- `<number> K|M|G or %`: Reduces container size by the indicated size in kilobyte (`K`), megabyte (`M`), gigabyte (`G`), or by the percentage rate.
- `STOP`: Stops a running `REDUCE` operation.

Analyze Tablespace Storage

To determine which tablespaces have large gaps and are therefore suitable for the reclaim operation, you can consult the different values for the tablespace pages, as well as the HWM. The following statement enables you to get the current values for the total number of usable pages (`TBSP_USABLE_PAGES`), the number of free pages (`TBSP_FREE_PAGES`), and the HWM (`TBSP_PAGE_TOP`):

```
db2 "select substr(TBSP_NAME,1,14) as TS_NAME,
     TBSP_USABLE_PAGES as Usable_pages,
     TBSP_PAGE_TOP as High_water_MARK,
     TBSP_FREE_PAGES as Free_pages
     from table ( MON_GET_TABLESPACE ( NULL , -1 ) ) as ts
     where reclaimable_space_enabled = '1'"
```

The output of this statement only shows those tablespaces supporting reclaimable storage; thus, SMS tablespaces and tablespaces older than DB2 for LUW 9.7 aren't shown. Figure 3.30 shows you an extract of the statement output for a DB2 for LUW with an SAP system.

Then, the aim is to determine which free pages are lying above and which are below the HWM:

- `free pages above HWM ≈ Usable_pages – High_water_MARK` This free area above HWM can be returned to the file system without extent movements.
- `free pages below HWM = Free_pages – (Usable_pages – High_water_MARK)`

This number, being certainly of greater interest, indicates the pages released from below HWM during operation. It quantifies the gaps in the tablespace. This free storage capacity can only be eliminated and returned to the file system by extent movements.

```
vm32:db2t32 72> db2 "select substr(TBSP_NAME,1,14) as TS_NAME, TBSP_USABLE_PAGES as Usable
_pages, TBSP_PAGE_TOP as High_water_MARK, TBSP_FREE_PAGES as Free_pages from table ( MON_G
ET_TABLESPACE ( NULL , -1 ) ) as ts where reclaimable_space_enabled = '1'"

TS_NAME        USABLE_PAGES         HIGH_WATER_MARK      FREE_PAGES
-------------- -------------------- -------------------- --------------------
SYSCATSPACE                  104446               104376                   70
SAPTOOLS                      81552                78842                 2710
SAPEVENTMON                    3198                  664                 2534
SYSTOOLSPACE                   2046                 1760                  286
T32#DDICD                    139262               139246                   16
T32#DDICI                     32766                30834                 1932
T32#USER1D                     2046                  528                 1518
T32#USER1I                     2046                  534                 1512
T32#POOLD                    462846               460972                 1878
T32#POOLI                    219134               217626                 1518
T32#PROTD                     30718                30646                   72
T32#FACTD                      8150                 6476                 3164
T32#FACTI                      2046                  516                 1530
  34 record(s) selected.
vm32:db2t32 72>
```

Figure 3.30 Analysis of Tablespaces on Free Pages

After determining the free pages of a tablespace, you can decide if it's reasonable to release storage capacity to the file system. According to this formula, the tablespace T32#FACTD from Figure 3.30 has approximately 1,490 free pages below HWM and approximately 1,674 pages above HWM. Thus, a reclaim operation with the `REDUCE MAX` option would lead to 3,164 pages * 16KB (PAGESIZE) ≈ 50MB to return to the file system.

Note

The alternatives for reclaim storage explained here can only be applied with DMS or Automatic Storage tablespaces that were created with version 9.7 or newer DB2 for LUW versions. Tablespaces that were created with older versions can't use these functions, even after a database upgrade. A migration from old to new tablespaces isn't possible. To use reclaim storage, you have to delete tablespaces and create new ones.

In an SAP environment, you can use the DB6-ABAP Tool DB6CONV to move the complete data from one (old) to another (new) tablespace.

3.4.6 Table Compression

Effects

The IBM DB2 database software (and its different derivatives) has always been a pioneer in the different approaches for data compression in a database. The aim always was and still is to save hardware. A positive side effect of compression is performance enhancement in most cases. Due to compression, writing and reading operations require less I/O accesses. In sum, these savings clearly outbalance the slightly higher CPU usage.

At this point, describing all compression alternatives that the DB2 for LUW database supports is beyond the scope of this book. Nevertheless, we'll briefly present the different DB2 for LUW compression alternatives and outline the key points for administrators.

Alternative Row Format

Value compression

The value compression option already exists since DB2 for LUW 8.1. DB2 for LUW offers the possibility to save table rows in standard or alternative format. The storage format determines how the table row is stored in a page. The alternative storage of rows enables you to store 0 values and data types with variable length, such as `VARCHAR`, more efficiently; this explains the term "value compression." In this way, all 0 values of a table with alternative row format don't occupy storage pages, for instance.

Value compression is activated during table creation with the option `CREATE TABLE … VALUE COMPRESSION`. However, it's also possible to activate or deactivate value compression subsequently by use of `ALTER TABLE … ACTIVATE VALUE COMPRESSION` or `DEACTIVATE VALUE COMPRESSION`. If you use DB2 for LUW for SAP systems, value compression is automatically set by default. Thus, nearly all SAP tables in a DB2 for LUW database work with the alternative row format.

Row Compression

Classical row compression

As the first compression type, the classical row compression was launched with DB2 for LUW 9. Row compression uses a compression dictionary to save recurring patterns in data rows separately and only once. Figure 3.31 illustrates this approach.

Figure 3.31 Classical Row Compression

The compression dictionary is responsible for the whole table, and is therefore also referred to as *table-level compression dictionary*. Since DB2 for LUW 9.7, compression captures all data of a table, except LOB data. In a DB2 for LUW database, the classical row compression of tables is activated by `CREATE TABLE` with the option `COMPRESS YES (STATIC)`. The element `STATIC` must only be used for DB2 for LUW 10.1; otherwise, adaptive compression is activated (discussed in more detail later in this section).

Row compression can be activated subsequently for existing tables, with the statement `ALTER TABLE … COMPRESS YES (STATIC)`. All previous data of the table remains uncompressed, until a reorganization or table movement in another tablespace is executed.

Automatic Dictionary Creation

Before DB2 for LUW 9.5, row compression led to considerable administrative efforts because the compression dictionary had to be created manually with the command `REORG TABLE <Table_Name> RESETDICTIONARY`. Naturally, this was only reasonable when tables reached a least minimum quantity of data records. With version 9.5, Automatic Dictionary Creation (ADC) was introduced. This feature ensures that the compression dictionary is created automatically when compression is activated for a table and when a basic quantity of data is loaded into the table using `INSERT` or `LOAD` operations. Usually, ADC starts when a table has reached a size of

1–2MB. All data loaded before ADC remains uncompressed until a full reorganization or data row change is executed.

Another issue you have to keep in mind as administrator is the degeneration of compression due to high alternation rates in the table or rapid table growth. Because the compression dictionary is created only once, dictionary contents don't reflect the actual table contents in an optimal way; that is, there are certain patterns missing and the compression rate declines. The larger the alternation rate of the table with different data records, the faster the compression rate declines. Under some circumstances, it may be favorable to completely abandon compression of tables with very high alternation rates.

Adaptive Compression

Adaptive compression through page-level compression

With DB2 for LUW 10.1, the classical row compression was extended and improved. The enhanced row compression is called adaptive compression. As enhancement to the classical approach, a second, page-based dictionary level was added. For this reason, it's also referred to as *page-level compression dictionary*. Thus, a table contains one dictionary for the whole table and another dictionary for each data page that is nearly 100% in use. Figure 3.32 shows an example.

Activate adaptive compression

Equivalent to classical table creation, adaptive compression is activated with the command `CREATE TABLE` with the option `COMPRESS YES`. For DB2 for LUW 10.1, another option is required because adaptive compression is used by default. In an SAP environment, however, the registry parameter `DB2_ROWCOMPMODE_DEFAULT=STATIC [DB2_WORKLOAD]` sets the default compression to the classical row compression (because the adaptive compression isn't compatible with the statement `REORG ... INPLACE`).

If the database isn't upgraded to version 10.1, the adaptive compression can also be activated subsequently for existing tables, with the statement `ALTER TABLE … COMPRESS YES ADAPTIVE`. Here too, all previous data of the table remains uncompressed, until a reorganization or table movement in another tablespace is executed. This means that no "old" data comes into the *table-level dictionary*, but the pages, which satisfy a filling level of nearly 100%, will be completely included (with the old data) into the newly created *page-level dictionary*.

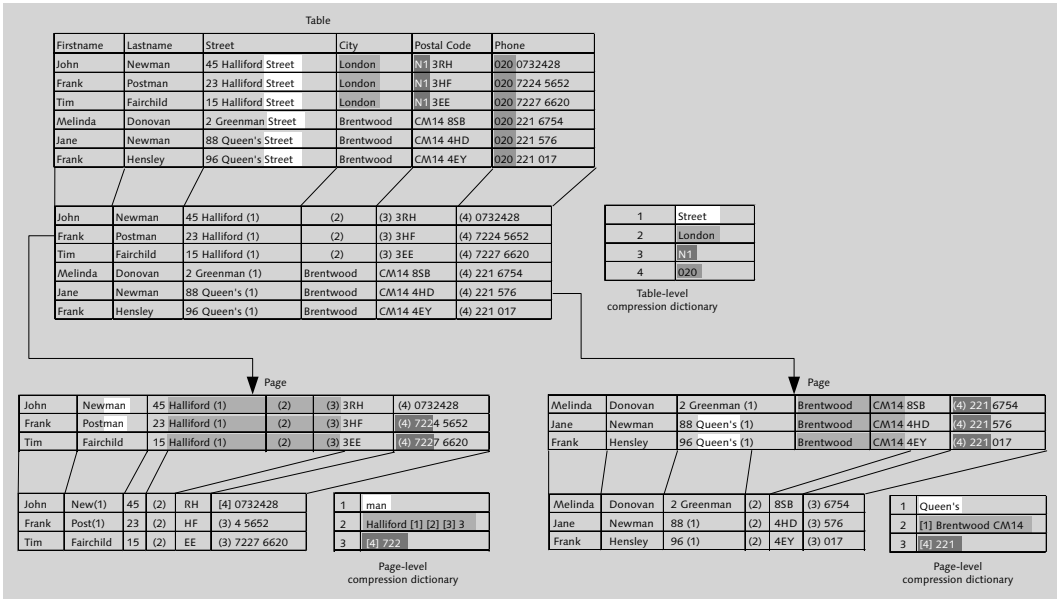


Figure 3.32 Adaptive Compression

The reorganization is executed with the same commands as the classical row compression.

Information about compression of a table

If you want to identify the compression a table uses, you can apply the following small SQL query:

```
db2 "SELECT SUBSTR(TABSCHEMA, 1, 10) AS TABSCHEMA,
     SUBSTR(TABNAME, 1, 20) AS TABNAME,
     COMPRESSION, ROWCOMPMODE
     FROM SYSCAT.TABLES WHERE TABNAME='<TableName>'
     AND TYPE='T'"
```

Figure 3.33 shows query extracts over all tables of an SAP system.

The table column COMPRESSION can have the following values:

- **V (value)**
  The table uses the alternative row format but no row compression.
- **R (row)**
  The table uses the standard row format and row compression (classical or adaptive).

- **B (both)**
  The table uses the alternative row format and row compression (classical or adaptive).
- **N (none)**
  The table uses the standard row format but no row compression.

```
vm23:db2pr2 148> db2 "SELECT SUBSTR(TABSCHEMA, 1, 10) AS TABSCHEMA, S
UBSTR(TABNAME, 1, 20) AS TABNAME, COMPRESSION, ROWCOMPMODE FROM SYSCA
T.TABLES WHERE TYPE NOT LIKE 'V'"

TABSCHEMA  TABNAME              COMPRESSION ROWCOMPMODE
---------- -------------------- ----------- -----------
SAPPR2     TGUI                 B           S
SAPPR2     THEXS                B           S
SAPPR2     TSINH                B           A
SAPPR2     TSINI                B           A
SAPPR2     VSEOCLIF             N
SAPPR2     VSEOCLSDF            N
SAPPR2     USREXTIDT            R           S
SAPPR2     USVAR                R           S
SAPTOOLS   DBH_STG_ME           V
SAPTOOLS   DBH_STG_CF           V
```

Figure 3.33 Table Attributes for Compression

The column ROWCOMPMODE indicates the type of row compression—A (adaptive) stands for adaptive compression, and S (static) for classical row compression.

Note

For an integrated DB2 for LUW and SAP installation, classical row compression is set automatically for all tables for any actual version of the DB2 for LUW database. The classical row compression is integrated in the SAP tool R3load.

There are different ways to analyze which table is suitable for compression by use of SQL queries and the administrative function `ADMIN_GET_TAB_INFO`. The function `ADMIN_GET_TAB_COMPRESS_INFO` enables you to analyze how much storage capacity the compression of a table saves. As a DB2 for LUW administrator with SAP access, there are many comfortable options based on the DBA Cockpit (see Chapter 5, Section 5.2.3).

Index Compression

Index compression is the second important compression alternative after the row compression. This alternative was launched with DB2 for LUW 9.7 and operates with three different compression alternatives:

Working of index compression

- **Variable slot directory**
  Each index page has a slot directory, in which each slot has a byte offset for each index key. During this process, the slot size isn't statically defined but is variable for each entry size.
- **RID list compression**
  Each index always contains a key value and a key data field in the form of one or more Record-IDs (RIDs). In this process, the RID—8 bytes long by default—is saved in a compressed form. To be more precise, only the delta of the previous RID is stored for a RID.
- **Prefix compression**
  Basically, this is the counterpart to RID list compression because the key value of an index entry is compressed here. This process checks the key values for common prefixes to remove their redundancies.

The database automatically determines which compression process to use, so it's possible that several processes will be applied in parallel. Index compression runs in unique as well as in secondary indexes. Block indexes and XML path indexes can't be compressed.

Index compression essentially operates similar to ADC; that is, compression is regularly ensured by the database, and manual administration isn't necessary.

Row compression leads to index compression

If one of the two row compressions is activated for a table, all the indexes of the table are compressed automatically as well. If compression was already activated during table creation, the administrator has no other steps to take—compression is executed automatically. However, if compression is activated later on with the statement `db2 ALTER INDEX <IndexName> COMPRESS YES` or indirectly by activating the compression for the table, the index must be reorganized so that already existing data is compressed. For this purpose, the following command has to be used for all indexes: `db2 REORG INDEXES ALL FOR TABLE <TableName>`.

The following short statement indicates whether an index is compressed:

```
db2 "SELECT CHAR(INDNAME,20) AS INDEX,
      COMPRESSION FROM SYSCAT.INDEXES
      WHERE TABNAME='<TableName>'"
```

There is also an administrative function for indexes called `ADMIN_GET_INDEX_COMPRESS_INFO`. It enables you to access information on index compression, for example, the percentage rate of storage capacity savings.

Manage compression with the DBA Cockpit

You can use an IBM management tool for index and table compression management, which is particularly suitable for SAP. It's very useful when you can't use an actual version of the DBA Cockpit. By use of SAP Note 980067, you can load this tool from the SAP Service Marketplace and install it as an ABAP program in the system. The tool offers you a GUI for managing table and index compression in your DB2 for LUW database. Figure 3.34 shows the entry into the compression tool.

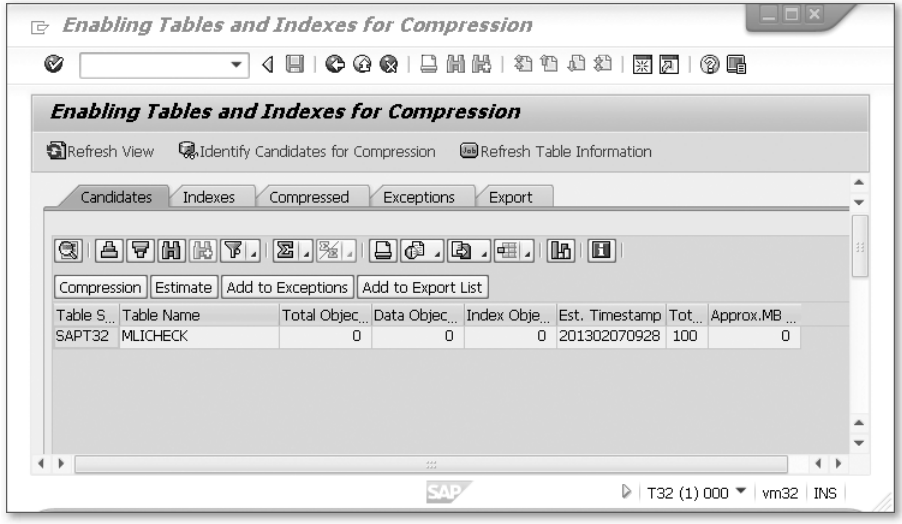


Figure 3.34 Compression Tool for DB2 for LUW on SAP

Other Compressions in DB2 for LUW

More compression possibilities

There are other compression alternatives in the DB2 for LUW database, which we'll discuss in this subsection for the sake of completeness.

The backup compression alternative was launched with DB2 for LUW 8. By use of the COMPRESS option, it's possible to define for each backup operation, except a file-level backup (with BACKUP DATABASE ... USE SNAPSHOT), that the created database image becomes compressed. The backup compression is successful, even if row and index compression is already used. Backup compression also includes LOB data.

Starting with DB2 for LUW 10.1 onward, it's also possible to compress archive log files. In parallel with other compressions, this compression type enables you to save more storage capacity. The activation of log archive compression is executed by the database parameters LOGARCHCOMPR1 and LOGARCHCOMPR2.

Another possibility to save storage capacity is the LOB inlining alternative. Usually, the storage of LOB objects (data types BLOB, CLOB, and DBCLOB) only includes the storage of descriptors in a table referring to the actual storage location. The descriptor size increases with the LOB size and accordingly uses storage capacity. For very small LOB objects, it's therefore more effective to save them directly in the table together with regular data. This direct storage process can be used for inline LOBs from DB2 for LUW 9.7 onward; in the SAP surrounding, it's also set by default.

The compression of temporary objects constitutes another compression type that can be used from DB2 for LUW 9.7. This compression type is activated automatically, if the DB2 Storage Optimization license is installed, and can't be deactivated. It operates like the classical row compression, inclusively ADC. The essential differences are that the ADC only becomes active from a table size of up to 100MB and sets up a compression dictionary.

All compression alternatives named in this section are supported in the SAP environment and are therefore usable with a corresponding DB2 for LUW release.

Note

Some of the compression features described in Section 3.4.4 require a separate license for the DB2 Storage Optimization feature. However, this license is part of the SAP OEM license for DB2. For this reason, as the SAP client, you can use all compression alternatives of DB2 for LUW.

The DBA Cockpit is the major tool for monitoring and maintaining the DB2 database in your SAP system. In addition to describing the DBA Cockpit, we'll explain some of the tools at the operating system level you can use without running the SAP system.

5 Administration Tools Inside and Outside the SAP System

The SAP strategy is to shift more and more functionality from operating system commands to an easy-to-understand and easy-to-use web frontend, the DBA Cockpit. The administrator can easily make changes to improve the system performance from within the same screen. All database key performance indicators (KPIs) can be accessed easily. Therefore, the biggest part of this chapter is describing all of the functions available in the DBA Cockpit.

Nevertheless, you'll still need to use some of the toolset at the operating system level in your daily work when administrating a DB2 database. We'll describe these tools first, then get into the details of the DBA Cockpit, and finally put a special focus on troubleshooting.

5.1 Operating System Tools

Not all functions can be performed in the DBA Cockpit; a lot of tasks, especially maintaining, starting, and stopping the database, have to be performed at the operating system level. In this section, we'll explain how to perform these actions, even when the SAP system is offline.

5.1.1 Starting the Database

Starting and stopping your DB2 database can be accomplished in many ways. SAP offers the Landscape Virtualization Manager (LVM) to centrally start, stop, or relocate your systems. For more information about the SAP LVM, refer to *http://help.sap.com/nwlvm*.

startsap script

SAP provides the operating system script `startsap` to start the SAP server and the DB2 database. With these parameters, `startsap` can be used for the tasks shown in Table 5.1.

Starting a database

ParameterWild Card	Description of the Task
`db`	Start ABAP database
`jdb`	Start Java database
`all`	Start database and SAP instance
`r3/j2ee`	Start SAP instance only
`check`	Check the status of database and SAP system instances
`startupsrv`	Start `sapstartsrv`

Table 5.1 Parameter startsap

If you use the `startsap` script with the parameter `db` to start your database, the script `startdb` is run. `startdb` issues the DB2 command `db2start` to start the database instance first and activates the database afterward (see Figure 5.1).

```
vml3:prdadm 52> startsap db
Checking db6 Database
Database is not available via R3trans
-------------------------------------------
01/31/2013 14:03:16     0   0   SQL1063N  DB2START processing was successful.
SQL1063N  DB2START processing was successful.
Database activated
```

Figure 5.1 startsap db

db2start command

If you need to use the DB2 commands and can't use the SAP script, start and activate the database with the DB2 commands `db2start` and `db2 activate database`.

db2start can be executed as a command line processor (CLP) or as a system command as shown in Figure 5.2.

```
vml3:prdadm 62> db2start
01/31/2013 12:36:54     0   0   SQL1063N  DB2START processing was successful.
SQL1063N  DB2START processing was successful.
```

Figure 5.2 db2start

If the database is stopped, you can start the DB2 instance layer with db2start. The per-instance Engine Dispatchable Units (EDUs) such as db2wdog, db2sysc, db2tcpcm, and db2ipccm are started.

The command ACTIVATE DATABASE allows you to explicitly activate a selected database (see Figure 5.3). **Activate database**

```
vml3:prdadm 63> db2 activate db PRD
DB20000I  The ACTIVATE DATABASE command completed successfully.
```

Figure 5.3 db2 ACTIVATE DATABASE Command

After the database layer is started, the per-database EDUs like db2loggr, db2loggw, db2dlock, db2pfchr, and db2pclnr are started. The EDUs are explained in detail in Chapter 3, Section 3.2.1.

Another option to start a database is to establish a connection to the database with the command as shown in Figure 5.4.

```
vml3:prdadm 66> db2 connect to PRD

   Database Connection Information

 Database server        = DB2/LINUXX8664 10.1.0
 SQL authorization ID   = PRDADM
 Local database alias   = PRD
```

Figure 5.4 db2 connect to DBSID

5.1.2 Stopping the Database

Similar to the script startsap, SAP provides the script stopsap to stop your database and/or your system. Depending on the parameters used, you can stop your ABAP database, a Java database, an SAP instance, sapstartsrv, or both database and SAP instance.

stopsap: stopping the database

As you can see in Figure 5.5, the script `stopsap` checks to see if the database is running. If so, the stop procedure is started. The first step is the deactivation of the database. After deactivation has been successfully completed, the `db2stop` command is issued to stop the database instance.

```
vml3:prdadm 52> stopsap db
Checking db6 Database
Database is running
-------------------------------------------
Database is running
Continue with stop procedure
DB20000I  The DEACTIVATE DATABASE command completed successfully.
01/31/2013 12:18:55     0   0   SQL1064N  DB2STOP processing was successful.
SQL1064N  DB2STOP processing was successful.
Database sucessfully stopped
Checking db6 Database
Database is not available via R3trans
-------------------------------------------
```

Figure 5.5 stopsap

Deactivate database

The command `db2 deactivate database` deactivates the specified database (see Figure 5.6). Unsaved buffer pool content is copied back to DB2 containers, all files are closed, and all necessary database services are stopped.

```
vml3:prdadm 64> db2 deactivate db PRD
DB20000I  The DEACTIVATE DATABASE command completed successfully.
```

Figure 5.6 db2 Deactivate Database

Terminate database

The `terminate` command performs an internal commit and causes the database connection to be lost (see Figure 5.7). All other existing DB connections, such as running SAP work processes, are not affected. The database would shut down if you terminated the last remaining database connect and the database was not explicitly activated using command `ACTIVATE DB`.

```
vml3:prdadm 70> db2 terminate
DB20000I  The TERMINATE command completed successfully.
```

Figure 5.7 db2 terminate

db2stop

The command `db2stop` can be issued, as shown in Figure 5.8, if the database isn't active anymore. It stops the database instance including the per-instance EDUs.

```
vml3:prdadm 62> db2stop
01/31/2013 16:33:58     0   0   SQL1064N  DB2STOP processing was successful.
SQL1064N  DB2STOP processing was successful.
```

Figure 5.8 db2stop

5.1.3 DB2 Command Line Processor

Because the DBA Cockpit is integrated into the SAP system, it's only available if the SAP system is online. When the SAP system is offline, you can still use the command line processor (CLP) to issue SQL statements or DB2 commands, and call stored procedures. The CLP is essentially a command processor or shell environment that is customized for working with DB2.

You can use the DB2 CLP as your primary interface to interact with your DB2 instances and databases. It can be an alternative to using GUIs such as the DBA Cockpit in the SAP world. For the most part, the interface is only used occasionally when the GUI isn't available. The DB2 CLP provides maximal support for working with DB2 instances. It's a good choice for database administrators and application developers who prefer a more traditional command interface.

The CLP is mostly used to do the following:

- Stop a DB2 instance.
- Start a DB2 instance.
- Issue DB2 commands for configuring a DB2 instance or database.
- Establish a database connection.
- Execute SQL statements.
- Run DB2 tools and utilities.

You use the `db2` command to start the CLP. As we mentioned, the CLP is used to execute database utilities, SQL statements, and online help. It offers a variety of command options. You can use the CLP in the following modes:

Modes

- **Interactive input mode**
  The interactive input mode is characterized by the `db2 =>` [input prompt] (see Figure 5.9).

```
(c) Copyright IBM Corporation 1993,2007
Command Line Processor for DB2 Client 10.1.0

You can issue database manager commands and SQL statements from the command
prompt. For example:
    db2 => connect to sample
    db2 => bind sample.bnd

For general help, type: ?.
For command help, type: ? command, where command can be
the first few keywords of a database manager command. For example:
 ? CATALOG DATABASE for help on the CATALOG DATABASE command
 ? CATALOG          for help on all of the CATALOG commands.

To exit db2 interactive mode, type QUIT at the command prompt. Outside
interactive mode, all commands must be prefixed with 'db2'.
To list the current command option settings, type LIST COMMAND OPTIONS.

For more detailed help, refer to the Online Reference Manual.

db2 =>
```

Figure 5.9 CLP Interactive Input Mode

When you start the DB2 CLP in interactive input mode, you can enter your `db2` commands at the `db2` prompt. In the interactive input mode, the db2 shell or command processor is specially configured and initialized for processing the DB2 commands and SQL statements, as well as for returning the output of the processing.

- **Command mode**
  You can use this mode by simply using the prefix `db2` when executing DB2 commands or SQL statements (see Figure 5.10).

```
vml3:db2prd 8> db2 get dbm cfg | grep DISPATCHER
 WLM dispatcher enabled                  (WLM_DISPATCHER) = NO
```

Figure 5.10 CLP Command Mode

When using the CLP command mode, the shell or command processor environment is maintained. This indicates to the operating system that the command that follows has to be handled by the DB2 CLP. The processor or shell environment isn't initialized; therefore, operating system commands can be still issued as before.

- **Batch mode**
  The batch mode uses the `-f` file input option and the `-z` file output option (see Figure 5.11).

```
vm13:db2prd 9> db2 -z output.out -f MyScript.sql
```

Figure 5.11 CLP Batch Mode

The batch mode enables you to specify a file with a set of DB2 commands or SQL statements. These commands are processed by the CLP. The batch mode is an extension of the command mode.

Using the shell command `!` allows you to execute operating system commands from the interactive mode or the batch mode. For example, on UNIX-based systems, you can run `!ls -ltr` and on Windows operating systems `!dir`.

CLP syntax

CLP Syntax
For commands:
`db2 -[option-flag], [db2-command` or `sql-statement]`
To request help:
db2 ? [phrase, message, sqlstate or classcode]

You can use the CLP help commands to learn about the available options or to gain information from the CLP (see Table 5.2). Note that a blank space must separate the question mark from the variable name.

Statement	Description
`?`	View CLP general help.
`? phrase`	View the help text associated with a specific command or topic. If there is no information for your requested information, the general help screen is displayed. `? help`, for example, requests information on how to use and read the help screens.
`? sqlstate`	You can request help for a message specified by a valid `SQLSTATE`.
`? message`	You can request information for a message specified by a valid `SQLCODE`.
`? class-code`	You can request information for a message specified by a valid class code.

Table 5.2 CLP Help Options

When you want to stop the CLP, use the command `QUIT` or `TERMINATE`. `QUIT` just stops the CLP. `TERMINATE` stops the CLP, removes any associated backend process and frees memory that is being used. Before stopping the database manager (`db2stop`), it's recommended that you issue `TERMINATE`. If you change database configuration parameters, it might be ostensible to issue `TERMINATE` so that these changes take effect.

CLP options

The CLP options you see in figure are the default settings. They can be changed by setting the DB2OPTIONS environment variable or by using command line flags. Under Linux you can use the command `setenv DB2OPTIONS -m` to turn on that the number of rows affected is displayed.

```
vm13:db2prd 54> db2 list command options

     Command Line Processor Option Settings

 Backend process wait time (seconds)        (DB2BQTIME) = 1
 No. of retries to connect to backend        (DB2BQTRY) = 60
 Request queue wait time (seconds)          (DB2RQTIME) = 5
 Input queue wait time (seconds)            (DB2IQTIME) = 5
 Command options                           (DB2OPTIONS) =

 Option  Description                                 Current Setting
 ------  ------------------------------------------  ---------------
   -a    Display SQLCA                               OFF
   -c    Auto-Commit                                 ON
   -d    Retrieve and display XML declarations       OFF
   -e    Display SQLCODE/SQLSTATE                    OFF
   -f    Read from input file                        OFF
   -i    Display XML data with indentation           OFF
   -l    Log commands in history file                OFF
   -m    Display the number of rows affected         OFF
   -n    Remove new line character                   OFF
   -o    Display output                              ON
   -p    Display interactive input prompt            ON
   -q    Preserve whitespaces & linefeeds            OFF
   -r    Save output to report file                  OFF
   -s    Stop execution on command error             OFF
   -t    Set statement termination character         OFF
   -v    Echo current command                        OFF
   -w    Display FETCH/SELECT warning messages       ON
   -x    Suppress printing of column headings        OFF
   -z    Save all output to output file              OFF
```

Figure 5.12 db2 list command options

Setting the DB2OPTIONS environment variable overrides the default settings. Using a flag in the command line operation overrides the DB2OPTIONS. In case you want to override the system defaults, the settings in DB2OPTIONS and the settings used in the command line flags during an interactive session or a batch job you can use the `UPDATE COMMAND`

OPTIONS command. Afterwards, the settings will revert to the settings chosen before.

Now that we've discussed your non-DBA Cockpit options, it's time to get into the core of this chapter.

5.2 DBA Cockpit

The DBA Cockpit is an SAP tool used to manage the underlying databases. It's a part of every SAP NetWeaver-based system. Initially, it was developed by IBM and SAP for managing DB2 databases. Now all SAP-supported database platforms can use it as a central point for database administration tasks and monitoring. The DBA Cockpit is a consolidated interface to standard DBA functionality, most of which has existed in SAP transactions for years. As shown in Figure 5.13, you can run the DBA Cockpit locally on an SAP NetWeaver-based system by calling Transaction DBACOCKPIT.

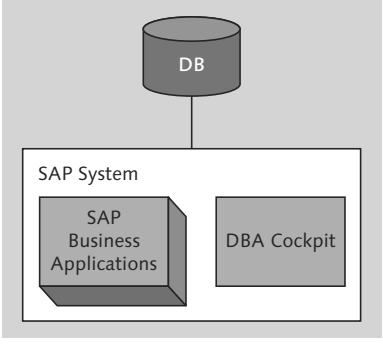


Figure 5.13 Running a DBA Cockpit on a Local SAP System

Alternatively, as shown in Figure 5.14, you can also run the DBA Cockpit on your SAP Solution Manager system, where you can access all databases in your system landscape using remote connections.

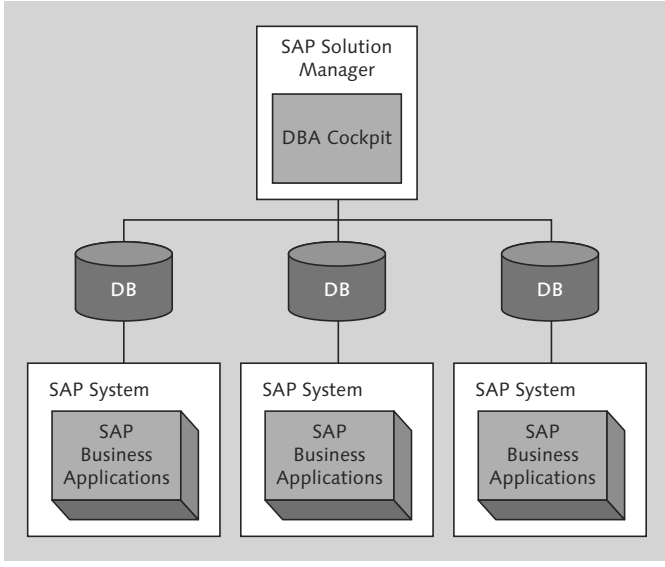


Figure 5.14 A Central DBA Cockpit as Part of SAP Solution Manager

If you use the DBA Cockpit as part of the SAP Solution Manager system, you can update and administer all databases from a central system rather than logging on to each individual system separately.

Now that you have an understanding of the DBA Cockpit and how it can be used with your SAP system, we'll get you familiar with the UI and walk you through the multitude of screen options and tools.

5.2.1 DBA Cockpit UI

In older versions of the DBA Cockpit, the classical SAP GUI-based user interface was used. The new web browser UI differs from this classical approach with regards to the overall screen layout and navigation, customizing of the UI, and additionally provided functions.

Areas of the UI

You can find the following areas in the web browser UI as shown in Figure 5.15.

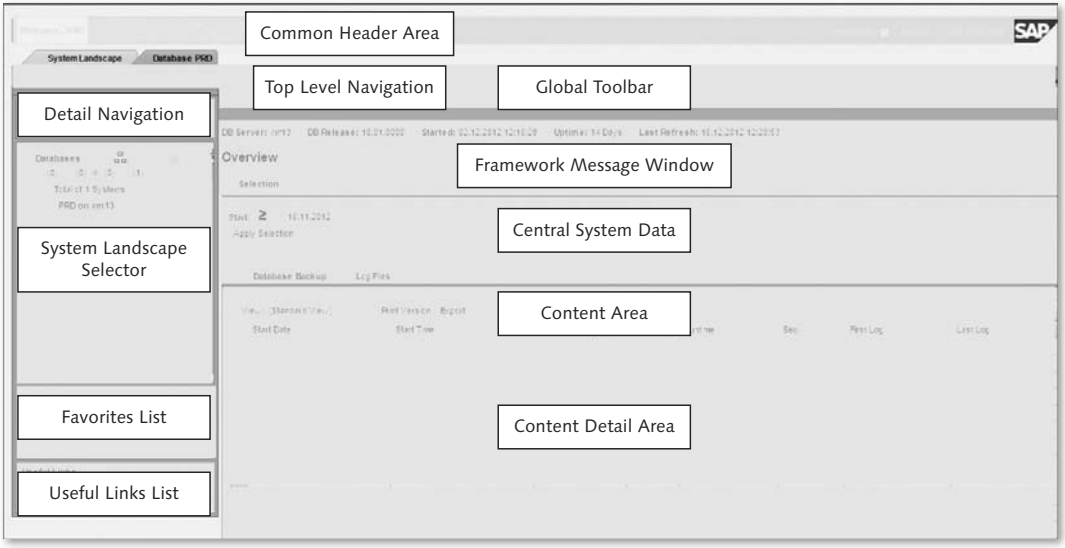


Figure 5.15 Areas in the Web Browser UI

Following are the navigation and screen layout of the browser UI areas:

- Common Header Area
  This area provides a set of standard functions. You can, for example, customize the layout or log off from the DBA Cockpit.
- Top Level Navigation
  In the top-level navigation (including second-level navigation), you can switch between the cross-system area on the System Landscape tab page and the database-specific area on the Database tab page. The System Landscape tab page provides information about the overall system landscape. The Database tab page provides information about the selected database.

  In the second-level navigation, the main tasks areas for your database administration are provided. For fast navigation, pull-down menus corresponding to the related detail levels are available.
- Detail Navigation
  Here you find the main actions of the main task areas. Depending on your selected main action, a set of related actions is available. For example, if you choose Performance – History, the subactions Database and Tables become available.

- System Landscape Selector
  This provides you a quick overview of all configured systems.
- Favorites List
  This list contains links to special tools and actions. If you want to add a specific tool or action to your list, choose Personalize • Add Favorite in the common header area. You can rename or delete favorites by choosing Personalize • Organize Favorites.
- Useful Links List
  You'll find a link to the IBMDB2 Information Center and to the SAP on DB2 for LUW in the SCN: SAP Community Network.
- Framework Message Window
  Messages that are provided by the framework are displayed in the message window. The windows contain a complete history of all messages that are sent during a session.

  When a new message is generated, the message windows are automatically expanded. If there is no message, the window is collapsed by default.
- Global Toolbar
  This toolbar provides a set of globally available functions for navigation and content-related functions.
- Central System Data
  In this area, you find, for example, the time of the last refresh, the startup time, or the database name.
- Content Area
  The content area displays details of the current selected action. Depending on the chosen action, you can see the following areas:
  - **Selection area**
    In this area, you can enter selection criteria for the content to be displayed.
  - **Summary area**
    This area provides a summary overview of the selected data.
  - **Content**
    Depending on the screen and action you have chosen, the content is displayed here.

- Content Detail Area
  If the content in the content area is displayed as a table, you can select a table row and display more details of this table entry in the detail area.

DBA Cockpit Correction Note for SAP Basis 7.02/7.30/7.31

SAP Note 1456402 is a collection of corrections for the DBA Cockpit in SAP Basis Release 7.02/7.30/7.31. This note includes other required SAP notes, and it's regularly extended and updated.

Implement and update the note using Transaction SNOTE (refer to Chapter 9).

Now that you're familiar with the UI, we'll delve into the different screen areas and tools that are available to help you manage the database KPIs and maintain them if necessary.

5.2.2 System Landscape

The System Landscape area provides tools for all systems connected to your DBA Cockpit. The following tools and options are available for you to use to configure and manage your system landscape:

- System configuration
- Central Calendar
- Database connections
- DB Connection Monitor
- Self-monitoring
- SLD System Import

We'll discuss each of these tools in greater detail in the following subsections.

System Configuration

If you don't use SAP Solution Manager for monitoring, you can also use the DBA Cockpit to monitor your databases. Therefore, you have three possibilities, which we discuss in the next sections.

Create Configuration Connection Manually

You can create the system configuration and database connection manually, if the monitored database has not been configured during the integration of a system using SAP Solution Manager. This setup includes the basic setup necessary to connect to the monitored database.

Prerequisites

As a prerequisite, the user for the database connection needs to have sufficient permission. And, if they don't exist, you need to create the tablespace SAPTOOLS and SAPEVENTMON in the monitored database.

Create tablespaces

To create the tablespaces that don't exist as shown in Figure 5.16, choose the button SET UP TABLESPACE SAPTOOLS and SET UP TABLESPACE SAPEVENTMON. Instead of using these buttons (i.e., if you don't have the necessary patch level) you can go to the SPACE/TABLESPACES area. Here you can click the ADD button to see the screen shown in Figure 5.17 and to create the tablespaces.

Figure 5.16 Tablespaces that Don't Exist

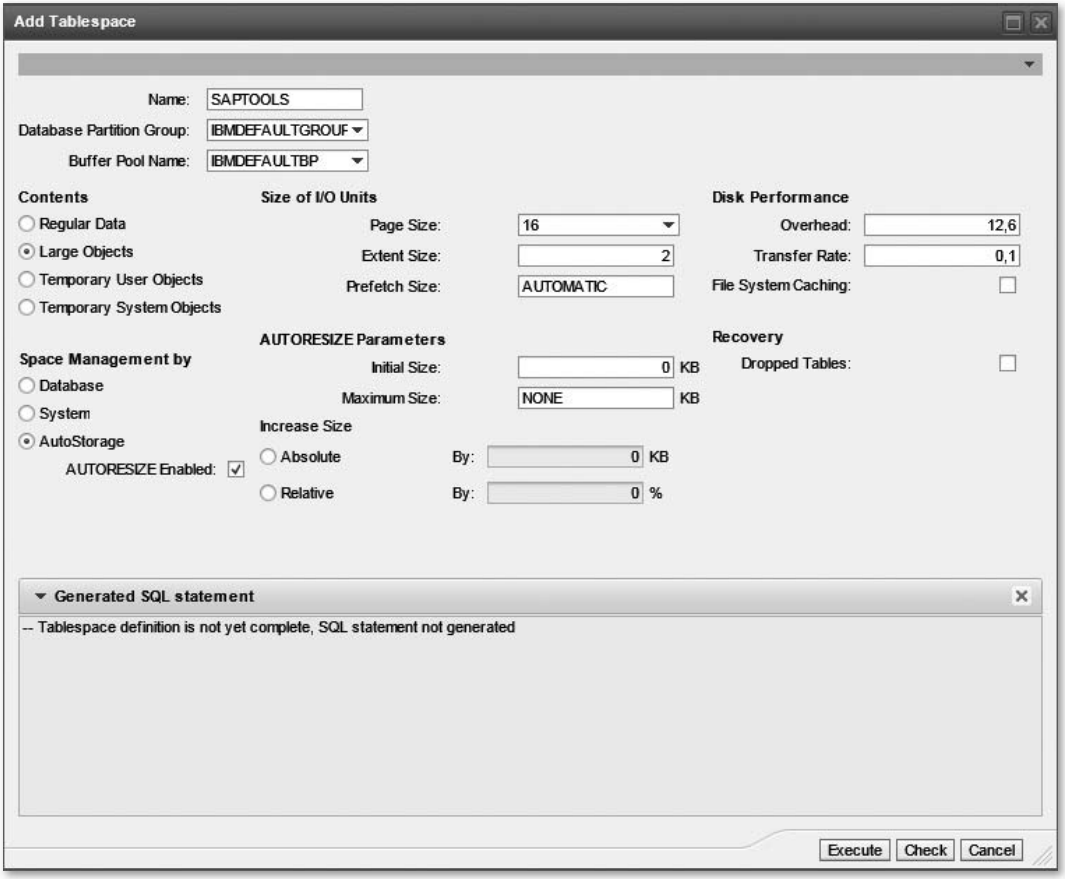


Figure 5.17 Creating Missing Tablespaces

Enter "SAPTOOLS" in the NAME field. In the CONTENTS area, select LARGE OBJECTS. In the SPACE MANAGEMENT BY area, choose AUTOSTORAGE. Click the CHECK button, and if you don't run into an error, click EXECUTE. In a second step, repeat the same procedure for table SAPEVENTMON.

Add system

To add a system, call the DBA Cockpit, and go to the SYSTEM CONFIGURATION. Here you see a list of all available systems and the current system status. Click the ADD button, and a wizard will guide you through the following screens as shown in Figure 5.18.

Figure 5.18 Integrate a System: Introduction

The INTRODUCTION step provides you an overview of the configuration steps. No actions have to be taken here; just click CONTINUE to move to the screen shown in Figure 5.19.

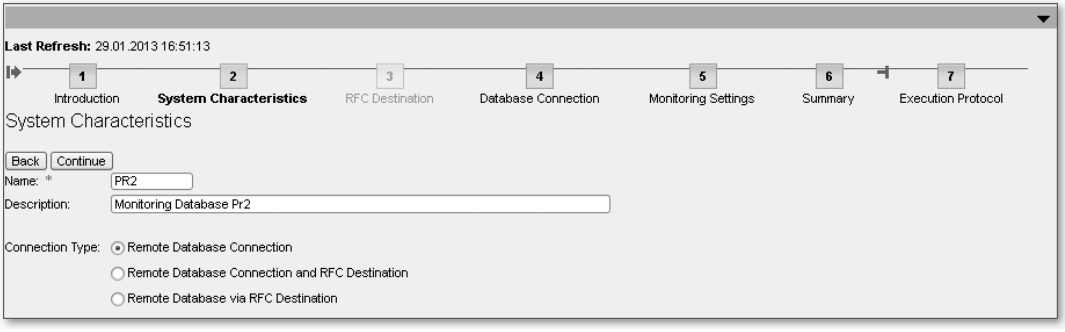


Figure 5.19 Integrate a System: System Characteristics

Here you specify the following:

- The name of the system you want to monitor. This name is a unique ID and doesn't have to be the SAP system ID.
- A description of the monitored system.
- The connection type you want to use. You can use REMOTE DATABASE CONNECTION, REMOTE DATABASE CONNECTION AND RFC DESTINATION, or REMOTE DATABASE CONNECTION VIA RFC DESTINATION.

Specifying the RFC destination is an optional step that is only necessary if you have chosen a connection type that requires an RFC destination.

The RFC destination must already exist and be available. You can test the connection by clicking Test Connection as shown in Figure 5.20. Click Continue to move to the screen shown in Figure 5.21.

Figure 5.20 Integrate a System: RFC Destination

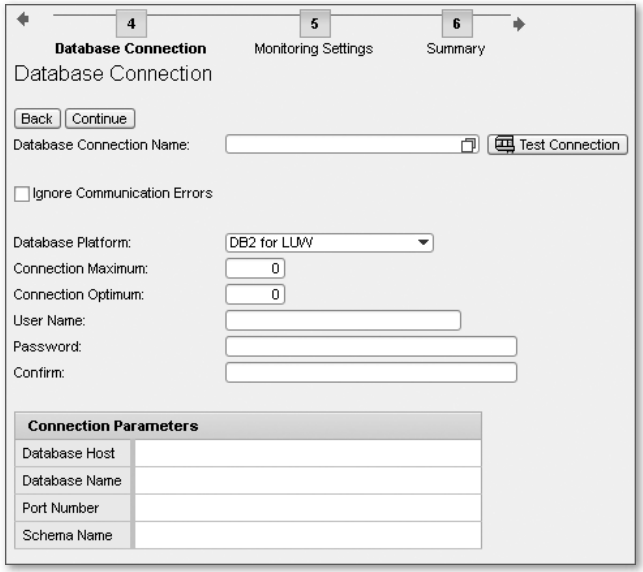


Figure 5.21 Integrate a System: Database Connection

Here you can pick an existing database connection or create a new one. See the "Database Connections" section later in this chapter to find details on how to create a database connection.

Collect monitoring data

In the screen shown in Figure 5.22, you specify how you want to collect monitoring data:

- Activate Alert Monitoring
  If you use the RZ20 alert monitor, use this option. If you're using the

DBA Cockpit in SAP Solution Manager 7.1, you should switch to E2E alerting instead. This doesn't need any specific setup.

- Collect Space and Performance History Data
  Check this option if the monitoring data has to be collected by the remote system.
- Show scheduled jobs in central planning calendar
  Check this option if your actions should appear in the Central Planning Calendar.

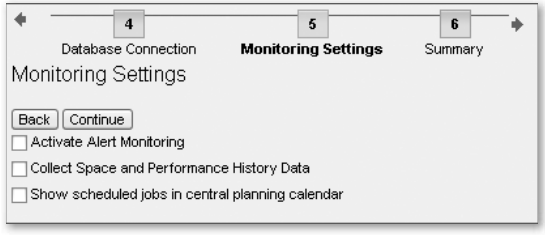


Figure 5.22 Integrate a System: Monitoring Settings

On the last screen, you get a summary of all performed actions.

Use the System Landscape Directory

The second option for system configuration is to use database information that is stored in the system landscape directory (SLD) and generate or update your system information automatically. For more details see Section 5.2.3.

Set Up Database Connections

You can use the Database Connection screens starting in Figure 5.21 to set up database connections that are used for non-monitoring components.

Central Calendar

The Central Calendar screen visualizes actions on all of the databases of your SAP systems connected to your DBA Cockpit. This is a single point to manage your DBA actions in an integrated SAP environment. Depending on the database platforms you use next to DB2 10.1 the available actions may differ, but they way in which you use it is always the same.

By using the selection screen (Figure 5.23), you can decide from which date you want to see the performed actions and what factory calendar to use.

Figure 5.23 Central Calendar Selection

In the CENTRAL CALENDAR screen (Figure 5.24), you can only check the results of the performed action. If you need to schedule, change, delete, or execute actions, you have to jump into the SAP system's DBA Planning Calendar, which we explain next.

Thursday	Friday	Saturday	Sunday
December, 27	December, 28	December, 29	December, 30
003 PR2 001	002 PR2 002	002 PR2 002	002 PR2 002
001 PRD 001	001 PRD 001		001 PRD 001
January, 03	January, 04	January, 05	January, 06
002 PR2 002	002 PR2 002	002 PR2 002	002 PR2 002
001 PRD 001	001 PRD 001		001 PRD 001
January, 10	January, 11	January, 12	January, 13
002 PR2 002	002 PR2 002	002 PR2 002	002 PR2 002
001 PRD 001	001 PRD 001		001 PRD 001
January, 17	January, 18	January, 19	January, 20
002 PR2 002	002 PR2 002	002 PR2 002	002 PR2 002
001 PRD 001	001 PRD 001		001 PRD 001

Figure 5.24 Central Calendar

Steps to Access the Central Calendar

You start the Central Calendar by choosing CENTRAL CALENDAR on the SYSTEM LANDSCAPE tab page. The Central Calendar displays an overview of past actions and planned actions.

Figure 5.25 shows you the overview of a day you get when you open the Central Calendar. This example shows you two affected systems: PRD and PR2. On system PR2, three actions are planned for Thursday the 27th of December. One action is running right now.

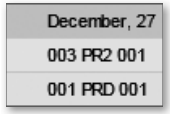


Figure 5.25 Central Calendar Overview Day

On system PRD, only one action is planned, but it isn't executed yet. By clicking the LEGEND button, you can see the color-coded statuses (Table 5.3) that are used in the Central Calendar.

Status

Color	Status
Light blue	Planned
Dark blue	Running
Green	Finished successfully
Yellow	Finished with warning
Red	Finished with error
Dark yellow	No longer available
Dark red	Scheduling failed

Table 5.3 Color-Coded Statuses in the Central Calendar

To see a summary of the actions for a day, click the DAY header. In the DETAILS area, you see a summary of the actions and the status for each system as shown in Figure 5.26.

Figure 5.26 Central Calendar Details

Scheduled job statistics

You have several options you can use to obtain new statistics for the scheduled jobs. One option is to choose the Refresh button. If you have a lot of actions planned, this might take a long time, so this option isn't recommended.

To schedule the refresh, choose the Refresh in the Background button, and use the dialog box shown in Figure 5.27.

Figure 5.27 Schedule Refresh Dialog Box

This dialog box provides the following options:

- Run in Dialog
  This runs the refresh in the dialog mode. This option isn't recommended.
- Start Immediately in the Background
  The job is immediately started as a background job.
- Schedule
  The refresh job runs in the background on the date you choose.

Database Connections

In this area, you can set and maintain technical attributes for remote database connections. These connections are used for administration, monitoring, or application programs that use secondary connections to external databases. New connections can be created in this screen, but you can also create them using the System Configuration page (see the previous "System Configuration" section).

When you call the DATABASE CONNECTIONS screen, a list (see Figure 5.28) of available database connection definitions is displayed.

Figure 5.28 Overview Database Connections

By default, the database connections that are defined in the local system are displayed as detailed in Table 5.4.

Column	Description
REMOTE DATABASE CONNECTION	Name of the database connection. You can choose this unique name.
DB NAME	Name of the database.
DB HOST	Name of the database host.
DB SCHEMA	Name of the monitored database schema.
USER NAME	Name of the connect user.
PERMANENT	Is the connect user permanently available?
MAX. CONNECTIONS	Maximum allowed numbers of open connections.
OPT. CONNECTIONS	Optimal number of connections.

Table 5.4 Columns used for Database Connections

If you want to add a database connection, click the ADD button, and the dialog box shown in Figure 5.29 appears.

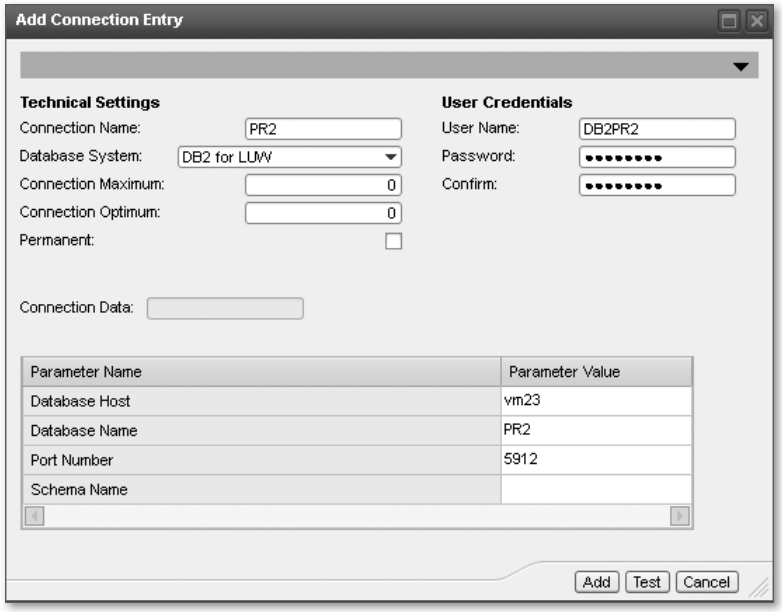


Figure 5.29 Add New Database Connection

In this dialog box, you can enter the values that are detailed in Table 5.5.

Attribute	Description
Connection Name	Unique name you choose freely (except names reserved by SAP).
Database System	Name of the database system.
Connection Maximum	Limits the number of database connections that are currently held.
Connection Optimum	Optimal number of connections.
Permanent	Set this parameter if the connection is absolutely required to run your system. If this parameter is set, the database connection is handled like a local one, so if the database connection isn't available, a work process that wants to use it can't run.

Table 5.5 Attributes for Adding a Database Connection

Attribute	Description
User Name	Name of the connect user (authorizations required).
Password + Confirm	Password for the connect user.
Database Host	Name of the remote database server.
Database Name	Name of the database.
Port Number	Number of the port.
Schema Name	Name of the schema to be monitored. If you leave this field blank, the SAP connect user is used as the schema.

Table 5.5 Attributes for Adding a Database Connection (Cont.)

To change one of your database connections, select the connection and choose the Change button. Now you can change the field you like. To check if you have entered the correct user and password, click the Test button. The result is displayed in a protocol in the content detail area. To save, click the Save button. If you want to delete a connection, choose the connection and click the Delete button. Be aware that you can only delete connections that aren't used by a system registered in the DBA Cockpit.

DB Connection Monitor

Monitor database connections

The DB Connection Monitor allows you to monitor all database connections from all connected application servers. All disconnected database connections are hidden by default. The information in Table 5.6 is displayed.

Column	Description
Work Process	Number of the work process.
Handle	Each work process can have several connections; this is an internal number of each connection.
Connection Name	Name of the connection.
Connection ID	ID of the connection.

Table 5.6 Columns in the DB Connection Monitor

Column	Description
Connection State	This is the state of the connection. Possible states are active, inactive, connecting, and disconnected.
TX (Changing Operation)	Tells you if a changing operation was executed.
BC (Block Commit)	Indicates if `COMMIT` statements are blocked.
HC (Cursor with Hold)	The connection has a cursor with hold.
PRM - Permanent Connection	The connection is permanent.
RCT - Reconnect	Indicates that the connection automatically reconnects after it is closed.
FRC - Force Reconnect	The connection is forced to reconnect after it is closed.
TIM - Timeout	Timeout when the connection is closed.
Maximum	Maximum allowed open connections.
Optimum	Optimal number of connections.
Date	Date when the connection was established.
Time	Time when the connection was established.
DB Host	Host name.
Program	Program that opened the connection.

Table 5.6 Columns in the DB Connection Monitor (Cont.)

SLD System Import

Outdated function

Before we briefly explain the SLD system import, note that we don't recommend that you use this outdated function anymore. Instead, you should use SAP Solution Manager for the setup and monitoring of the system landscape. If you want to use it anyway, choose the SLD System Import Button.

The SLD System Import screen appears, and you see the following nodes (depending on your landscape):

- New Database Systems in the SLD
- Changed Systems from Earlier SLD Imports
- Systems No Longer Registered in the SLD
- Systems Identical in the SLD and in the DBA Cockpit
- Unsupported Database Systems in the SLD

Click the Change button to import the database system data. All actions allowed are displayed now. Select the action you want to execute for the selected database, and choose the Import button.

Users and passwords aren't available via the SLD, so you might need to add some connection information.

5.2.3 Performance

Performance is the first point on the Database tab of the DBA Cockpit. The goal of the performance tuning is to maximize the use of your system resources to perform work as efficiently and rapidly as possible. The DB2 database is designed to manage work efficiently, but it is possible to improve the performance by customizing settings and the configuration of your database according to your setting.

Therefore the following areas are available on the tab:

- Performance Warehouse
- Time Spent Analysis
- Workload Statistics
- Top SQL Statement Analysis
- Snapshots
- Critical Activities
- Utilities
- History

We'll explain each of these areas in the following subsections.

Performance Warehouse

Historical performance analysis

The Performance Warehouse screen allows you to analyze the historical performance data of your database system and your SAP applications. All relevant performance indicators that are collected from all SAP systems are stored in your central Business Intelligence (BI) system. The historical data can then be mined, trended, and analyzed, using powerful SAP NetWeaver Business Warehouse (BW) interfaces with charts, dashboards, and drilldown capabilities.

Performance Warehouse: Configuration

This screen is only available in the DBA Cockpit of your SAP Solution Manager system if the Database Performance Warehouse has been configured.

SMD

The prerequisite to use the Performance Warehouse option is to have an SAP Solution Manager system with enabled Solution Manager Diagnostics (SMD). You use the SMD Setup Wizard to configure the extraction of data into the SMD BI component. Your data is stored in the SMD BI. You can access the Configuration screen of the performance warehouse by choosing Performance Warehouse • Configuration. You can configure all necessary parameters on the Performance Warehouse Configuration screen. The available tab pages include Configuration, Web Reports, and Report Categories, which we'll explained in the following list:

- Configuration
  You can view or modify the configuration parameters for the monitored system. Depending on your database platform, the displayed selection of values can vary. For all database platforms, the parameters BI Server, Managing DBA Cockpit, and Reporting Time Zone are displayed. The BI Server parameter designates which BI server to use for performance data. The Managing DBA Cockpit-parameter indicates which DBA Cockpit is allowed to change data collectors or configuration for this database. The DBA Cockpit of the SAP Solution Manager system is the default setting. For all reports, the performance data timestamps are converted to one global time zone.
- Web Reports
  You can configure the display on the reporting screen by modifying the integrated SAP Business Explorer (BEx) web templates. For each

report category, you can view or modify the views by expanding the appropriate report category. The main report categories are displayed. To dive deeper into the details of a view, select it in the table. Then you can display the name, detailed description, category of the report, database platform used, and data providers.

- Report Categories
  You can view and modify the categories for BEx web templates. You can change the sequence of the categories on the Reporting screen using the Up and Down buttons.

Time Spent Analysis

Performance tuning

The Time Spent Analysis area has been available since DB2 version 9.7. It's a starting point for performance tuning or to identify time-based problems of your database. By picking a time frame and metrics, you can create an aggregated view on the time line of the current or past performance situation of your database. This allows you to analyze specific workload situations in real time. Besides using the standard database KPIs such as the buffer pool hit ratio, you can also identify how much time is spent on the different kinds of database operations.

The Time Spent Analysis area of the tab is divided into the following two subareas:

- The Selection area (see Figure 5.30) lets you decide if you want to use the data collected during the periodic data collection or ad-hoc data to run the analysis. You can choose the time frame, the involved service classes, the applied metrics and how to drill down the analysis.

Figure 5.30 Time Spent Analysis Selection

- The Chart view Time Spent in DB2 by Metric includes the options to display it as a pie chart, as shown in Figure 5.31, or to display it as a time line histogram.

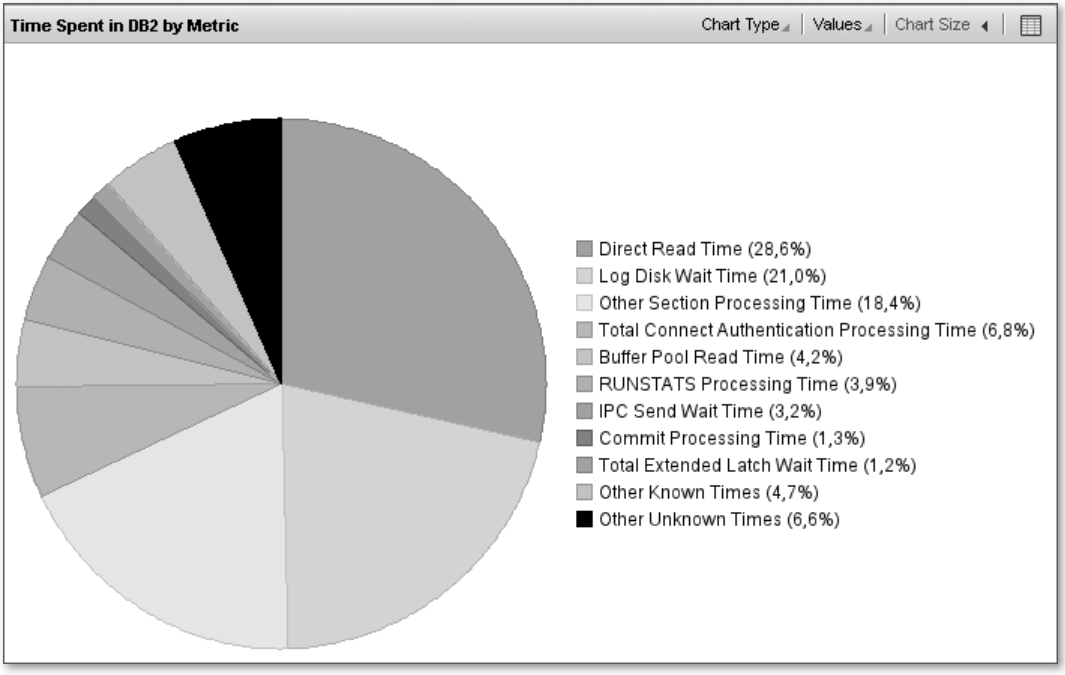


Figure 5.31 Time Spent Analysis: Pie Chart

Workload Statistics

Database system behavior

The WORKLOAD STATISTICS area has been available since DB2 V9.5. These basic statistics describe the overall behavior of the database system. For example, you can use them to do the following:

- Investigate a system slowdown and understand the type of slowdown.
- Monitor your service level agreements (SLAs).
- Define workload management (WLM) thresholds using the high watermark metrics.
- Search for members, time frames, or service classes in which queries are running long.

In the SELECTION area (see Figure 5.32), you can choose the time frame and the service classes. By choosing ADVANCED SELECTION, you see all available service classes and the parent service class. If you have a distributed database system, you can also filter by members and service classes.

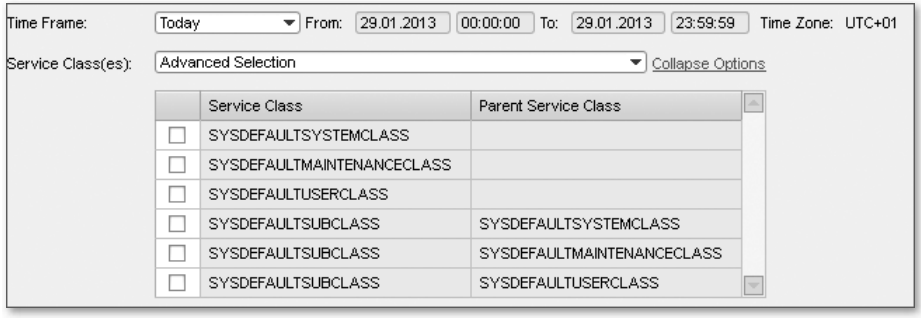


Figure 5.32 Workload Statistics Selection

After applying the selection, you can see the results in the Activity Lifetimes (shown in Figure 5.33) and the Statistics tab. You can choose between three different histogram types.

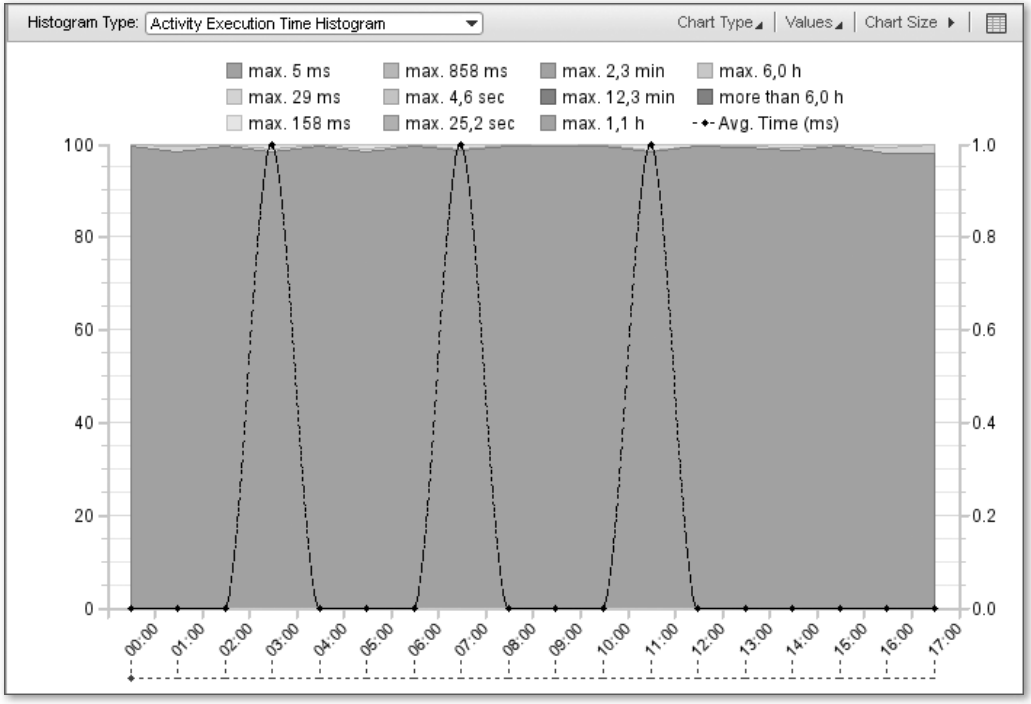


Figure 5.33 Activity Lifetime Histogram

Histograms

The following histogram types are available:

- Activity Execution Time Histogram
  This histogram displays the execution times of all activities that were executed in the selected service class and time frame. *Execution time* means the time an activity has spent executing. It doesn't include the time spent during initialization, queuing, or between cursor operations.
- Activity Lifetime Histogram
  This histogram displays the lifetime of all activities that were executed in the selected service class and time frame. *Lifetime* means the total elapsed time of an activity, including the time spent during initialization, queuing, or between cursor operations.
- Activity Queue Time Histogram
  This histogram displays the time activities spent in a WLM queue in the selected service class and time frame.

Below the histogram chart, an additional chart is displayed with the total number of activities that have been completed, rejected, or aborted as shown in Figure 5.34.

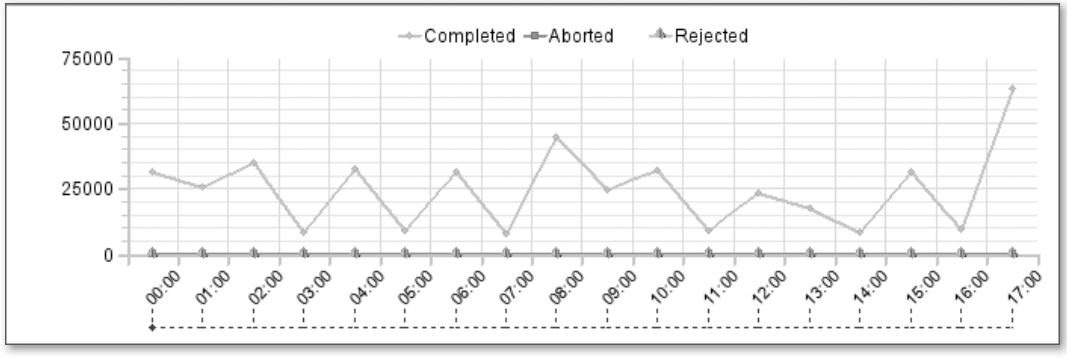


Figure 5.34 Workload Statistics Chart

The Statistics tab provides you with average metrics and a range of high watermarks. The information from Table 5.7 is displayed.

Column	Description
Time from	Time and date the statistics interval begins.
Time to	Time and date the statistics interval ends.
Member	The member that has captured the statistics.
Service Superclass	Name of the service superclass for which the statistics were created.
Service Subclass	Name of the service subclass for which the statistics were created.
Coord. Act. Lifetime Top	High watermark for the coordinator activity lifetime (in milliseconds).
Coord. Act. Lifetime Avg.	Arithmetic mean of the coordinator activity lifetime (in milliseconds).
UOW Total Time Top	High watermark for the unit of work lifetime (in milliseconds).
Act. Temp. Tablespace Top	High watermark for the temporary tablespace usage of a single Data Manipulation Language (DML) activity (in kilobytes).
Agg. Temp. Tablespace Top	High watermark for the aggregate temporary tablespace across all DML activities (in kilobytes).
Activity CPU Time Top	High watermark for processor time used (in milliseconds).
Rows Read Top	High watermark for the number of rows read.
Rows Returned Top	High watermark for the number of rows returned.

Table 5.7 Workload Statistics

Top SQL Statement Analysis

Analyze metrics

The Top SQL Statement Analysis area of the screen has been available for databases as of release DB2 V9.7. This tools helps you display information about SQL statements that are executed either very often or that are very expensive with regards to specific metrics. With this information, you can identify SQL statements that consume a lot of resources and can

think about fine-tuning these statements to improve the performance of the database.

Let's examine the analysis steps and how to work with this tool.

Analysis steps

The Package Cache data collector of the data collection framework (DCF) captures the required data and dumps it into the history tables. The data collector performs a top "n" analysis and only stores the data of those statements that are in the top "n" based on the defined metric in the history tables. Therefore, the data provided in this screen isn't a 100% view on all executed SQL statements on the database server because it includes only the top "n" statements.

In the SELECTION area (see Figure 5.35), you specify the time frame and the way the top SQL statements are chosen. You can select either the TOP SQL STATEMENTS BY or the CUSTOM radio button.

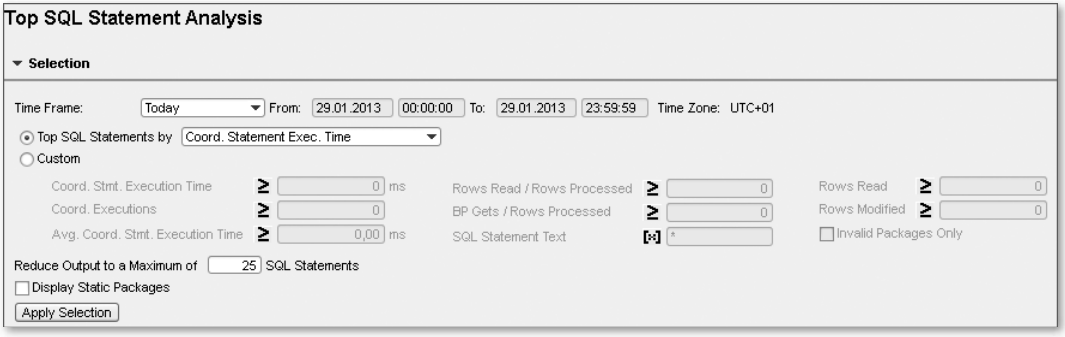


Figure 5.35 Top SQL Statement Analysis Selection

SQL STATEMENTS BY lets you choose a ranking criterion from a dropdown list. This will be the first metric column and is used for ranking. The CUSTOM radio button allows you to define your own filter criteria. You can also limit the number of SQL statements and use the DISPLAY STATIC PACKAGES option, if you want to analyze static packages.

By clicking the APPLY SELECTION button, the information detailed in Table 5.8 is displayed.

Column	Description
Statement Text	Text of the SQL statement.
Ranking metric	Default: Coord. Statement Exec. Time = Execution time by coordinator agent. This metric column is used for ranking. You choose what metric to use in the Selection area.
No. of Coord. Executions	Number of executions by coordinator agent.
Avg. Coord. Stmt. Exec. Time	Average execution time for the statement in milliseconds.
Total CPU Time	Total CPU time in milliseconds.
Rows Read/Rows Processed	Ratio of rows read from the base table compared to the rows processed (`SELECT`, `UPDATE`, `INSERT`, or `DELETE` statement). A high value indicates statements with inefficient access; the value "1" stands for optimal access.
BP Gets/Rows Processed	The average number of pages read from the buffer pool per rows processed (`SELECT`, `UPDATE`, `INSERT`, or `DELETE` statement).

Table 5.8 Top SQL Statement Analysis

SQL Statement Details area

By double-clicking on an SQL statement, you open the SQL Statement Details area as shown in Figure 5.36.

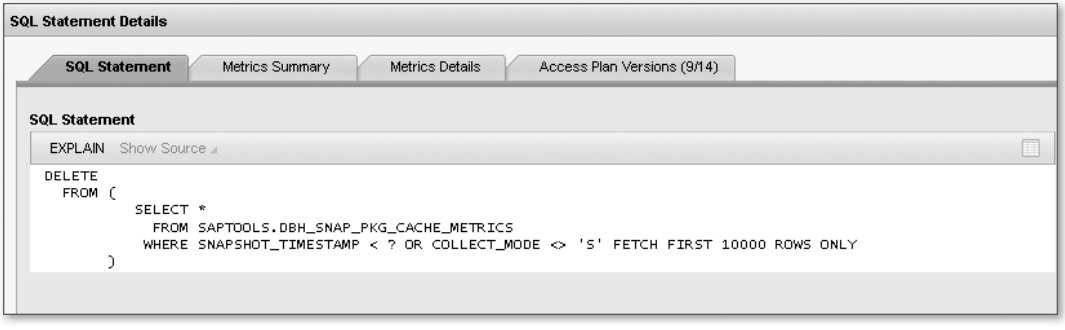


Figure 5.36 Top SQL Statement Analysis Details: SQL Statement

On the SQL Statement tab page, you can see the complete SQL statement as a text. To jump into the current access plan of the statement, click the EXPLAIN button (also see the "EXPLAIN Access Plan" section).

Metrics Summary

The Metrics Summary tab displays a chart with the time spent information for the SQL Statement as shown in Figure 5.37. Under the chart, the following information is displayed:

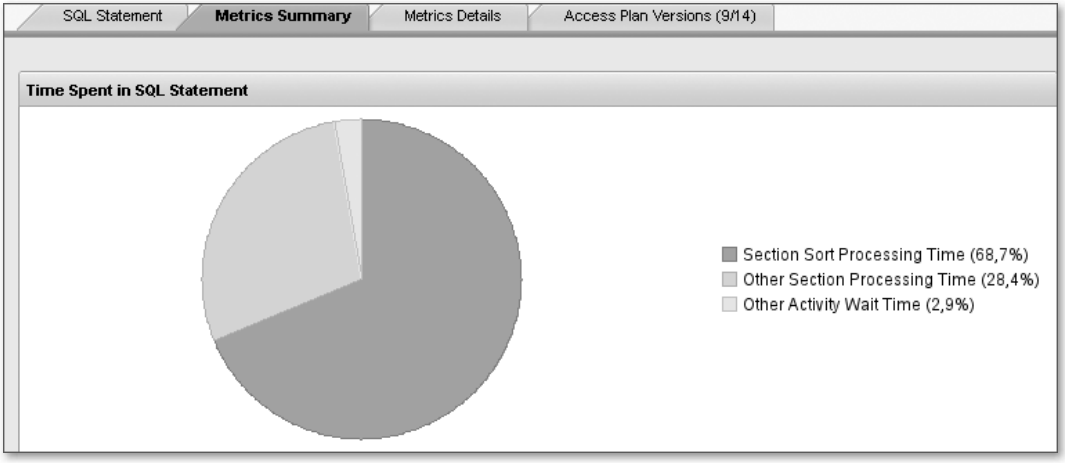
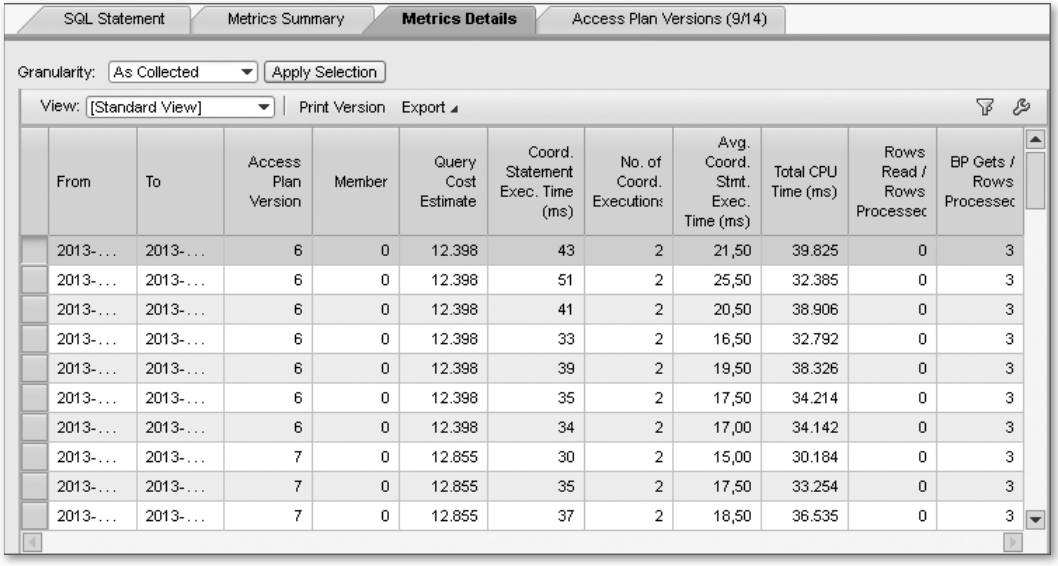


Figure 5.37 Top SQL Statement Analysis Details: Metrics Summary

- Statement metadata
- Miscellaneous metrics
- Sorts
- Locking
- Logger
- Buffer pool
- Direct I/O
- Metrics Details

The Metrics Details tab page provides you with time slices and detailed information for each time slice as shown in Figure 5.38. When highlighting a time slice, the following information is displayed:



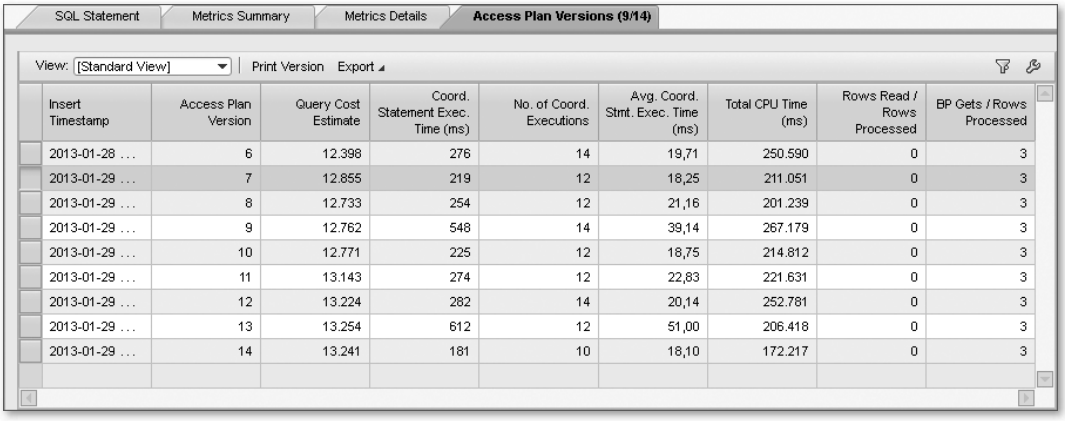
From	To	Access Plan Version	Member	Query Cost Estimate	Coord. Statement Exec. Time (ms)	No. of Coord. Executions	Avg. Coord. Stmt. Exec. Time (ms)	Total CPU Time (ms)	Rows Read / Rows Processed	BP Gets / Rows Processed
2013-...	2013-...	6	0	12.398	43	2	21,50	39.825	0	3
2013-...	2013-...	6	0	12.398	51	2	25,50	32.385	0	3
2013-...	2013-...	6	0	12.398	41	2	20,50	38.906	0	3
2013-...	2013-...	6	0	12.398	33	2	16,50	32.792	0	3
2013-...	2013-...	6	0	12.398	39	2	19,50	38.326	0	3
2013-...	2013-...	6	0	12.398	35	2	17,50	34.214	0	3
2013-...	2013-...	6	0	12.398	34	2	17,00	34.142	0	3
2013-...	2013-...	7	0	12.855	30	2	15,00	30.184	0	3
2013-...	2013-...	7	0	12.855	35	2	17,50	33.254	0	3
2013-...	2013-...	7	0	12.855	37	2	18,50	36.535	0	3

Figure 5.38 Top SQL Statement Analysis Details: Metrics Details

- Statement metadata
- Miscellaneous metrics
- Sorts
- Locking
- Logger
- Buffer pool
- Direct I/O

Access Plan Versions

The Access Plan Versions tab gives you an overview of the different generated access plans for this SQL statement in the history (see Figure 5.39). An access plan version is a virtual version number. If the SQL statement is flushed from the package cache and recompiled, a new version is used. The estimated query costs can be used as an indicator for similar access plans. To find out more about the access plan, just select the line you want to dive deeper into, and the Access Plan Details area appears.



Insert Timestamp	Access Plan Version	Query Cost Estimate	Coord. Statement Exec. Time (ms)	No. of Coord. Executions	Avg. Coord. Stmt. Exec. Time (ms)	Total CPU Time (ms)	Rows Read / Rows Processed	BP Gets / Rows Processed
2013-01-28 ...	6	12.398	276	14	19,71	250.590	0	3
2013-01-29 ...	7	12.855	219	12	18,25	211.051	0	3
2013-01-29 ...	8	12.733	254	12	21,16	201.239	0	3
2013-01-29 ...	9	12.762	548	14	39,14	267.179	0	3
2013-01-29 ...	10	12.771	225	12	18,75	214.812	0	3
2013-01-29 ...	11	13.143	274	12	22,83	221.631	0	3
2013-01-29 ...	12	13.224	282	14	20,14	252.781	0	3
2013-01-29 ...	13	13.254	612	12	51,00	206.418	0	3
2013-01-29 ...	14	13.241	181	10	18,10	172.217	0	3

Figure 5.39 Top SQL Statement Analysis Details: Access Plan Versions

The SQL Statement tab is displayed first when you go into the Access Plan Details screen (see Figure 5.40). You see the SQL Statement and information about the Compilation Environment.

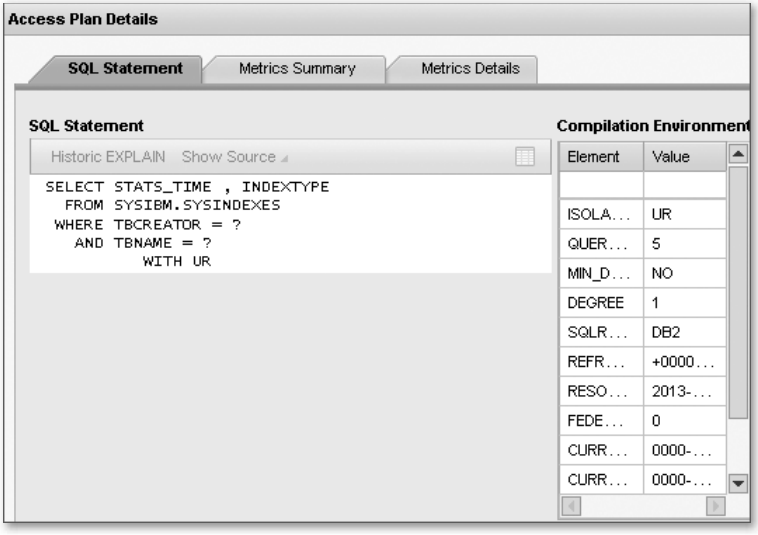


Figure 5.40 Top SQL Statement Analysis Details with Access Plan Versions and SQL Statement

You can take a look at the access plan of the statement by clicking the Historic EXPLAIN button. History in this context means that you see

an access plan from the time it was generated. To find out more about the EXPLAIN function, see the "EXPLAIN Access Plan" subsection in Section 5.2.11.

In the Metrics Summary tab, you see a chart view of Time Spent in SQL Statement during the time frame defined in the selection area (see Figure 5.41). Under the chart, the following information is displayed:

- Statement metadata
- Miscellaneous metrics
- Sorts
- Locking
- Logger
- Buffer pool
- Direct I/O

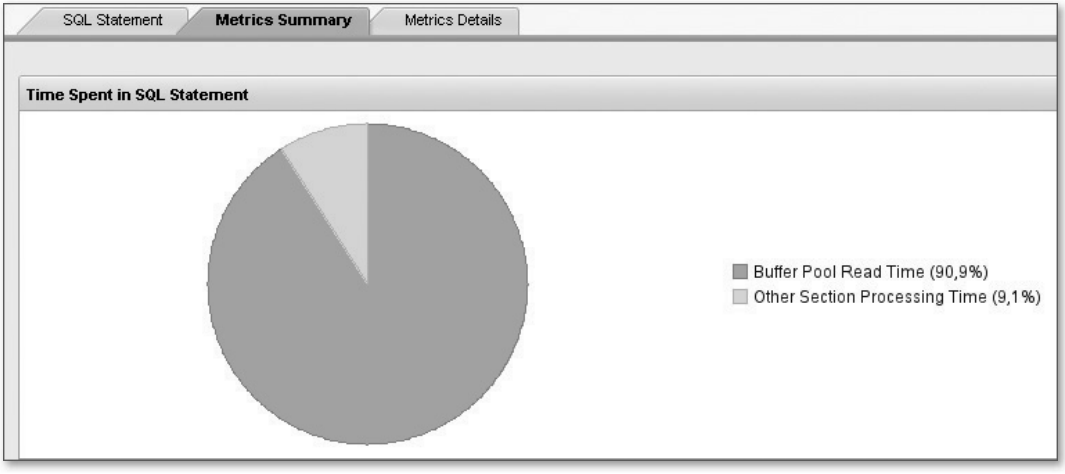
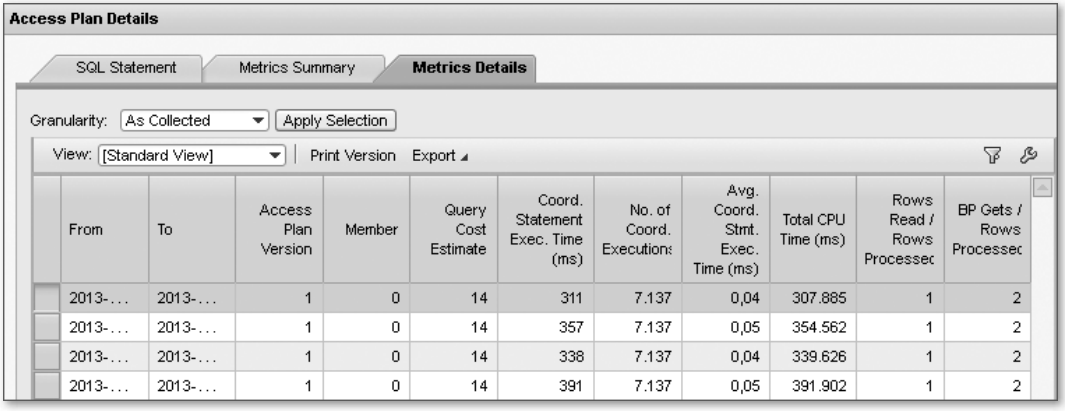


Figure 5.41 Top SQL Statement Analysis Details with Access Plan Versions and Metric Summary

The Metrics Details tab (see Figure 5.42) allows you to display the following detailed information for specific time slices for the access plan version:

- Statement metadata
- Miscellaneous metrics
- Sorts
- Locking
- Logger
- Buffer pool
- Direct I/O



From	To	Access Plan Version	Member	Query Cost Estimate	Coord. Statement Exec. Time (ms)	No. of Coord. Executions	Avg. Coord. Stmt. Exec. Time (ms)	Total CPU Time (ms)	Rows Read / Rows Processed	BP Gets / Rows Processed
2013-...	2013-...	1	0	14	311	7.137	0,04	307.885	1	2
2013-...	2013-...	1	0	14	357	7.137	0,05	354.562	1	2
2013-...	2013-...	1	0	14	338	7.137	0,04	339.626	1	2
2013-...	2013-...	1	0	14	391	7.137	0,05	391.902	1	2

Figure 5.42 Top SQL Statement Analysis Details with Access Plan Versions and Metric Details

Snapshots

Monitor data

In the Snapshots task area, you can handle various snapshots to monitor data in an unfiltered and raw format. Snapshot data is collected right after you start the database until now. Data is accumulated over a long time period, so it isn't the best tool to investigate short-term performance or workload problems. If you want to use it for a shorter time period, you can set an explicit starting point using the Reset and Since Reset buttons. Or if your snapshot data is collected by the DCF, you can specify a time interval for your analysis. This option is applicable for the following snapshot options:

- Database
- Buffer Pools

- Tablespaces
- Containers
- Tables
- SQL Cache
- System Resources
- Transaction Log

The following options can't be used with DCF support:

- Schemas
- Applications

When choosing Performance/Snapshots in the top level navigation, the Database snapshot is opened by default, and the snapshot options (see Figure 5.43) appear on the left.

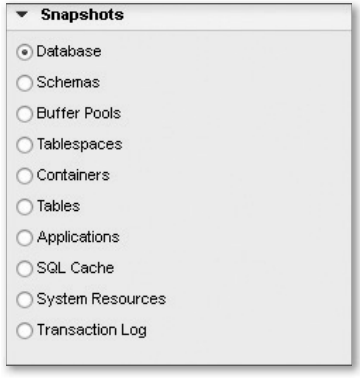


Figure 5.43 Snapshots Options

Database

Snapshot: Database option

When you look at your database snapshot, depending on the version of your database and if you configured the data collection framework, you either have a database snapshot with or without DCF support.

Database snapshots without DCF support

If you don't have DCF support, your Database screen provides you a selection of performance data that you can use to identify performance-critical partitions before starting a more detailed analysis. For each partition of your database system, the following information is shown:

- Number and total size of buffer pools
- Number of data and index logical reads
- Number of data and index physical reads
- Average physical read and write time required to read data from and write data to
- The buffer pool
- Executed SQL statements
- Package cache size
- Package cache quality

Database snapshots with DCF support

In contrast to the database snapshot without DCF support, if you do have DFC support, you can view the current data and monitor the periodically collected data.

The Database screen provides the following:

- A selection area where you can specify the time frame
- An overview table with the most important key figures aggregated over the selected time period
- A detailed results list of selected monitoring data
- A graphical view of the history data (see Figure 5.44)
- A detailed view of all historical data collected during a specified time frame

To analyze your database snapshot data, first identify the period in which you experienced problems. These problems can be long-running background jobs or bad user response time, for example.

In the Selection Area, you specify the time frame. Now refresh your monitoring data by clicking Apply Selection. You now see information about each partition. To find out more about a partition, select the line in the overview table for detailed information. Select the Summary tab to see a graphical view of the history (Figure 5.45) and a summary of many relevant figures (Figure 5.44).

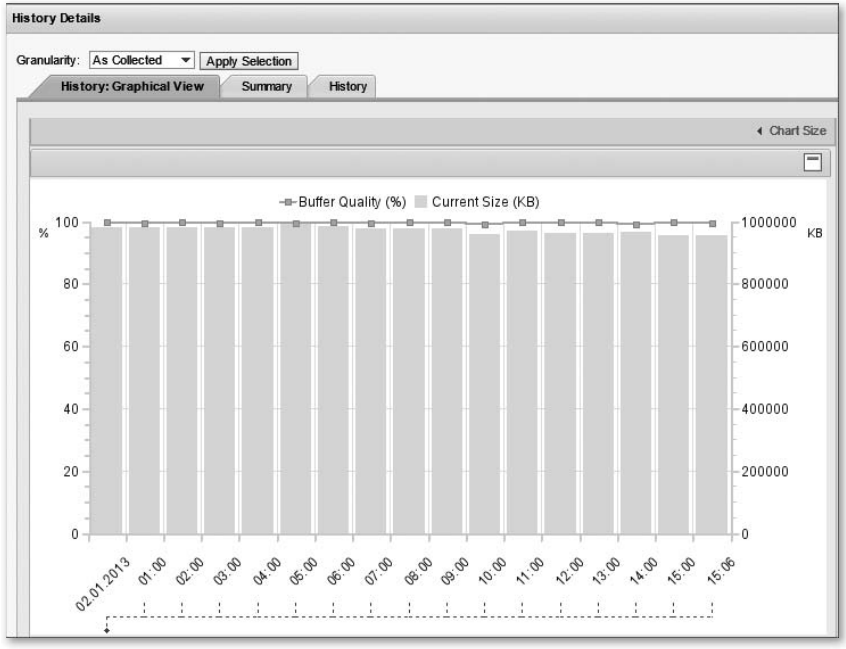


Figure 5.44 Snapshot: History

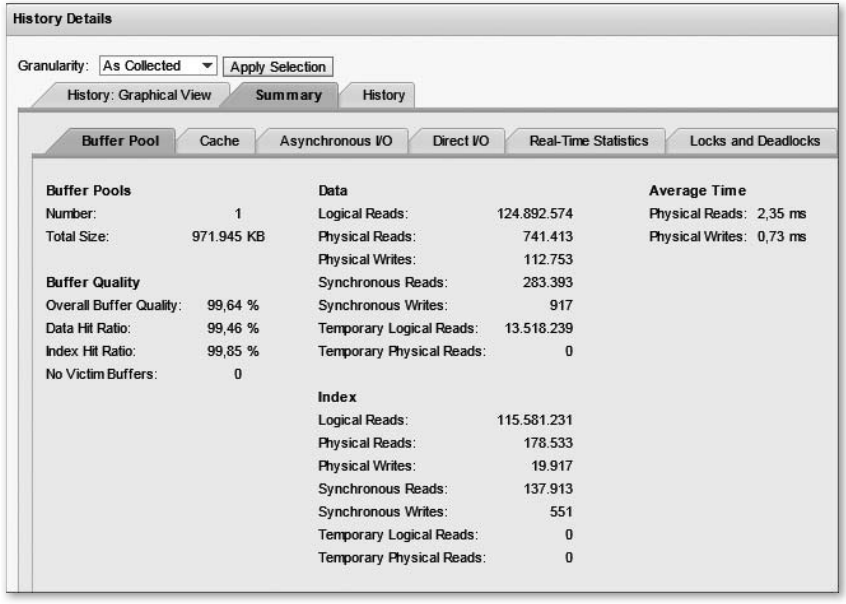


Figure 5.45 Snapshot: Database Summary

Schema

Snapshots: schema

Using the SCHEMA option allows you to research performance problems if more than one SAP component is installed within the same database. You can identify the workload distribution among the components and identify performance-critical components. The information detailed in Table 5.9 is provided for your components.

Column	Description
USER	Name of the connection user
PARTITION	Number of the partition (only displayed if you use a multi-partition database)
SAP COMPONENT	Shows you if your connection user is related to the SAP component
DATA LOGICAL READS	Number of read accesses to data stored in the buffer pool
DATA LOGICAL READS (%)	Percentage of logical data read accesses
INDEX LOGICAL READS	Number of read accesses to index data stored in the buffer pool
INDEX LOGICAL READS (%)	Percentage of index logical data read accesses
DATA PHYSICAL READS	Number of read accesses to data on disk (I/O)
DATA PHYSICAL READS (%)	Percentage of physical data read accesses
INDEX PHYSICAL READS	Number of read accesses to index data stored on disk (I/O)
INDEX PHYSICAL READS (%)	Percentage of physical index data read accesses

Table 5.9 Component Information Displayed in Snapshot: Schema

Buffer Pools

Snapshot: buffer pools

If you want to investigate your databases buffer pools, use the BUFFER POOLS option (see the BUFFER POOL tab shown in Figure 5.46). You can use this snapshot with or without DCF support.

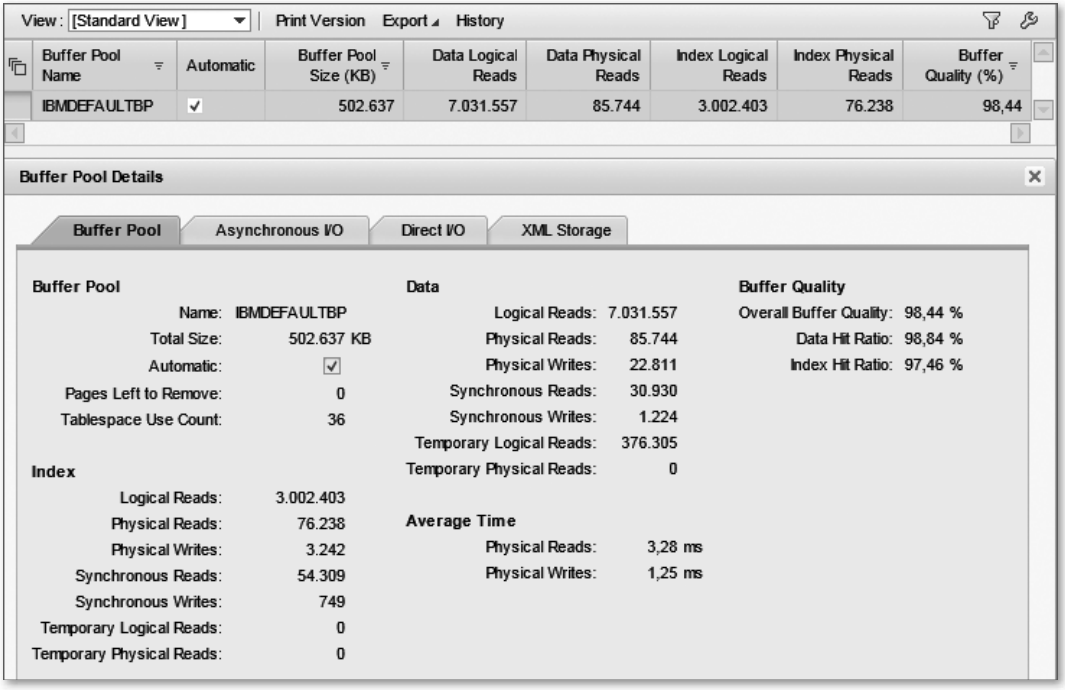


Figure 5.46 Snapshot: Buffer Pools without DCF Support

As shown in Figure 5.46, the BUFFER POOL screen gives you an overview of several key indicators of buffer pools.

When displaying the BUFFER POOLS screen with DCF support, you see the current global buffer pool snapshot just as you do without DCF support, and you see the periodically collected monitoring data. In the SELECTION area, you can define the time frame you want to investigate. The most important key figures are displayed in an overview table. These figures are aggregated over the selected time period. When selecting an entry from the overview table, you get a detailed view on the selected data. You can jump into the history data and get a graphical view on this data (see Figure 5.47).

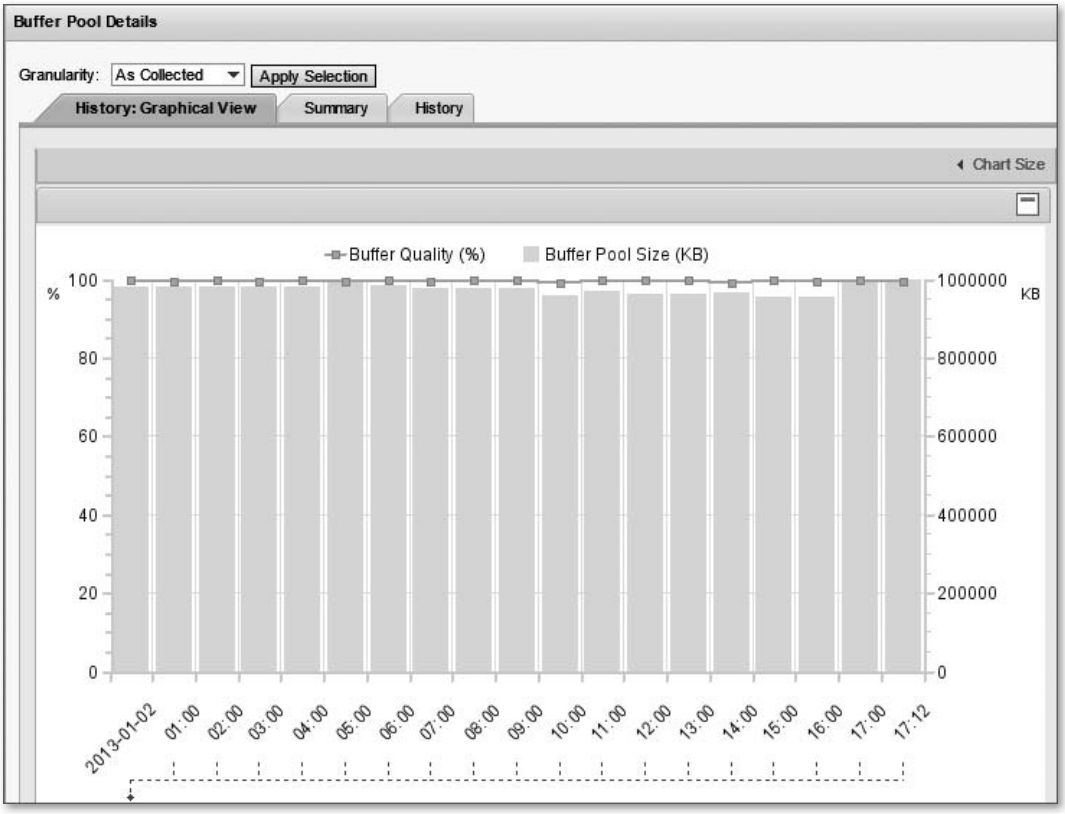


Figure 5.47 Snapshot: Buffer Pools with DCF Support

Tablespaces

To display information about each tablespace in your database, choose the TABLESPACES option. The tablespace snapshots also offer DCF support.

Snapshot: tablespace without DCF Support

When running a tablespace snapshot without DCF support (see Figure 5.48), you get an overview of your database activities. This includes the tablespace name, the member for distributed systems, the buffer quality, the average physical read and write time, the data logical reads, the data physical reads, the index logical reads, the index physical reads, and the buffer pool name.

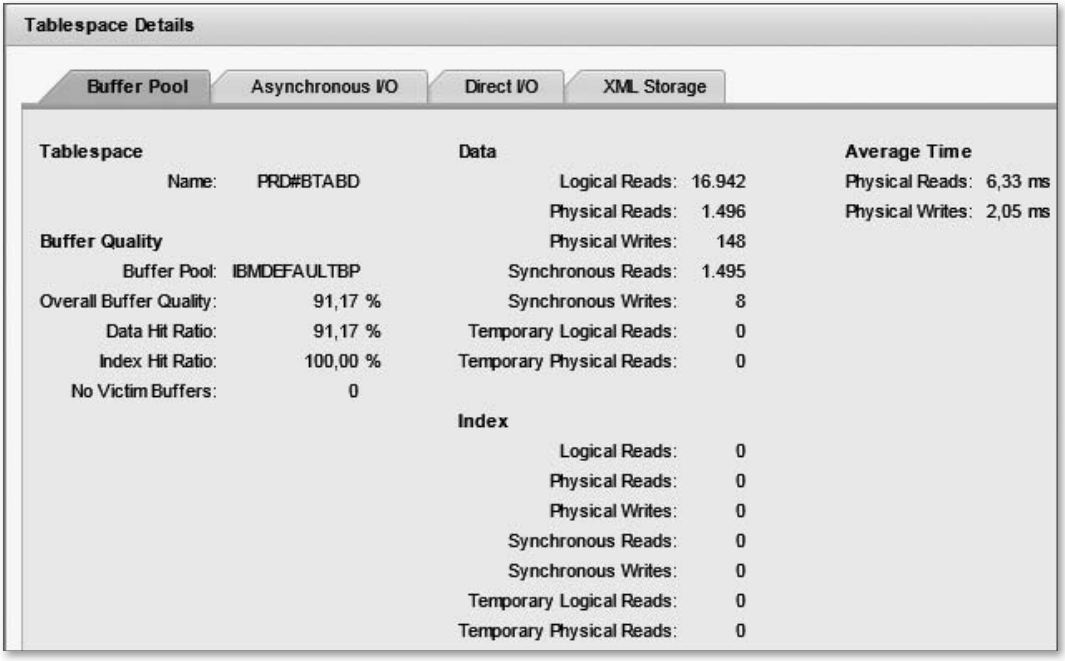


Figure 5.48 Snapshot: Tablespace without DCF Support

Snapshot: tablespace with DCF Support

When displaying the tablespace snapshot screen with DCF support, you see the current global tablespace snapshot data just as you do without DCF support, and you see the periodically collected monitoring data. In the SELECTION area, you can define the time frame you want to investigate how to sort your tablespaces and the maximum number of rows (see Figure 5.49).

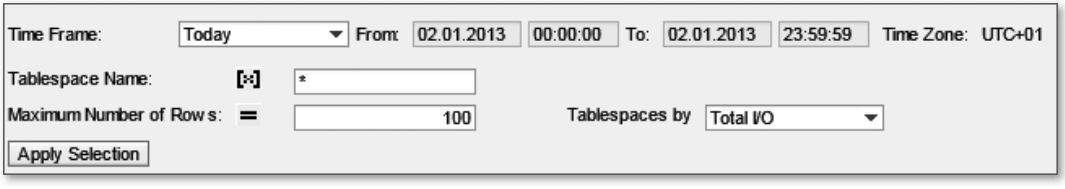


Figure 5.49 Snapshot: Tablespace Selection

The most important key figures are displayed in an overview table. These figures are aggregated over the selected time period. When selecting an entry from the overview table, you get a detailed view of the selected

data. You can jump into the history data and get a graphical view of this data (see Figure 5.50).

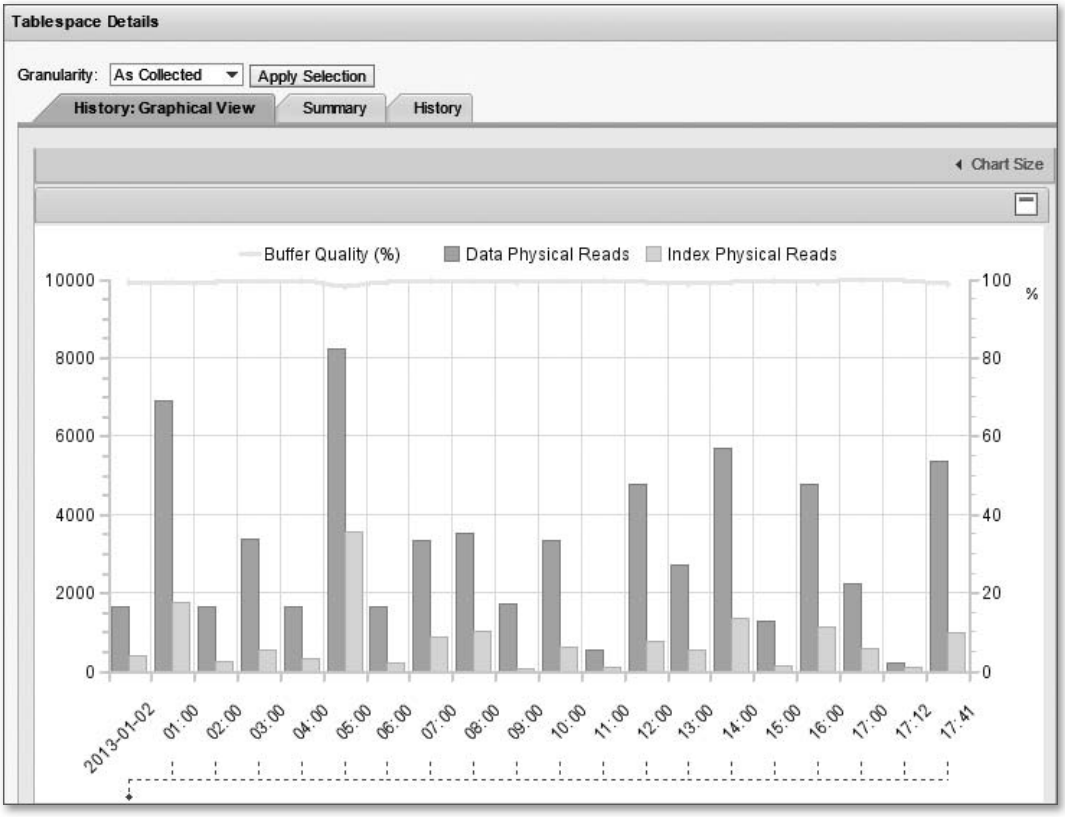


Figure 5.50 Snapshot: Tablespace with DCF Support

Container List

Snapshot: container list

The snapshot CONTAINER option is available for DB2 V9.7 or higher. Your DCF needs to be set up correctly, and your database configuration parameter `MON_OBJ_METRICS` needs to be a value other than `NONE`.

Containers are physical storage objects in which the DB2 tablespaces store their data. A tablespace can span one or many containers.

With the container list, you can gain an overview of the containers of your database. Data within the tablespaces are striped evenly across all containers. The information given in Table 5.10 is displayed.

Column	Description
Tablespace Name	Name of the tablespace.
Container Name	Name of the container.
Accessibility	Tells you if the container is accessible: 1 means that the container is accessible, and 0 mean the container isn't accessible.
Stripe Set	Stripe set of the container.
Direct Writes	Write operations not using the buffer pool of the container.
Direct Reads	Read operations not using the buffer pool of the container.
Direct Read Time	Elapsed time for read operations not using the buffer pool of the container in milliseconds.
Direct Write Time	Elapsed time for write operations not using the buffer pool of the container in milliseconds.
Est. Time per Write	Estimated time for write operations in milliseconds.
Pages Read	Number of pages read from the container.
Pages Write	Number of pages written to the container.

Table 5.10 Snapshots: Container Lists

Tables

The snapshot Tables option can be used to analyze the tables of your database. You can find out which tables are candidates for reorganization, which tables are accessed the most and need to be reorganized, or which tables have frequent update operations and need new statistics. You can use this option with or without DCF support.

Snapshots: tables without DCF Support

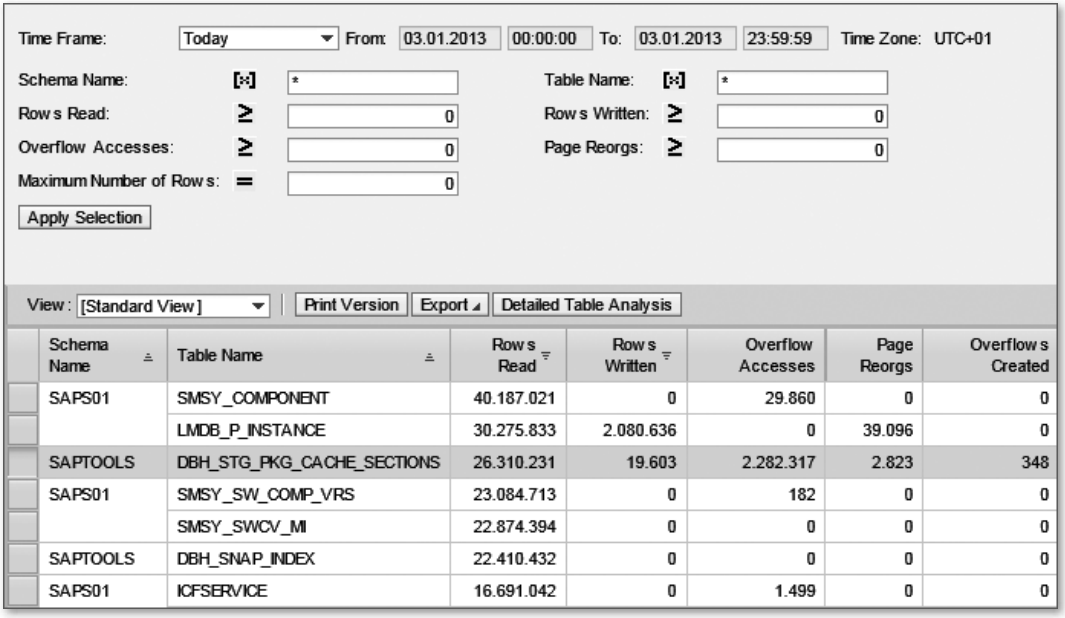
You see an overview of your database's tables and relevant figures. For each table, you can jump into Single Table Analysis by clicking the Detailed Table Analysis button. To find out more about the Single Table Analysis option, read Section 5.2.4. The information that can be displayed in this option is shown in Table 5.11.

Column	Description
Table Schema	Name of the schema.
Table Name	Name of the table.
Partition	In multi-partition databases, the number of partitions is displayed.
Rows Written	Number of changed rows by `INSERT`, `DELETE`, or `UPDATE` in the table.
Rows Read	Number of rows read from the table.
Overflow Access	Number of reads and writes to overflowed rows of the table. A high number is an indicator for a necessary reorganization.
Page REORGs	Number of page reorganizations executed for the table. A high number of these reorganizations adversely affect performance.
Overflows Created	Number of automatically created page overflows.
Table Scans	Number of executed scans, which gives you an indicator for a missing index for a table or a bad access plan.

Table 5.11 Snapshot: Tables

Snapshots: tables with DCF support

When displaying the Tables screen with DCF support, you retrieve a list of the current tables snapshot just as you do without DCF support, and you also see the periodically collected monitoring data. In the Selection area, you can define the time frame you want to investigate, the schema, the table, the number of rows read and written, the overflow access, the page reorganizations, and the maximum number of rows. The most important key figures are displayed in an overview table. These figures are aggregated over the selected time period. When selecting an entry from the overview table, you get a detailed view on the selected data. You can jump into history data and get a graphical view on this data (see Figure 5.51).



Schema Name	Table Name	Rows Read	Rows Written	Overflow Accesses	Page Reorgs	Overflows Created
SAPS01	SMSY_COMPONENT	40.187.021	0	29.860	0	0
	LMDB_P_INSTANCE	30.275.833	2.080.636	0	39.096	0
SAPTOOLS	DBH_STG_PKG_CACHE_SECTIONS	26.310.231	19.603	2.282.317	2.823	348
SAPS01	SMSY_SW_COMP_VRS	23.084.713	0	182	0	0
	SMSY_SWCV_MI	22.874.394	0	0	0	0
SAPTOOLS	DBH_SNAP_INDEX	22.410.432	0	0	0	0
SAPS01	ICFSERVICE	16.691.042	0	1.499	0	0

Figure 5.51 Snapshots: Tables

Applications

Snapshots: applications

The snapshot Applications option provides you with information about applications that are currently connected to your database. You don't see historic data but only the most recent unit of work of the selected application.

The first step is to specify the filter criteria in the Selection area from which service class you want to retrieve information. Additionally, you can say whether you want to display information about DB2 autonomic processes by selecting the Show DB2 Autonomics and Utilities checkbox. Now click the Apply Selection button.

In the Summary area (see Figure 5.52), you can set a filter on the retrieved snapshot data. For the state of an application, the following values are possible:

- Active
  The application is active and executing activities on the database.
- Inactive (having an uncommitted activity)
  Currently, the application is inactive and not executing activities on

the database. In the past, the activity was active, and some past activities have not been committed.

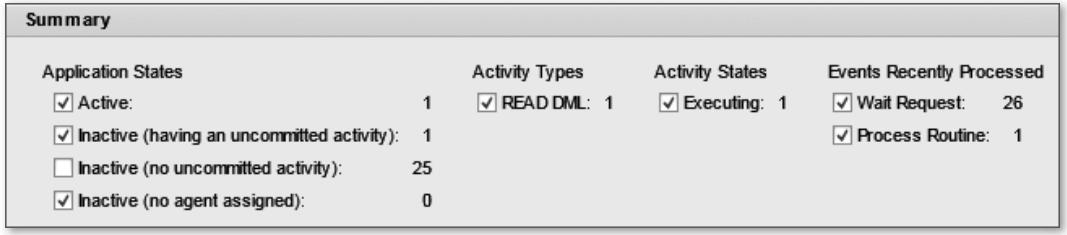


Figure 5.52 Snapshots: Applications Summary

- INACTIVE (NO UNCOMMITTED ACTIVITY)
  Currently, the application is inactive and not executing activities on the database. All activities in the past have been committed.
- INACTIVE (NO AGENT ASSIGNED)
  No agent/coordinator is assigned to the application.

Activity types, activity states, and events recently processed, are only displayed for activity types that are currently processed. For activity types, the following values are possible:

- LOAD
- READ DML
- WRITE DML
- DDL
- CALL
- OTHER

Activity states can be the following, for example:

- CANCEL PENDING
- EXECUTING
- IDLE
- INITIALIZING
- QUEUED
- TERMINATING

Some of the following events recently processed are possible:

- Process Request
- Process Routine
- Wait Request
- Acquire Lock
- Wait WLM Queue

The application information given in Table 5.12 is displayed when accessing the screen.

Column	Description
Application Handle	A unique ID for the application.
Application State	State of the application.
Application Name	Name of the application.
UOW Runtime	Total runtime of the current unit of work in milliseconds.
Entity	Type or name of the system entity.
Client PID	Process ID of the client application making the connection to the database.
Client Workstation	Client workstation (TPMON special register).
Client User ID	In an SAP ABAP system, the client user ID is the SAP user name. This ID is specified in the TPMON special register.
Client Application	The client application as specified in the TPMON special register. In an ABAP system, you use "ptype BTC" for batch work processes, "ptype SPO" for spool work processes, "ptype UPD" or "ptype UPD2" for update work processes, and the SAP transaction code for dialog work processes.
Service Superclass	Name of the service superclass to which the coordinating agent is assigned.
Service Subclass	Name of the service subclass to which the coordinating agent is assigned.

Table 5.12 Snapshots: Applications

Activity

Trace activity

If you want to start investigations on a long-running activity, you can trace, capture, or cancel the activity. By using the TRACE ACTIVITY button, you start the APPLICATION ACTIVITY TRACE in a separate web browser. In the ASSIGNED AGENTS tables, you see all of the agents that are currently assigned to the application handle. The UNIT OF WORK PROGRESS table shows collected data by the trace about the application handle. By choosing OPEN DIALOG SETTINGS, you can configure the metrics you're interested in.

Capture activity

To capture an activity, click the CAPTURE ACTIVITY button. Now go to CRITICAL ACTIVITIES – THRESHOLD VIOLATIONS (see the "Critical Activities" section). Use the filter function to find your manually captured activity more easily.

Cancel activity

To cancel an activity, click the CANCEL ACTIVITY button. If the cancellation of the activity is successful, an SQL error SQL4725N with the status SQLSTATE 57014 is returned.

SQL Cache

Snapshots: SQL cache

The SQL CACHE option allows you to investigate SQL statements that are executed very often and therefore are stored in the SQL cache of your system. You can find out which statements consume a lot of resources and if fine-tuning these statements can improve the performance of your database.

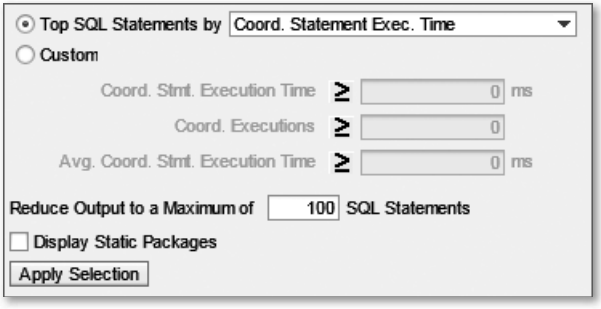


Figure 5.53 Snapshots: SQL Cache Selection

As you can see in this excerpt of the SELECTION screen (Figure 5.53), you have various options to filter your result set. Without filtering, you might get a very large, hard-to-overview result set. The TOP SQL STATEMENTS BY

dropdown box offers you different ranking criteria. The value you choose here is the first metric column in the result set. If you choose the CUSTOM radio button instead of TOP SQL STATEMENTS BY, you can specify your own filter criteria for relevant fields. By choosing the APPLY SELECTION button, the information is displayed as detailed in Table 5.13.

Column	Description
STATEMENT TEXT	Text of the SQL statement
RANKING METRIC	Default: Coord. Statement Exec. Time = Execution time by coordinator agent. This metric column is used for ranking. You choose the metric to use in the SELECTION area.
NO. OF COORD. EXECUTIONS	Number of executions by coordinator agent.
AVG. COORD. STMT. EXEC. TIME	Average execution time for the statement in milliseconds.
TOTAL CPU TIME	Total CPU time in milliseconds.
ROWS READ/ROWS PROCESSED	Ratio of rows read from the base table compared to the rows processed (`SELECT`, `UPDATE`, `INSERT`, or `DELETE` statement). A high value indicates statements with inefficient access; the value `1` stands for optimal access.
BP GETS/ROWS PROCESSED	The average number of pages read from the buffer pool per rows processed (`SELECT`, `UPDATE`, `INSERT`, or `DELETE` statement).

Table 5.13 Snapshots: SQL Cache

To get detailed information about an SQL statement, select the relevant line from the result list. You'll see an SQL STATEMENT and a STATEMENT METRICS tab as shown in Figure 5.54.

SQL statement

On the SQL STATEMENT tab, you can see the SQL statement text and the compilation environment. The access plan of the statement can be accessed by clicking the EXPLAIN button. (For more details, see the "EXPLAIN Access Plan" section.) To jump into the ABAP source code, use the SHOW SOURCE button. You can choose between opening it in an SAP GUI or a web frontend.

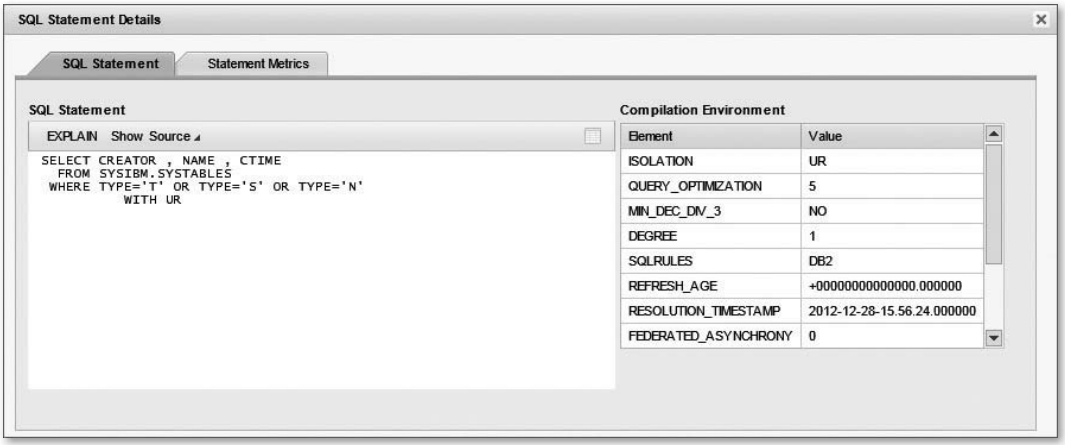


Figure 5.54 Snapshots: SQL Cache and SQL Statement Tabs

The Statement Metrics tab displays the chart view (see Figure 5.55) Time Spent in SQL Statement. In the same area, you can also find the following information:

Statement metrics

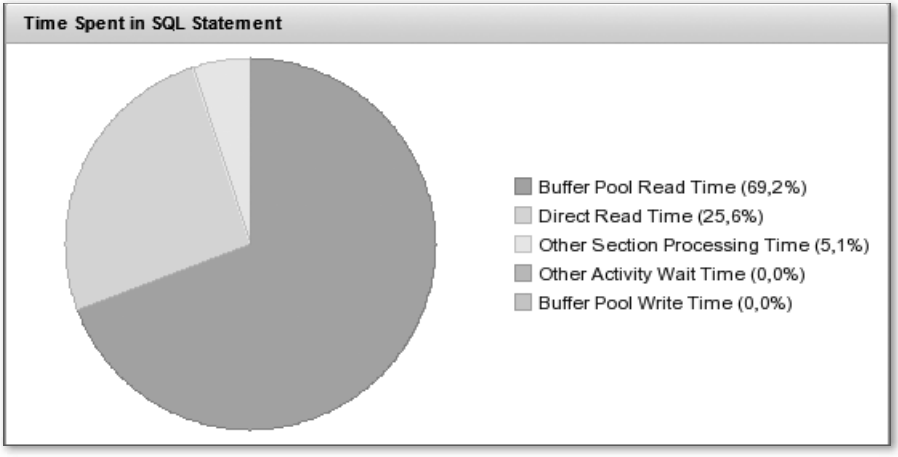


Figure 5.55 Snapshot: SQL Cache Statement Metrics

- Statement metadata
- Miscellaneous metrics
- Sorts
- Locking

- Logger
- Buffer pool
- Direct I/O

System Resources

Snapshots: system

To get an overview of your system resources, navigate to the Snapshots – System Resources option. On this screen, you find the information shown in Table 5.14 for every currently monitored database system.

Column	Description
Host Name	Name of the host.
Operating System	Operating system name.
Members	Members located on this host.
CPU Usage	Current CPU usage on the host in percent.
CPU Load Short	In UNIX systems, you see this information. It's a short-period CPU load indicator; for example, it's loading samples for the past 5 minutes.
CPU Load Medium	This information is also only available on UNIX systems. It's a medium-period CPU load indicator; for example, it's loading samples for the past 10 minutes.
CPU Load Long	This information is also only available on UNIX systems. It's a long-period CPU load indicator; for example, it's loading samples for the past 15 minutes.
CPUs Online	Number of CPUs currently online.
Memory Total	Total size of physical memory in megabytes.
Memory Free	Free physical memory in megabytes.

Table 5.14 Snapshots: System

Critical Activities

The Critical Activities task area lets you define thresholds on resources to gain information or cancel database activities that are behaving abnormally. These thresholds can be controlled proactively before the action begins or reactively while the action is running and using resources. The

activities Thresholds Violations and Threshold Configuration are available.

Threshold violations

In the Threshold Violations screen, you gain information to analyze threshold violations that occur in your database. You can specify a certain time frame and for this time frame the information shown in Table 5.15 is displayed.

Column	Description
Violation Time	Time the threshold violation occurred
Member	The member where the violation occurred
Predicate	Predicate that was violated
Violated Value	Value that violated and exceeded the threshold predicate
Service Superclass	Service superclass in which the violation occurred
Service Subclass	Service subclass in which the violation occurred

Table 5.15 Critical Activities: Threshold Violations

To display detailed information about a threshold, click on the line. You'll then get a tab with general information about the activity that violated the threshold.

You'll also see a tab called SQL Statement, which displays the executed SQL statement that is part of the activity. To dive deeper into the access plan of the SQL statement, choose EXPLAIN and the access plan will open in a separate window (for more details, see the "EXPLAIN Access Plan" section).

Threshold configuration

All defined thresholds for your database system are displayed on the Threshold Configuration screen. Here you can enable or disable your threshold, create a new threshold, or drop an existing threshold.

Utilities

You can access the Database Utilities, Database Utility History, and Inplace Table Reorganization options by choosing Performance • Utilities and then picking one of the options as shown in Figure 5.56.

Figure 5.56 Utilities Options

We'll explain each of these options in the following subsections.

Database Utilities

By clicking the DATABASE UTILITIES option, you can monitor the influence of utilities on your database performance. Depending on the utility, the performance can be influenced by high physical I/O or by locking situations of database resources. Following are examples of different type of utilities:

- ASYNC_INDEX_CLEANUP
- BACKUP
- CRASH_RECOVERY
- RESTART_RECREATE_INDEX
- RESTORE
- ROLLFORWARD_RECOVERY
- RUNSTATS

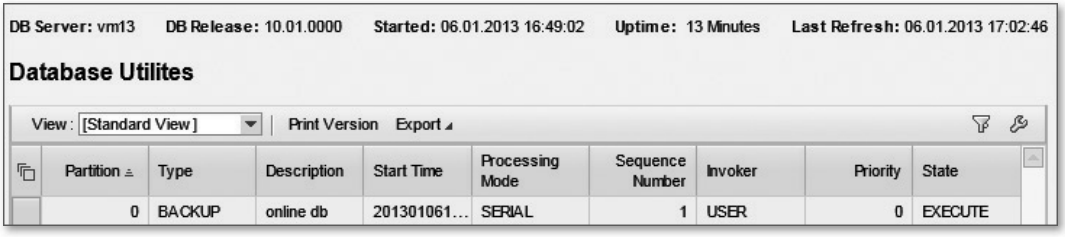


Figure 5.57 Utilities: Database Utilities

After running utilities as shown in Figure 5.57, the information given in Table 5.16 is displayed.

Column	Description
Partition	Partition number on which the utility is running
Type	Type of utility
Description	What the utility is working on
Start Time	Date and time the utility started
Processing Mode	The mode of the utility
Sequence Number	The current phase the utility is working on
Invoker	Whether the utility was started automatically or manually by the user
Priority	The relative importance of a throttled utility
State	The current state of the utility

Table 5.16 Utilities: Database Utilities

In the content detail area, you'll find an overview table with further information concerning whether the processing mode is concurrent or serial. The following processing modes are possible:

- Concurrent
  Any element of the progress list can be updated at any time due to concurrent processing.
- Serial
  The elements of the list where one task needs to be fully completed before the next task starts.
- <space>
  Only one phase is used without parallelism.

In this overview table that displays the running phases, you find information about the partition, the start time, the sequence number, the action description, the progress in percent, the number of processed and total units, metrics of the work units, and the current execution state of the utility.

Database Utility History

The Database Utility History screen allows you to get an overview of all of the logged database utility operations for a certain time frame (see Figure 5.58). With this option, you can analyze operations that were

performed on your database system in this time frame. To use this option, you must have set up the DCF correctly.

When you access this option the starting and ending time of the event, the type of utility and of the affected object, the schema and the name of the affected, the type of utility operation and the invoker that indicated if the utility was started automatically or manually is displayed.

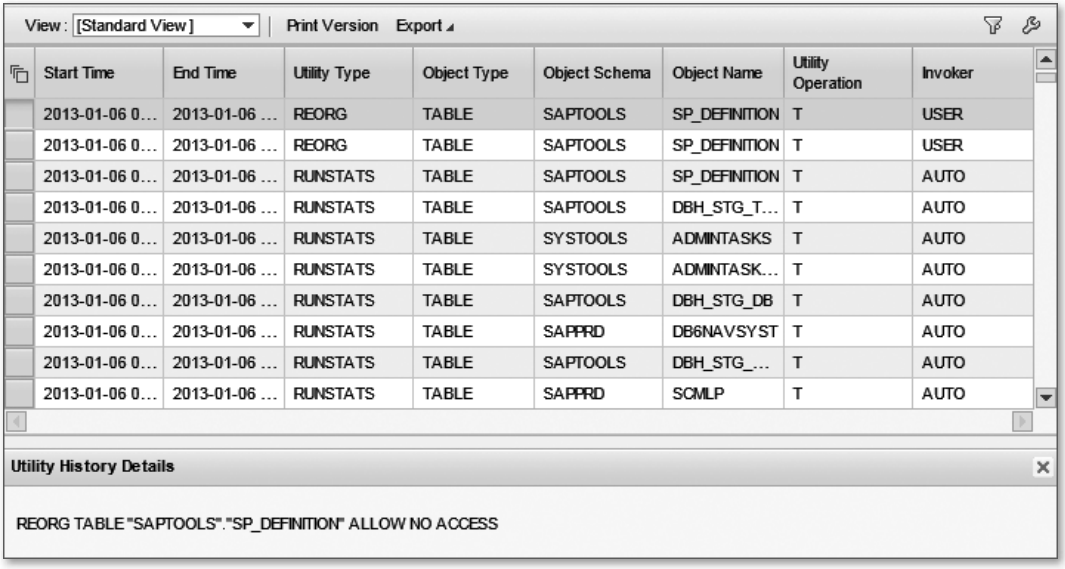


Figure 5.58 Utility History Details Area

Inplace Table Reorganization

In the Inplace Table Reorganization screen, you can get an overview of currently running and interrupted inplace table reorganizations. This allows you to access the tables while they are being reorganized.

For an inplace reorganization, you can perform the following actions:

- Pause
  Pause a running inplace reorganization.
- Resume
  Resume a previously paused inplace table reorganization.

- Suspend
  Suspend a table reorganization.

The overview shown when choosing the Inplace Table Reorganization option displays the information given in Table 5.17.

Column	Description
Table Schema	Name of the table schema of the table that is currently being reorganized
Table Name	Name of the table
Partition	Number of the partition
REORG Status	Status of the inplace table reorganization
Progress	Progress of the reorganization in percent
Start Date	Start date of the reorganization
Start Time	Start time of the reorganization
Access Mode	Displays the access mode for other users trying to access the table while reorganization is taking place
Tablespace	Name of the affected tablespace

Table 5.17 Utilities: Inplace Table Reorganization

History

As an administrator, you should take action to prevent potential problems before they occur. Historical trends help you make proactive analysis easier. When you navigate to Performance • History, you can access the history of either your database or your tables (see Figure 5.59).

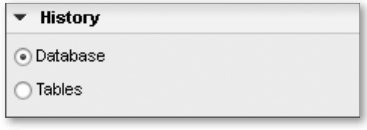


Figure 5.59 History: Options

History: database

To see historic data, you first need to configure your database to collect this information (see Section 5.2.6). After everything is configured, your

database provides a day-by-day trend analysis of your database activities when you click on the Database option. For each day, the values shown in Table 5.18 are monitored.

Column	Description
Partition	In a multi-partition database number of the monitored partition
Date	Date of the monitoring
Avg. Phys. Read Time	Average physical read time
Avg. Phys. Write Time	Average physical write time
Data Logical Reads	Read accesses to data in the buffer pool
Data Physical Reads	Read accesses to data on disc (I/O)
Data Physical Writes	Write accesses to data on disc (I/O)
Index Logical Reads	Read accesses to index data in the buffer pool
Index Physical Reads	Read accesses to index data on disc (I/O)
Index Physical Writes	Write accesses to index data on disc (I/O)
COMMIT Statements	Number of `COMMIT` statements
ROLLBACK Statements	Number of `ROLLBACK` statements
Lock Waits	Number of times applications or connections waited for locks
Lock Wait Time	Elapsed time applications or connections waited for locks
Deadlocks	Number of deadlocks
Lock Escalations	Number of locks that have been escalated from several row locks to a table lock
Exclusive Lock Escalations	Number of locks that have been escalated from several row locks to an exclusive table lock

Table 5.18 Performance: History Database

To find out more details about a specific day, select the row, and a details area appears with a Snapshot and an Interval tab. Measured values from the selected day are displayed in the Snapshot tab. The Interval tab shows the delta values of the measurements provided under Snapshot.

This provides you with a day-to-day analysis of your table activities (see Figure 5.60).

History: tables

Date	Table Schema	Table Name	Rows Written	Rows Read
05.01.2013	SYSIBM	SYSINDEXES	0	18.446.744.073.710.840.893
		SYSROUTINES	0	18.446.744.073.710.270.025
		SYSTABLESPACES	0	18.446.744.073.709.614.628
		SYSNODEGROUPDEF	0	18.446.744.073.709.562.260
	SAPPRD	DDNTT	0	9.223.372.036.855.243.775
		DDNTF	0	9.223.372.036.854.813.557
	SAPTOOLS	DBH_STG_PKG_CACHE_SECTIONS	0	81.277.201
	SAPPRD	E071	0	29.817.713
	SAPTOOLS	DBH_STG_CFG_DB	0	28.674.036
	SYSIBM	SYSTABLES	0	26.886.527
	SAPPRD	DD25L	0	24.955.066
	SAPTOOLS	DBH_STG_CFG_DBM	0	18.774.189
		DBH_STG_INDEX	0	14.347.285
		DBH_SNAP_PKG_CACHE_METRICS	0	11.556.533
	SAPPRD	DD03L	0	10.776.388
	SAPTOOLS	DBH_STG_PKG_CACHE_METRICS	0	9.529.928
		DBH_STG_CFG_REG	0	8.280.051
	SYSTOOLS	HMON_ATM_INFO	0	8.051.930
	SAPPRD	DD04L	0	7.689.155

Figure 5.60 Performance: History Tables

In this trend analysis, the information shown in Table 5.19 is displayed.

Column	Description
Date	Date of the monitored data
Table Schema	Name of the schema
Table Name	Name of the table
Partition	In a multi-partition database number of the monitored partition
Rows Written	Numbers of rows written
Rows Read	Numbers of rows read
Overflow Accesses	Numbers of read accesses that caused an overflow
Page REORGs	Numbers of internal page reorganizations during `INSERT`

Table 5.19 Performance: History Tables

5.2.4 Space

Space and storage are very important aspects of your database performance configuration. In general, the disk I/O is the slowest part of your machine. This can be the bottleneck if you have a poor data and storage layout. In the Space section, you'll see the following options:

- Overview
- Automatic Storage
- Database
- Tablespaces
- Containers
- File Systems
- Tables and Indexes
- Single Table Analysis
- Performance Warehouse

In the following subsections, we'll explain each of these options and how to manage them to provide optimal speed.

Overview

The Overview option gives you general information about the space of your database. The Databases and Tablespaces and Tables and Indexes tab pages are available.

Database and Tablespaces tab

As shown in Figure 5.61, on the Database and Tablespaces tab, you see the date and time of the last analysis, the total number, the total size, the free space, the used space, the minimum used space, and the maximum used space in a tablespace.

Tables and Indexes tab

On the tab Tables and Indexes tab (see Figure 5.62), you also see the date and time of the last analysis, the total number of tables, the total amount of used space, the total number of indexes, the total size of indexes, the oldest REORG check, and the latest REORG check.

Figure 5.61 Space: Database and Tablespaces Tab of the Overview Option

Figure 5.62 Space: Tables and Indexes Tab of the Overview Option

Automatic Storage

If you've enabled automatic storage management during the SAP system installation, or if you have activated it later, this function is available. Automatic storage simplifies the storage management for table spaces. To find out more details about automatic storage, please refer back to Chapter 3, Section 3.4.1.

In this screen you can access information about the automatic storage file systems of your database. You can add, change, or delete the storage paths by selecting the line and clicking the corresponding button. For more background information about storage groups, refer to Section 3.4.2. Figure 5.63 shows an example of how to change your storage paths.

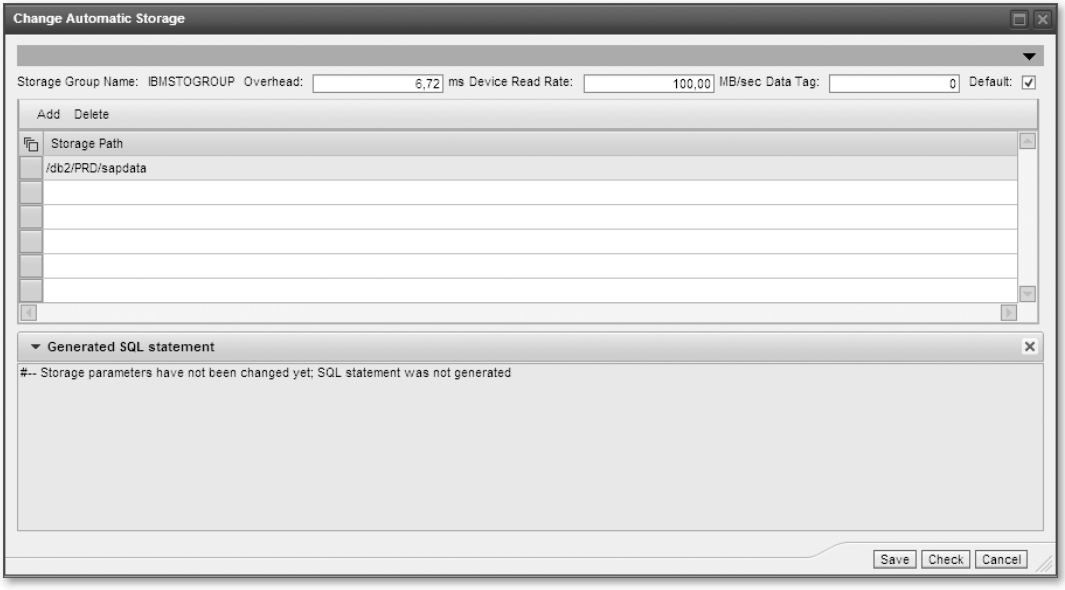


Figure 5.63 Automatic Storage: Changing the Storage Path

For each of your storage paths, the information shown in Table 5.20 is displayed.

Column	Description
Storage Group Name	Name of the storage group
Storage Path	Full path name
FS ID	ID of the related file system
Total Size (GB)	Total size available in the file system
Used Size (GB)	Use size in the file system
Available Size (GB)	Free size available in the file system
Overhead (ms)	Specifies the I/O controller usage and disk seek and latency time
Device Read Rate (MB/sec)	Specifies the device specification for the read transfer rate in megabytes per second

Table 5.20 Automatic Storage

By selecting a line, you can jump into the automatic storage details.

Database

Monitor space

By choosing Space/Database, you can monitor the space that is consumed by your database. Not only can you check the already-consumed space, but you can also consider the progress of space growth. With this information, you can predict whether your database is running out of space and your archiving and reorganization operations are successful, and you can identify short-term growth triggered by specific activities.

To analyze your database, first identify the time period you want to investigate. In the Selection area, you specify the time frame and apply the selection. To dive deeper into historic data, select a line, and the History Details area is displayed. Here you can analyze the space growth in more detail.

Tablespaces

To gain information and alter your tablespaces, use the tablespaces tools. Depending on how you set up your tablespaces, you can access the Automatic Storage and/or DMS/SMS Tablespaces tabs. Automatic Storage is available if you enabled automatic storage management (see Section 3.4.2) for the tablespace. If you have chosen manual maintenance of your tablespace, the DMS/SMS TABLESPACES tab is available (for more information on DMS/SMS tablespaces, see Section 3.4).

On the tabs, the values shown in Table 5.21 are displayed.

Column	Description
Tablespace Name	Name of the tablespace
Partition	Number of the partition (only displayed if you use a multi-partition database)
TS Type	Type of tablespace (only with DMS/SMS)
Contents	Contents of tablespaces
TS State	Status of the tablespace
KB Total	Total space used by the tablespace

Table 5.21 Tablespaces

Column	Description
Page Size (KB)	Size of a page
No. Containers	Number of containers
KB Free	Amount of free space
High-Water Mark (KB)	Maximum value of reached used pages
Percent Used	Used space in relation to available space
AUTORESIZE	Indicator if the tablespace is enabled for automatic resizing (only with DMS/SMS)
Pending Free Pages	Number of free pages in the status pending

Table 5.21 Tablespaces (Cont.)

To get more detailed information about tablespace, click a line of the overview table (see Figure 5.64).

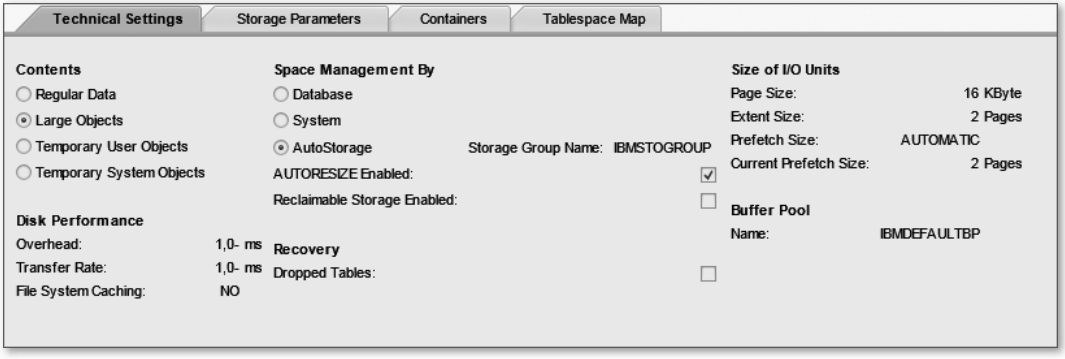


Figure 5.64 Tablespaces: Details Area

In the details area, you find some general information about the tablespace such as the name, the database partition group, the total space, the used space in percent, and the free space.

Technical Settings tab

The Technical Settings tab provides the information shown in Figure 5.64. The different radio button options are as follows:

- Contents
  This area includes options that describe the type of data stored in the tablespace. This can be Regular Data, Large Objects, Temporary User Objects, or Temporary System Objects.

- Space Management by
  This area shows whether the tablespace containers are managed by the Database (DMS), the file system (SMS), or if the automatic storage management is enabled. If automatic storage management is enabled, the AUTORESIZE checkbox is selected by default. This marked checkbox means that tablespace containers are automatically extended using the space in the file system where they are located. The Reclaimable Storage Enabled checkbox indicates that the tablespace is enabled for reclaimable storage. Unused extents can be released to the system for reuse.
- Size of I/O Units
  In this area, Page Size, Extent Size, and Current Prefetch Size are displayed.
- Disk Performance
  In this area, you can monitor disk performance. You get information about the Overhead, Transfer Rate, and File System Caching.
  - Overhead displays I/O controller overhead, disk seek, and latency time in milliseconds.
  - Transfer Rate displays the time needed to read one page into memory in milliseconds. These two values can be used to determine the cost of I/O during query optimization.
  - File System Caching indicates how you use the file system caching. YES indicates that file system caching has been explicitly enabled, and you use the `FILE SYSTEM CACHING` clause of the `CREATE` or `ALTER TABLESPACE` statement. NO says that file system caching has been explicitly disabled. OPERATING SYSTEM indicates that the default of the file system is used, and no explicit file system caching clause has been specified.

Additional Information

If you need further information to decide if you should turn off the file system caching, refer to SAP Note 1353421 (DB6: How to disable file system caching for tablespaces).

Storage Parameters tab

Values for tablespaces that have AUTORESIZE enabled or that are completely managed by automatic storage management are shown on the

Storage Parameters tab. Under Settings, you see the Initial Size allocated when the tablespace is created and you see how the database increases the tablespace whenever it gets full (see Figure 5.65). This can be done using an automatically chosen value: Automatic, Absolute by a certain size, or Relative by a certain set percentage. It's also possible to set the maximum size of the tablespace.

Figure 5.65 Tablespaces: Storage Parameters

Under Size, you see the current size. If there was a resize, you see the date and time of the last automatic resize operation. If the last automatic resize operation failed, the SQL error is displayed in the lower half of the screen.

Containers tab

The Containers tab displays information about the containers of the selected tablespace.

Tablespace Map tab

The Tablespace Map tab gives you a simple graphical overview of the tablespace. You can see size all of containers and how they are allocated to stripe sets.

This overview helps you find out if data is evenly distributed if you have multiple containers. If you're planning to expand storage space, you can figure out how a container or stripe set can be added without incurring the cost of rebalancing the containers.

Maintaining tablespaces

On the Tablespace screen, you can maintain your tablespaces, change tablespaces and container settings, convert regular to large tablespaces, rebalance tablespaces, start and stop the reduction of the high-water mark, add new tablespaces, and delete tablespaces.

To delete a tablespace, select the tablespace and choose the DELETE button. Be aware that you can only delete tablespaces that aren't in use by the SAP system. If a tablespace is obsolete and should be deleted, you have to delete the corresponding data class before you can delete the tablespace.

To change a tablespace, select the tablespace and click the CHANGE button.

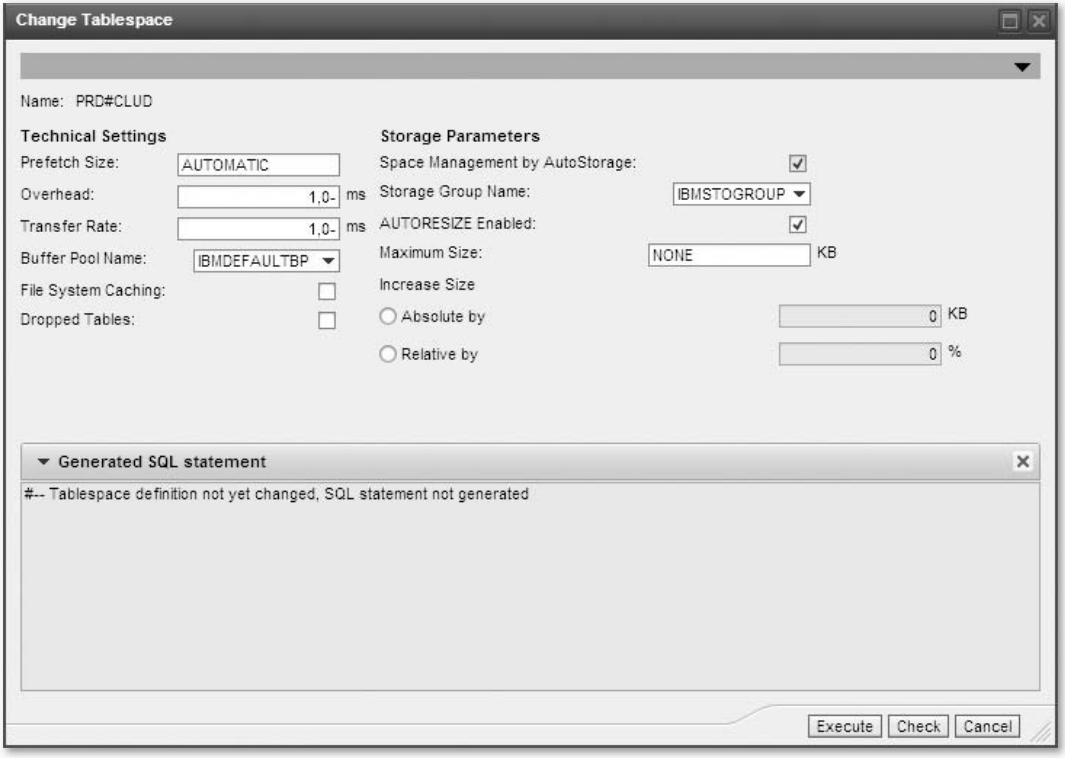


Figure 5.66 Tablespaces: Change

In the CHANGE TABLESPACE (see Figure 5.66) dialog box you can change the following settings:

Change tablespace

- TECHNICAL SETTINGS
  - PREFETCH SIZE
    Numbers of pages to be prefetched.
  - OVERHEAD
    I/O controller overhead and disk seek and latency time in milliseconds.

- Transfer Rate
  Time to read one page.
- Buffer Pool Name
  Name of the allocated buffer pool.
- File System Caching
  Activates or deactivates the file system caching.
- Dropped Tables
  Using this checkbox you can enable the opportunity to recover dropped tables using the `RECOVER DROPPED TABLE ... TO` option of the `ROLLFORWARD DATABASE` command. Dropped table recoverability is not a standard setting of tablespaces in SAP systems.

- Storage Parameters
  - Space Management by Auto Storage
    Activate Automatic Storage for a tablespace that isn't yet automatically managed.
  - Storage Group Name
    Assign a new storage group here (available as of DB2 V10.1).
  - AUTORESIZE Enabled
    Allows DB2 to automatically enlarge tablespace containers in the file system where they reside. This option can be activated/deactivated online for DMS tablespaces.
  - Maximum Size
    Enables you to set an absolute value in kilobytes that will not be exceeded by automatic extensions. If, as in Figure 5.66, you use the value NONE, there is no maximum size limit. Containers will be extended until the file system is completely occupied.
  - Increase Size
    Enables you to specify the size in kilobytes or in percent by which the tablespace will be extended if it's full.
  - Containers
    Optional field that appears if your tablespace isn't managed by automatic storage management. You can add or delete containers. Be aware that adding or changing containers might have a heavy impact on your system performance because it can cause a rebalancing.

Add tablespace

When adding a tablespace (choose the ADD button), you have to set a name for your tablespace (see Figure 5.67). Then you can define the CONTENTS, the SIZE OF I/O UNITS, the DISK PERFORMANCE, the SPACE MANAGEMENT, and the AUTORESIZE parameters. For more details, review the information on changing tablespaces, or for background information go to Section 3.4.

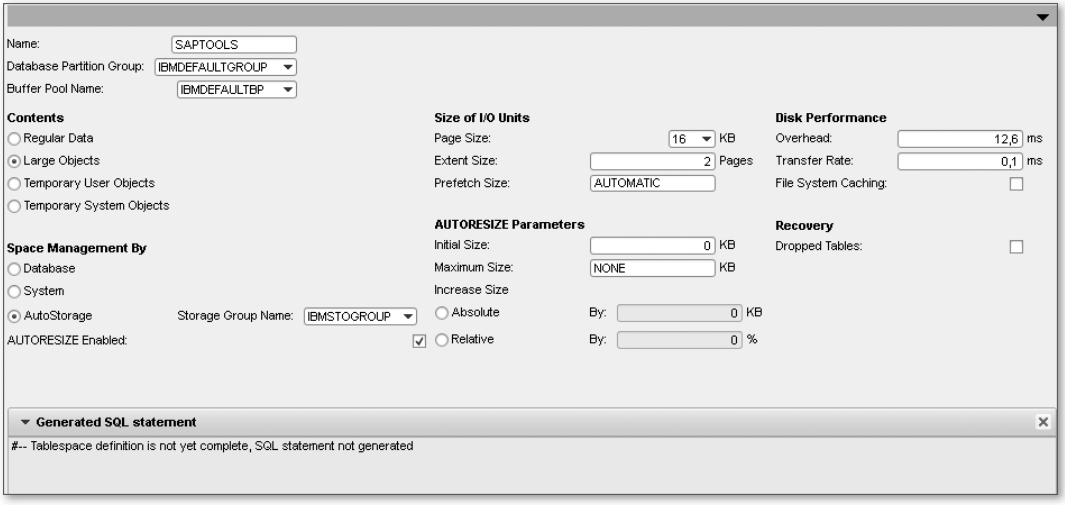


Figure 5.67 Tablespaces: Add

Containers

The CONTAINERS page is very similar to the TABLESPACES page. The overview page shows the tablespaces and the corresponding containers. From here, you can change, add, and delete tablespaces containers. The information in Table 5.22 is displayed on the CONTAINERS OVERVIEW screen.

Column	Description
TABLESPACE NAME	Name of the tablespace
PARTITION	In a multi-partition database, the number of the monitored partition
STRIPE SET	The number of the strip set the container belongs to
CONTAINER NAME	Name of the container in which the tablespace data is stored

Table 5.22 Containers

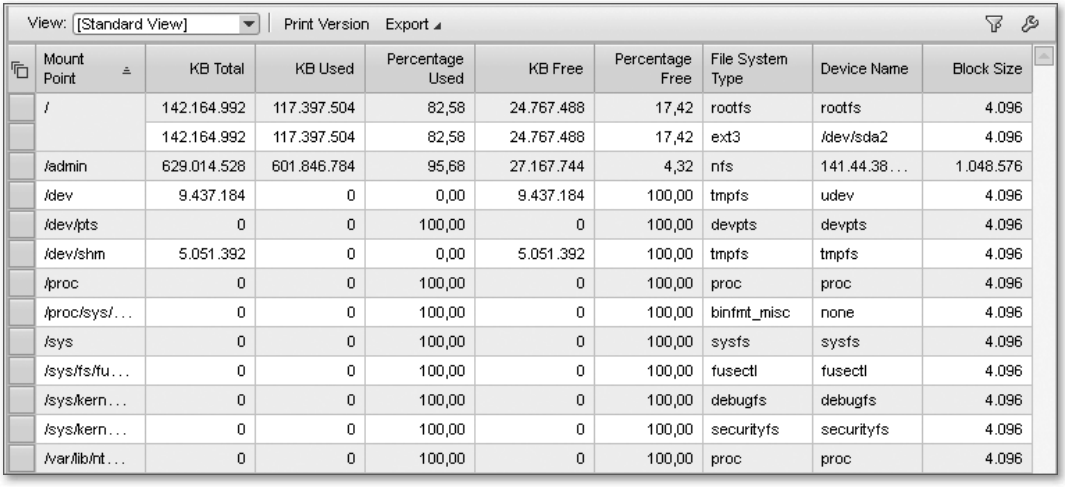
Column	Description
Type	Type of the container
KB Total	Total size of the container
Pages Total	Total amount of pages
Accessible	Indicates whether the container is accessible or not
FS ID	File system ID
FS Free Size (GB)	Free space in the file system

Table 5.22 Containers (Cont.)

File Systems

The File Systems function is only available if you aren't using the remote data connection.

As shown in Figure 5.68, this function lets you access information about the file systems. You can find out how much free space is available in your file system, which you would need to know if you wanted to extend tablespaces, for example. The information shown in Table 5.23 is displayed.



Mount Point	KB Total	KB Used	Percentage Used	KB Free	Percentage Free	File System Type	Device Name	Block Size
/	142.164.992	117.397.504	82,58	24.767.488	17,42	rootfs	rootfs	4.096
	142.164.992	117.397.504	82,58	24.767.488	17,42	ext3	/dev/sda2	4.096
/admin	629.014.528	601.846.784	95,68	27.167.744	4,32	nfs	141.44.38...	1.048.576
/dev	9.437.184	0	0,00	9.437.184	100,00	tmpfs	udev	4.096
/dev/pts	0	0	100,00	0	100,00	devpts	devpts	4.096
/dev/shm	5.051.392	0	0,00	5.051.392	100,00	tmpfs	tmpfs	4.096
/proc	0	0	100,00	0	100,00	proc	proc	4.096
/proc/sys/...	0	0	100,00	0	100,00	binfmt_misc	none	4.096
/sys	0	0	100,00	0	100,00	sysfs	sysfs	4.096
/sys/fs/fu...	0	0	100,00	0	100,00	fusectl	fusectl	4.096
/sys/kern...	0	0	100,00	0	100,00	debugfs	debugfs	4.096
/sys/kern...	0	0	100,00	0	100,00	securityfs	securityfs	4.096
/var/lib/nt...	0	0	100,00	0	100,00	proc	proc	4.096

Figure 5.68 File Systems

Column	Description
Mount Point	Location in the operating system directory structure where the file system appears
KB Total	Size of the file system
KB Used	Amount used on the file system
Percentage Used	Percentage of used space of the total size in the file system
KB Free	Amount free on the file system
Percentage Free	Percentage of free space of the total size in the file system
File System Type	Type of the file system
Device Name	Name of the device
Block Size	Block size used in the file system

Table 5.23 File Systems

Tables and Indexes

To access data about your tables and indexes, choose Tables and Indexes, and then choose the Fitting option. In these areas, you can find the largest indexes and tables, as well as information on which tables and indexes to reorganize regarding space reclaim, performance degradation, and where data compression could make sense.

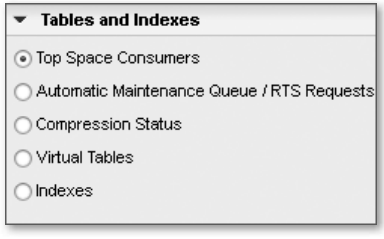


Figure 5.69 Tables and Indexes: Options

The following options displayed in Figure 5.69 are available:

- Top Space Consumers
  Used to identify the top space consumers in your database for a specified time period.

- Automatic Maintenance Queue/RTS Requests
  Displays information about objects that are in the queue for automatic maintenance (including RUNSTAT runs) or have a real time statistics request.
- Compression Status
  Allows you to check the compression status of a table or determine whether a table is a suitable candidate for compression.
- Virtual Tables
  Used to check whether or not a table can be virtualized or materialized to save space.
- Indexes
  Provides an overview of all indexes in your database.

We'll go intro greater detail about these screen options in the following subsections.

Top Space Consumers

The Top Space Consumers screen is used to access information about the largest tables of your database. A prerequisite for this is that you have the DCF set up correctly.

The screen provides a selection area where you can specify filter criteria for the collected data, an overview table (see Figure 5.70) displaying the retrieved data, and a detail area where you can get more information about the collected historic data during the selected time frame.

View: [Standard View] | Print Version | Export | Detailed Table Analysis

Partition	Schema Name	Table Name	Data Tablespace	Total Table Size (KB)	Total Growth (KB)	Log. Index Size (KB)	No. of Indexes	Log. Data Size (KB)	Log. XML Size (KB)	Log. Long Data Size (KB)
0	SAPPRD	REPOSRC	PRD#E...	2.090.432	0	76.160	12	235.584	0	0
0	SAPPRD	WDR_...	PRD#S...	594.048	0	1.024	6	2.464	0	0
0	SAPPRD	DOKCLU	PRD#E...	305.952	0	21.408	6	284.544	0	0
0	SAPPRD	REPOL...	PRD#E...	284.992	0	1.152	6	2.848	0	0
0	SAPPRD	DD03L	PRD#E...	220.960	0	149.440	48	71.520	0	0
0	SAPPRD	SMIMC...	PRD#B...	193.056	0	7.328	6	185.728	0	0
0	SAPPRD	E071	PRD#S...	181.056	0	78.944	12	102.112	0	0

Figure 5.70 Tables and Indexes: Top Space Consumers

Automatic Maintenance Queue/RTS Requests

The Automatic Maintenance Queue/RTS Requests tab is new as of DB2 10.1. In the Selection area, you can choose the maximum number of rows. The Automatic Maintenance Queue tab displays information about all queued and running automatic maintenance jobs for all members. To find out where you plan and maintain automatic maintenance, check Section 5.2.6.

The Runstats Candidates tab displays objects that are candidates for the `RUNSTATS` command. This command updates statistics about the characteristics of a table and/or statistical views or associated indexes.

Under the tab Request for Real Time Statistics (RTS) you find information about all real-time statistics requests that are currently being processed and that are pending in the system.

Compression Status

The Compression Status screen provides information about already-compressed tables or about tables that are candidates for compression. On the Compressed Tables tab, you see tables and indexes that are enabled for compression and that are compressed (see Figure 5.71).

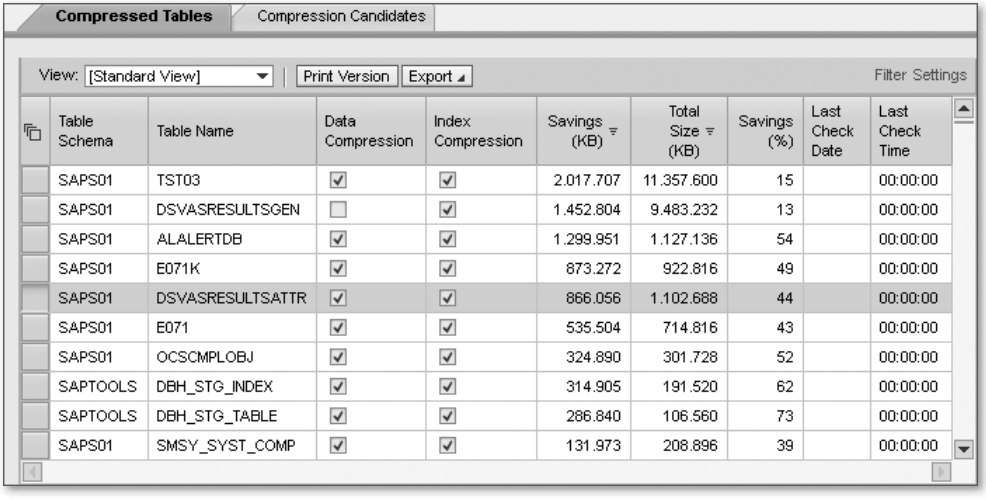


Table Schema	Table Name	Data Compression	Index Compression	Savings (KB)	Total Size (KB)	Savings (%)	Last Check Date	Last Check Time
SAPS01	TST03	☑	☑	2.017.707	11.357.600	15		00:00:00
SAPS01	DSVASRESULTSGEN	☐	☑	1.452.804	9.483.232	13		00:00:00
SAPS01	ALALERTDB	☑	☑	1.299.951	1.127.136	54		00:00:00
SAPS01	E071K	☑	☑	873.272	922.816	49		00:00:00
SAPS01	DSVASRESULTSATTR	☑	☑	866.056	1.102.688	44		00:00:00
SAPS01	E071	☑	☑	535.504	714.816	43		00:00:00
SAPS01	OCSCMPLOBJ	☑	☑	324.890	301.728	52		00:00:00
SAPTOOLS	DBH_STG_INDEX	☑	☑	314.905	191.520	62		00:00:00
SAPTOOLS	DBH_STG_TABLE	☑	☑	286.840	106.560	73		00:00:00
SAPS01	SMSY_SYST_COMP	☑	☑	131.973	208.896	39		00:00:00

Figure 5.71 Tables and Indexes: Compression Status

The Compression Candidates tab page includes the following:

- Tables that have already been compressed but more space could be saved by recompressing it based on a better compression dictionary
- Tables that have been enabled for compression but contain records that aren't compressed yet
- Tables with indexes that aren't yet enabled for index compression
- Tables with compressed indexes, where you can save more space by recompressing the table
- Already classically compressed tables that save space using the new adaptive compression

On the Compressed Tables and the Compression Candidates overview tabs shown in Figure 5.71, that data explained in Table 5.24 is displayed.

Column	Description
Table Schema	Name of the table schema.
Table Name	Name of the table.
Data Compression	Indicates if the table is enabled for data compression.
Index Compression	Indicates if the indexes are enabled for index compression.
Savings (MB)/ Estimated Savings (MB)	Total savings caused by data or index compression/Total savings caused by data or index compression that can be archived.
Total Size (MB)	Total size of the table/if the compression would be performed.
Savings (%)/Estimated Savings (%)	Percentage of the savings/Percentage of the estimated savings.
Last Check Date	Last check date.
Last Check Time	Last check time.

Table 5.24 Compressed Tables

By choosing a line of the overview table, you can display details about the compression status. Here you can see the current saving and the estimated saving when compressing the database. These estimated savings are calculated when running the compression check. A compression check can be started by using the action button Start Compression Check... You

can start the check immediately in the background or schedule it. After the compression check completes, the Compression Candidates table is updated. Now you have the option to compress the table by marking the table and using the Compress button.

Virtual Tables

Save system space

An SAP system contains thousands of empty tables consuming a lot of space. To save this space, you can replace them with virtual tables. Virtual tables are actually views; with the first `WRITE` operation on such a view, the virtual table is automatically replaced with a table. This action is performed by the database shared library (DBSL) in your SAP system. A remote or CLP access does not materialize a virtual table.

On this screen, you have the Virtual Tables and the Candidates for Virtualization tab. The Virtual Tables screen contains a list of all virtual tables in your system. By choosing Materialize, you can materialize tables.

The Candidates for Virtualization tab shows a list of tables that could be recreated as virtual tables because they are empty, not volatile, don't have a partitioning key, and don't use multidimensional clustering (MDC). By clicking Convert Empty Tables, these tables are dropped and recreated as virtual tables.

Indexes

Improve database performance

Information about the indexes of your database can be accessed on the Indexes screen. Effective indexes are the best way to improve the performance of a database. Without an index, the SQL Server engine is like a reader trying to find a word in a book by examining each page. By using the index, the reader can complete the task in a shorter time. Without an index in a database, a table has to be scanned, and every row has to be examined. Sometimes scans are unavoidable, but especially on large table, scans have a terrific impact on the performance. Optimizing and tuning the indexes of your database therefore is a very important task for every database administrator.

In the Selection area, you can specify filter criteria, a time frame, and how your indexes are sorted. According to your filter criteria, the information from Table 5.25 is displayed in the overview tables.

Column	Description
Table Name	Name of the table
Table Schema	Name of the table schema
Index Name	Name of the index
Index Schema	Name of the index schema
Number of Leafs	Number of leafs in the index
Cardinality	Cardinality of the index
Last used	Date when the index was last used
Est. Phys. Index Size (KB)	Estimated physical index size
Est. Log. Index Size (KB)	Estimated logical index size

Table 5.25 Tables and Indexes: Index

Single Table Analysis

The Single Table Analysis screen allows you to analyze and maintain single tables. You can check and maintain statistics information for each table and index such as cardinality, table size, and number of overflow records.

As you can see in Figure 5.72, you enter the schema and the name of the table you want to analyze. After you click the Apply Selection button, you'll see information in the summary area below about the size of the table, administrative actions being taken, and current savings by compression.

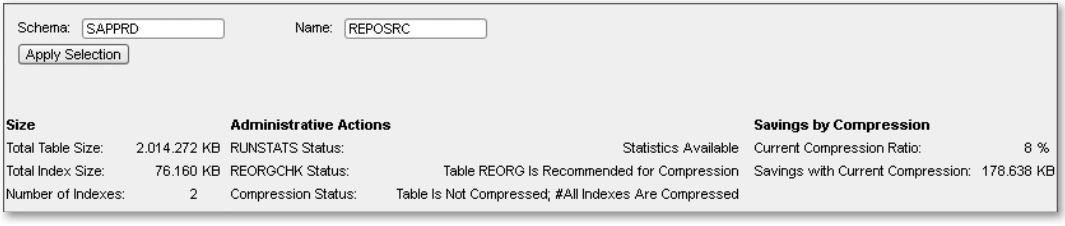


Figure 5.72 Single Table Analysis: Selection

Optimize table/indexes

Above the selection area, the following buttons are available to optimize your table and its indexes:

- RUNSTATS
  Schedule a RUNSTATS job for a single table.
- REORG
  Schedule a REORG job for a table.
- COMPRESSION CHECK
  Schedule a job to check whether a table would benefit from compression.
- COMPRESSION ON/OFF
  Switch the compression of a table on or off.
- VOLATILE ON/OFF
  Change the VOLATILE attribute of a table. Volatile tables are not processed by automatic RUNSTATS.
- COUNT
  Count the number of rows.

System Catalog tab

On the SYSTEM CATALOG tab, you find information that's available from the system catalog entry of the selected table. Here you find information about space management, technical attributes, statistical data, RUNSTATS profiles, and compression settings (see Figure 5.73).

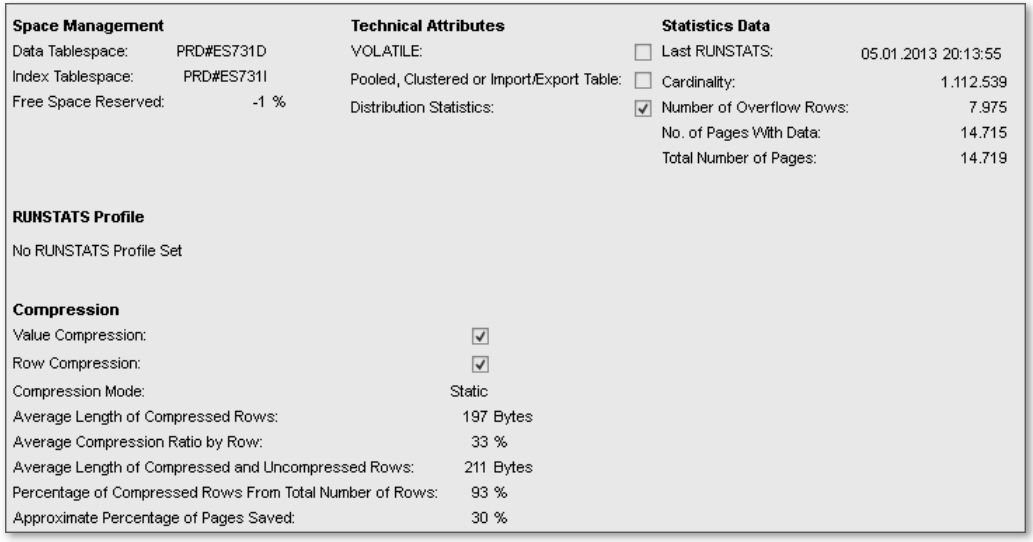


Figure 5.73 Single Table Analysis: System Catalog

Table Structure tab

Figure 5.74 shows the TABLE STRUCTURE tab, which provides an overview of the database columns of your table and detailed information about each column. In the COLUMN DISTRIBUTION table, you get an overview of the most frequently occurring values of the table.

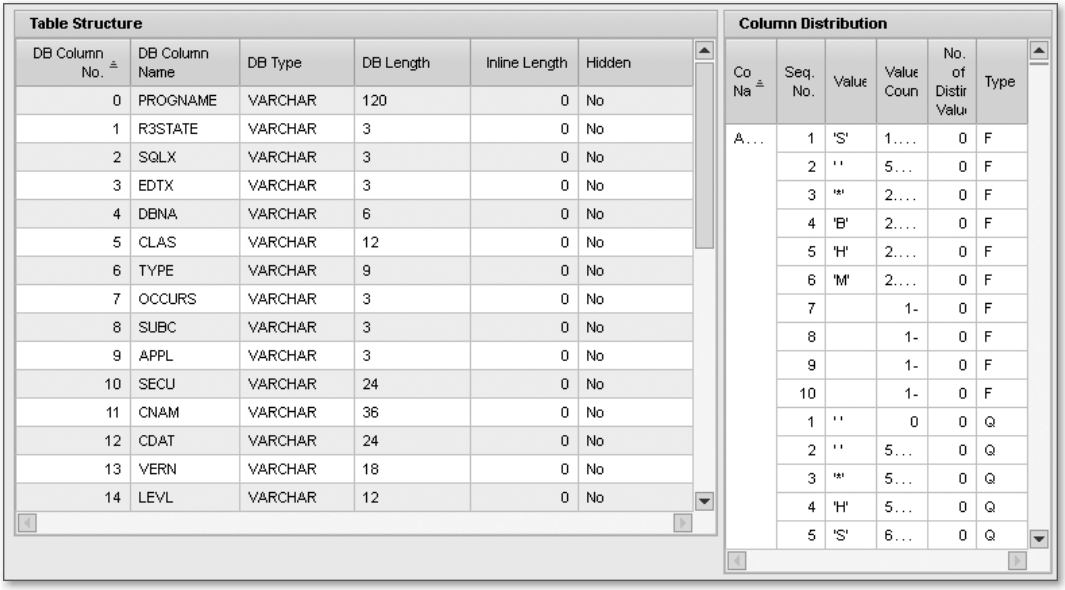


Table Structure

DB Column No.	DB Column Name	DB Type	DB Length	Inline Length	Hidden
0	PROGNAME	VARCHAR	120	0	No
1	R3STATE	VARCHAR	3	0	No
2	SQLX	VARCHAR	3	0	No
3	EDTX	VARCHAR	3	0	No
4	DBNA	VARCHAR	6	0	No
5	CLAS	VARCHAR	12	0	No
6	TYPE	VARCHAR	9	0	No
7	OCCURS	VARCHAR	3	0	No
8	SUBC	VARCHAR	3	0	No
9	APPL	VARCHAR	3	0	No
10	SECU	VARCHAR	24	0	No
11	CNAM	VARCHAR	36	0	No
12	CDAT	VARCHAR	24	0	No
13	VERN	VARCHAR	18	0	No
14	LEVL	VARCHAR	12	0	No

Column Distribution

Co Na	Seq. No.	Value	Value Coun	No. of Distir Valu	Type
A...	1	'S'	1....	0	F
	2	' '	5...	0	F
	3	'*'	2....	0	F
	4	'B'	2....	0	F
	5	'H'	2....	0	F
	6	'M'	2....	0	F
	7		1-	0	F
	8		1-	0	F
	9		1-	0	F
	10		1-	0	F
	1	' '	0	0	Q
	2	' '	5...	0	Q
	3	'*'	5...	0	Q
	4	'H'	5...	0	Q
	5	'S'	6...	0	Q

Figure 5.74 Single Table Analysis: Table Structure

Indexes

The INDEX tab page provides a list of all indexes that are defined for the chosen table. For each index the index name, the rule for index uniqueness, the type of index, and the date and time of the last RUNSTATS run are displayed. You can access detailed information using the following tabs under the overview:

- SYSTEM CATALOG
  Provides you with general index information, statistics data, and technical attributes (see Figure 5.75).
- INDEX STRUCTURE
  Displays the structure of the index and provides more information about the compression and the index size.

- Compression Status
  Indicates whether the index is compressed, and provides the values of your saved pages and saved leaf pages.
- REORGCHK
  Displays information about the cluster ratio, allocated space, total numbers of existing and deleted entries in the index, and cardinality.

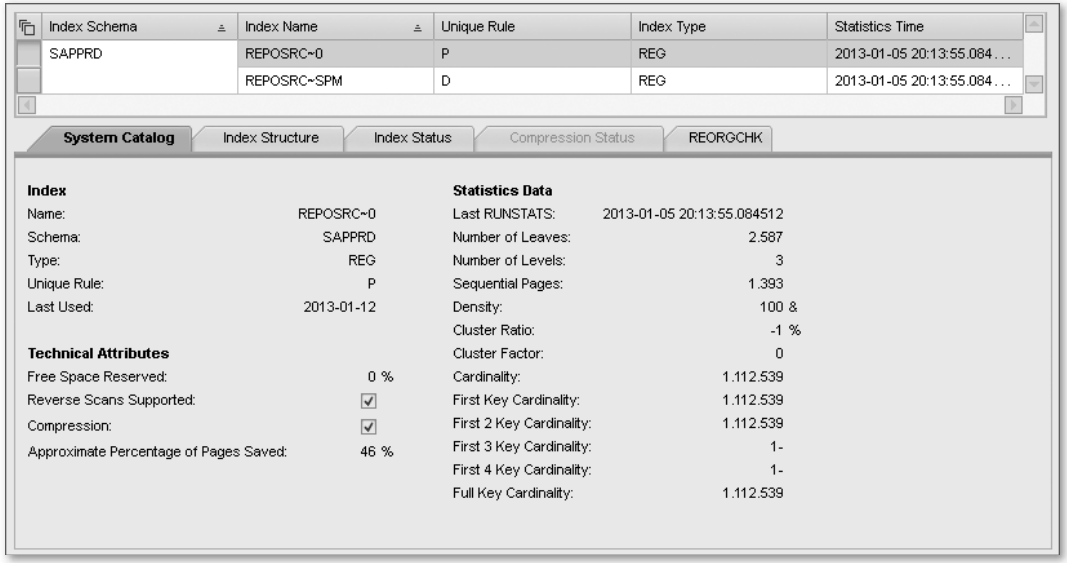


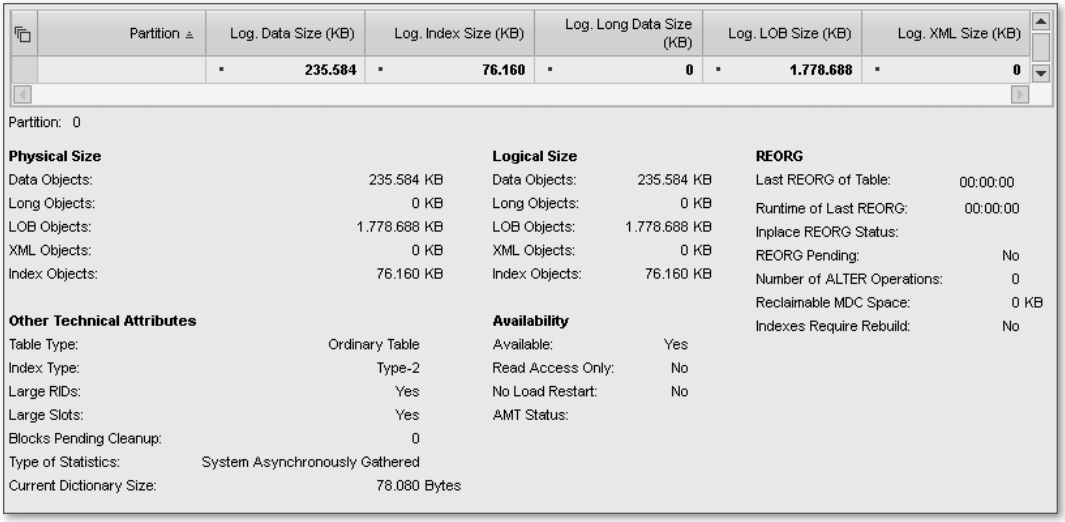
Figure 5.75 Single Table Analysis: Indexes

Table Status tab

The Table Status tab gives you an overview of the physical size, logical size, REORG information, availability, and other technical attributes (see Figure 5.76).

Compression Status tab

On the Compression Status tab, you can see if the compression is enabled, what code path was taken to build the dictionary, the build time, and the estimated compression.



Partition	Log. Data Size (KB)	Log. Index Size (KB)	Log. Long Data Size (KB)	Log. LOB Size (KB)	Log. XML Size (KB)
	235.584	76.160	0	1.778.688	0

Partition: 0

Physical Size

Data Objects:	235.584 KB
Long Objects:	0 KB
LOB Objects:	1.778.688 KB
XML Objects:	0 KB
Index Objects:	76.160 KB

Logical Size

Data Objects:	235.584 KB
Long Objects:	0 KB
LOB Objects:	1.778.688 KB
XML Objects:	0 KB
Index Objects:	76.160 KB

REORG

Last REORG of Table:	00:00:00
Runtime of Last REORG:	00:00:00
Inplace REORG Status:	
REORG Pending:	No
Number of ALTER Operations:	0
Reclaimable MDC Space:	0 KB
Indexes Require Rebuild:	No

Other Technical Attributes

Table Type:	Ordinary Table
Index Type:	Type-2
Large RIDs:	Yes
Large Slots:	Yes
Blocks Pending Cleanup:	0
Type of Statistics:	System Asynchronously Gathered
Current Dictionary Size:	78.080 Bytes

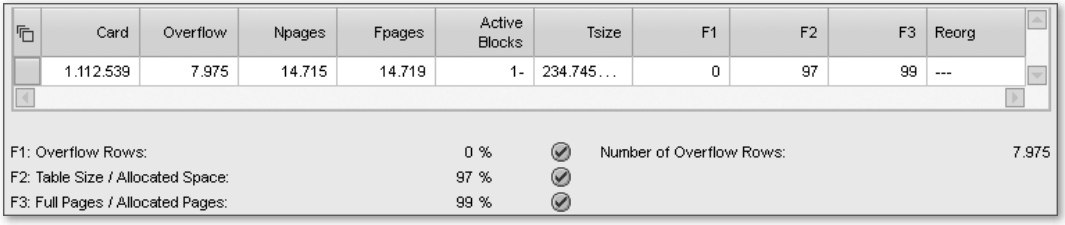
Availability

Available:	Yes
Read Access Only:	No
No Load Restart:	No
AMT Status:	

Figure 5.76 Single Table Analysis: Table Status

REORGCHK tab

On the REORGCHK tab, you can see the percentage of overflow rows, the table size divided by allocated space, and the full pages divided by allocated pages as a percentage (see Figure 5.77). The number of records that have overflowed is also shown. An overflow row can be created when rows are updated and the new rows contain more by bytes than the existing ones. Overflow rows can also be created when you add columns to already existing tables. The recommended threshold for overflow rows is under 5% of the total numbers of rows.



Card	Overflow	Npages	Fpages	Active Blocks	Tsize	F1	F2	F3	Reorg
1.112.539	7.975	14.715	14.719	1-	234.745...	0	97	99	---

F1: Overflow Rows:	0 %	Number of Overflow Rows:	7.975
F2: Table Size / Allocated Space:	97 %		
F3: Full Pages / Allocated Pages:	99 %		

Figure 5.77 Single Table Analysis: REORGCHK

Create Statement tab

On the CREATE STATEMENT tab, the SQL statement to create the table is displayed.

Index

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W