

DB2 System Performance Introduction

Introduction	1-2
DB2 address spaces	1-3
Storage groups	1-7
Databases	1-8
Tablespaces	1-9
Tables	1-12
Indexes	1-13
Pages	1-15
Bufferpools	1-17
SQL types	1-19
Memory usage	1-20
Resource limit facility database	1-23
Work file database	1-24
DB2 and the Parallel Sysplex	1-25

Introduction

This section reviews the principal components of a DB2 system and serves to provide us with the ‘big picture’ of the resources which will be covered on this course.

The DB2 family of products includes:

- DB2 Warehouse Manager feature. This contains the tools to build, manage, govern, and access DB2 for OS/390 and z/OS-based data warehouses. It includes:
 - DB2 Warehouse Center:
 - DB2 Universal Database Version 7 Release 1 Enterprise Edition
 - Warehouse agents for UNIX, Windows, and OS/390
 - Information Catalog
 - QMF Version 7
 - QMF High Performance Option
 - QMF for Windows

- DB2 Management Clients Package:
 - DB2 Control Center
 - DB2 Stored Procedure Builder
 - DB2 Installer
 - DB2 Visual Explain (neat)
 - DB2 Estimator

- Net Search Extender for in-memory text search for e-business applications

- Net.Data for secure Web applications

DB2 Connect

WebSphere

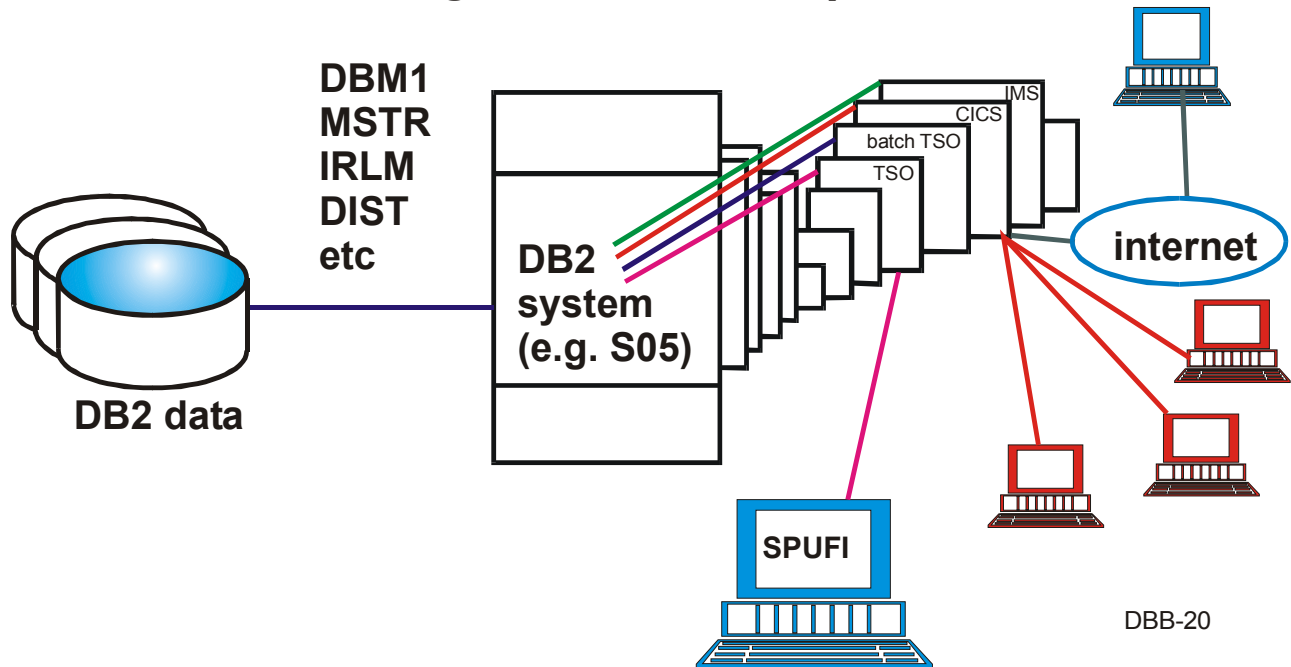
Omegamon

Platinum

DB2 address spaces

Each DB2 system is made up of a series of related address spaces.

Accessing DB2 address spaces

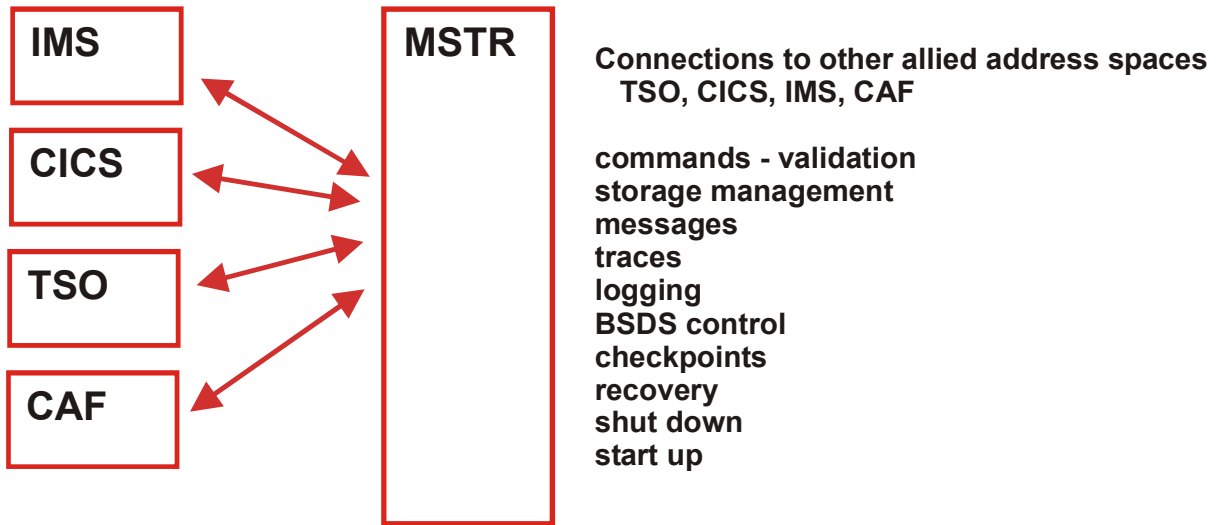


NP	JOBNAME	StepName	ProcStep	JobID	Owner	C	Pos	DP	Real	Paging	SIO
	DSN1SPAS	DSN1SPAS	IEFPROC	STC02545	START2	NS	F5	969	0.00	0.00	
	DSN1DIST	DSN1DIST	IEFPROC	STC02544	START2	NS	F5	732	0.00	0.00	
	DSN1DBM1	DSN1DBM1	IEFPROC	STC02543	START2	NS	F7	12T	0.00	0.00	
	DB2IRLM	DB2IRLM		STC02542	START2	NS	FD	350	0.00	0.00	
	DSN1MSTR	DSN1MSTR	IEFPROC	STC02541	START2	NS	FE	1927	0.00	0.00	

NP	JOBNAME	CPU%	ASID	ASIDX	EXCP-Cnt	CPU-Time	SR	Status	SysName	SPag	SCP
	DSN1SPAS	0.01	77	004D	668	45.65			P390		0
	DSN1DIST	0.00	256	0100	2,048	6.33			P390		0
	DSN1DBM1	0.01	257	0101	27,218	357.88			P390		0
	DB2IRLM	0.04	258	0102	440	385.77			P390		0
	DSN1MSTR	0.16	259	0103	23,529	1104.15			P390		0

NP	JOBNAME	U%	Workload	SrvClass	SP	ResGroup	Server	Quiesce	ECP	ECPU-Time	ECP
	DSN1SPAS	8	DB2	DB2LOW	1		NO			45.65	0.
	DSN1DIST	8	DB2	DB2LOW	1		NO			6.33	0.
	DSN1DBM1	8	DB2	DB2MED	1		NO			357.88	0.
	DB2IRLM	8	DB2	DB2HI	1		NO			385.77	0.
	DSN1MSTR	8	SYSTEM	SYSSTC	1		NO			1104.16	0.

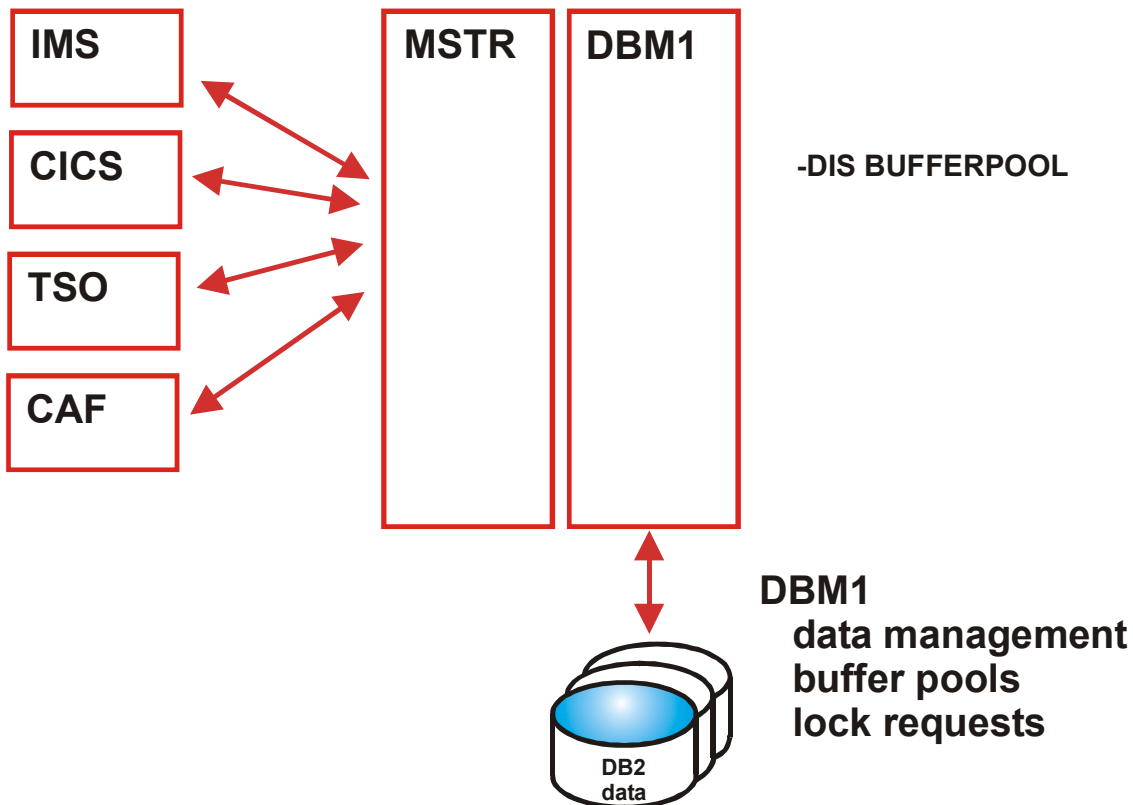
DB2 Regions



MSTR
system services for DB2

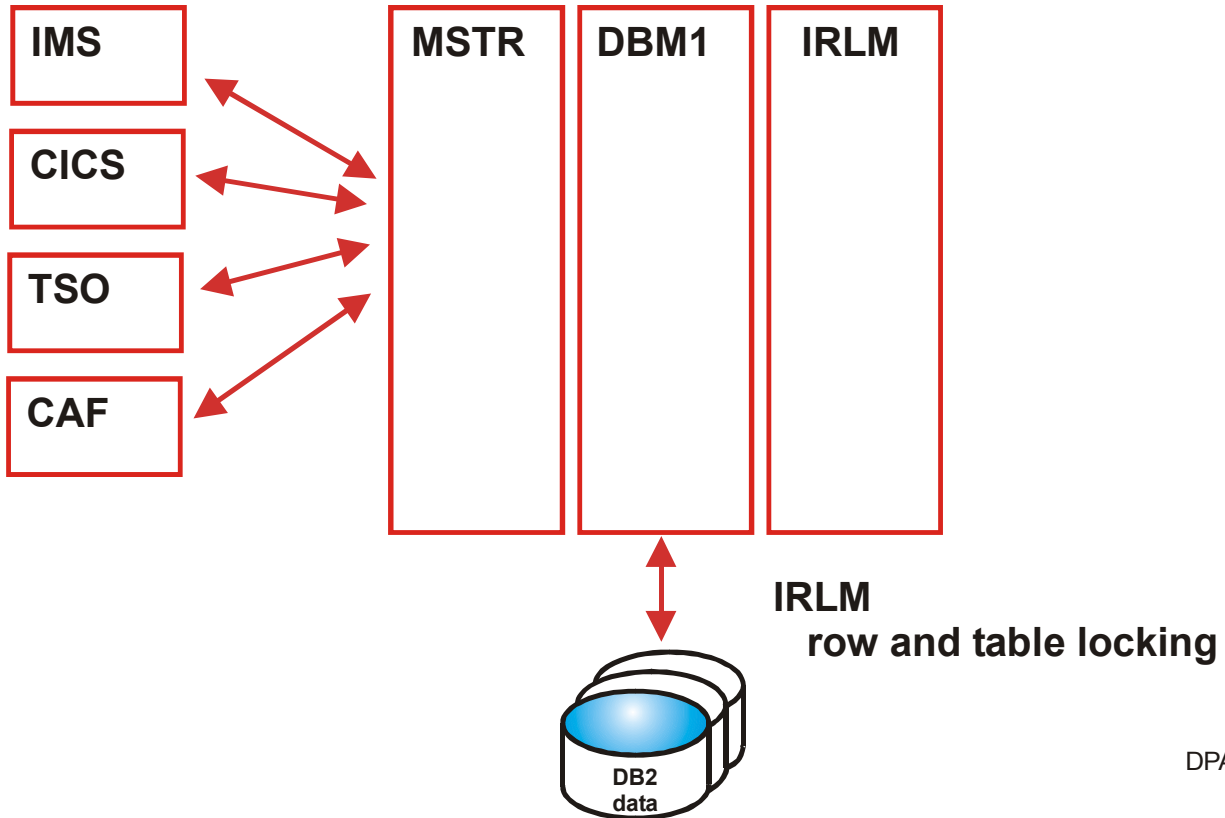
DPA-180

DB2 Regions



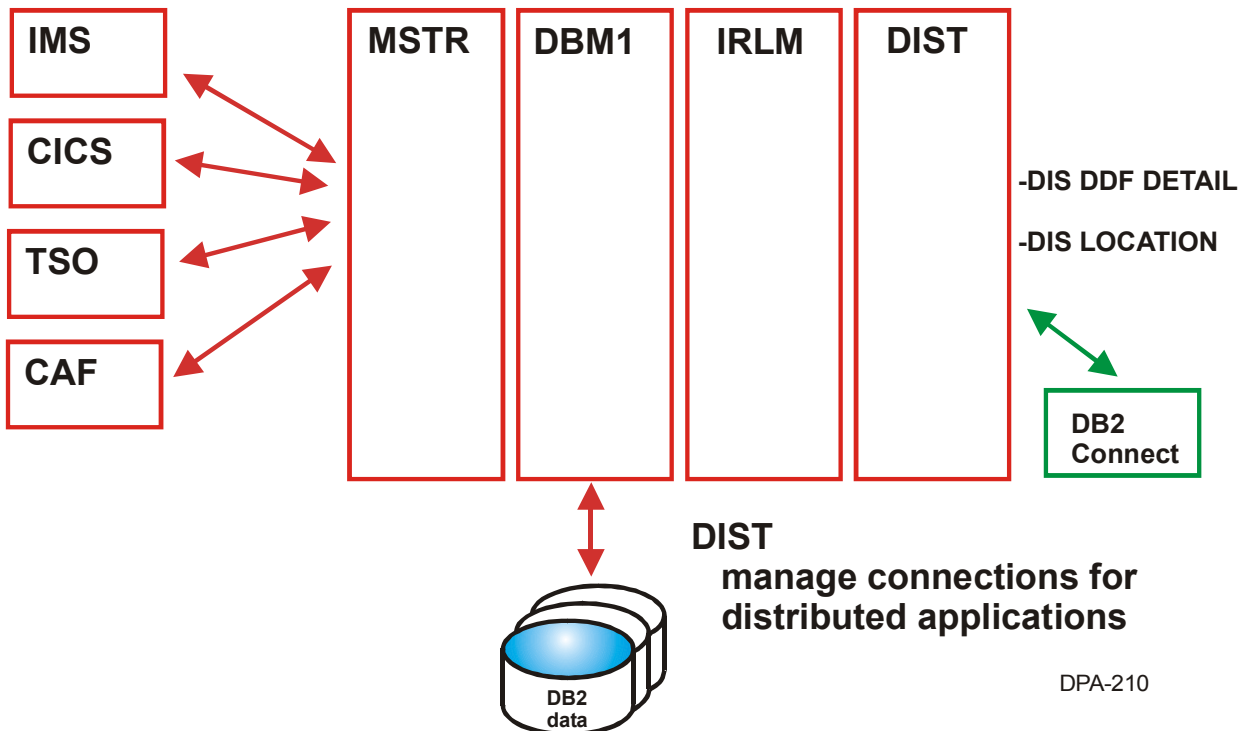
DPA-190

DB2 Regions



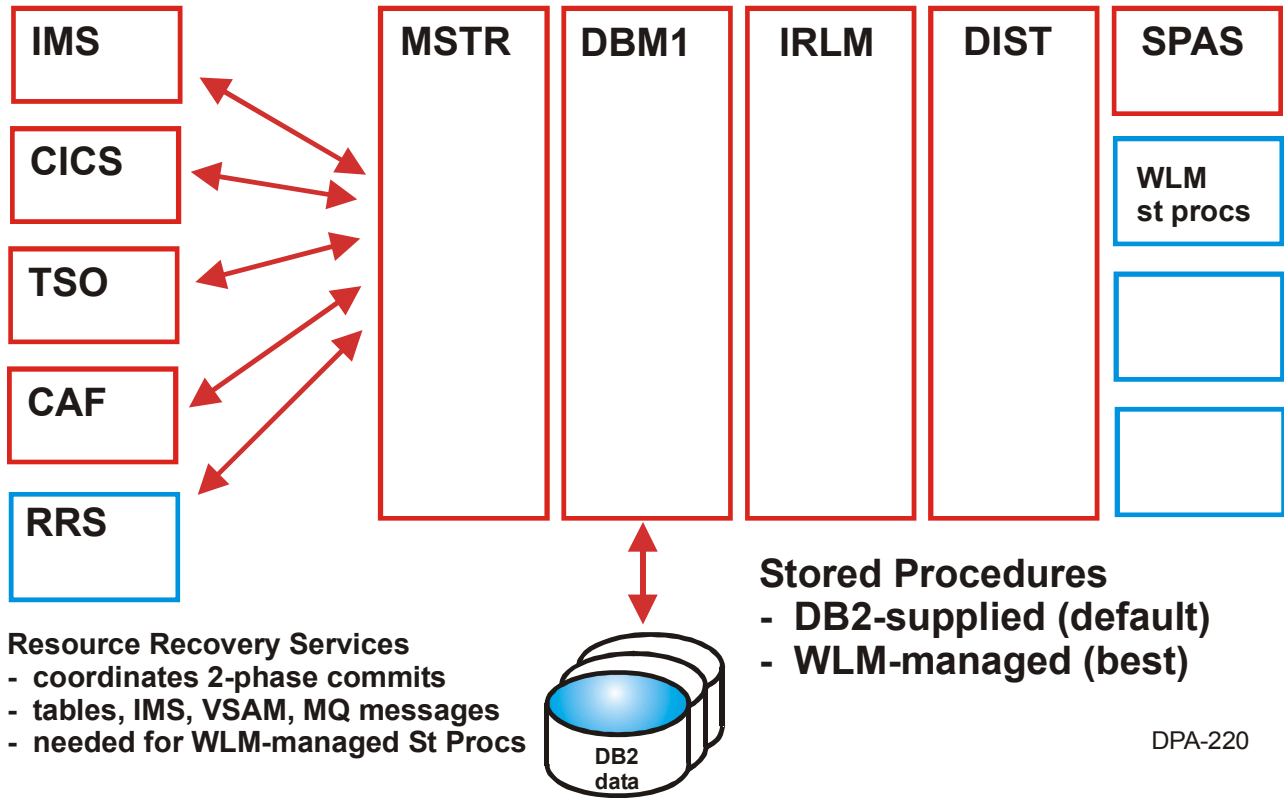
DPA-200

DB2 Regions



DPA-210

DB2 Regions



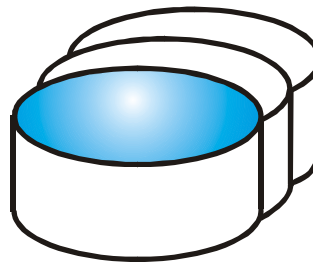
DB2 resources: Storage Groups

Before we can process the data, we must create the table and all its pre-requisites. The systems programmer will first allocate one or more disks to be used by DB2. This will create a **storage group**.

Different storage groups may use disks of different characteristics e.g. disk types. Names may be up to 8 characters.

Storage group

- * group of disks
- * defined by administrator
- * **CREATE STOGROUP ...**



DBB-30

Storage groups should be spread horizontally across arrays of disk volumes. This reduces contention problems if two busy indexes are placed on the same logical volume.

```
select *
      from sysibm.sysstogroup
```

NAME	CREATOR	VCATNAME	VPASSWORD	SPACE	SPCDATE	IBMREQD	CREATEDB
TSTSG001	JBCF00	JSDCOF		0		N	JBCF00
DBXSG001	JBCF00	JSDDBX		0		N	JBCF00
DBXSG002	JBCF00	JSDDBX		0		N	JBCF00
CPMSGTRN	HBS03	HBSCPM		0		N	HBS03
CPMSG01	HBS03	HBSCPM		0		N	HBS03

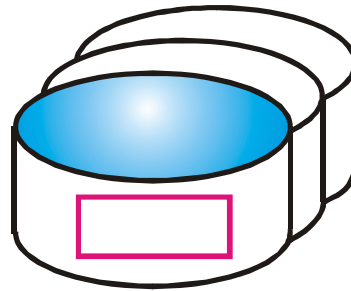
Databases

Once the storage group has been created, the systems programmer (or DBA) will create a **database** which uses one or more storage groups.

The term ‘database’ is not the same in DB2 as it is in IMS. With DB2, the database is the unit of start/stop for operations. Each database may (eventually) contain several related tables of data.

Databases use Storage Groups

- * unit of Start/Stop for operations
- * holds one or more tables
- * CREATE DATABASE ...
- * no space specification



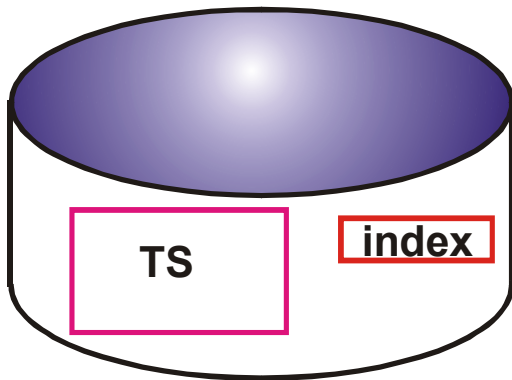
DBB-40

```

DSNT360I &D2TA *****
DSNT361I &D2TA * DISPLAY DATABASE SUMMARY
              * GLOBAL
DSNT360I &D2TA *****
DSNT362I &D2TA DATABASE = ZT01DD01 STATUS = RW
              DBD LENGTH = 4028
DSNT397I &D2TA
NAME        TYPE PART STATUS          PHYERRLO PHYERRHI CATALOG  PIECE
-----
ZT01TS01 TS          RW,CHKP
OAK555I0 IX          RW
OAK901I0 IX          RW
***** DISPLAY OF DATABASE ZT01DD01 ENDED *****
DSN9022I &D2TA DSNTDDIS 'DISPLAY DATABASE' NORMAL COMPLETION
***
    
```

The DBD LENGTH parameter often has a value around 80K. Seibel is nearer 1MB.

Tablespaces



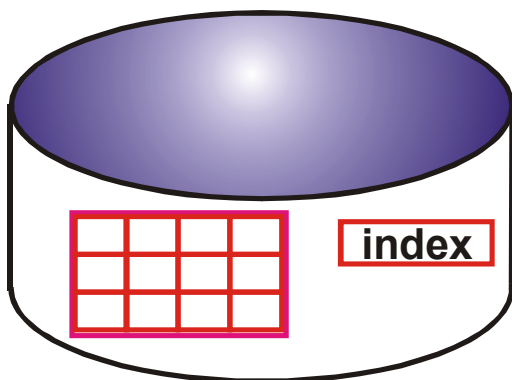
Tablespace

- * unit of recovery
- * holds tables
- * can be locked for update
- * **CREATE TABLESPACE ...**

DBB-50

The next step is to define a **tablespace** which belongs to the database. Tablespaces are important: you will take backups of the tablespace, not a database. If there is a failure, you will restore the tablespace, not the entire database. Each database, therefore, consists of one or more tablespaces. Ideally, less than 100 TS per DB.

Segmented tablespaces divide their space into groups of pages called segments which contain rows from a single table. If all the table's rows are to be processed, DB2 scans only the segments which hold that table.



Segmented Tablespace

TS divided into segments

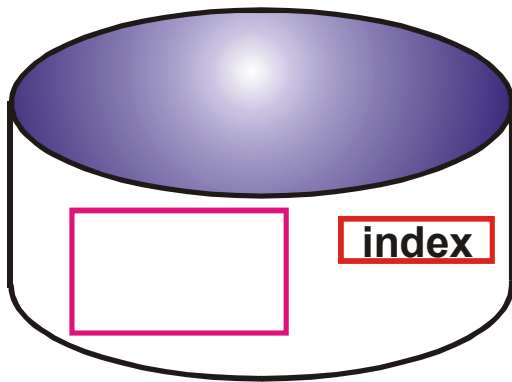
Each segment used for one table only

When processing a table, DB2 ignores other segments

TS has a segment map

DBB-51

Simple tablespaces may contain more than one table, but have no segments defined.



Simple Tablespace

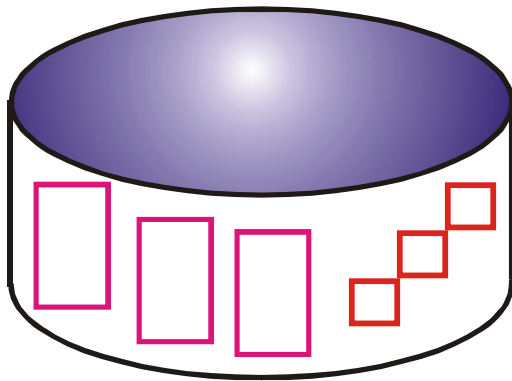
No segments

Can have multiple tables

Locking implications

DBB-52

Large tablespaces can be **partitioned**. This means that one partition could be processing a DB2 utility whilst other partitions are available for SQL processing. Each partition contains rows from one table. Partitions can be in different storage groups.



Partitioned Tablespace

For large tables

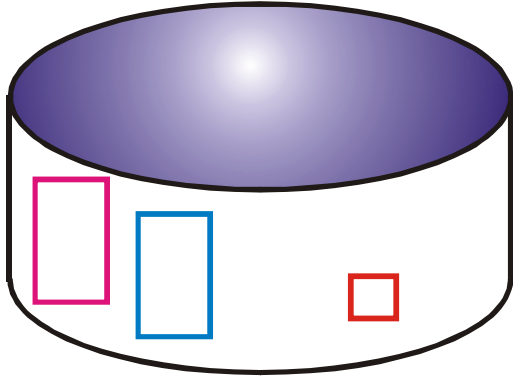
Can be processed separately

Parallel processing option

Index control key ranges

DBB-53

Large objects such as video, audio, and large text columns are held in an auxiliary table in a **LOB tablespace**. The base table must have a column of type ROWID.



LOB Tablespace

Contains auxiliary tablespace:

video

audio

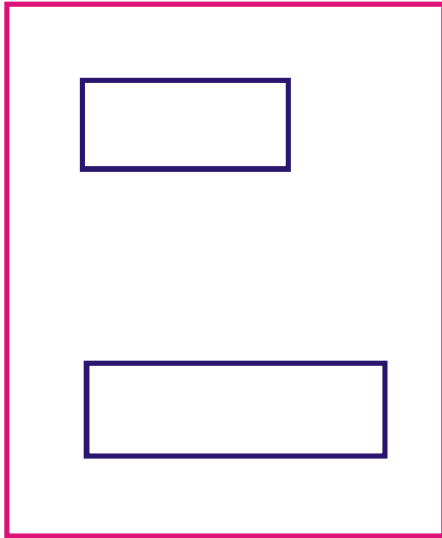
large text columns

Uses ROWID to link to base table

DBB-54

Tables

Tables



- * store data
- * rows and columns (max 750)
- * manipulated by SQL
- * maximum size 64 GB (V4)
1024 GB (V5)
16 TB (V6)
- * space provided by DB2
- * index(es) for improved access

DBB-60

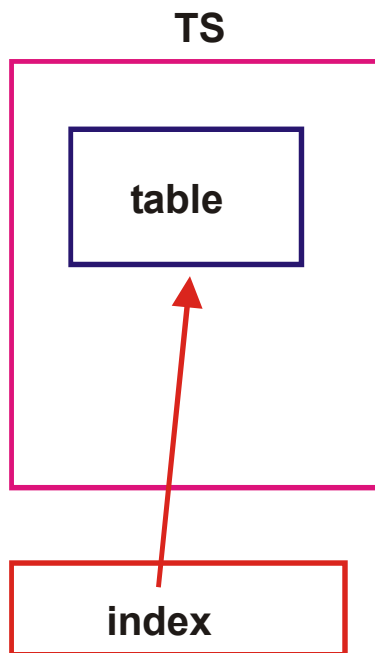
Within a tablespace, there will be **tables**. Your programs will process data in the tables, and the SQL statements will manipulate the data in the tables.

The greatest permitted size of a DB2 table has increased from 64 GB with the early versions of DB2 to 16 Terabytes with Version 6. The space for the tables is automatically provided by DB2 (though it is possible to set up your own VSAM datasets which are not managed by DB2).

Partitioned tables allow parallel processing for queries.

If you need very large tables, use the extended-addressability (EA) function of dfSMS. This can be done by specifying a DSSIZE larger than 4GB in the CREATE TABLESPACE statement.

Indexes



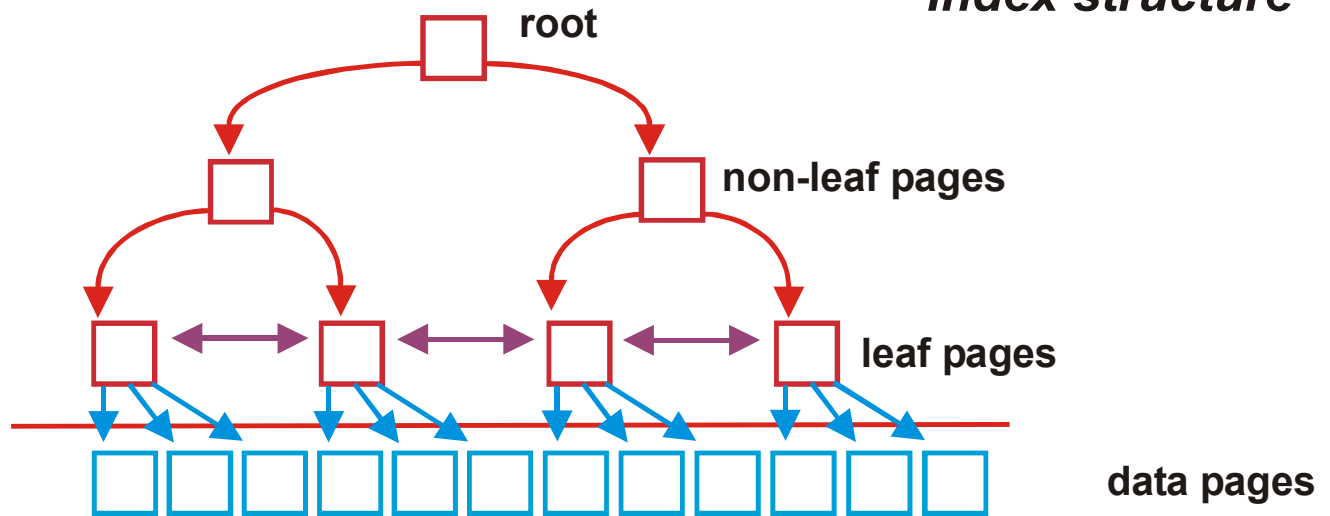
Index

- * faster access to a column
- * based on one or more columns
- * can be unique
- * ascending or descending sequence
- * **CREATE INDEX ...**
- * reserves own space

DBB-80

CREATE INDEX automatically creates a unique indexspace. Names are up to 18 characters long. Freespace (PCTFREE) can be reserved for future insertions.

If the data is generally accessed on a particular column, it would be more efficient to create an **index** based on that column's values. A table may have several indexes. Some of these may be unique.

Index structure

DPA-10

The greatest efficiency arises when the physical arrangement of the data closely matches the index sequence. This type of index is known as the **clustering index**. A table can have at most one clustering index. A table with a clustering index should be the only table in a tablespace. Always define a clustering index.

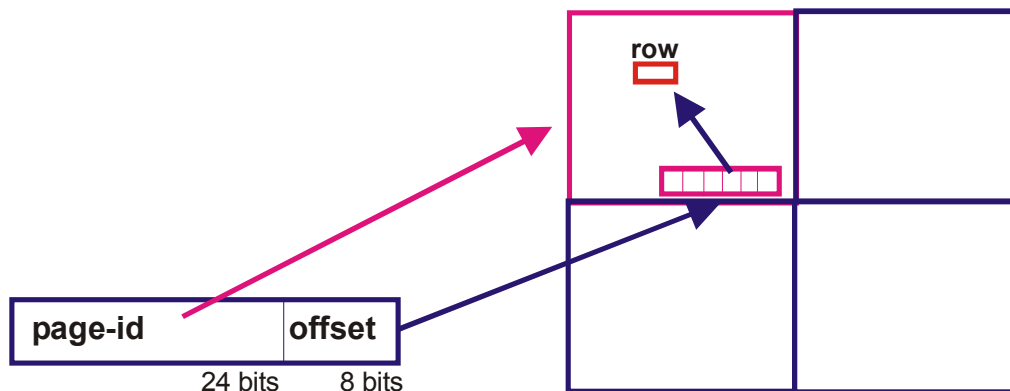
The PIECESIZE parameter enables you to divide an index into several datasets. This applies to non-partitioned indexes. So, for example, if you had an index of 8MB and wanted to spread it over 4 datasets, you should specify PIECESIZE 2 M.

If you have a partitioned table, you can use ALTER to adjust the key ranges of the index.

Pages

Data is held in pages. The page size can be 4K, 8K, 16K or 32K. Rows do not span pages. Each page uses 24 bytes for control information. Each row has a fixed overhead of 8 bytes plus additional overhead depending on the column definitions. In practice, if the database records are greater than 4054 bytes, a larger page size must be used. A page can hold up to 127 rows (255 with data compression).

Each row has a four byte record-id (RID). The first three bytes form the **page-id**, and the last byte holds an **offset** from the page foot. The value in the offset tells DB2 exactly where the data is stored within the page. This technique allows DB2 to move the data around within the page without affecting its RID.

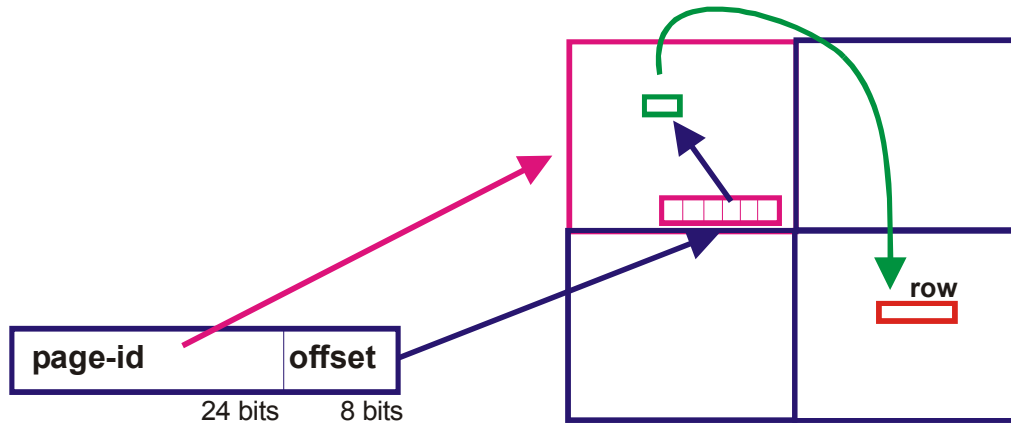


Each Record-id (RID) is 4 bytes long

DBB-70

If Extended Addressability is used (e.g. for tables larger than 64GB), the RID is five bytes long.

If the row becomes too long for the current page, it is moved to a different page and an overflow pointer is placed in the original page. Again, the RID is unchanged.



Overflow pointer points to new location

DBB-75

When inserting and deleting data, all freespace is managed by DB2:

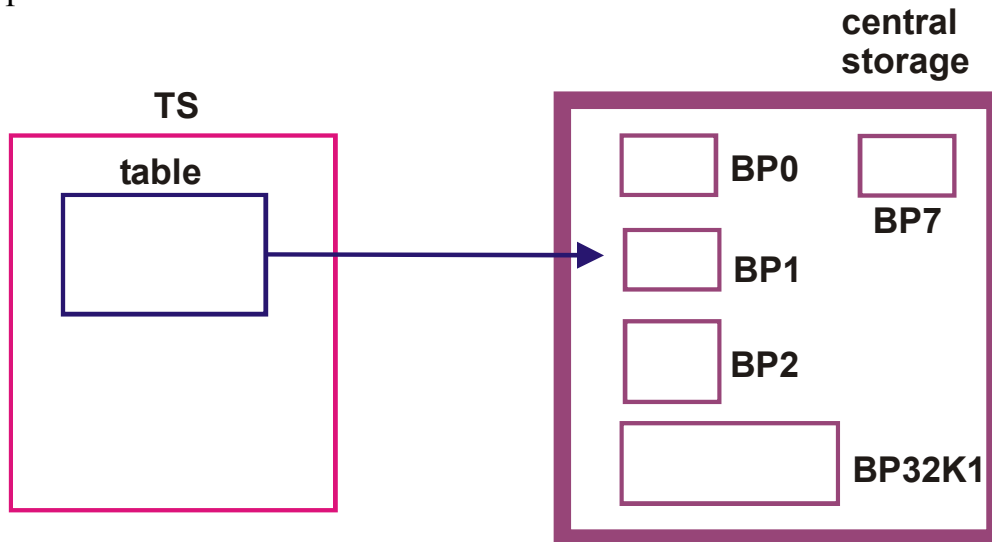
- FREESPACE 5 tells DB2 to keep every 5th page free at load time
- PCTFREE 10 tells DB2 to keep 10% of a page free at load time

LOB (large object) data is held in an auxiliary table which is located in a LOB tablespace. LOB data may span pages. A page contains data from one LOB item only.

Row-level locking was introduced with DB2 Version 4.

Bufferpools

Data is transferred to and from the tables using **bufferpools** which are areas of virtual storage. When an update is made, the data is written back to disk (eventually). If a read operation takes place, DB2 checks the buffers to see if the required data is already in the pool. This reduces I/O as DB2 doesn't have to re-read unless necessary.



Data is transferred using Bufferpools

DBB-90

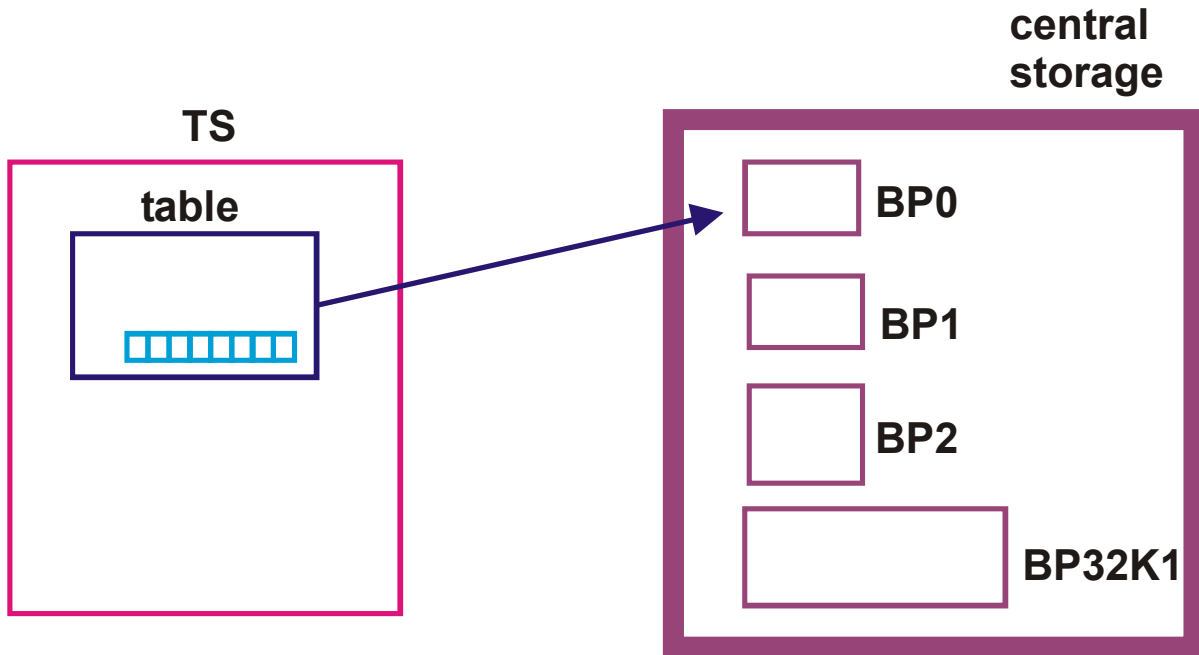
The buffer pools are BP0, BP1, BP2, etc. to BP49 for 4K buffers; the 32K buffers use pools called BP32K0, BP32K1 to BP32K9. Buffers can also be 8K and 16K and have names of the form BP8Kn and BP16Kn. The number of buffers in each pool is dynamically controlled by DB2. The `DISPLAY BUFFERPOOL` command shows the current settings.

```

DSNB401I  -D1WP BUFFERPOOL NAME BP1, BUFFERPOOL ID 1, USE COUNT 358
DSNB402I  -D1WP VIRTUAL BUFFERPOOL SIZE = 80000 BUFFERS
           ALLOCATED      =      80000   TO BE DELETED   =          0
           IN-USE/UPDATED =      2932
DSNB406I  -D1WP VIRTUAL BUFFERPOOL TYPE -
           CURRENT        = DATASPACE
           PENDING        = DATASPACE
           PAGE STEALING METHOD = LRU
DSNB403I  -D1WP HIPERPOOL SIZE = 0 BUFFERS, CASTOUT = YES
           ALLOCATED      =          0   TO BE DELETED   =          0
           BACKED BY ES   =          0
DSNB404I  -D1WP THRESHOLDS -
           VP SEQUENTIAL  = 80   HP SEQUENTIAL          = 80
           DEFERRED WRITE = 50   VERTICAL DEFERRED WRT = 10,0
           PARALLEL SEQUENTIAL = 50   ASSISTING PARALLEL SEQT= 0
DSNB405I  -D1WP DATASPACE NAME(S) - D1WPD000
    
```

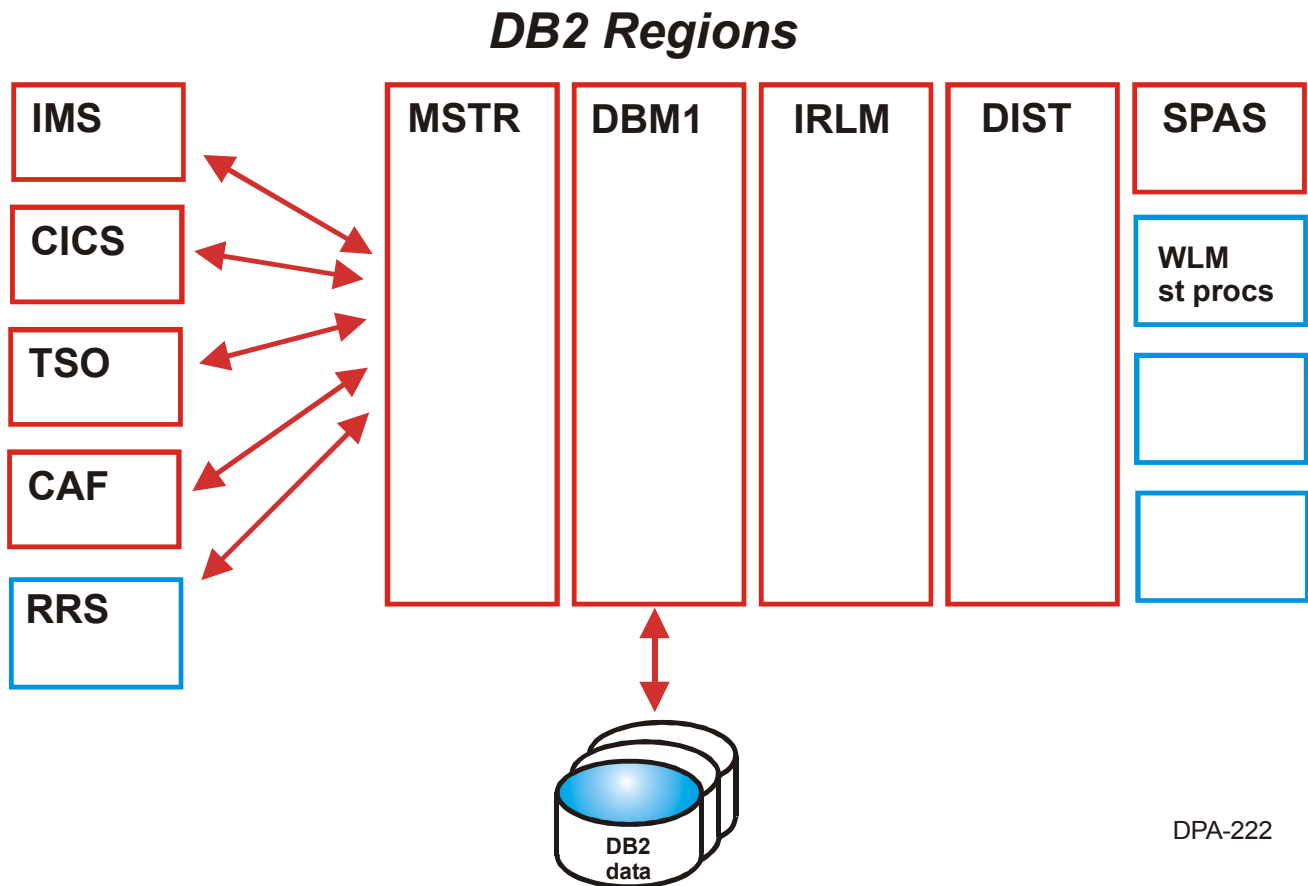
Hiperpools are extensions to bufferpools; up to 8GB of expanded storage can be used for data buffers.

Normally, a page is read one-at-a-time. If sequential processing is being performed, DB2 can opt to use a **prefetch** which will read up to 64 pages at a time.



Sequential prefetch reads multiple rows

DBB-92



SQL types:

- **STATIC** request is compiled
- **DYNAMIC** request is passed to DB2 at run time
 PREPARE c.f. BIND
 EXECUTE IMMEDIATE
- **Interactive SQL** using SPUFI

SQL can be placed in **stored procedures** – especially useful in web and client/server applications.

ODBC can be used by C and C++ applications in client/server environments.

SQLJ allows JAVA programs to produce static SQL.

JDBC allows JAVA programs to produce dynamic SQL.

Memory usage

Each address space has a virtual storage limit of 2 GB as a result of 31-bit addressing. This corresponds to a limit of 16 MB below the line, and 2032 MB above the line. DB2 manages storage into several subpools by issuing GETMAINS and FREEMAINS.

The DBM1 address space uses most storage. Components which need storage above the line include:

- Virtual Buffer Pool
- EDM pool
- User threads
- System threads
- Compression dictionaries
- Local dynamic statement cache

If your system is running near the 2 GB limit, you may receive Short on storage conditions at GETMAIN time and a critical DB2 task may fail.

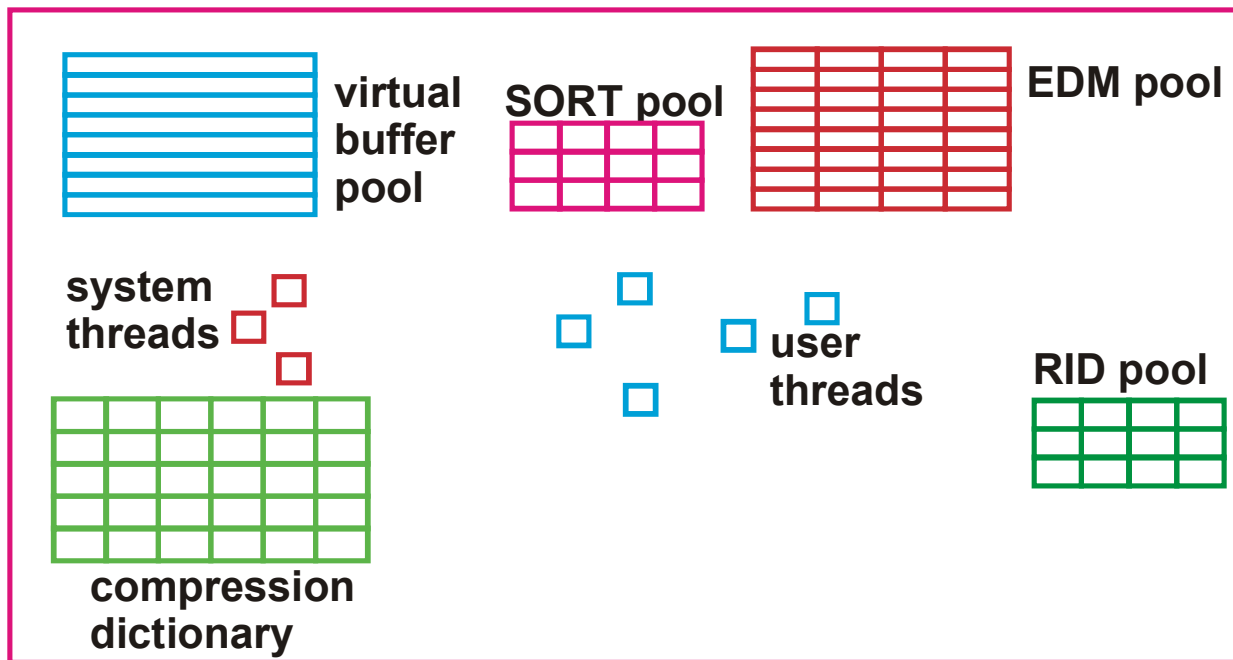
The RMF Virtual Storage Private Area Report can show potential storage shortages and trends of storage consumption (as can products such as Omegamon).

If the amount of storage still available above the line is less than 200 MB, you should consider taking some action.

Eventually, the problem will disappear in version 8 when DB2 uses the 64-bit architecture of z/OS.

Typical virtual storage usage

Virtual buffer pools	40 to 800 MB
EDM pool	20 to 400 MB
User thread storage	0.1 to 2 MB per active user thread
System thread storage	50 KB per system thread
Compression dictionary	up to 64 KB for each open compressed data set
Local dynamic statement cache	0 to 300 MB
Others	200 to 400 MB



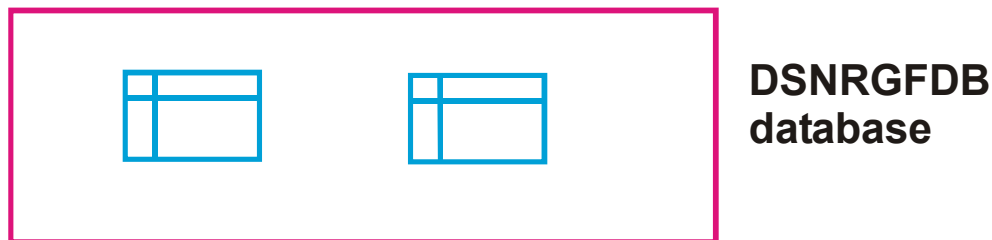
DBM1 virtual storage components

DPA-25

Data definition control support database

This is automatically created during installation. This database is a collection of tables used by data definition control support to **restrict** the submission of specific DB2 DDL statements to selected application identifiers (plans or collections of packages).

After this database is created, you must populate the tables to use of this facility. The system name for this database is DSNRGFDB.



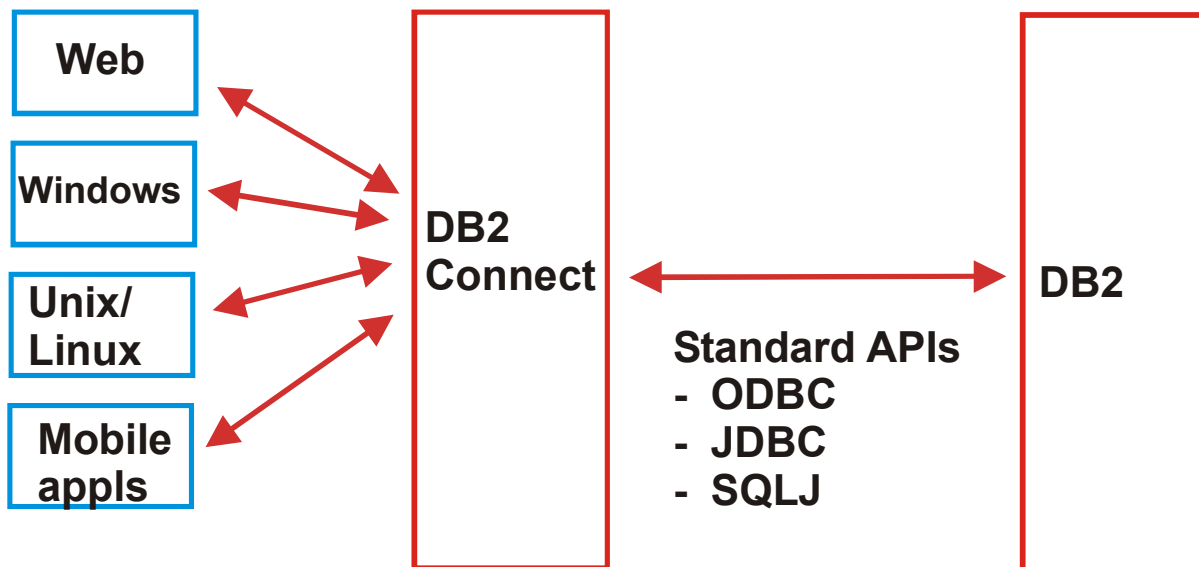
DDL → plans / packages

Data definition control support database

DPA-30

It is often used in conjunction with DB2 Connect.

DB2 Connect



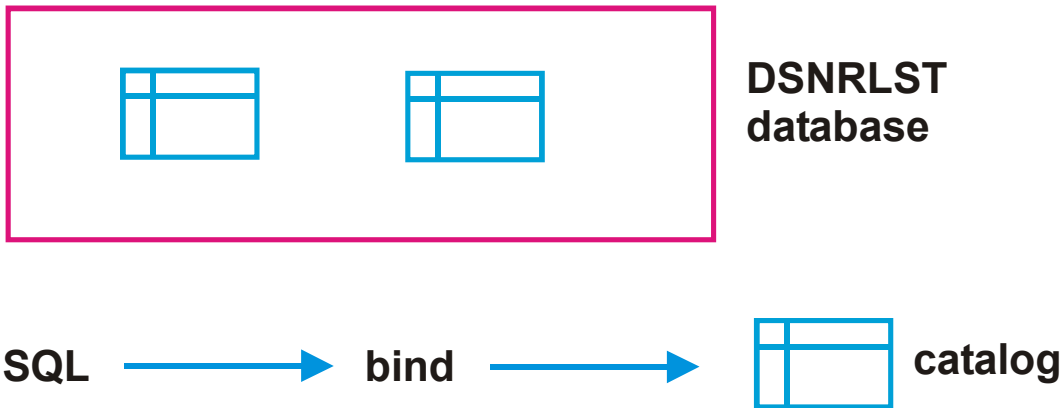
DPA-260

Resource limit facility database (DSNRLST)

This enables you to control the amount of processor resources used by dynamic SELECT statements. This means that you could choose to disable bind operations during critical times of day to avoid contention with the DB2 catalog.

You can establish a single limit for all users, different limits for individual users, or both. You can choose to have these limits applied before the statement is executed (**predictive governing**), or while a statement is running (**reactive governing**).

You can opt to use both modes of governing. The limits are defined in one or more resource limit specification tables (RLST).



Resource limit facility database

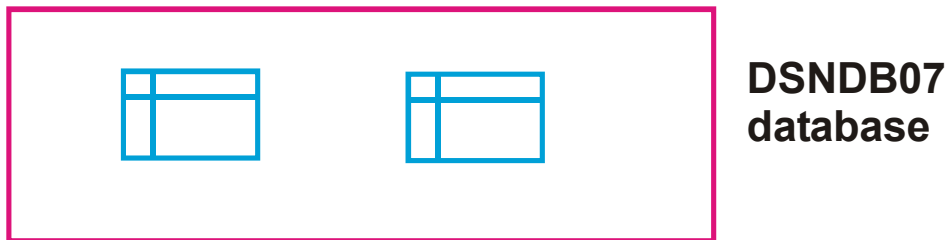
DPA-40

Work file database

This is used when processing SQL statements which need some working space for sort space.

DB2 creates a work file database at installation time, though you can dynamically create additional work file table spaces using CREATE TABLESPACE statements.

In a non-data-sharing environment, the work file database is called DSNDB07. In a data sharing environment, each DB2 member in the data sharing group has its own work file database. Put this work file in its own bufferpool e.g. BP7.

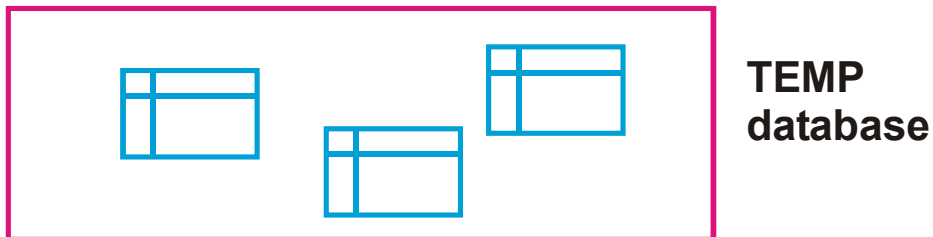


Work file database

DPA-50

TEMP database

The TEMP database is used to hold declared temporary tables. All the declared temporary tables are placed in this database. You can create one TEMP database for each DB2 subsystem or data sharing member.



TEMP database holds declared temporary tables

DPA-60

DB2 and the Parallel Sysplex

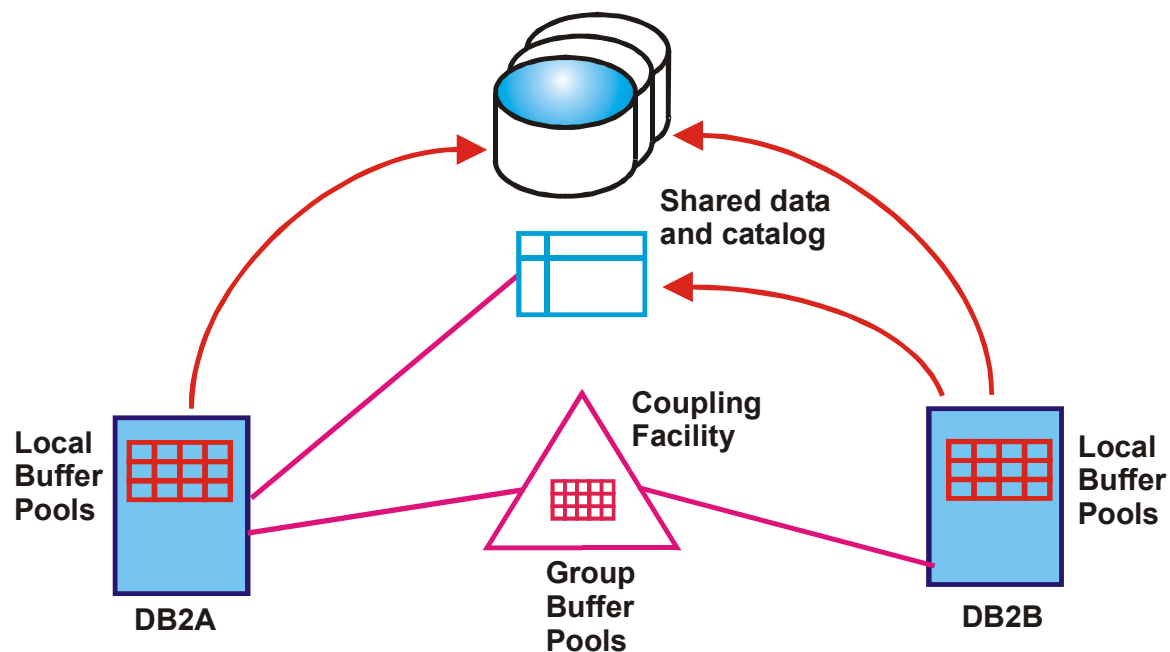
DB2 can take advantage of the processing capabilities of a Parallel Sysplex. If you use two or more processors to share the same data, you can:

- maximize performance while minimizing cost
- improve system availability and concurrency
- expand system capacity
- configure your system environment more flexibly

With data sharing, applications running on more than one DB2 subsystem can read from and write to the same set of data concurrently.

Sharing DB2s belong to a DB2 data sharing **group**. A data sharing group is a collection of one or more DB2 subsystems (members) accessing shared DB2 data.

Members of a particular group use the same shared DB2 catalog and directory.



DB2 Data Sharing Group

DPA-172