DB2 UDB for UNIX, Windows, OS/2 Database Administration

Certification Preparation

DB2 UDB Data Management Consulting Services
Toronto, Canada

© IBM CORPORATION 2000

Certification Preparation Topics

- **Overview of test #512 (Fundamentals) material.**
- **In depth review of test #513 (Database Administration) material. From the certification guide:**
  - Chapter 9 - Data Storage Management
  - Chapter 10 - Maintaining Data
  - Chapter 11 - Database Recovery
  - Chapter 12 - Monitoring and Tuning
- **Your responsibility?? Ask questions.**

Certification Preparation Overview of test #512 materials (Fundamentals)

- Knowledge and ability to use the DB2 UDB GUI tools and the Command Line Processor (CLP)
- Database Administration Server (DAS)
- Authorities and Privileges
- Creating and accessing DB2 databases
- Basic SQL usage

Fundamentals
DB2 UDB GUI Tools and CLP

- **Database administrators have a number of graphical-based tools they can use to manage and administer DB2 databases. Alternatively, a DBA can also use script-based tools to administer the database environment.**

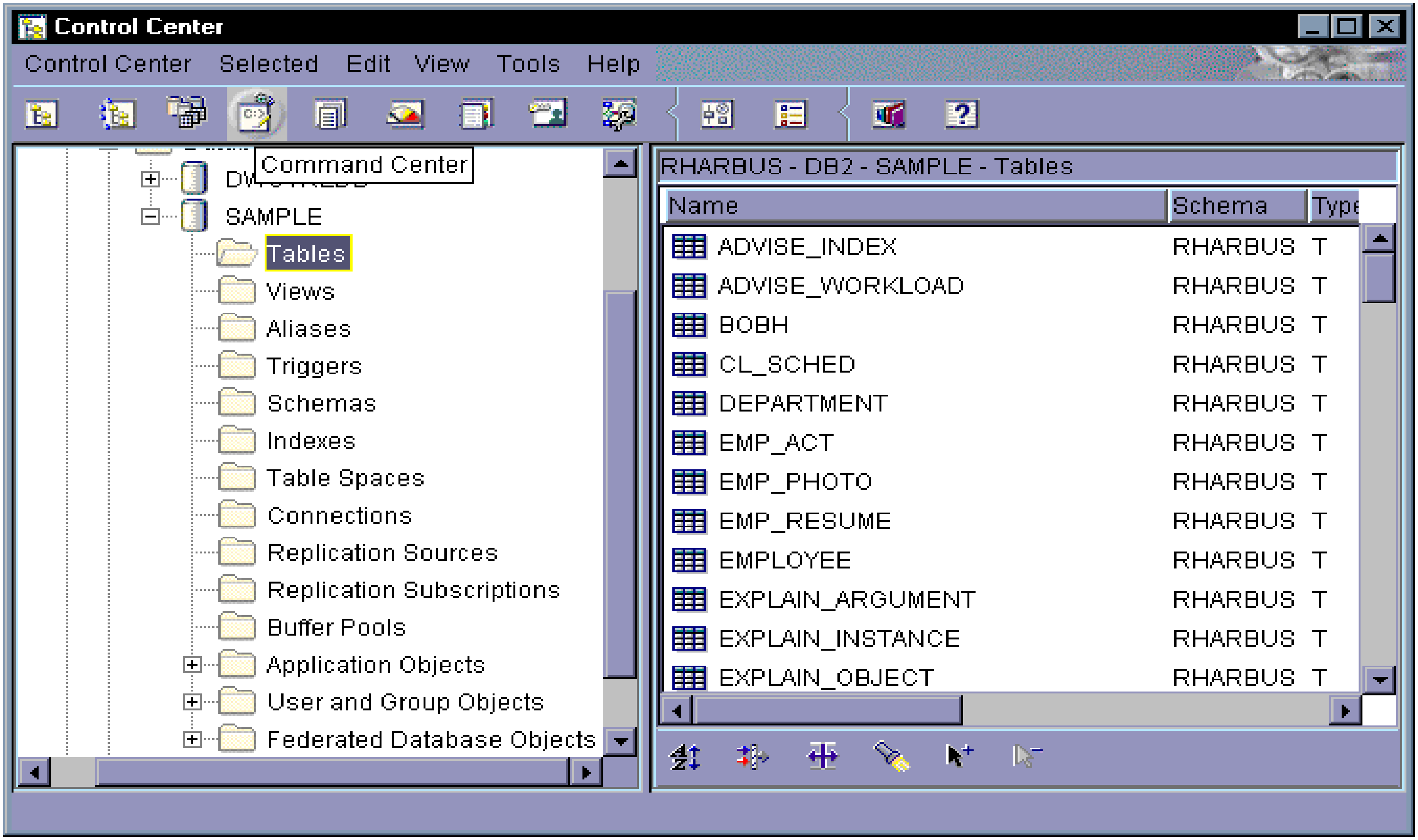
  - Control Center
  - Command Center
  - Command Line Processor
  - Script Center
  - Data Warehouse Center
  - Wizards
  - Visual Explain

- **We look at the first couple in this overview.... let's start with the Control Center ...**

Fundamentals
DB2 UDB Control Center

- **The Control Center is the central point of administration for DB2.**
  - Allows you to administer all platforms, including OS/390 and also setup and maintain your replication environment.
  - Overall launch point for all other tools
  - Java based application and applet
  - You can use it to perform all DBA tasks including:
    - configuration
    - recovery
    - connectivity
    - managing your media
    - monitoring
  - BUT - it only has very minimal SQL ability

Fundamentals
What does the Control Center look like



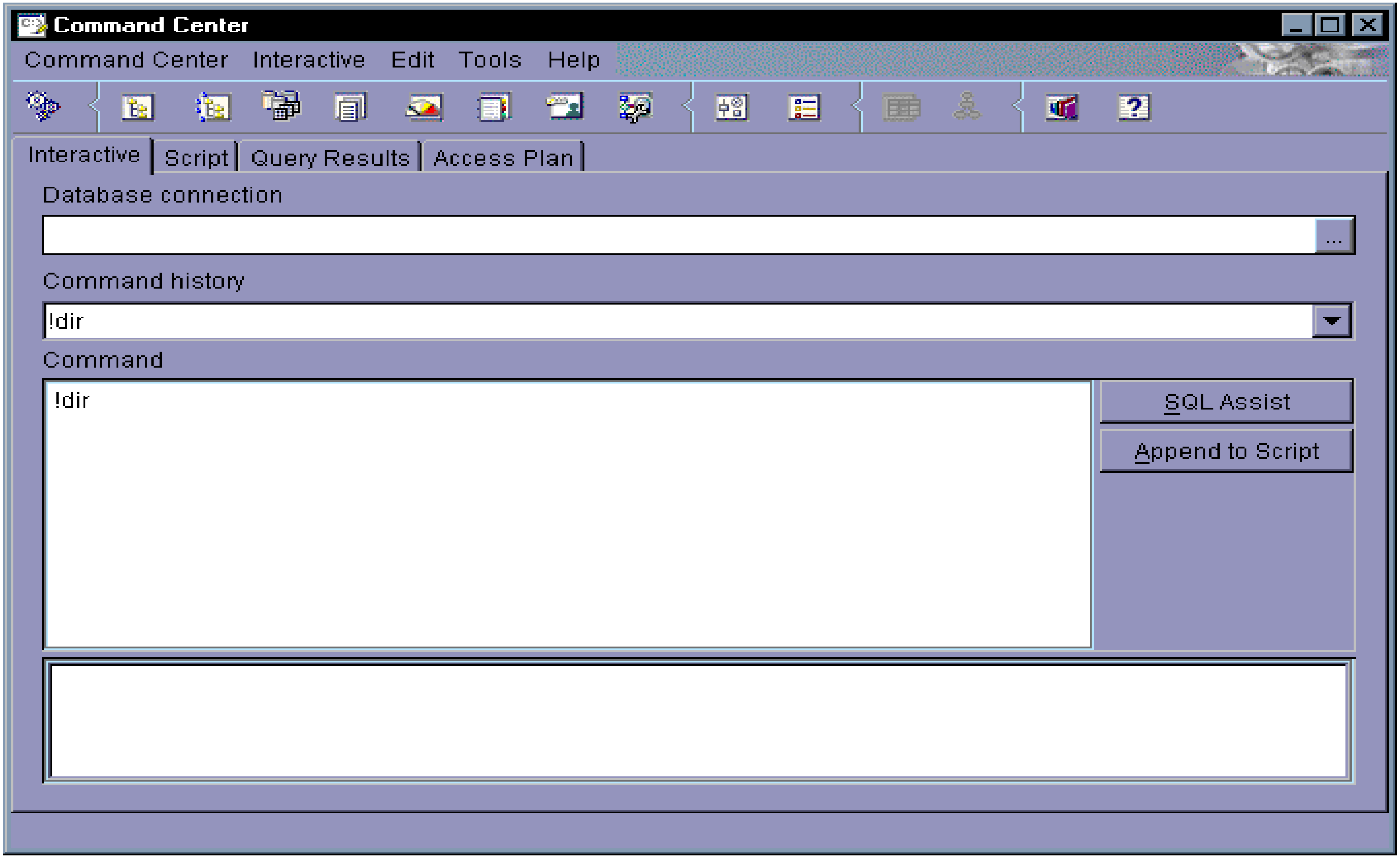
RHARBUS - DB2 - SAMPLE - Tables

Name	Schema	Type
ADVISE_INDEX	RHARBUS	T
ADVISE_WORKLOAD	RHARBUS	T
BOBH	RHARBUS	T
CL_SCHED	RHARBUS	T
DEPARTMENT	RHARBUS	T
EMP_ACT	RHARBUS	T
EMP_PHOTO	RHARBUS	T
EMP_RESUME	RHARBUS	T
EMPLOYEE	RHARBUS	T
EXPLAIN_ARGUMENT	RHARBUS	T
EXPLAIN_INSTANCE	RHARBUS	T
EXPLAIN_OBJECT	RHARBUS	T

Fundamentals
DB2 UDB Command Center

- **This provides an interactive window allowing input of SQL statements or DB2 commands, viewing of execution results and explain information.**
  - Can be invoked from the Control Center or from the start panel (on Intel)
  - You can enter SQL statements, DB2 commands or operating system commands
  - Visual explain can be invoked after SQL invocation to view the access plan information.
  - Essentially the same as the Command Line Processor but more feature rich... you can use it to enter text from an interactive screen.
  - Interacts with the Script Center ... you can create scripts, import existing scripts, execute scripts and schedule scripts.

Fundamentals
What does the Command Center look like?



Database connection

Command history

```
!dir
```

Command

```
!dir
```

Fundamentals
DB2 UDB Command Line Processor

- **The Command Line Processor (CLP) is a component common to all DB2 products.**

- **It is a text-based application that is used to execute SQL statements and DB2 commands. For example, you can create a database, catalog a database and issue dynamic SQL statements from the CLP. You can also execute operating system commands.**

- **You have three modes you can use:**
  - **Interactive mode**
  - **Command line input**
  - **File input**

- **All SQL statements issued from the Command Line Processor are dynamically prepared and executed on the database server. The output, or result, of the SQL query is displayed on the screen by default.**

- **Has is own set of options settings ... you tailor your environment.**

- **Know the GUI tools and what each can do for you ... now lets move on to instances and security**...

Security and Instances

Database Administration Server (DAS)

- There are two types of instances: a DB2 Administration Server (DAS) instance and a DB2 instance. Both of these instances may be created during the installation process.

- The DB2 Administration Server (DAS) instance is a special DB2 instance for managing local and remote DB2 servers. The DAS instance is used by the DB2 administration tools including the Control Center and the Client Configuration Assistant to satisfy requests. The db2admin commands allow you to start, stop, and configure the DAS instance and catalog nodes.

  The DAS instance must be running on every DB2 server that you want to administer remotely or detect using the client configuration search method. The DAS instance provides remote clients with the information required to set up communications to access the DB2 server instances. You can create only one DAS instance for each machine. When the DAS instance is created, the DB2 global-level profile registry variable DB2ADMINSERVER is set to the name of the DAS instance.

Security and Instances
Database Administration Server continued...

- The DB2 Administration Server instance will:

  - Obtain the configuration of the operating system
  - Obtain user and group information
  - Start/Stop DB2 instances
  - Set up communications for a DB2 server instance
  - Attach to an instance to perform administration tasks for a database
  - Provide a mechanism to return information about the DB2 servers to remote clients
  - Collect information results from DB2 Discovery

- The DB2 Administration Server instance will not:

  - Support databases

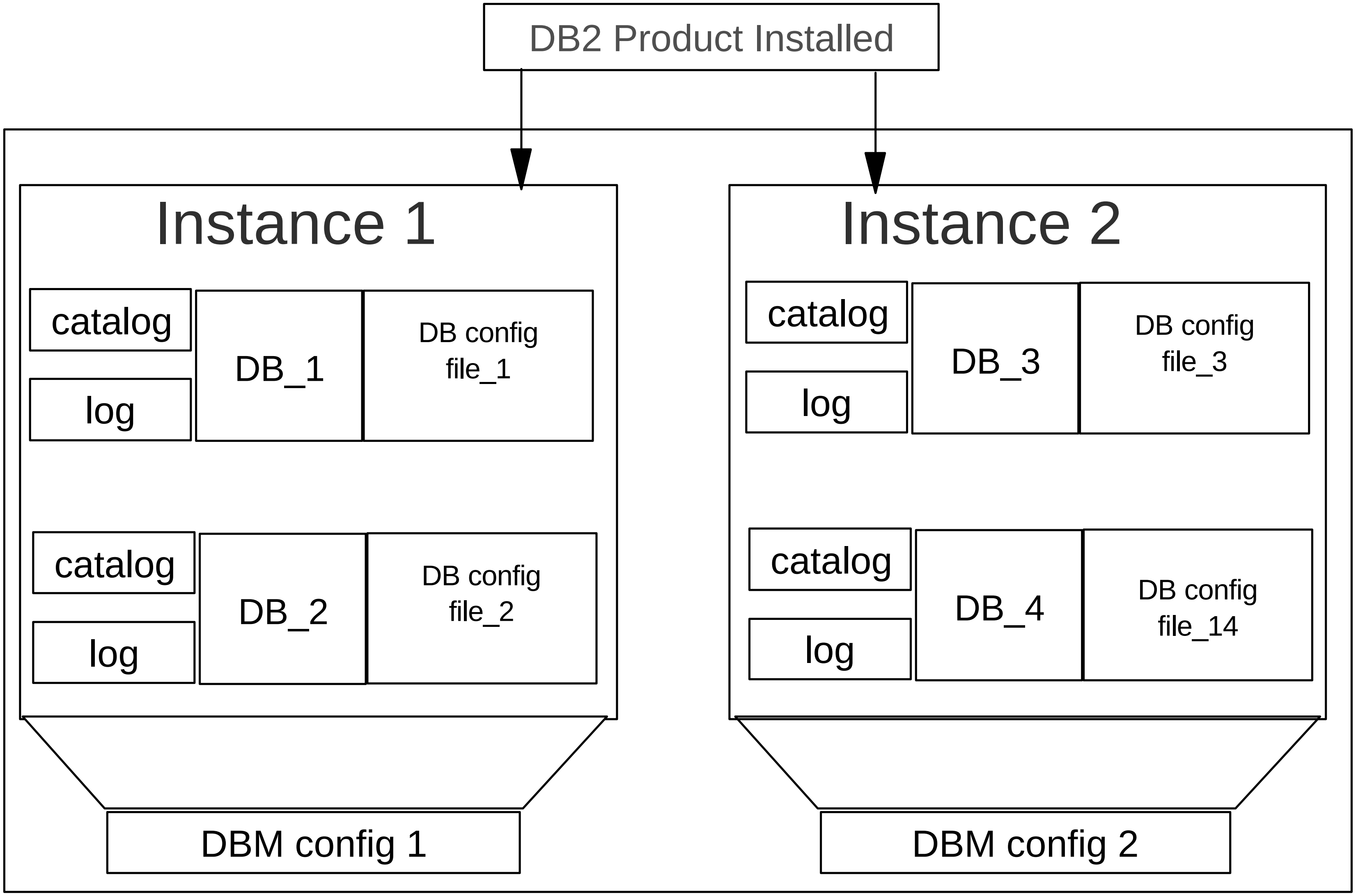
Security and Instances
DB2 Instance

- A DB2 instance is defined as a logical database server environment. DB2 databases are created within DB2 instances on the database server. The creation of multiple instances on the same physical server provides a unique database server environment for each instance. For example, you can maintain a test environment and a production environment on the same machine.

- Each instance has an administrative group associated with it. This administrative group must be defined in the instance configuration file known as the database manager configuration file. Creating user IDs and user groups is different for each operating environment.

- The installation process creates a default DB2 instance. This is the recommended method for creating instances. However, instances may be created (or dropped) after installation.

Security and Instances
Instance overview



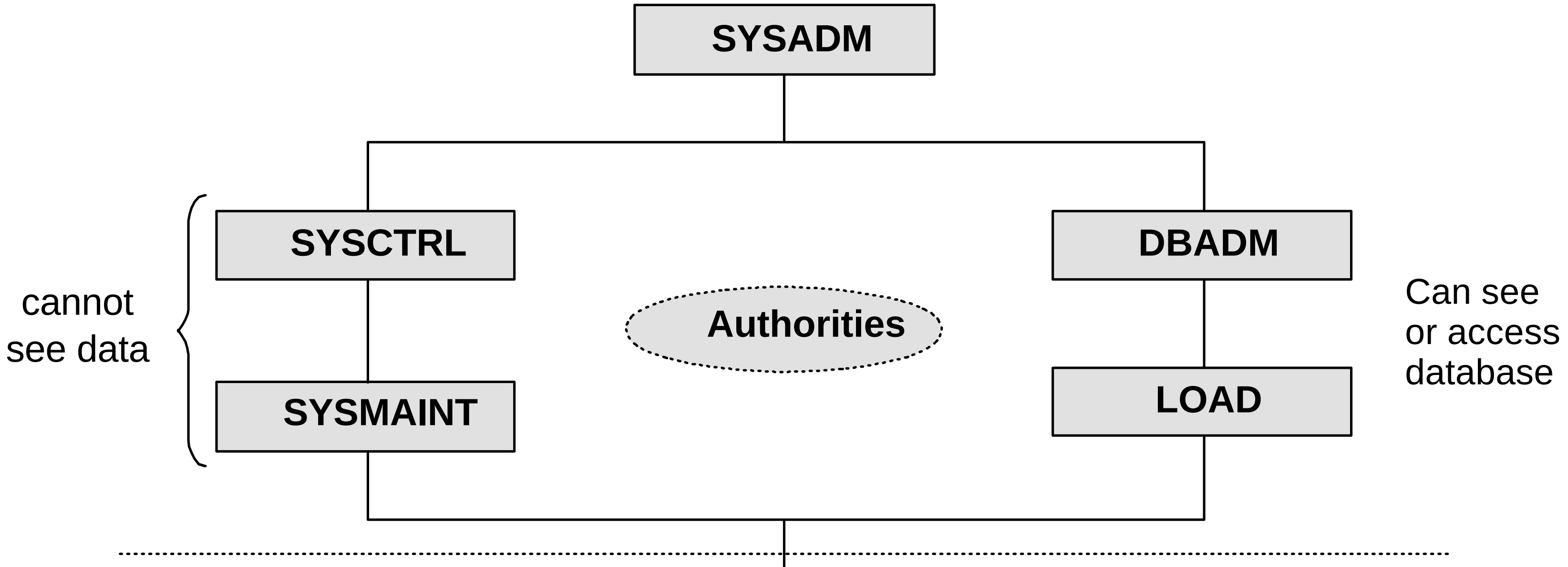
Security and Instances
Security

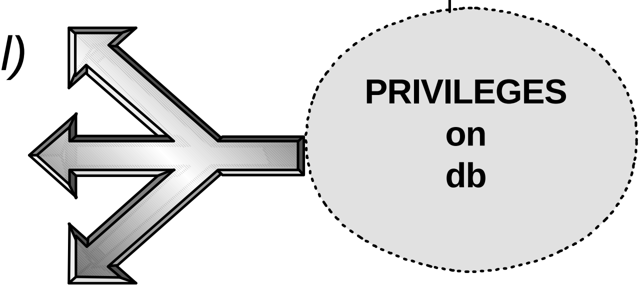
- There are three levels of security that control access to a DB2 system. The first level controls the access to the instance. The second level controls the access to the database. The last level relates to the access of data or data-associated objects within the database.

- All access to the instance is managed by a security facility external to DB2. The security facility can be part of the operating system or a separate product. It allows the system to make sure that the user really is who he or she claims to be and may also control access to other objects like files and programs. Users and groups to which they may belong are normally used by these security facilities. This security level is termed Authentication.

- Access to a database and its data objects is controlled at the database level by the DB2 database manager. Here, administrative authorities and user privileges defined in the database are used to control access.

Security and Instances
Security: Authentication



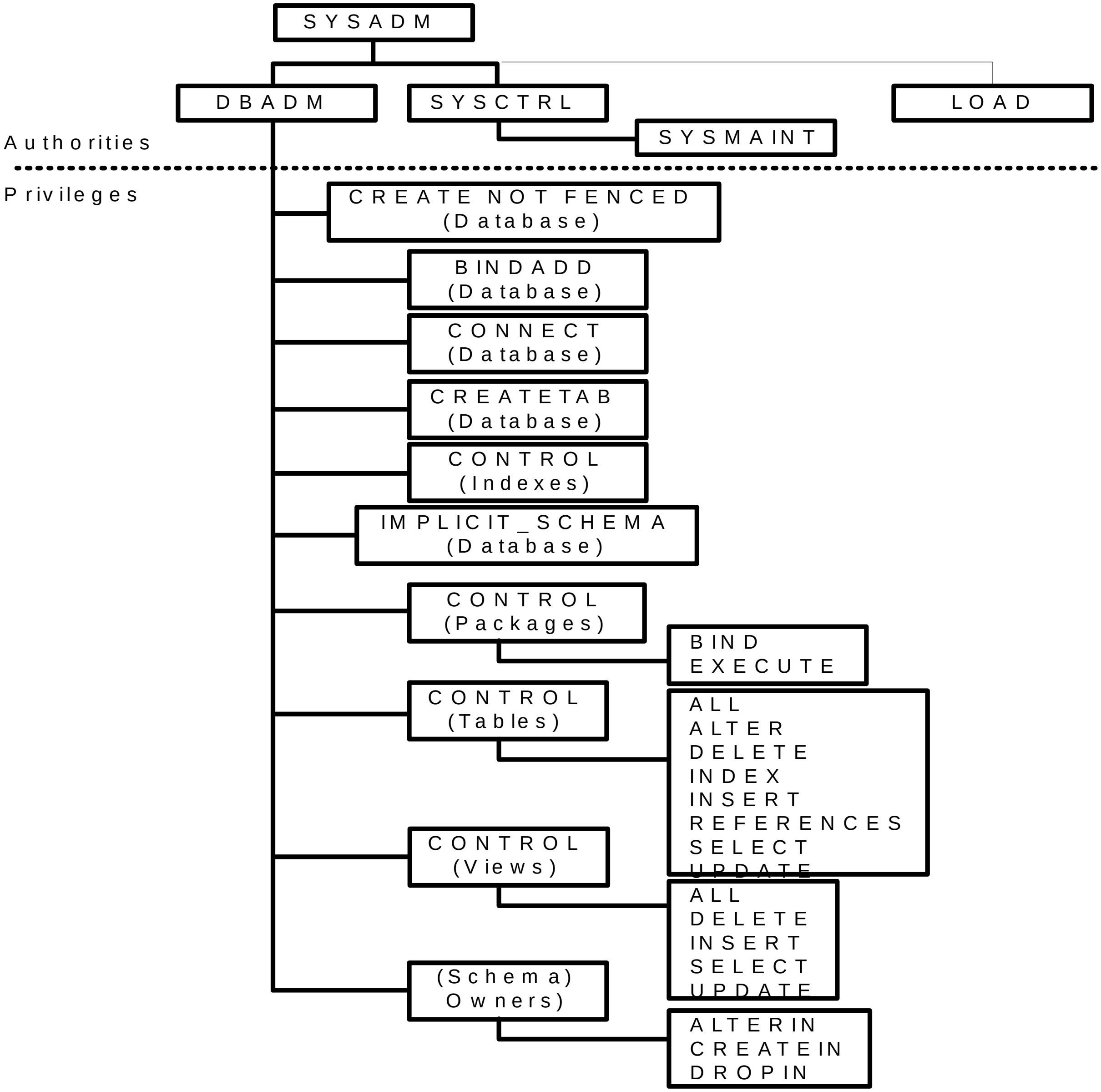


- *Ownership (Control)*
- *Individual*
- *Implicit*

Security and Instances
Security: Controlling Access

Authorities set at group level

db2 update dbm cfg using sysadm_group adm1
db2 update dbm cfg using sysctrl_group ctrl1
db2 update dbm cfg using sysmaint_group maint1



SYSADM

DBADM

SYSCTRL

SYSMAINT

LOAD

Authorities

Privileges

CREATE NOT FENCED (Database)

BINDADD (Database)

CONNECT (Database)

CREATETAB (Database)

CONTROL (Indexes)

IMPLICIT_SCHEMA (Database)

CONTROL (Packages)

BIND
EXECUTE

CONTROL (Tables)

ALL
ALTER
DELETE
INDEX
INSERT
REFERENCES
SELECT
UPDATE

CONTROL (Views)

ALL
DELETE
INSERT
SELECT
UPDATE

(Schema) Owners

ALTERIN
CREATEIN
DROPIN

Security and Instances
Security: Privileges

- A privilege is the right to create or access a database object. DB2 authorities and privileges on database objects are hierarchical in nature. These rights are controlled by users with SYSADM (system administrator) authority or DBADM (database administrator) authority or by creators of objects.

- Privileges can be obtained explicitly or implicitly.
  - **Explicitly granted or implicitly from group membership.**

- Different levels of Privileges
  - **Database**
  - **Schema**
  - **Table and View**
  - **Package**
  - **Index**
  - **Alias, Distinct Type (UDT), User-Defined Function (UDF)**
  - **Nickname and Server (not covered here).**

Security and Instances

Security: Privileges continued.

Privileges can be granted to users or groups

SYSCAT.DBAUTH	Database privileges
SYSCAT.INDEXAUTH	Index privileges
SYSCAT.PACKAGEAUTH	Package privileges
SYSCAT.TABAUTH	Table and view privileges

Implicit Privileges

db2 connect to eddb

db2 grant dbadm on database to user fred

db2 revoke dbadm on database from user fred

What privileges does fred retain?

Special group: PUBLIC

All users belong to this group

Has certain implicit privileges against a database

Unique role in package preparation

DCL (Data Control Language)

Used to provide data object access control

GRANT/REVOKE

Security and Instances
New V7.1 Authorities

- ## LOAD Authority

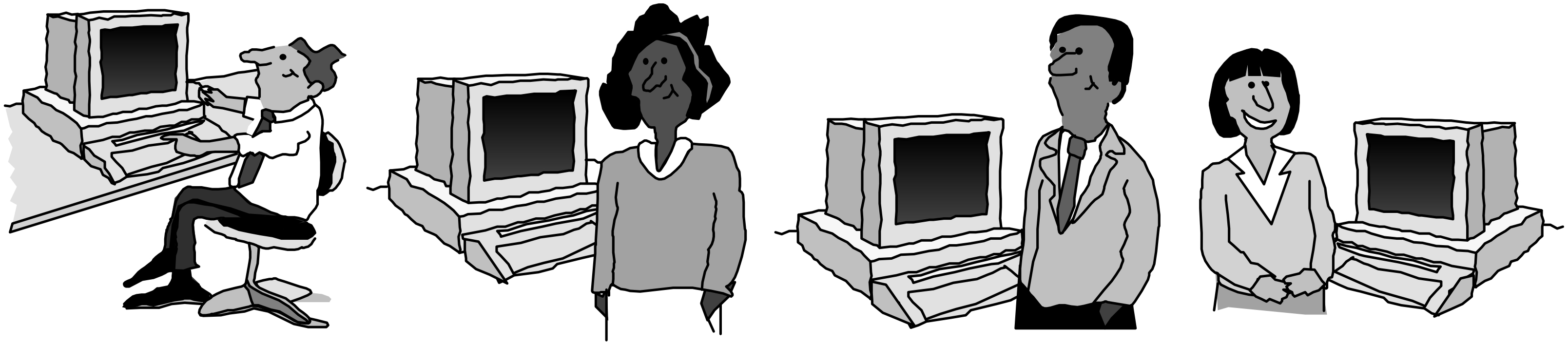
  Users granted LOAD authority can run the LOAD utility without the need for SYSADM or DBADM. This allows users to perform more DB2 functions and gives database administrators more granular control over the administration of their database.

- ## USE OF TABLE SPACE Authority

  This authority allows users to create tables only table spaces that they have been granted access to.

  - owner of table space has USE privilege and ability to GRANT
  - at db creation time, USE priv given to PUBLIC for USERSPACE1
  - not valid with SYSCATSPACE
  - not valid with any TEMPORARY table spaces

Security and Instances
Authority & Privilege Scenario



Bob - End user who executes a program *app1* and use a table to track personal addresses

Clara - Application developer who will develop a a program *app1*

Steve - Needs to load tables

Susan - Wants to be able to create a database to store personal information

NEEDS

EXECUTE on package
CONTROL on table
bob.personal

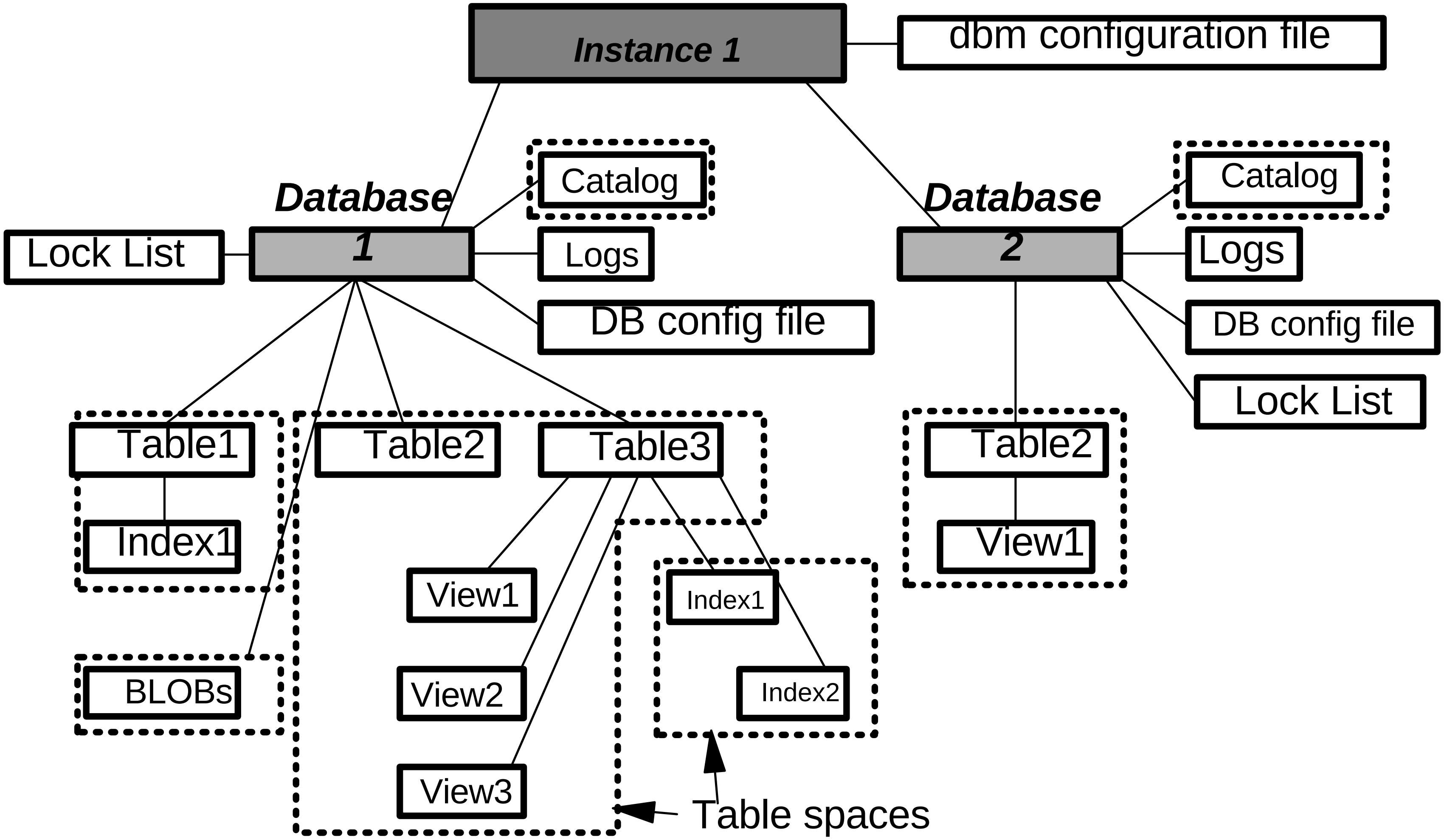
BINDADD on database
access to req'd objects
SELECT, INSERT, UPDATE
DELETE ***on various tables***

DBADM on the **database**
or LOAD authority

SYSADM for the instance

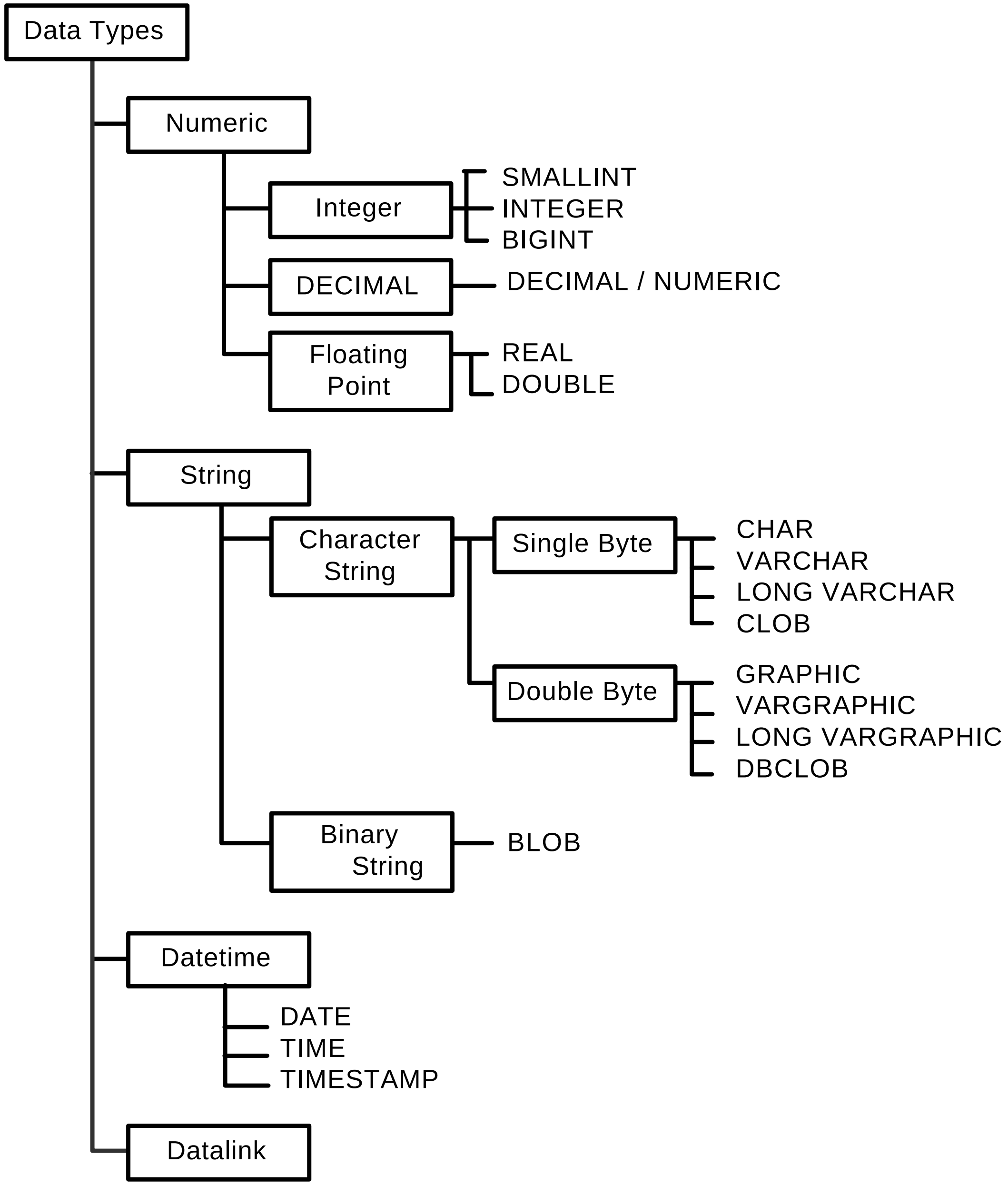
All users require **CONNECT** privilege on the database

Creating and Accessing DB2 Databases Database Objects



- DDL (Data Definition Language)
  Used to create, modify, or drop database objects.

Creating and Accessing DB2 Databases
Data Types



User Defined Types

Need to establish context for values

DB2 enforced typing

IDENTITY Columns

Creating and Accessing DB2 Databases
Storage Architecture

```
<some_directory>/<my_instance>/NODE0000/SQL00001/
   SQLBP.1/2                   // bufferpool info
   SQLSPCS.1/2                 // table space info
   SQLDBCON                    // database configuration
   SQLINSLK/SQLTMPLK           // lock files
   db2rhist.asc/bak            // history file
   SQLOGCTL.LFH                // log control file
   SQLOGDIR/SQL00000.LOG       // log files
           /SQL00001.LOG
   SQLT0000.0/SQL00001.DAT
             /SQL00002.DAT     // SYSTABLES - data pages
             /SQL00002.INX     //           - index pages
             /SQL00002.LB      //           - LOBs
             ...
             /SQLTAG.NAM       // tag file for this container
   SQLT0001.0/SQLTAG.NAM       // tag file for this container
   SQLT0002.0/SQLTAG.NAM       // tag file for this container
```

When you create your database:

- logs are created

- default and additional table spaces are created

- you are creating a basic outline

A DB2 database must exist before any of the database objects can be created in it.

Creating and Accessing DB2 Databases
Object Definition

- You can CREATE or DROP the following objects:

  Table
  Bufferpool
  UDF
  Index
  View
  Schema
  Trigger
  UDT
  Alias
  Event Monitor
  Table Space
  Stored Procedure

- You can DECLARE a TEMPORARY table to be used during the connection.

- But you can only ALTER:

  Table
  Type
  View
  Table Space
  Bufferpool

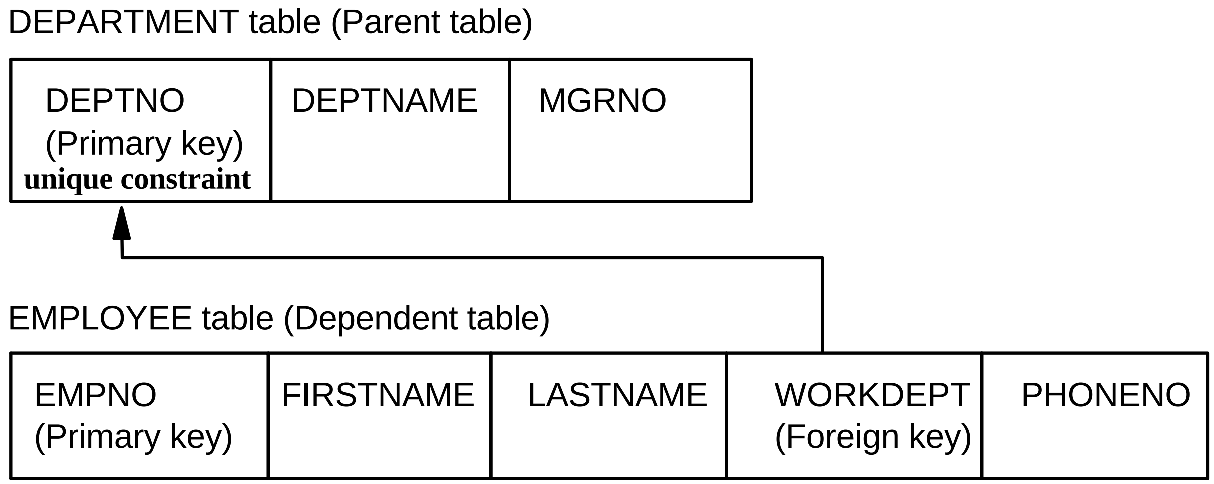
Creating and Accessing DB2 Databases
Constraints: Referential Integrity

- **Referential Constraints are established with the**
  - **Primary Key clause**
  - **Unique constraint clause**
  - **Foreign Key clause**
  - **References clause**

- **in the CREATE/ALTER TABLE statements**

```
create table artists (artno          .............
   primary key (artno)Insert Rules
   foreign key dept (workdept)
   references department on delete no action
```

- **Can be defined on typed-tables**



DEPARTMENT table (Parent table)

DEPTNO (Primary key) **unique constraint**	DEPTNAME	MGRNO

EMPLOYEE table (Dependent table)

EMPNO (Primary key)	FIRSTNAME	LASTNAME	WORKDEPT (Foreign key)	PHONENO

Insert Rules

Rule is implicit, backout
if not found

Delete Rules

Restrict

Cascade

No Action

Set Null

Update Rules

Restrict

No Action

Creating and Accessing DB2 Databases
Constraints: Check Constraints

- Enforce data integrity at a table level.
- Once defined every update/insert must conform, otherwise it will fail.

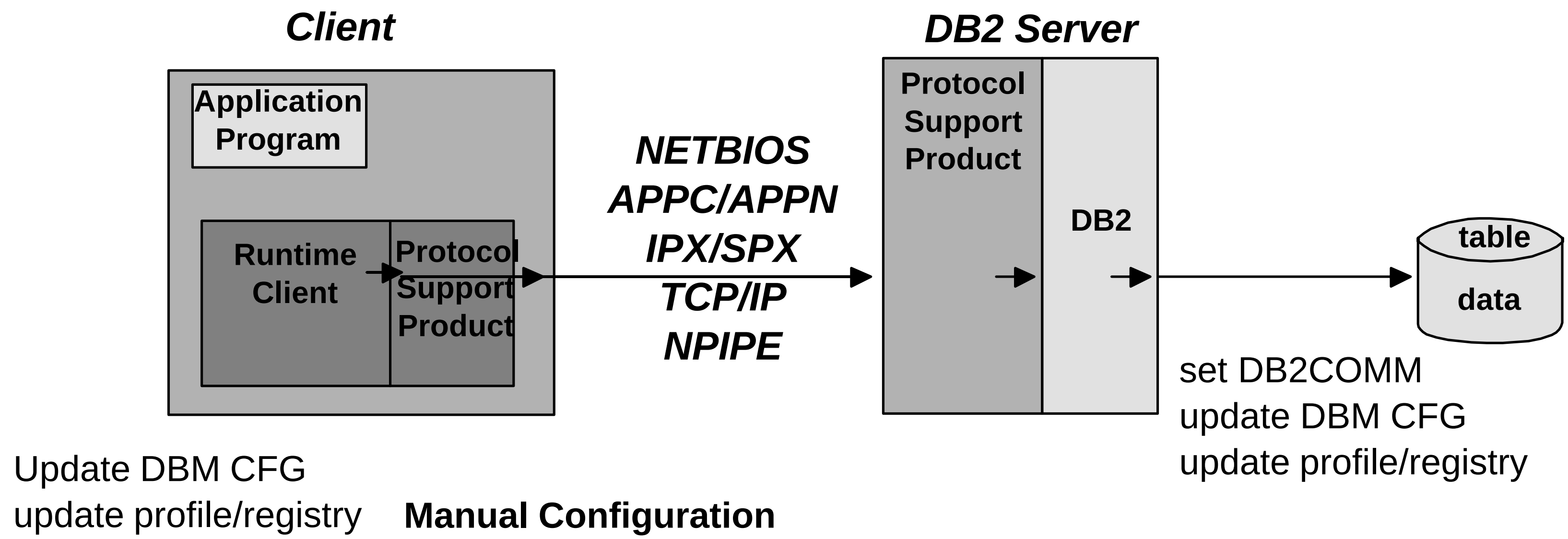
```
        Create table artists
(artno          smallint not null,
 name            varchar(50) with default 'abc',
 classification  char(1) not null,
 bio            clob(100K) logged,
 picture         blob(  2M) not logged compact)
CONSTRAINT classify
CHECK (classification in ('C','E','P','R'))
 in dms01
```

If some rows do not meet the constraint then it will fail.

You can turn off checking, add the data and then add the constraint, but the table will be placed in CHECK PENDING.

To modify a constraint you must drop it and create a new constraint.

Creating and Accessing DB2 Databases Remote Client Connections



Manual Configuration

- **Catalog NODE**
  - **catalog tcpip node mynode remote server1 server 50000**
  - **catalog netbios node jeremy remote N01FCBE3 adapter 0**
  - **catalog npipe node mynode remote jeremy instance db2**
  - **catalog ipxspx node mynode remote * server 00000001.08005AB80EE4.879E**
  - **catalog appc node mynode remote db2cpic00 security same**
- **Catalog DATABASE**
  - **catalog database sample as mysamp at node mynode**
- **Use the Control Center or Client Configuration Assistant**

DB2 Discovery using the Client Configuration Assistant

- **Two Methods, Known or Search**
- **Requires Administration Server to be running on Client and Server**

Using an Access Profile

SQL Usage
Basic SQL

- The SQL Language has been developed around 4 basic verbs used for 4 major tasks:
  - Data Retrieval: SELECT
    - Queries come in many styles: full table, projection, ordering columns, restricting, joins, sorting output
  - Data Addition: INSERT
    - Can insert one or more rows at a time
    - Insert into particular columns
    - You can use subselect to determine values
  - Data Modification: UPDATE
    - Full table
    - Searched with a WHERE clause
    - Positioned using a CURSOR in a program
  - Data Removal: DELETE
    - can apply to single or multiple rows
    - searched or positional

Note: You always need the correct privileges on the object being accessed

SQL Usage
View classifications

- **Views are logical tables that are created using the CREATE VIEW statement. Once a view is defined, it may be accessed using DML statements, such as SELECT, INSERT, UPDATE, and DELETE, as if it was a base table. A view is a temporary table and the data in the view is only available during query processing.**

- **Views can be classified as one of 4 types depending on the operations they allow:**

  - DELETABLE
  - UPDATEABLE
  - INSERTABLE
  - READ-ONLY

- **We will cover more on views in a later section ...**

SQL Usage
Advanced SQL: Triggers

- A trigger is a set of actions that will be executed when a defined event occurs. The triggering events can be the following SQL statements:
  - INSERT
  - UPDATE
  - DELETE

- A trigger can be fired before or after an event.
- Triggers used for
  - VALIDATION
    - Similar to constraints but more flexible
  - CONDITIONING
    - Allows new data to be modified/conditioned to a predefined value.
  - INTEGRITY
    - Similar to RI but more flexible

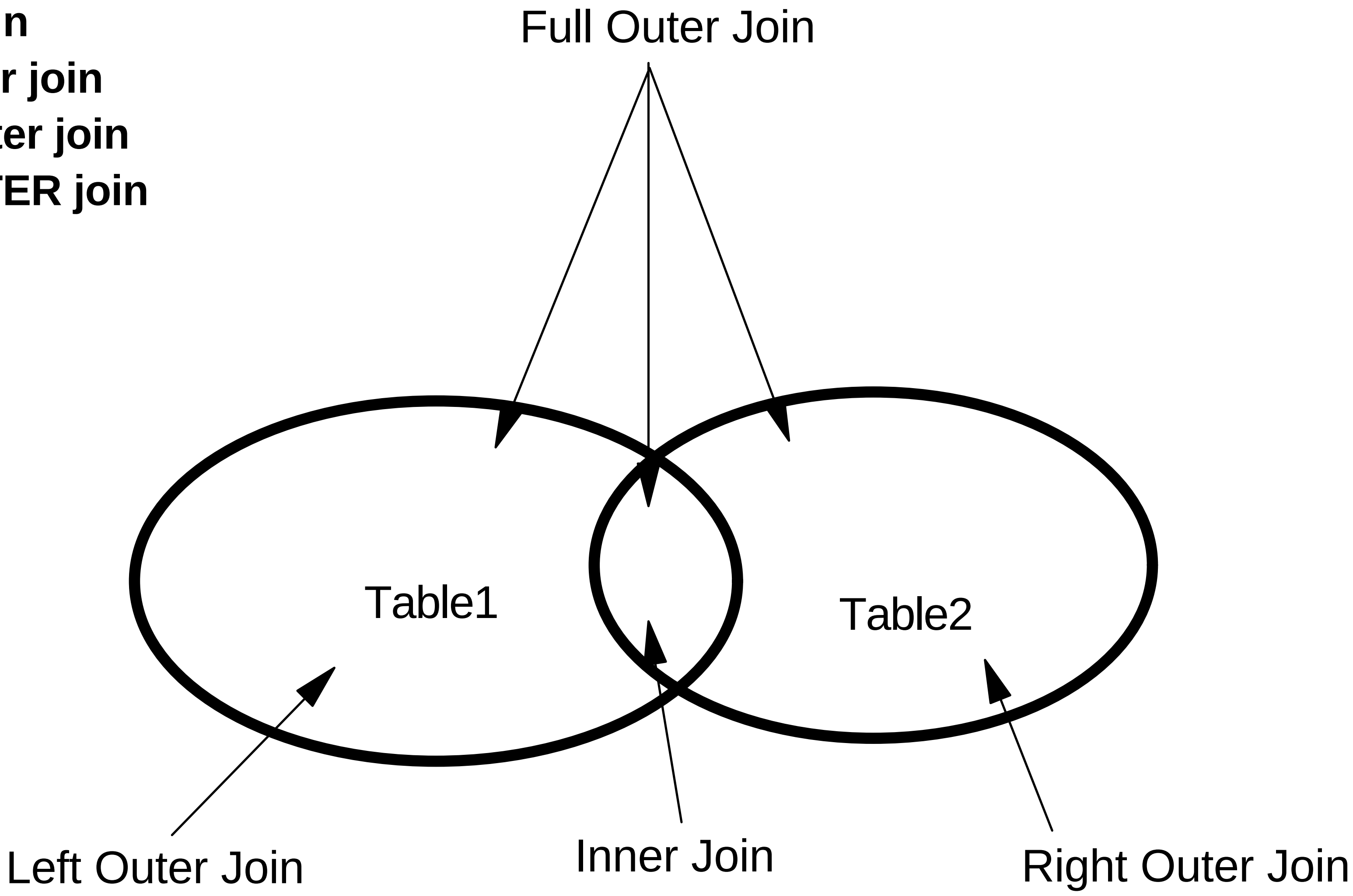
SQL Usage
Advanced SQL: Trigger (AFTER) example

- A trigger is defined to set the value of the column pass_fail dependent on the score attained.

```
CREATE TRIGGER passfail AFTER INSERT ON test_taken
    REFERENCING NEW AS N
    FOR EACH ROW MODE DB2SQL
UPDATE test_taken
SET PASS_FAIL = CASE
         WHEN N.SCORE >=
           (SELECT CUT_SCORE FROM TEST
             WHERE NUMBER = N.NUMBER) THEN 'P'
         WHEN N.SCORE <
           (SELECT CUT_SCORE FROM TEST
             WHERE NUMBER = N.NUMBER) THEN 'F'
         END
         WHERE N.CID = CID
          AND N.TCID = TCID
              AND N.NUMBER = NUMBER
          AND N.DATA_TAKEN = DATA_TAKEN
```

SQL Usage
Advanced SQL: Joins

- **INNER join**
- **OUTER join**
- **LEFT outer join**
- **RIGHT outer join**
- **FULL OUTER join**



SQL Usage

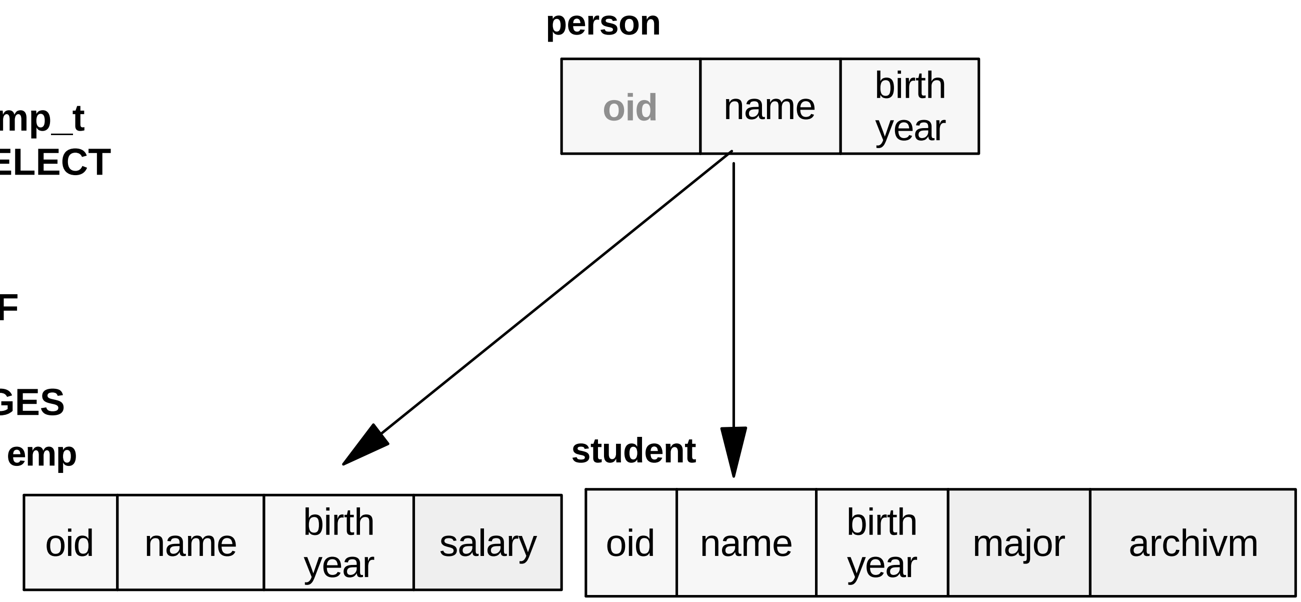
Advanced SQL: Typed Tables and Table Hierarchy

- **Tables that contain structured types are called 'Typed Tables'**
- **Typed tables can inherit attributes from parent table or supertable.**
- **Single inheritance only.**

```
CREATE TABLE person OF
Person_t(REF IS oid USER
GENERATED)

CREATE TABLE emp OF Emp_t
UNDER person INHERIT SELECT
PRIVILEGES

CREATE TABLE student OF
Student_t UNDER person
INHERIT SELECT PRIVILEGES
```



person

oid	name	birth year

emp

oid	name	birth year	salary

student

oid	name	birth year	major	archivm

SQL Usage
Advanced SQL: Typed Table Considerations

- A typed table hierarchy inherits:
  - All attributes
  - Primary key and unique index
    - Primary key or unique indexes can applied only on a root table
  - Check constraints
    - Check constraints are enforced at any level of a Type Hierarchy regardless where they were defined
- RI constraints are not supported for Typed Tables
- Triggers are not supported for Typed Tables
- Typed Tables can not be loaded
- Replication of Typed Tables is not supported
- Considerations for dropping the table(s)

SQL Usage:
Advanced SQL: OLAP Features

- Star Schema
- OLAP Indexes
- Star Joins
- Super Groupings
- Moving Functions

Database Concurrency
Locking

- Control undesirable effects
  - Lost updates/Multiple updates
  - Access to uncommitted data
  - Non repeatable reads
  - PhantomReads
- Locking Controlled by Isolation Level
- DB2 provides different levels of protection to
- Isolate the data.
  - Uncommitted Read
  - Cursor Stability
  - Read Stability
  - Repeatable Read
- For embedded SQL the level is set at bind time
- For dynamic SQL the level is set at run time
- Default is Cursor Stability
- Different types of locks, what you can lock on, how many and how long

Database Concurrency
Isolation Level: Uncommitted Read

- Also known as DIRTY READ.
- It is the lowest level of isolation.
- least restrictive/most concurrency
- It may return data that is never committed to the database.
- Non-repeatable reads & Phantom reads
- may occur when using this level.

Database Concurrency Isolation Level: Cursor Stability

- Locks on any row touched in a unit of work.
- It is the default for applications.
- Cannot return uncommitted data.
- Non-repeatable Read and Phantom reads
- may occur when using this level.

Database Concurrency
Isolation Level: Read Stability

- Locks only the rows an application
- retrieves within a unit of work.
- If the application issues the same query
- more than once within a unit of work,
- additional 'phantom' rows may be seen.
- Cannot return uncommitted data.

Database Concurrency
Isolation Level: Repeatable Read

- Locks all the rows an application retrieves within a unit of work; every row referenced is locked.(Note: optimizer may choose to do a TABLE lock)
- A SELECT statement issued by an application twice within a unit of work gives the same result each time.
- No other applications can update, delete, or insert a row that would affect the result table until the unit of work completes.

Database Concurrency Locking Information

- **DB2 uses record (row) level locking by default**
- **Locks can be held at both the row and table level**
- **The compatibility of a lock to another will determine whether or not the lock will be granted or whether the application requesting the lock will wait.**
- **Locks can be:**
  - ► Converted to a different mode
  - ► Escalated to a higher level (i.e. from row to table)
- **Lock Issues**
  - ► Lock Waits/Timeouts
  - ► Deadlocks

Database Concurrency
Declared Temporary Tables

- **a temporary table that is only accessible to SQL statements that are issued by the application which created the temporary table.**
- **does not persist beyond the duration of the connection of the application to the database.**
- **use them to possibly improve performance of your applications.**
  - no entry inserted into catalog tables ⇨ no catalog contention
  - no locking on table or its rows
  - no logging of table or its rows
  - not subject to name collision
- **BUT! ... you knew there had to be one ... no ALTER, COMMENT, GRANT, LOCK, RENAME or REVOKE ability or IMPORT/LOAD and of course do not forget that you need a USER TEMPORARY TABLE SPACE defined.**

Database Concurrency

Declared Temporary Tables -

YES there is more!

- **If you issues a ROLLBACK stmt for a transactions which contains a TEMPORARY table declaration ... that table is dropped.**
- **If you issue a DROP TABLE statement for a TEMPORARY table declaration, a ROLLBACK statement only restores the table as an empty table.**
- **When you COMMIT the transaction, the default behavior is to DELETE all rows in the table.**
  - avoidance? ... "**ON COMMIT PRESERVE ROWS"** option on create
- **OH BOY! If you attempt to modify the table contents using INSERT/UPDATE/DELETE and you do a ROLLBACK or one of the statements fails, all rows are deleted!**

Database Concurrency

Controlling Transactions with Savepoints

- Savepoint? A mechanism of undoing work by DB2 when a database request fails.
- Savepoint? Makes non-atomic db requests behave atomically.
- Error occurs during execution? Savepoint can be used to undo changes made by the transaction between the time of the savepoint and the time the savepoint rollback is requested.
- Similar to a compound SQL statement.

Certification Preparation
Moving on ...

- **We have just completed the quick review for the DB2 UDB V7.1 Fundamentals Exam.**
  - We have done a quick review
  - Ensure you are prepared

- **Questions?**

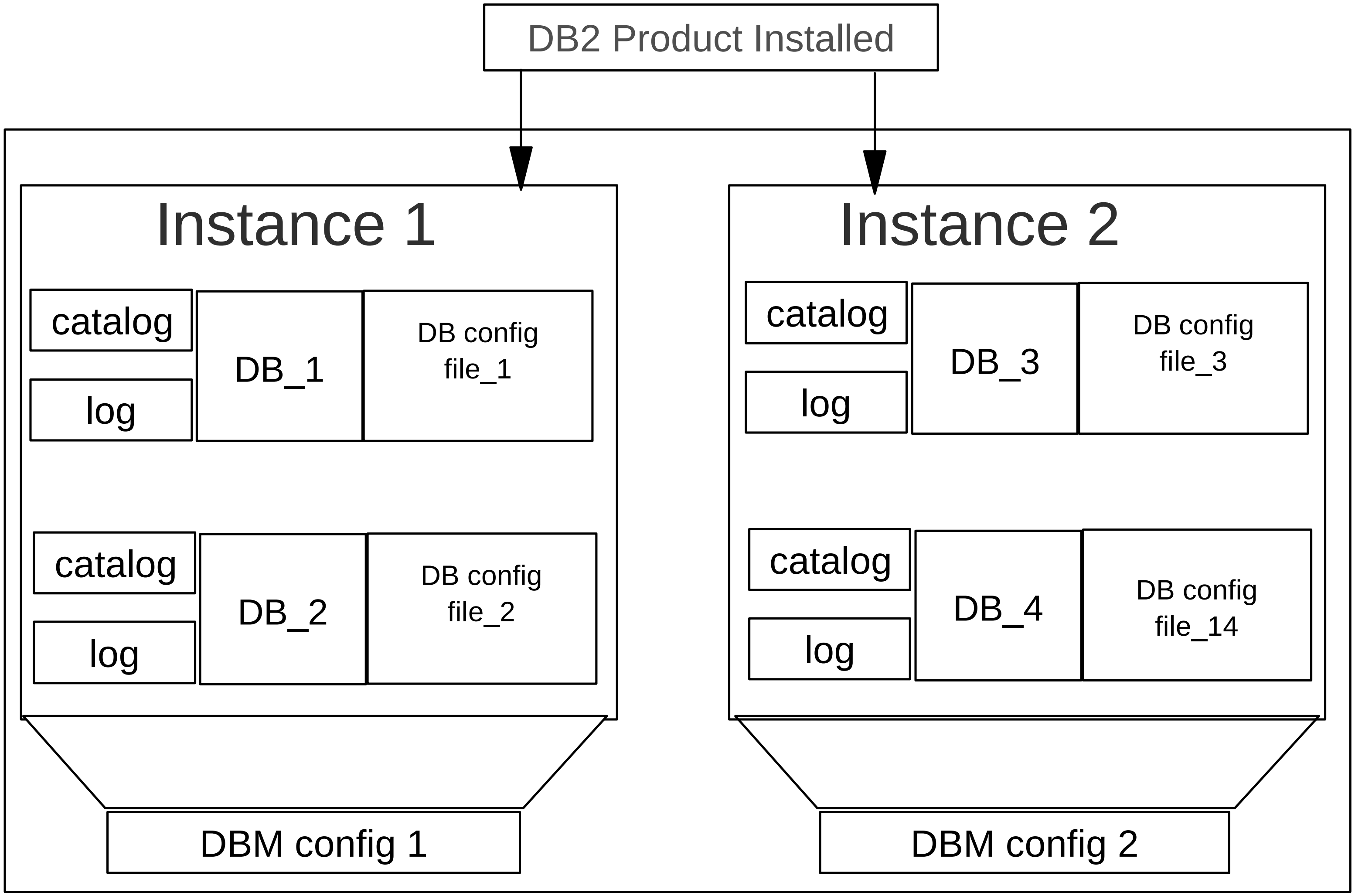
- **That was just the beginning ... Lets move on to Database Administration ...**

Database Administration

- The sections we are going to be covering for the Database Administration component today include:
  - DB2 Server Management - This is next
  - Data Placement
  - Database Access
  - Database Recovery and Maintenance
  - Monitoring DB2 Activity and Problem Determination

So as you can see we have a lot to go through...

DB2 Server Management
Configure and Manage your instances



DB2 Server Management
DB2 Instances

- **DB2 DAS vs. DB2 Instance**
  - What is each used for?
  - How do the two related to each other?
  - How to you maintain each? db2admin? db2? db2set?
    - **start and stop?**
    - **update configuration?**
  - Communications support?
  - Managing an instance using ATTACH

DB2 Server Management
Managing Groups and Users

- **Where are DB2 database users defined?**
  - Inside of DB2?
  - Outside of DB2?
- **Where are user authorities and privileges kept?**
  - System Catalog tables?
  - Configuration files?
  - Operating system files?
- **How can you maintain your users and groups?**
  - Control Center?
  - Command Line SQL statements?
  - Operating System commands?

DB2 Server Management Authorization Levels

There are five authorities in DB2:

➢ SYSADM	System Administration Authority
➢ SYSCTRL	System Control Authority
➢ SYSMAINT	System Maintenance Authority
➢ DBADM	Database Administration Authority
➢ LOAD	Load Table Authority

Authorities in DB2 provide a hierarchy for administration capabilities as we saw earlier.

Authorities are assigned to a group of users. Exceptions to this rule include the DBADM and LOAD authority since they can be granted to a user or a group.

If a certain authority is given to a group, each member of that group has the same DB2 authority unless they have had that authority revoked explicitly.

The LOAD authority was introduced in DB2 UDB Version 7.1. This allows database administrators to grant authority to users or developers for bulk loading of data into DB2, without requiring them to have DBA capabilities.... we covered this in more detail in the Fundamentals piece today.

DB2 Server Management
Managing your Authorization Levels

DBM Configuration file:

```
SYSADM group name (SYSADM_GROUP)              =
SYSCTRL group name (SYSCTRL_GROUP)            =
SYSMAINT group name (SYSMAINT_GROUP)          =
.
.
DB2 UPDATE DBM CFG USING SYSCTRL_GROUP DB2CNTRL
db2stop
db2start
.
.
SYSADM group name (SYSADM_GROUP)              =
SYSCTRL group name (SYSCTRL_GROUP)            =  DB2CNTRL
SYSMAINT group name (SYSMAINT_GROUP)          =
```

Using the GRANT/REVOKE statement:

Load and **DBADM** authorities are granted similar to other database privileges:

```
GRANT LOAD ON DATABASE TO USER BILL
```

DB2 Server Management
Setting user/group access to objects? (0)

A privilege is the right to create or access a database object.
DB2 authorities and privileges on database objects are hierarchical in nature... we know this.
Privileges are stored in the system catalog tables within the database... we already know this.

There are three groupings of privileges: **Ownership, Individual, and Implicit.**

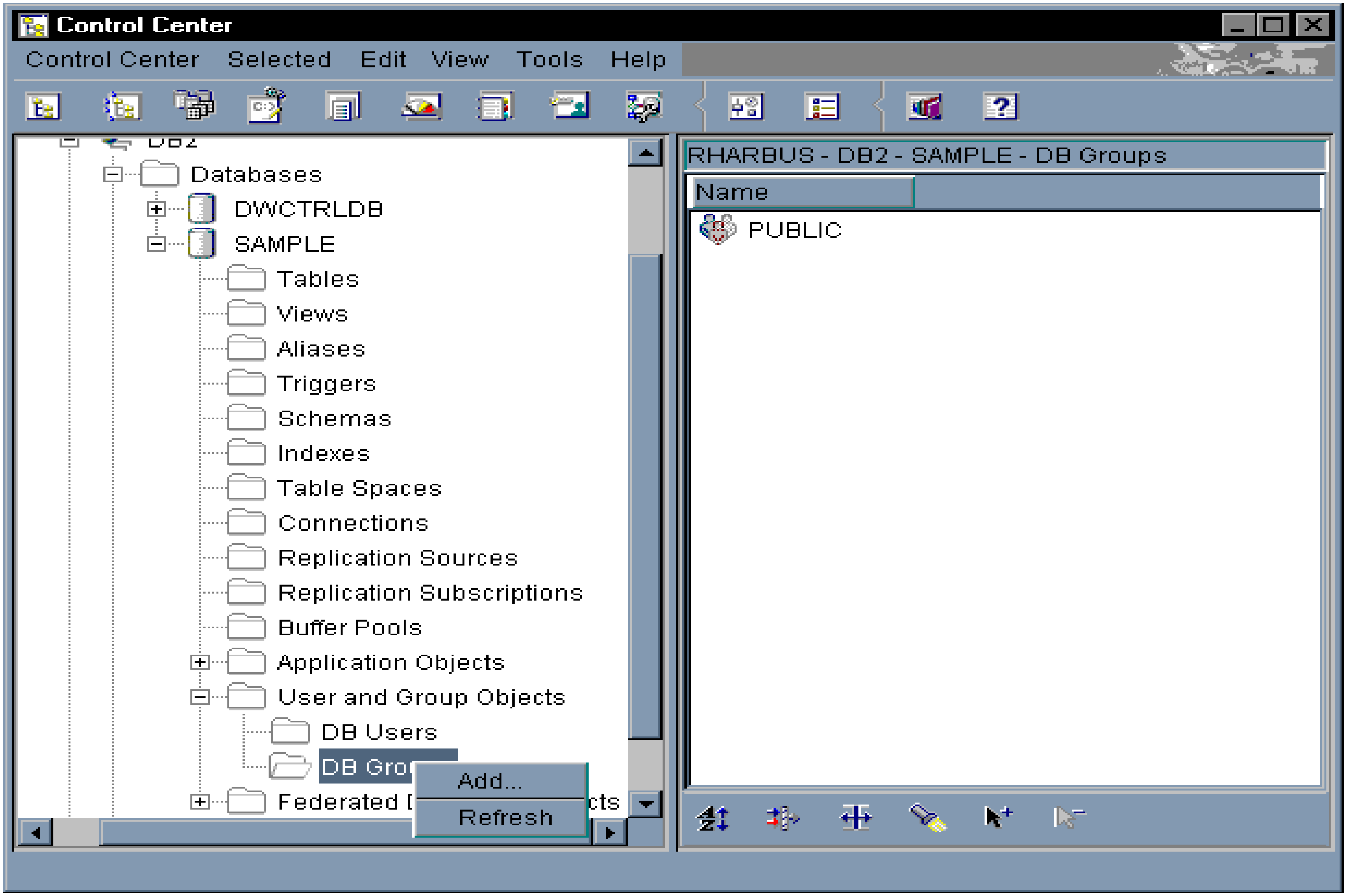
Ownership or **CONTROL** privileges - CONTROL privilege is automatically granted to the creator of an object. Having CONTROL privilege is like having ownership of the object. You have the right to access an object, give access to others, and give others permission to grant privileges on the object. Privileges are controlled by users with ownership or administrative authority. These users can grant or revoke privileges using GRANT or REVOKE SQL statements.

Individual privileges - These are privileges that allow you to perform a specific function, sometimes on a specific object. These privileges include SELECT, DELETE, INSERT, and UPDATE.

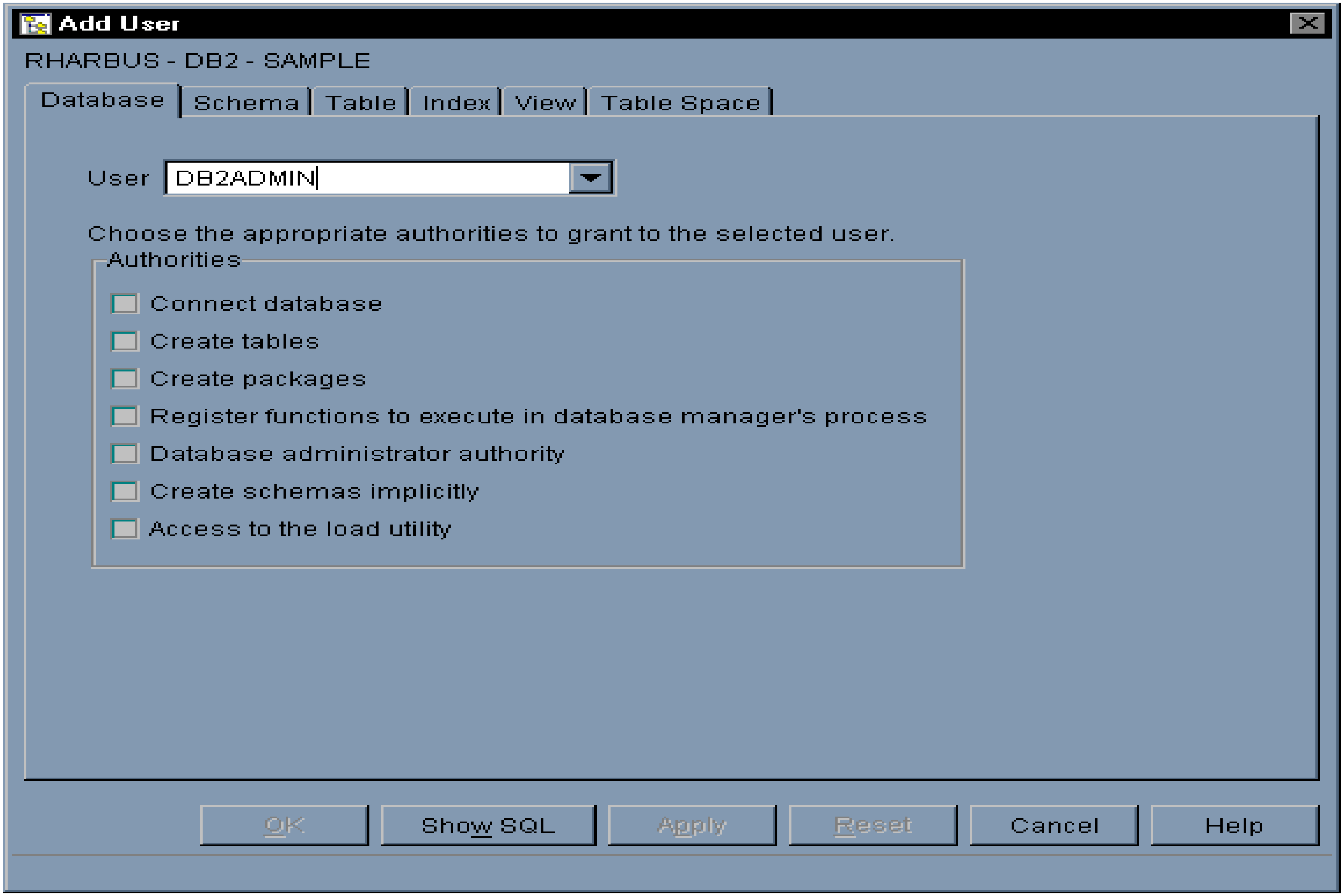
Implicit privileges - An implicit privilege is one that is granted to a user automatically when that user is explicitly granted certain higher level privileges. **These privileges are not revoked when the higher level privileges are explicitly revoked**.... take the CONTROL privilege for example and remember the REVOKE ALL statement.

In addition, an implicit privilege can also be associated with a package. As an example, when a user executes a package that involves other privileges, they obtain those privileges while executing the package. The do not necessarily require explicit privilege on the data objects accessed within the package. These privileges can also be called **indirect** privileges.

DB2 Server Management
Setting user/group access to objects? (1)



DB2 Server Management
Setting user/group access to objects? (2)



RHARBUS - DB2 - SAMPLE

User DB2ADMIN

Choose the appropriate authorities to grant to the selected user.

Authorities

- □ Connect database
- □ Create tables
- □ Create packages
- □ Register functions to execute in database manager's process
- □ Database administrator authority
- □ Create schemas implicitly
- □ Access to the load utility

DB2 Server Management
Setting up client/server connectivity

- **DB2 directories**
- **Enabling protocols to use**
- **Automated configuration using Discovery**
  - How to enable or limit what discovery can do
  - Searched discovery
  - Known discovery
- **Automated configuration using Access Profiles**
  - Server profile
  - Client profile
- **Manual Configuration**
  - Client Configuration Assistant
  - DB2 commands to catalog

Database Administration

- The sections we are going to be covering for the Database Administration component today include:
  - DB2 Server Management - DONE
  - Data Placement - This is next
  - Database Access
  - Database Recovery and Maintenance
  - Monitoring DB2 Activity and Problem Determination

Data Placement
Storage Concepts

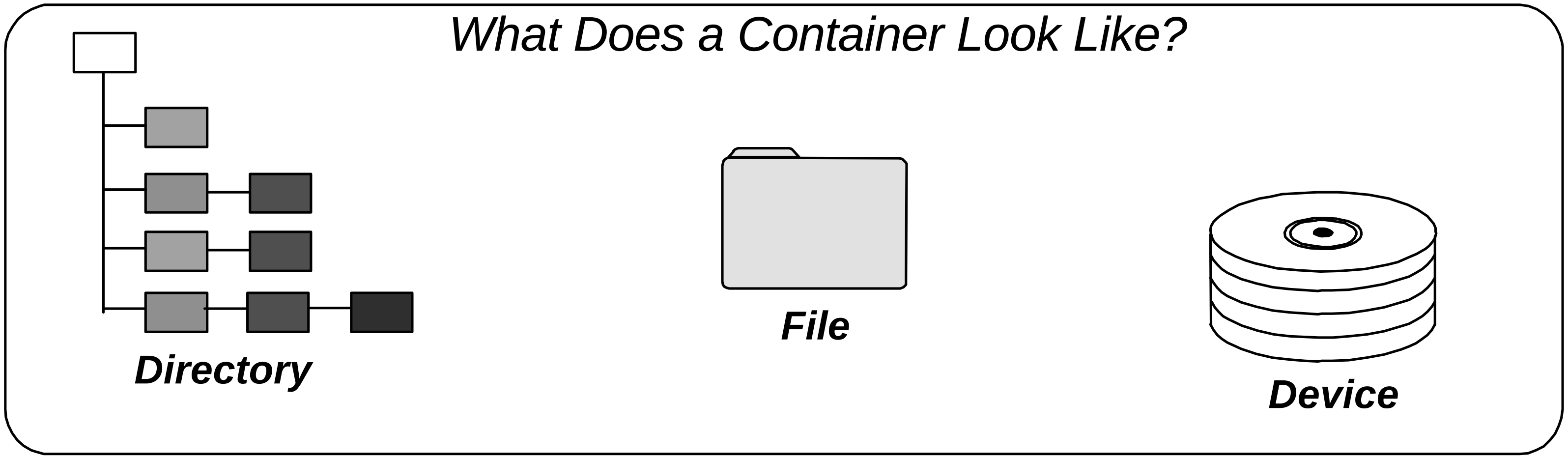
- **Tables, indexes, and catalogs reside in** *table spaces*
- **A** *table space* **consists of one or more containers**
- **A** *container* **is a file or (raw) device name and is assigned to a table space**
- **An** *extent* **is a unit of space allocation within a container**
- **A** *bufferpool* **caches the data in memory**
- **A** *table space* **is assigned to a** *bufferpool* **and a given** *bufferpool* **can have multiple** *table spaces* **assigned to it**
- **A** *database partition* **consists of tables (or horizontally partitioned table parts) and associated indexes & logs**
- **A** *nodegroup* **identifies a set of** *database partitions* **in a partitioned database (DB2 Extended - Enterprise Edition)**
- **A** *table space* **is assigned to a** *nodegroup* **(possibly default) and a given** *nodegroup* **can have multiple** *table spaces* **assigned to it**

Data Placement
Table spaces

- **All database objects are stored within table spaces**
  - REGULAR
  - LONG
  - TEMPORARY
    - DB2 V7.1 supports **system** temporary table spaces and **user** temporary table spaces. A system temporary table space must exist for DB2 to operate properly. User temporary table spaces can be used to place temporary tables.
- **Two types of table spaces:**
  - **DMS** (database managed space) and **SMS** (system managed space)
- **Tables paces are either 4K, 8K, 16K or 32K pages.**
  - 4K is default size.
  - Can not mix page sizes within a table space
  - Must be associated with a bufferpool with same page size
- **Table space composed of one or more containers.**
- **Data allocated by extents within containers.**
- **Three table spaces created by default (all SMS)**
  - SYSCATSPACE - system catalog tables.
  - USERSPACE1 - default user data.
  - TEMPSPACE1 - temporary data.

Data Placement
Containers and Table Spaces

- Container is an Allocation of Physical Space

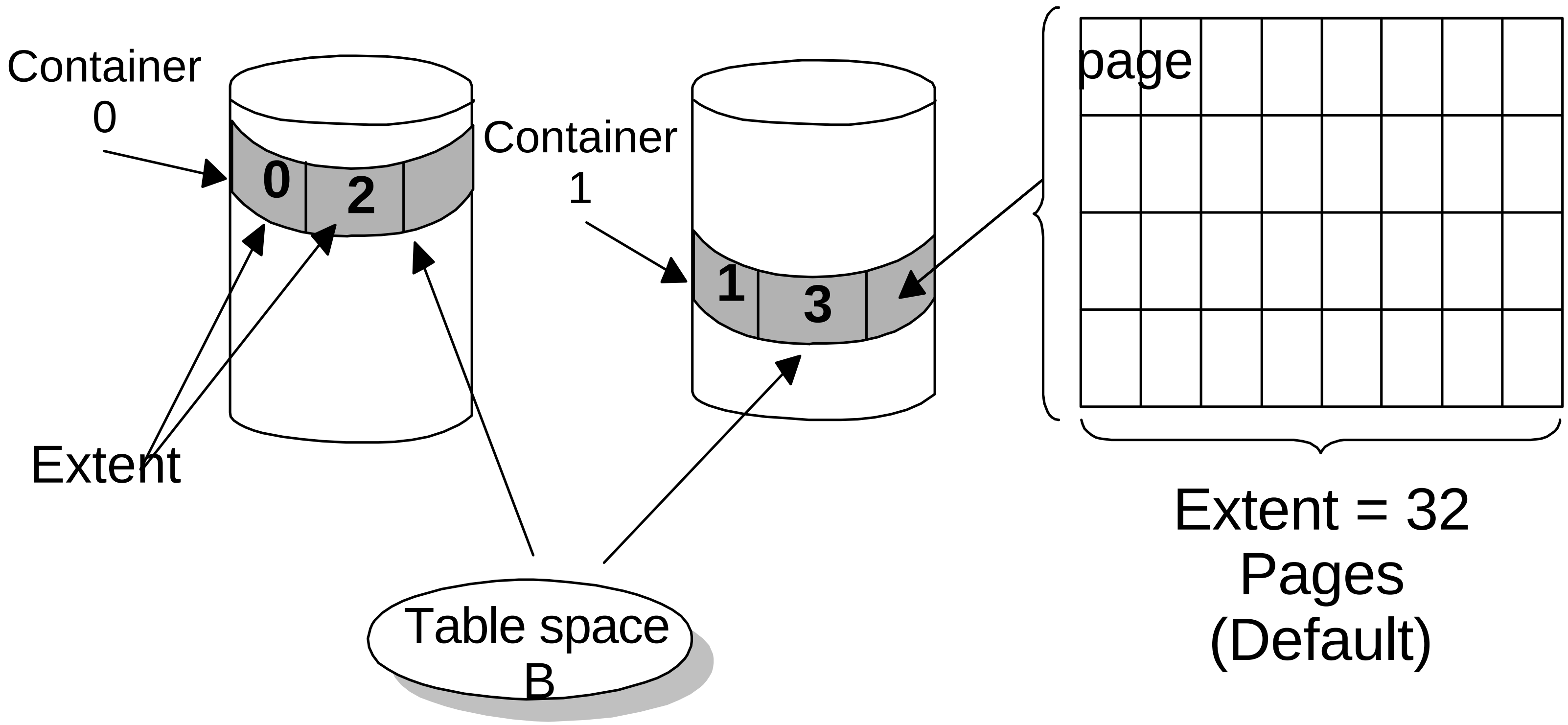


What Does a Container Look Like?

	File	Directory	Device
Intel	DMS	SMS	N/A
UNIX	DMS	SMS	DMS
Windows NT	DMS	SMS	DMS

Data Placement
Containers and Extents

- DFT_Extent_SZ defined at database level
- EXTENTSIZE defined per table space
- Once defined, EXTENTSIZE cannot be changed.
- Extents consist of multiple pages.

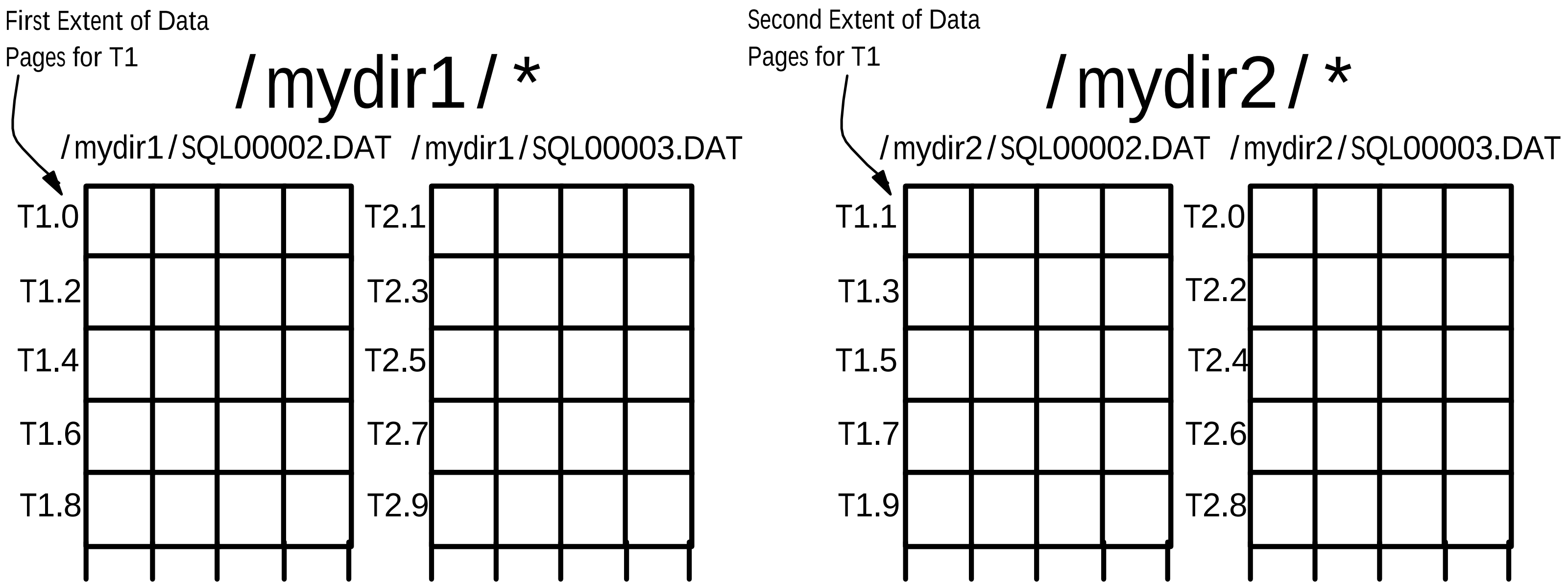


Data Placement
SMS Characteristics

- **Data is stored in files**
- **All table data and indexes share the same table space**
- **Dynamic file growth**
- **I/O parallelism is in effect**
- **Upper boundary on size governed by:**
  - Number of containers
  - Operating System limit on size of file system
  - Operating System limit on size of individual files
- **New containers cannot be added dynamically unless doing a redirected restore**
- **UNIX: file system size may be increased**
- **Very easy to administer**

Data Placement
SMS table spaces

- **What happens on disk during the following ?**
  - db2 create tablespace TS1 managed by system using ('/mydir1', '/mydir2'') extentsize 4
  - db2 create table T1 (c1 int ...) in TS1
  - db2 create table T2 (c1 float ...) in TS1



Data Placement
SMS table space summary

- **Containers are operating system directories.**
  - ►Can increase table space capacity be enlarging underlying operating system file system - if possible.
- **Data striped across container by extent.**
- **Disk space allocated *on demand* .**
  - ►One page at a time (default).
  - ►Use *db2empfa* utility to enable multiple page allocation.
- **Data "objects" (i.e. table data, indexes, LONG VARCHARs) located by operating system file name.**
- **Cannot separate data, index and large object data.**

Associate each container (i.e. directory) with a different file system
... otherwise table space capacity limited to that of a single file system

Ensure containers have equal capacity (roughly)
... excess in larger containers isn't exploited

Data Placement
DMS Characteristics

- **Space allocated at creation time**
- **Containers can be added (data is rebalanced)**
- **You "can" extend or resize a container or containers**
- **Automatic rebalancing if needed**
- **Capacity limited only by physical storage**
- **File system I/O used for DMS-file manipulation**
- **Direct I/O used for DMS-raw manipulation**
- **High performance potential (especially for OLTP)**
- **Flexible data placement**
- **Can split table objects (i.e. data, index, long field data) into different table spaces.**

Data Placement
DMS table space summary

- **Containers are either operating system files or raw devices.**
  - Can increase table space capacity with 'Alter tablespace add container' command or "Alter tablespace extend/resize .."
- **Data striped across containers by extent.**
- **Disk space allocated at tablespace *creation.***
  - Space Map Pages (SMP) keep track of what extends are used and which are free.
- **Data "objects" located by**
  - OBJECT TABLE - locates first extent in the object.
  - Extent Map Pages (EMPs) for the object - locate other extents in the object.

Associate each container with a different disk(s)
... enables parallel I/O, larger table space capacity

Data Placement
DMS table spaces

- **What happens on disk during the following ?**
  - db2 create tablespace TS2 managed by database using (file '/myfile' 1024, device '/dev/rhd7' 2048) extentsize 4 prefetchsize 8
  - db2 create table T1 (c1 int ...) in TS2
  - db2 create table T2 (c1 float ...) in TS2

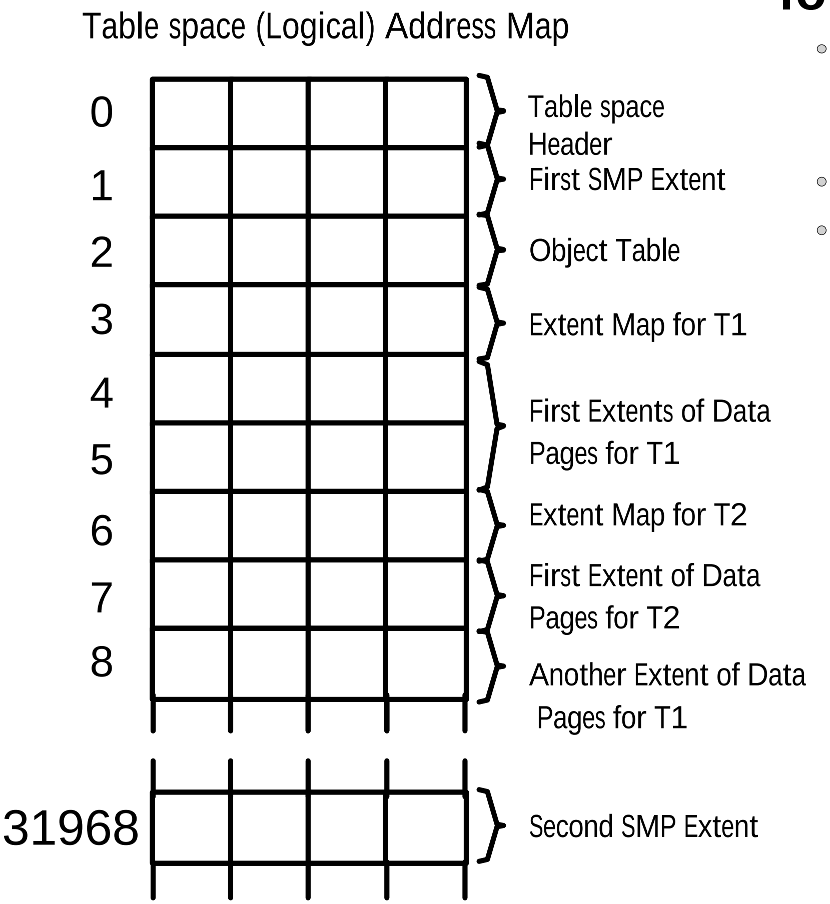
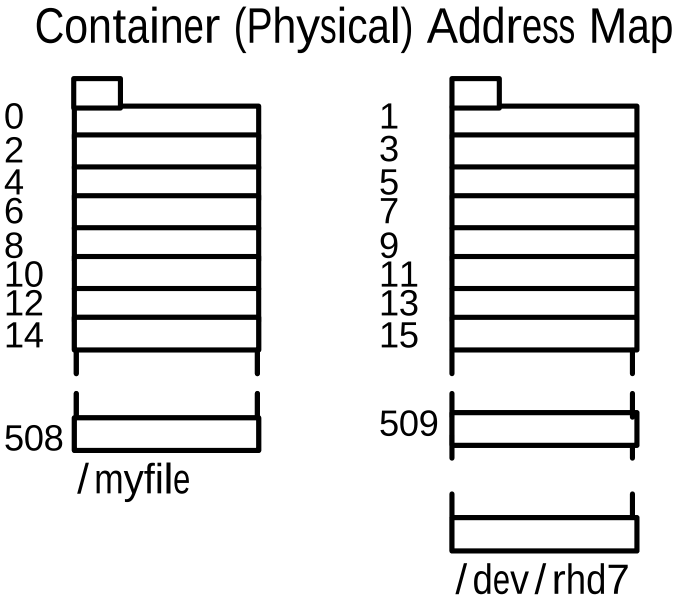


Table space (Logical) Address Map
0
1
2
3
4
5
6
7
8
31968
Table space Header
First SMP Extent
Object Table
Extent Map for T1
First Extents of Data Pages for T1
Extent Map for T2
First Extent of Data Pages for T2
Another Extent of Data Pages for T1
Second SMP Extent

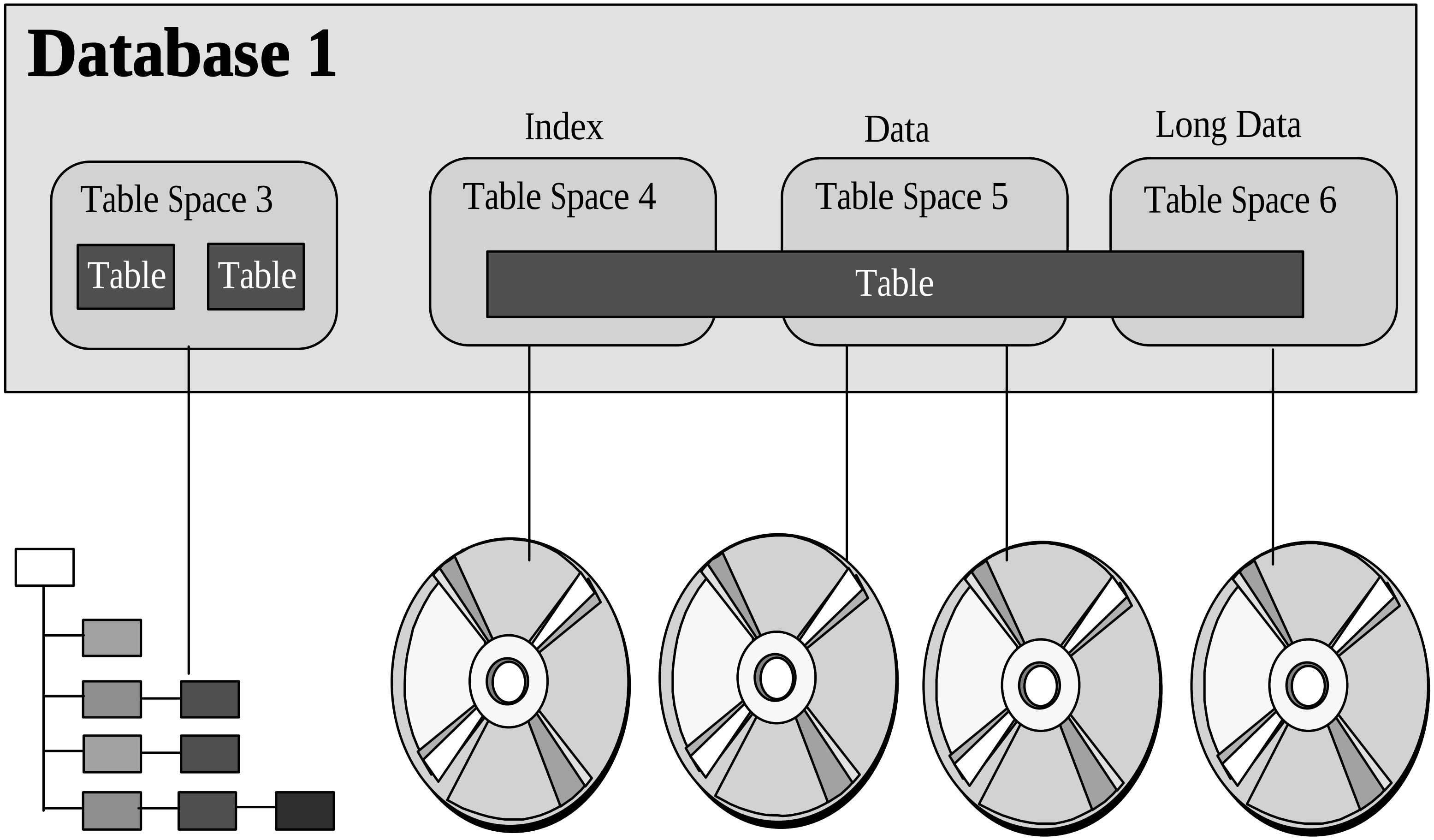


Data Placement
SMS versus DMS

	SMS	DMS
Striping	**Yes**	**Yes**
Object Management	**Operating system** (via unique file names)	**DB2** (Object table and EMP extents)
Space Allocation	**Grows/shrinks on demand**	**Preallocated**
Ease Of Administration	**Best** . Little/no tuning required (e.g.. OS prefetching often very good) . Enlarge file system(s) associated with containers	**Good** . Some tuning required (e.g.. EXTENTSIZE PREFETCHSIZE) . Can enlarge table space
Performance	**Very Good**	**Best** . Can achieve up to 5-10% advantage with raw containers. . Index, LOBs, Data for a single table can be spanned across table spaces.

- **Table spaces can now be renamed which gives ease of movement . Does not include SYSCATSPACE. NOTE: renaming does update the minimum recovery time.**

Data Placement
Table Spaces & Containers



Data Placement
Creating a database

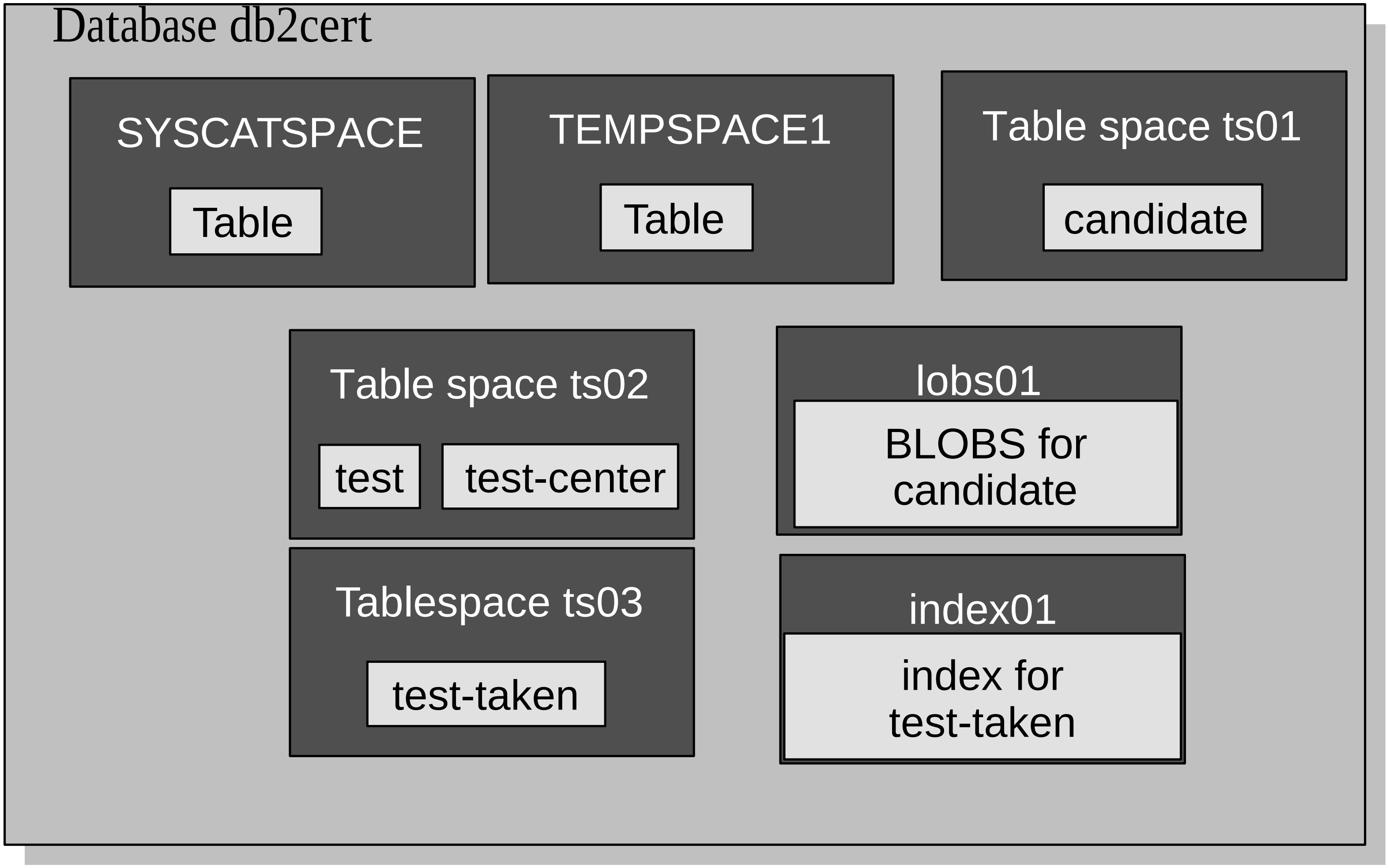
```
CREATE DATABASE DB2CERT
  DFT_EXTENT_SZ 4
CATALOG TABLESPACE
  MANAGED BY DATABASE USING
  (FILE’C:\CAT\CATALOG.DAT’ 2000   ,FILE’D:\CAT\CATALOG.DAT’ 2000)
  EXTENTSIZE 8
  PREFETCHSIZE 16
TEMPORARY TABLESPACE
  MANAGED BY SYSTEM USING
  (’C:\TEMPTS’   ,’D:\TEMPTS’)
USER TABLESPACE
  MANAGED BY DATABASE USING
  (FILE’C:\TS\USERTS.DAT’ 121)
EXTENTSIZE 24
PREFETCHSIZE 48
```

Data Placement
Planning the Environment

- Determine logical design
- Map design to Table spaces
- Create database
- Size the tables, indexes and table spaces
- Determine characteristics of containers
- Prepare physical environment
- Create table spaces
- Create tables and indexes

Database, tablespace, table, and index creation can be performed through DB2 UDB Command Line Processor script, DB2 Command Center, DB2 Control Center

Data Placement
Determine logical design



Database db2cert

SYSCATSPACE

Table

TEMPSPACE1

Table

Table space ts01

candidate

Table space ts02

test

test-center

lobs01

BLOBS for candidate

Tablespace ts03

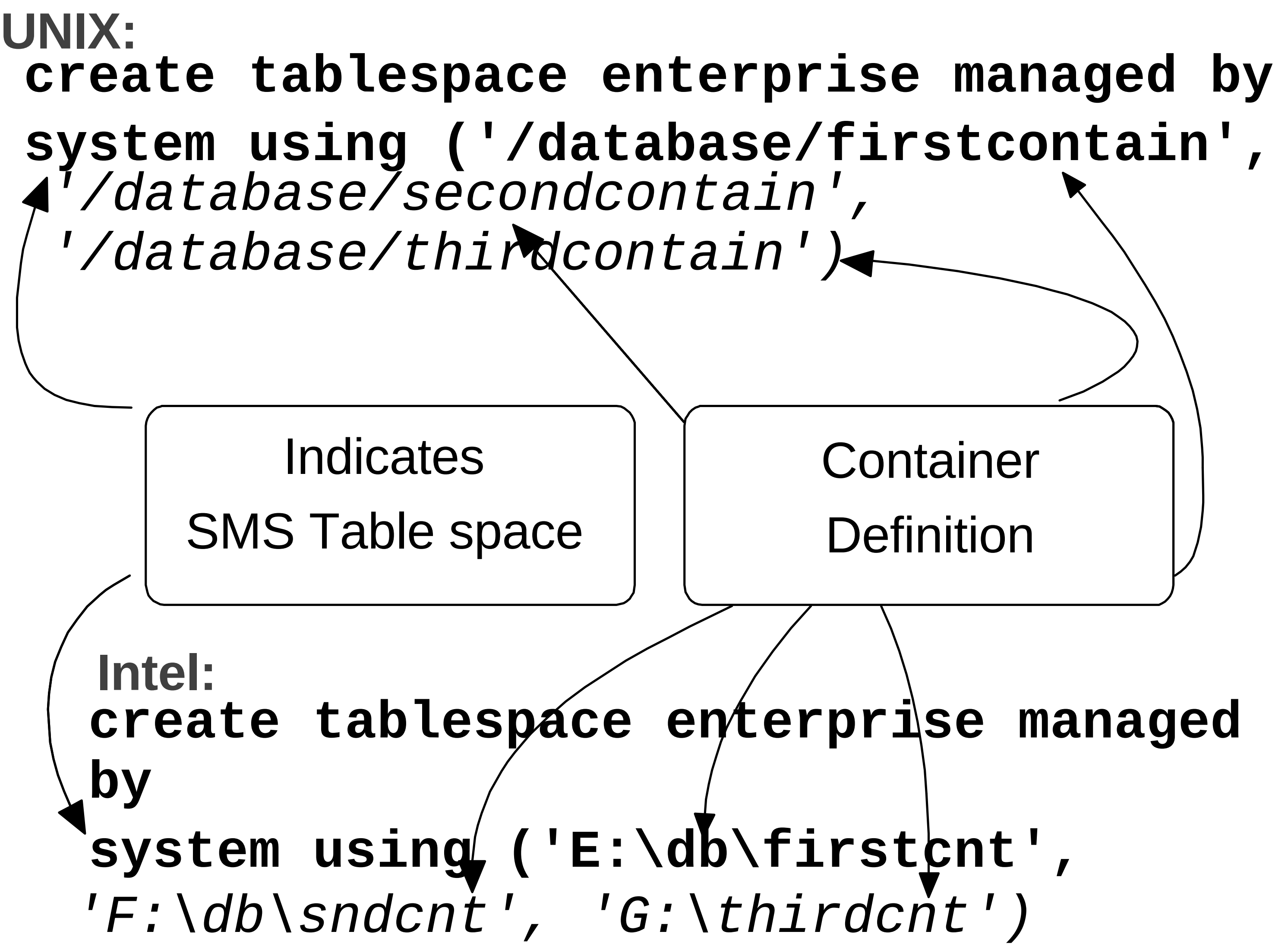
test-taken

index01

index for test-taken

Data Placement

Example: Creating SMS Table Spaces



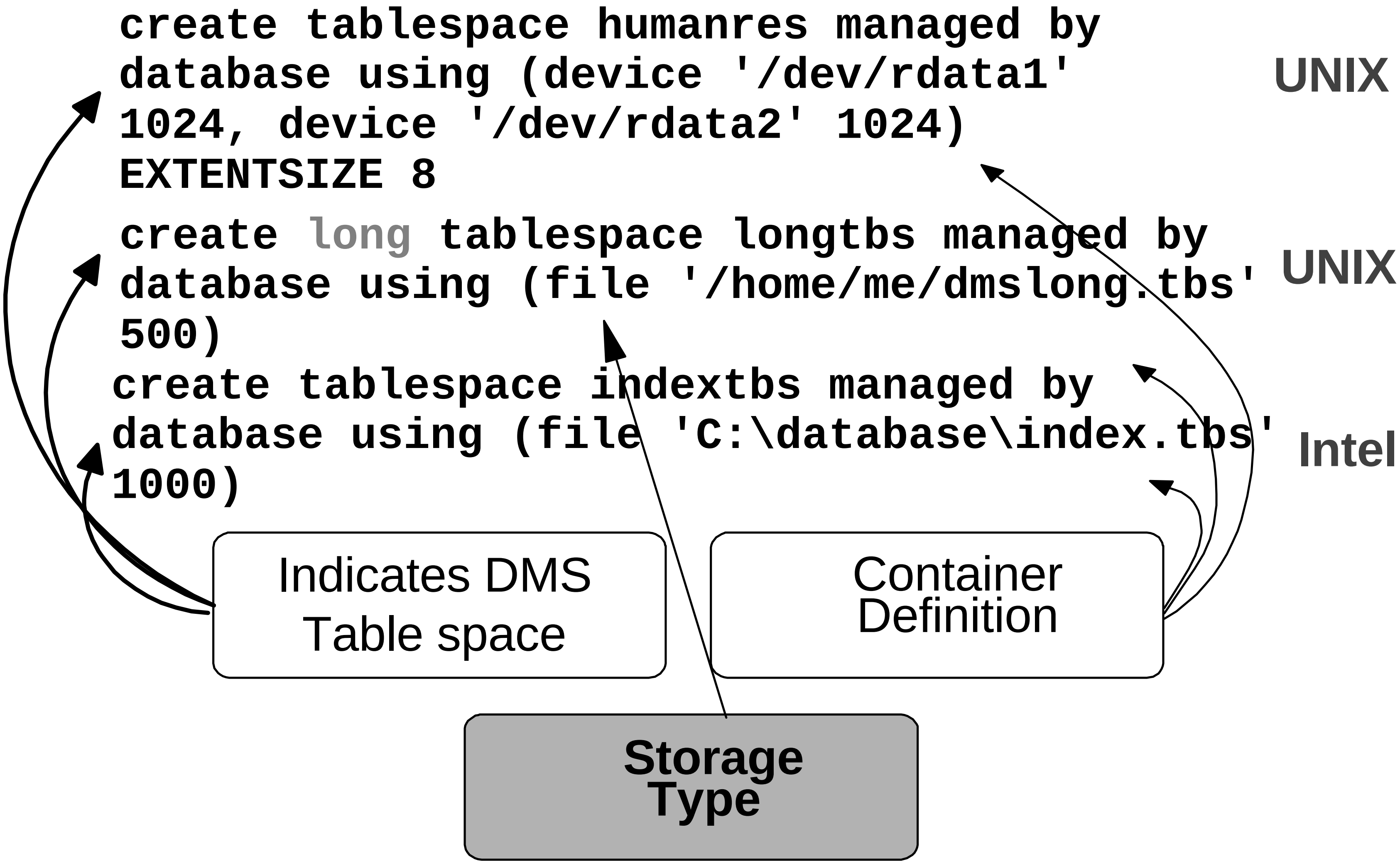
UNIX:

```
create tablespace enterprise managed by
system using ('/database/firstcontain',
'/database/secondcontain',
'/database/thirdcontain')
```

Intel:

```
create tablespace enterprise managed
by
system using ('E:\db\firstcnt',
'F:\db\sndcnt', 'G:\thirdcnt')
```

Data Placement
Example: Creating DMS Table Space



```
create tablespace humanres managed by
database using (device '/dev/rdata1'
1024, device '/dev/rdata2' 1024)
EXTENTSIZE 8
```

UNIX

```
create long tablespace longtbs managed by
database using (file '/home/me/dmslong.tbs'
500)
```

UNIX

```
create tablespace indextbs managed by
database using (file 'C:\database\index.tbs'
1000)
```

Intel

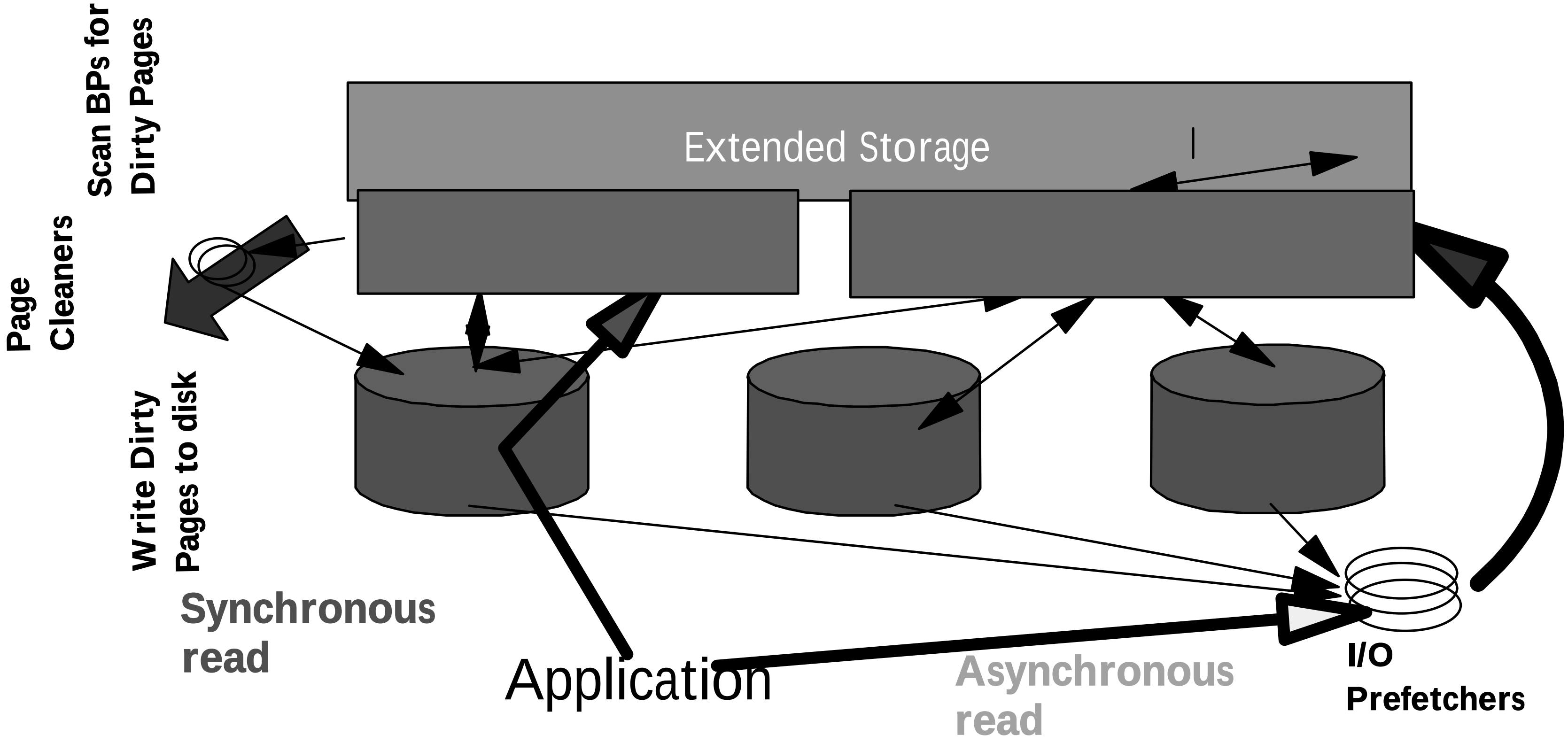
Indicates DMS Table space

Container Definition

Storage Type

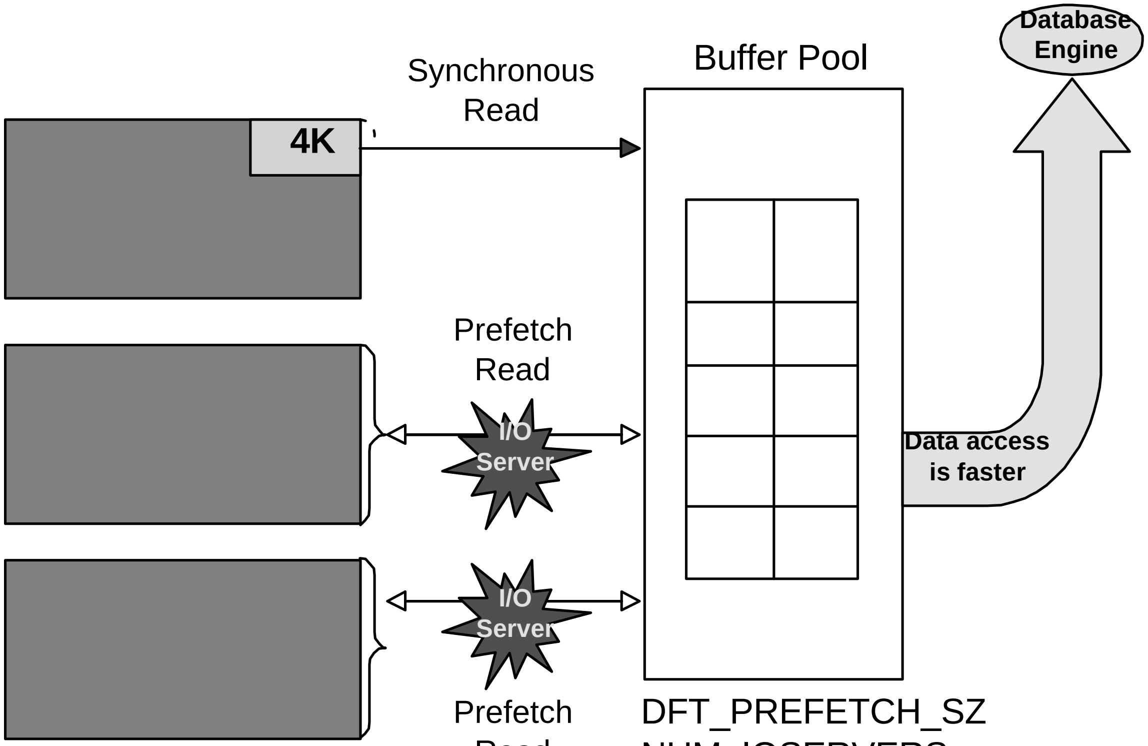
Data Placement
I/O Cleaners and Servers

- Separate threads of control for fetching and writing pages to/from hard drives.
- Can greatly enhance the performance of queries.
- Parameters are NUM_IOCLEANERS and NUM_IOSERVERS.
- Set num_ioservers to one or two more than the number of physical devices on which the database resides.
- Set num_iocleaners to be between one and the number of physical storage devices used for the database.



Data Placement
I/O Prefetch

- **Data is retrieved by I/O server tasks while previously retrieved data is processed by query tasks, thus reducing I/O bottleneck**
- **Prefetching, in order of best performance**
  - ➢Index prefetch occurs on index scans, RUNSTATs and REORG
  - ➢Sequential prefetch of data pages: occurs on table scans
  - ➢List prefetch of data pages: occurs on table scans of unclustered data



DFT_PREFETCH_SZ
NUM_IOSERVERS
SEQDETECT