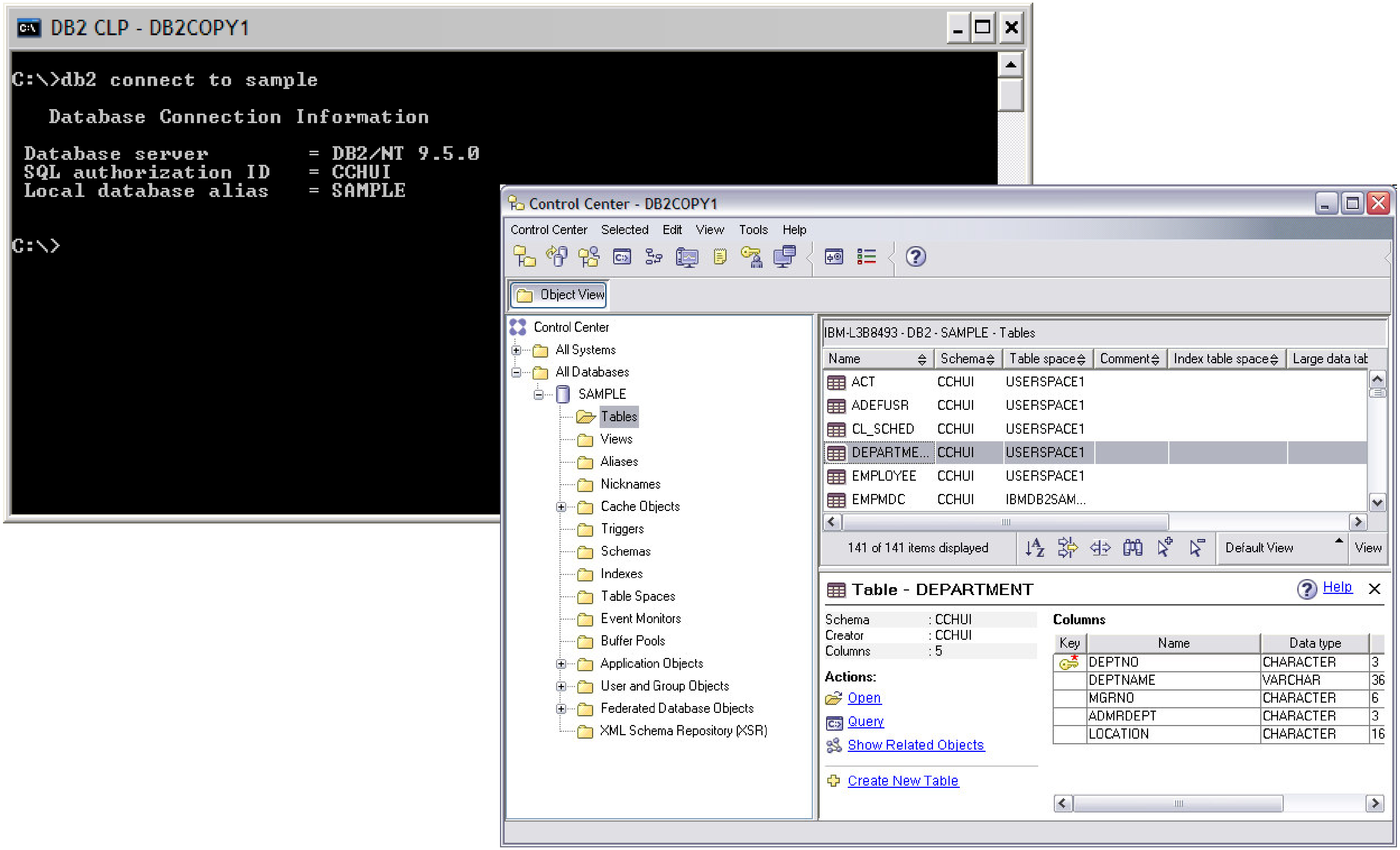
IBM Inter-University Programming Contest 2012 Training

Chapter 1: DB2

Chapter Outline

- Tools to Access DB2
- DB2 Instances and Databases
- Bufferpool
- Tablespace
- Schema
- Data Type
- Table
- Insert Statement
- Update Statement
- Delete Statement
- Select Statement
- Select Statement Options

Tools to Access DB2



DB2 CLP - DB2COPY1

```
C:\>db2 connect to sample

   Database Connection Information

 Database server        = DB2/NT 9.5.0
 SQL authorization ID   = CCHUI
 Local database alias   = SAMPLE


C:\>
```

Control Center - DB2COPY1

IBM-L3B8493 - DB2 - SAMPLE - Tables

Name	Schema	Table space	Comment	Index table space	Large data tat
ACT	CCHUI	USERSPACE1			
ADEFUSR	CCHUI	USERSPACE1			
CL_SCHED	CCHUI	USERSPACE1			
DEPARTME...	CCHUI	USERSPACE1			
EMPLOYEE	CCHUI	USERSPACE1			
EMPMDC	CCHUI	IBMDB2SAM...			

141 of 141 items displayed

Table - DEPARTMENT

Schema : CCHUI
Creator : CCHUI
Columns : 5

Actions:

Columns

Key	Name	Data type	
🔑	DEPTNO	CHARACTER	3
	DEPTNAME	VARCHAR	36
	MGRNO	CHARACTER	6
	ADMRDEPT	CHARACTER	3
	LOCATION	CHARACTER	16

Command Line Access

- DB2 Command Line Processor (CLP)
  - click on the DB2 Command Line Processor icon or enter db2 at a command prompt
  - an interactive input prompt appears:
    - db2 =>
  - no need to prefix commands with 'db2'
    - e.g. db2 => connect to sample
  - to end the interactive mode, enter quit
  - to disconnect from the database and terminate the DB2 backend process (db2bp), enter terminate
  - to execute OS commands, enter !<OS command>
- DB2 Command Line Window
  - click on the DB2 Command Window icon or from MS-DOS prompt, enter db2cmd
    - The command db2cmd is valid for Windows only
  - invoke the interpreter by prefacing commands and SQL with 'db2'
    - e.g. db2 connect to sample
    - e.g. db2 "select * from employee"
    - e.g. db2 -tvf createtab.db2
  - to end command line mode and terminate the DB2 backend process (db2bp), enter db2 terminate
  - all OS commands can be issued from the DB2 Command Window

Getting Help

- Obtain syntax and information for DB2 commands from the command line
  - db2 “?” list of all DB2 commands
  - db2 “? <db2-command>” get syntax help for a specific command
  - db2 “? sqlnnnn “ get message and brief description of a specific SQLCODE
  - db2 “? db2nnnn” get message and brief description of a DB2 error code
- For example:
  - db2 “? catalog tcpip”

```
CATALOG [ADMIN] TCPIP NODE node-name REMOTE hostname
    [SERVER service-name] [SECURITY {SOCKS}]
    [REMOTE_INSTANCE instance-name] [SYSTEM system-name]
    [OSTYPE os-type] [WITH "comment string"]
```

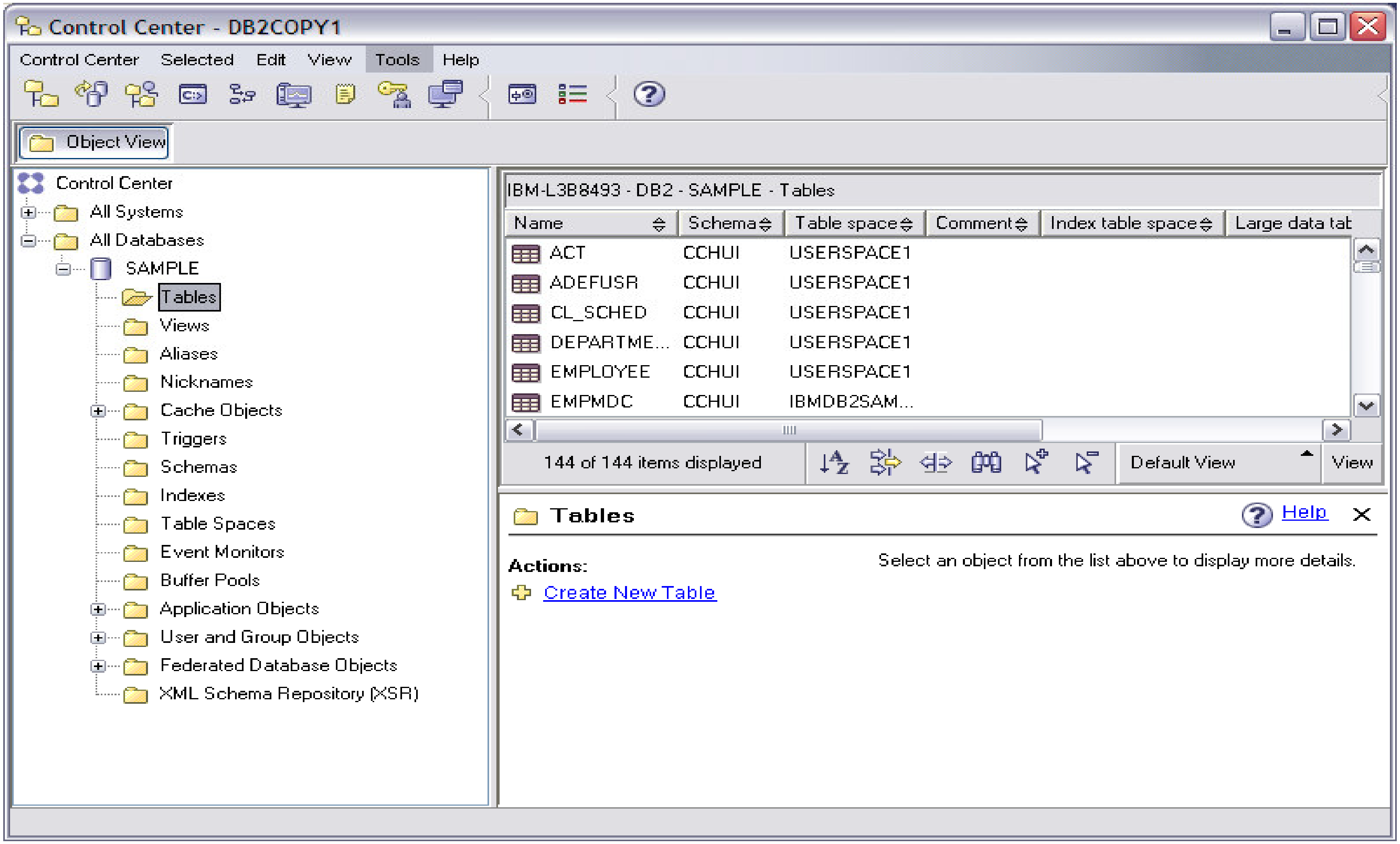
Command Line Options

- To list default command options
  - **LIST COMMAND OPTIONS**

```
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
   -e    Display SQLCODE/SQLSTATE                  OFF
   -f    Read from input file                      OFF
   -l    Log commands in history file              OFF
   -n    Remove new line character                 OFF
   -o    Display output                            ON
   -p    Display interactive input prompt          ON
   -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
```

- To change default command options.
  - For example, to turn auto-commit off: **set db2options=+c**
- To change command option for the current session
  - For example, to save output to file: **UPDATE COMMAND OPTIONS USING z ON output.txt**

Control Center



IBM-L3B8493 - DB2 - SAMPLE - Tables

Name	Schema	Table space	Comment	Index table space	Large data tat
ACT	CCHUI	USERSPACE1			
ADEFUSR	CCHUI	USERSPACE1			
CL_SCHED	CCHUI	USERSPACE1			
DEPARTME...	CCHUI	USERSPACE1			
EMPLOYEE	CCHUI	USERSPACE1			
EMPMDC	CCHUI	IBMDB2SAM...			

144 of 144 items displayed

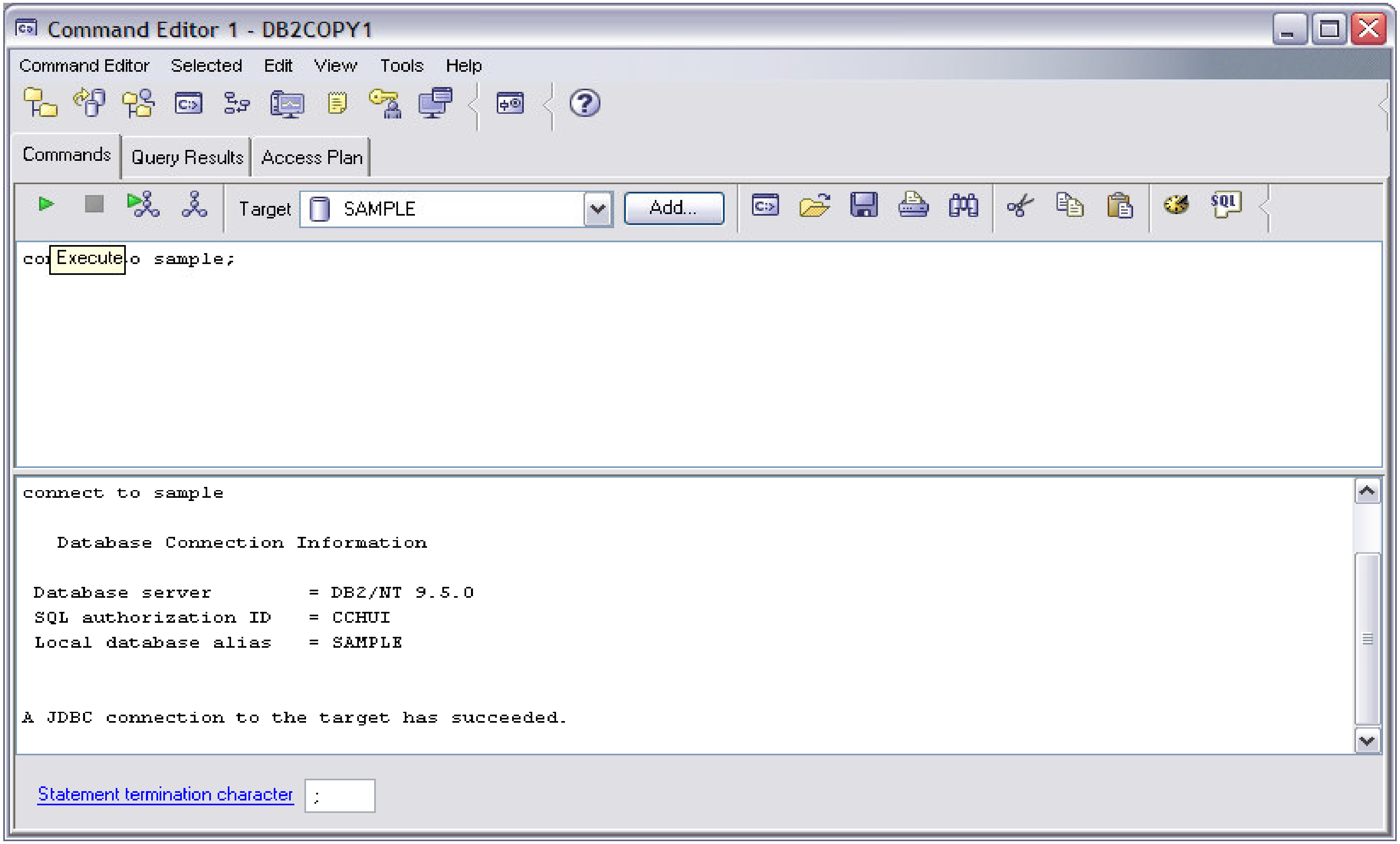
Tables

Actions:

Create New Table

Select an object from the list above to display more details.

Command Center

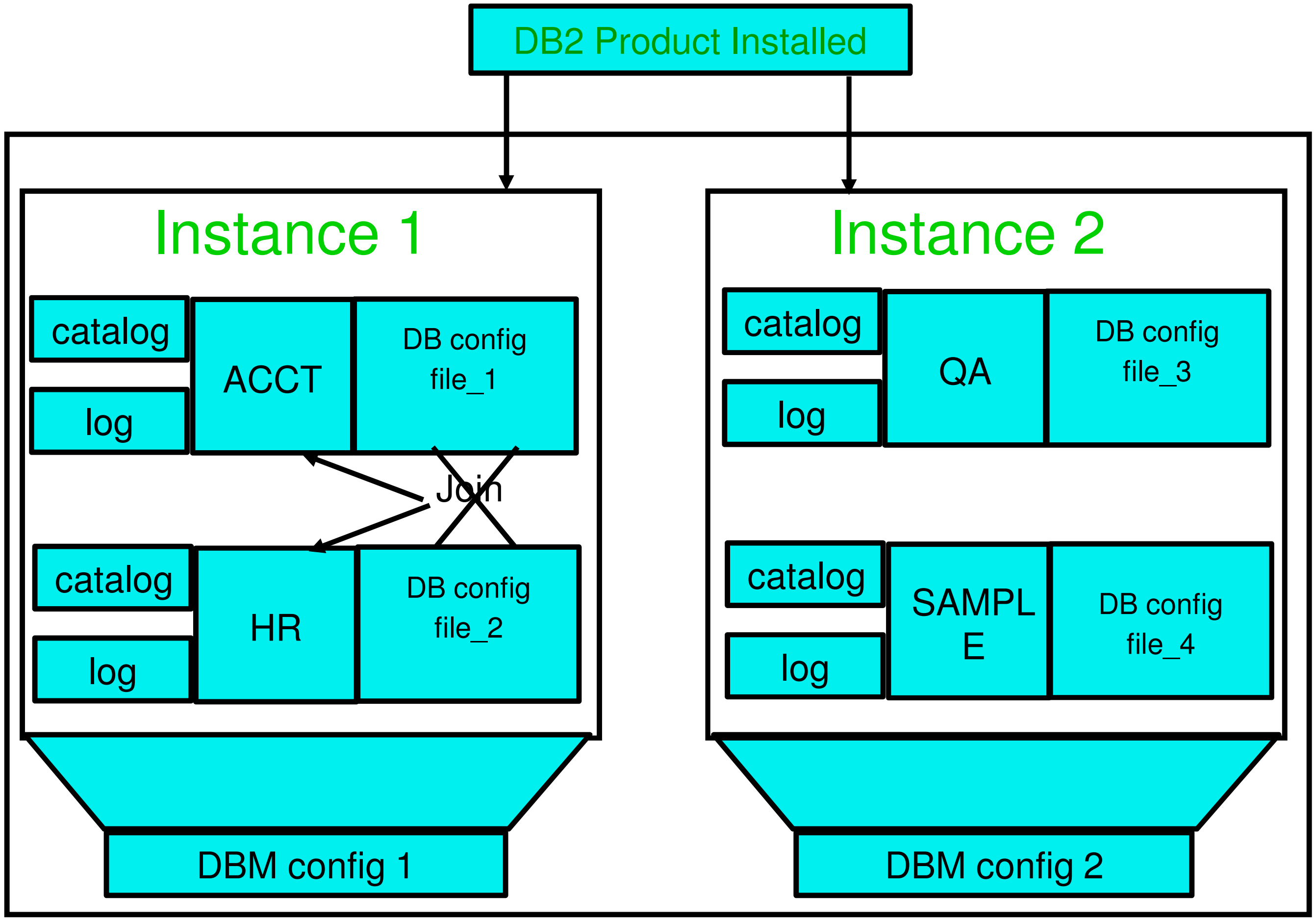


```
co[Execute]o sample;
```

```
connect to sample

   Database Connection Information

 Database server        = DB2/NT 9.5.0
 SQL authorization ID   = CCHUI
 Local database alias   = SAMPLE


A JDBC connection to the target has succeeded.
```

Statement termination character ;

DB2 Instances and Databases



DB2 Instances

- A DB2 instance is a logical context in which DB2 commands and functions are executed
- An instance manages access to database files
- More than one instance can be defined on a server machine
- Each instance is independent of the others
- An instance is created on install by default
  - db2inst1 (UNIX)
  - DB2 (Windows)
- To start an instance
  - db2start
- Starts a remote instance
  - db2start REMOTE <instance name>
- Start the instance in quiesced mode for administration purposes
  - db2start ADMIN MODE
- Terminate all database connections and stop an instance
  - db2stop force

Managing DB2 Instances

- To create additional instances
  - db2icrt -u <fenced user ID> <instance name>
    - must specify fenced user ID for UNIX platforms
  - db2icrt <instance name>
    - for Windows platforms
- Terminate all database connections and drop an instance
  - db2idrop -f <instance name>
- To list existing instances defined in a server
  - db2ilist
- To update a DB2 instance for access to functions associated with installation or removal of certain product options or fix patches
  - db2iupdt <instance name>
- To migrates an existing instance, for UNIX platforms only
  - db2imigr <instance name>
- To create, drop, update, or migrate an instance, root or administrative access is required

Create a Database

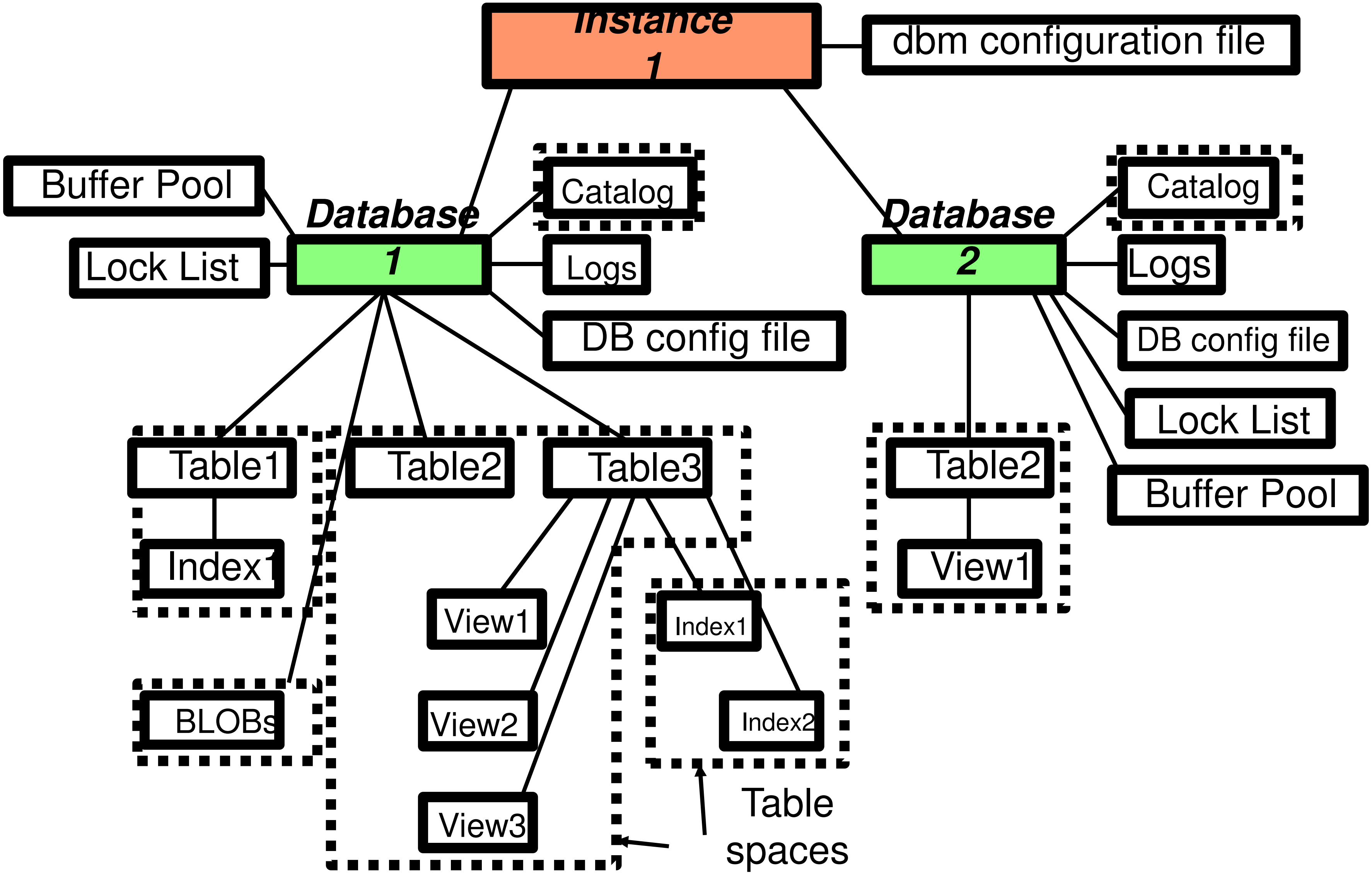
- Simplest format

```
CREATE DATABASE database-name
```

- More options:

```
CREATE DATABASE database-name
  [AUTOMATIC STORAGE {NO | YES}]
  [ON drive[{,drive}...][DBPATH ON drive]]
  [ALIAS database-alias]
  [USING CODESET codeset TERRITORY territory]
  [PAGESIZE integer [K]]
  [CATALOG TABLESPACE tblspace-defn]
  [USER TABLESPACE tblspace-defn]
  [TEMPORARY TABLESPACE tblspace-defn]
  [AUTOCONFIGURE [USING config-keyword value [{,config-keyword value}...]]
```

Database Object Hierarchy



Bufferpool

- Used to buffer data in memory to reduce the number of I/O operations to the physical database
- Keep often requested data/index pages in memory
- Keep infrequently accessed tables (e.g. random access into very large table) out of main memory
- Most data manipulation takes place in buffer pools, except for large objects and long field data
- IBMDEFAULTBP is the default bufferpool created with every database

Bufferpool Management

- Memory for the buffer pool is allocated when the database is activated or when the first application connects to the database
- Command to create buffer pool

```
CREATE BUFFERPOOL bpname IMMEDIATE SIZE sz PAGESIZE pgsz
CREATE BUFFERPOOL bpname DEFERRED SIZE sz PAGESIZE pgsz
```

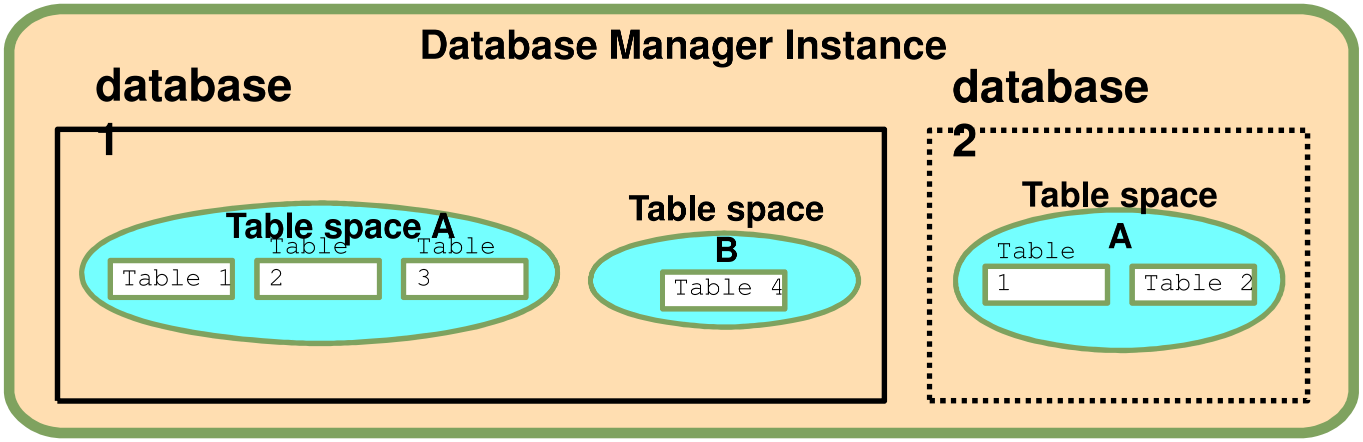
- IMMEDIATE
  - Bufferpool be created immediately
- DEFERRED
  - Bufferpool be created when the database is reactivated
- Information stored in SYSIBM.SYSBUFFERPOOLS
- Alter size of a bufferpool

```
ALTER BUFFERPOOL bpname IMMEDIATE SIZE sz
ALTER BUFFERPOOL bpname DEFERRED SIZE sz
```

- Memory is allocated as soon as if the memory is available.
- If the memory is not available, the changed occurs when the database is reactivated
- If the size of the bufferpool is decreased, memory is deallocated at commit time

Tablespaces

- Table space is a logical grouping of tables and other objects in a database
- All database objects are stored within table spaces
- Two types of storage:
  - System Managed Space (SMS)
  - Database Managed Space (DMS)
- A table space is composed of one or more containers
- Data allocated by extents within containers
- Table spaces are either 4K, 8K, 16K or 32K pages
  - 4K is default size
  - Must be associated with a buffer pool with same page size



Default Tablespaces in a Database

- With a simple CREATE DATABASE command:
  - CREATE DATABASE sample
- Three SMS table spaces are created automatically in default locations:
  - SYSCATSPACE - system catalog tables
  - USERSPACE1 - default user data
  - TEMPSPACE1 - temporary data
- Can change table space storage type and explicitly specify the locations of the containers:

```
CREATE DATABASE sample
   CATALOG TABLESPACE
      MANAGED BY SYSTEM
      USING ( 'c:\catdir1' );
   USER TABLESPACE
      MANAGED BY DATABASE
      USING ( FILE 'c:\db2files\usertbsp1 100, FILE 'c:\db2files\usertbsp2 100 )
   TEMP TABLESPACE
      MANAGED BY SYSTEM
      USING ( 'c:\tempspace' ) ;
```

Schema

- A schema is a collection of database objects such as tables, views, indexes, or triggers. It provides a logical classification of database objects
- How is a Schema name used?
  - Fully-qualified object name: "schemaname.tablename"
  - Can have multiple tables with the same name, but different schema names
    - eyerman.staff != jones.staff
- Reserved schema names
  - SYSCAT, SYSIBM, SYSSTAT, SYSFUN
  - Avoid schema names beginning with SYS
- If database object does not specify a schema name, table qualified with current authorization ID
- Alternate schema names can be specified using
  - SET CURRENT SCHEMA or SET CURRENT SQLID command
  - CREATE ALIAS <aliasname> FOR <tab/view name>
  - CREATE VIEW

System Schemas

- Created with every database and placed into the SYSCATSPACE table space
- SYSIBM
  - Base catalogs
  - Access not recommended
- SYSCAT
  - SELECT authority GRANTed to PUBLIC
  - Catalog Read-only Views
  - Recommended way to obtain catalog information
- SYSSTAT
  - Updateable Catalog Views - Influence the Optimizer
- SYSFUN
  - User-Defined Functions

System Catalog Tables and Views

Schema: Table = SYSIBM View = SYSCAT

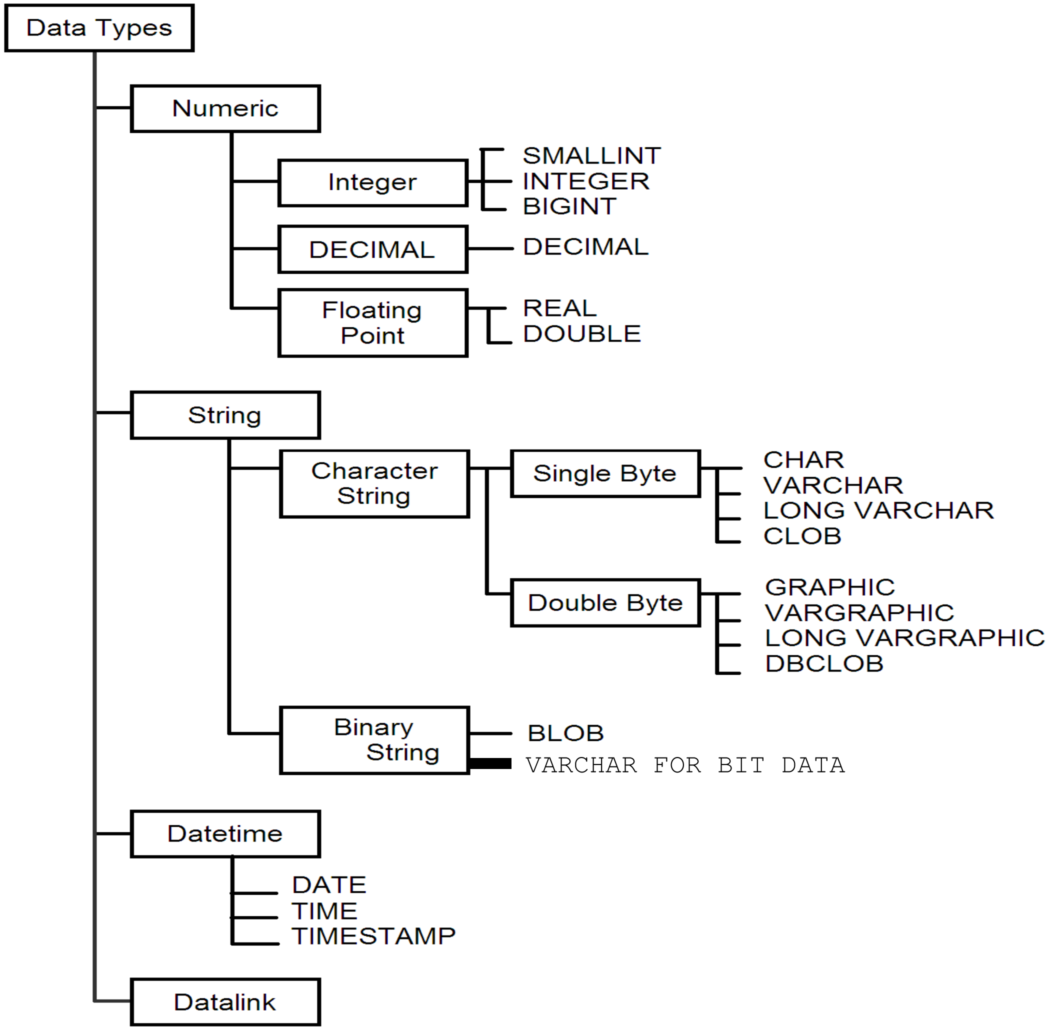
TABLE	VIEW	DESCRIPTION
SYSDBAUTH	DBAUTH	Authorities on database
SYSCHECKS	CHECKS	Check constraints
SYSCOLUMNS	COLUMNS	Column definitions
SYSCOLCHECKS	COLCHECKS	Columns referenced by check constraints
SYSCOLDIST	COLDIST	Detailed columns statistics
SYSKEYCOLUSE	KEYCOLUSE	Columns used in keys
SYSCONSTDEP	CONSTDEP	Constraint dependencies
SYSDATATYPES	DATATYPES	Datatype definitions (built-in & UDT)
SYSEVENTMONITORS	EVENTMONITORS	Event Monitor Definitions
SYSEVENTS	EVENTS	Events currently monitored
SYSFUNCPARMS	FUNCPARMS	Definitions of Parameters/Results of UDFs
SYSFUNCTIONS	FUNCTIONS	UDF definitions
SYSINDEXAUTH	INDEXAUTH	Index privileges
SYSINDEXES	INDEXES	Index definitions

System Catalog Tables and Views

Schema: Table = SYSIBM View = SYSCAT

TABLE	VIEW	DESCRIPTION
SYSPACKAGEAUTH	PACKAGEAUTH	Authorities on packages
SYSPACKAGEDEP	PACKAGEDEP	Package dependencies
SYSPACKAGES	PACKAGES	Package definitions
SYSREFERENCES	REFERENCES	Referential constraints definitions
SYSSTATEMENTS	STATEMENTS	Details of package SQL Statements
SYSTABAUTH	TABAUTH	Table Authorities
SYSTABCONST	TABCONST	Table constraint definitions
SYSTABLES	TABLES	Table definitions
SYSTABLESPACES	TABLESPACES	Table Space Definitions
SYSTRIGDEP	TRIGDEPEVENTS	Trigger dependencies
SYSTRIGGERS	TRIGGERS	Definitions of triggers
SYSVIEWDEP	VIEWDEP	View dependencies
SYSVIEWS	VIEWS	View definitions

Data Types


Data Types
Numeric
Integer
SMALLINT
INTEGER
BIGINT
DECIMAL
DECIMAL
Floating Point
REAL
DOUBLE
String
Character String
Single Byte
CHAR
VARCHAR
LONG VARCHAR
CLOB
Double Byte
GRAPHIC
VARGRAPHIC
LONG VARGRAPHIC
DBCLOB
Binary String
BLOB
VARCHAR FOR BIT DATA
Datetime
DATE
TIME
TIMESTAMP
Datalink

CREATE TABLE Command

- You must have SYSADM or DBADM authority or CREATETAB privilege on the database
- Example:

```
connect to eddb;
create table artists
 ( artno         SMALLINT NOT NULL
 , name          VARCHAR(50) WITH DEFAULT 'abc'
 , classification        CHAR(1) NOT NULL
 , bio           CLOB(100K) LOGGED
 , article               DATALINK LINKTYPE URL FILE
                 LINK CONTROL MODE DB2OPTIONS,
 , picture               BLOB(2M) NOT LOGGED COMPAT )
 INDEX IN indtbsp
 LONG IN longtbsp
 IN datatbsp;
```

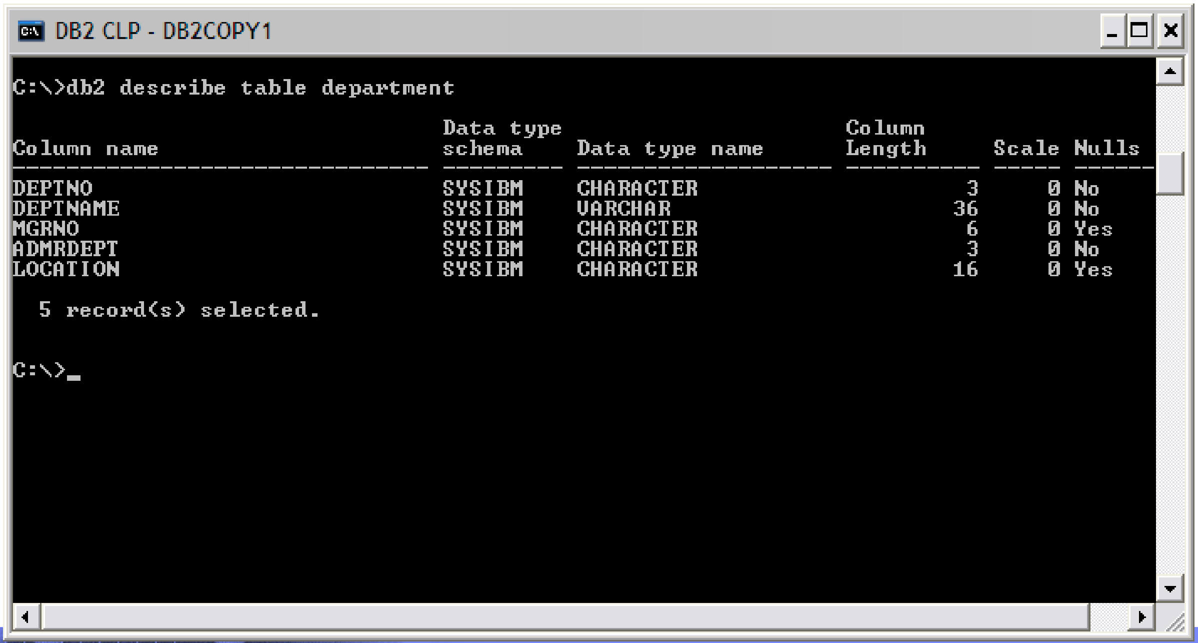
Table

- If a table is created without the IN clause, it will be placed:
  - In the IBMDEFAULTGROUP table space (if it exists and if the page size is sufficient)
  - In a user created table space which is of the smallest pagesize that is sufficient for the table
  - Then it will go in USERSPACE1 (if it exists and has a sufficient page size)
- Use IN, INDEX IN, and LONG IN clauses to specify the tablespaces to stored related information
- To duplicate a table:
  - CREATE TABLE tab1new LIKE tab1;
- No constraints, triggers, or indexes copied, and data not copied
- May specify table or view
- To define a table using SQL statement

  -
    ```
    CREATE TABLE t1new
            AS (SELECT c1, c8, c10 FROM t1)
            DEFINITION ONLY;
    ```
- Column attributes of defined table based upon referenced table
- Data not populated

Other Useful Table Commands

- LIST TABLES
  - List tables for the current user
- LIST TABLES FOR ALL
  - List all tables defined in the database
- LIST TABLES FOR SCHEMA <schema>
  - List tables for the specified schema
- DESCRIBE TABLE <tablename>
  - Show the structure of the specified table



```
C:\>db2 describe table department

                                Data type                     Column
Column name                     schema    Data type name      Length     Scale Nulls
------------------------------- --------- ------------------- ---------- ----- ------
DEPTNO                          SYSIBM    CHARACTER                    3     0 No
DEPTNAME                        SYSIBM    VARCHAR                     36     0 No
MGRNO                           SYSIBM    CHARACTER                    6     0 Yes
ADMRDEPT                        SYSIBM    CHARACTER                    3     0 No
LOCATION                        SYSIBM    CHARACTER                   16     0 Yes

  5 record(s) selected.

C:\>_
```

Insert Statement

- You need to have appropriate table or view privilege
- Can insert one or more rows at a time

```
INSERT INTO department (deptno, deptname, admrdept)
   VALUES ( 'E31', 'ARCHITECTURE', 'E01' ),
              ( ‘E32’, ‘INFRASTRUCTURE’, ‘E02)
```

- You can use subselect to determine values

```
INSERT INTO department (deptno, deptname)
   SELECT deptno, deptname FROM sales_depts
```

- May also be done to a subset of columns provided:
  - A column not specified accepts NULLs or
  - Columns defined WITH DEFAULT
- Large amounts of data? Look at using LOAD instead

Update Statement

- UPDATES come in 3 main varieties :
  - Full table

```
UPDATE employee SET salary = NULL
UPDATE employee SET (dept) = (
        SELECT deptname FROM department WHERE deptno = 1)
```

  - Searched with a **WHERE** clause

```
UPDATE employee SET salary = salary * 1.10 WHERE title = ‘DBA’
```

  - Positioned using a **CURSOR** in a program

```
EXEC SQL DECLARE C1 CURSOR FOR
      SELECT * FROM EMPLOYEE FOR UPDATE OF JOB;
EXEC SQL  OPEN C1;
EXEC SQL  FETCH C1 INTO ...     ;
if ( strcmp (change, "YES") == 0 )
  EXEC SQL  UPDATE EMPLOYEE
        SET JOB = :newjob
        WHERE CURRENT OF C1;
EXEC SQL  CLOSE C1;
end if
```

Delete Statement

- DELETES can apply to single or multiple rows
  - You can also use subselect to determine values
- Two forms of statement :
  - Searched DELETE form is used to delete one or more rows
    - Optionally determined by a search condition
  - Positioned DELETE form is used to delete exactly one row
    - Determined by the current position of a cursor
- Deleted rows not removed from table
  - Space is marked as unused
  - Reclaim space using the REORG command
- Example:

```
DELETE FROM DEPARTMENT WHERE DEPTNO = 'D11'
```

Select Statement

- SELECT is the QUERY element of the language
- Queries come in many styles:
  - Retrieving all table data
  - Limiting Columns or Rows retrieved
  - Cartesian Product
  - Inner Join
  - Set operators
  - Common table expressions
  - Other options:
    - derived columns
    - sub-query etc.
    - sorting
    - using functions
    - grouping values

ORDER BY Clauses

- ORDER BY clause
  - The ORDER BY clause specifies an ordering of the rows of the result table
  - If a single sort specification (one sort-key with associated direction) is identified, the rows are ordered by the values of that sort specification
  - If more than one sort specification is identified, the rows are ordered by the values of the first identified sort specification, then by the values of the second identified sort specification, and so on
  - Example:

```
SELECT c1, c2 FROM t1 ORDER BY c1;
```

GROUP BY, HAVING Clauses

- GROUP BY ... <grouping expression>
  - A GROUP BY clause contains a grouping expression, it specifies an intermediate result table that consists of a grouping of the rows of R. R is the result of the previous clause of the subselect.

```
SELECT workdept, MAX ( salary ) FROM employee GROUP BY workdept ;
```

- HAVING clause
  - The HAVING clause specifies an intermediate result table that consists of the groups of R for which the search-condition is true. R is the result of the previous clause of the subselect.

```
SELECT workdept, MAX ( salary ) FROM employee GROUP BY workdept
          HAVING MAX ( salary ) < ( SELECT avg ( salary FROM employee ) ;
```

- Obtain distinct value from a table

```
SELECT DISTINCT ( position ) FROM employee ;
```

- Use of LIKE

```
SELECT DISTINCT WORKDEPT FROM employee WHERE WORKDEPT LIKE ‘A%’;
```

- Use of EXISTS

```
SELECT empno, salary FROM employee
  WHERE NOT EXISTS ( SELECT empno FROM manager ) ;
```

- Use of IN, NOT IN clauses

```
SELECT empno, salary FROM employee
  WHERE department IN ( 'A00', 'B00', 'C00' ) ;
```

Questions
&
Answers