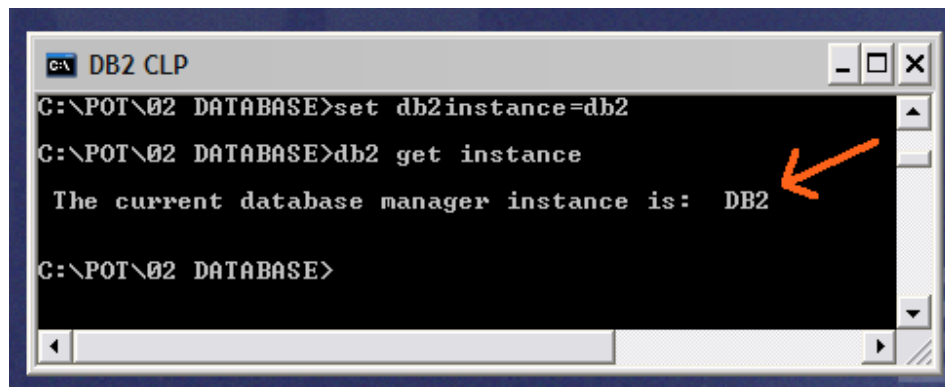# LAB 02 - DATABASE EXPLORATION

A. GENERAL DB2 DATABASE COMMANDS

Let's explore the database called sample in instance called DB2. First let's make sure we are pointing to the appropriate instance.

1. Make sure you position yourself in the **C:/POT/02 DATABASE/** directory before continuing

2. From a DB2 Command Window type:

   - **set db2instance=db2**

   - **db2 get instance**

3. Now let's get into the DB2 interactive mode, from a DB2 Command Window type: **db2**

4. To see the database configuration parameters, type: **get db cfg**

5. Notice you cannot get the db cfg without connecting to the database, to do so type: **connect to sample**

   - Full syntax: CONNECT TO SAMPLE USER [userid] USING [password]

   - Why can you do a "get dbm cfg", but not a "get db cfg" without a connect?

6. **list tables for schema administrator**

   - What is the difference between a user and a schema in DB2?

7. **describe table sysibm.systables**

8. **describe indexes for table sysibm.systables**

9. **select * from administrator.employee**

10. **describe output select * from administrator.employee**

   • Notice here DB2 describes the select itself

11. **list tablespaces show detail**

   • Pick a table space and note it's id

12. **list tablespace containers for [table space id] show detail**

   • Don't panic. This is not the only way to examine table spaces and containers!

13. **values user**

   • returns your userid

14. **values current date**

   • returns today's date

15. **runstats on table administrator.employee**

   • What is this like in Oracle?

16. **reorgchk update statistics on schema administrator**

   • This does a check for the need to REORG all tables in the schema

   • We could have used current statistics instead of update statistics

   • We can do every table in the database too with "on table all" instead of "on schema…"

17. **Quit**

Section Answers:

A5.   You can get DBM cfg because you are *attached* to the instance. You cannot get DB cfg because you were not *connected* to the database.

A6.   A user's default schema is always the username, however you don't have to have a user to exist in order to create a schema owned by that user. We'll explore that more later on.

B. THE DB2 SYSTEM DATABASE DIRECTORY

Try these commands from the DB2 Command Window:

1. **db2 list database directory > db_out1.txt**

   - Review file: **db_out1.txt**

   - This is a *system* database directory by the way, which means a directory of all databases cataloged for each instance system wide

   - The command: **db2 list database directory on c:** would give you a *local* database directory, which means all databases cataloged on the drive path.

2. **db2 connect to toolsdb**

   - Connect to another database

3. **db2 connect to sample**

4. **db2 uncatalog database toolsdb**

   - This uncatalogs the database

5. **db2 terminate**

   - This refreshes the directory cache

6. **db2 list database directory > db_out2.txt**

   - How has the directory listing changed?

7. **db2 connect to toolsdb**

   - What happens now? Can you connect to it?

Section Answers:

B6.   Uncatalogged database toolsdb no longer shows in the directory

B7.   No, you cannot connect to it because DB2 does not know it's there

---

C. CREATE A DATABASE

Try these commands from the DB2 Command Window:

1. **set db2instance=db2**

2. **db2 get instance**

3. **db2 create database sample2**

   - This will take approximately 90 seconds…while this is running, see explanation in box below for what this command is doing

> **Note: In DB2, the create database command does the following:**
>
> 1. Creates a subdirectory to hold the database information
> 2. Creates a database configuration file with default settings
> 3. Creates SYCATSPACE, TEMPSPACE1 and USERSPACE1 table spaces
> 4. Creates system catalog tables in the SYSCATSPACE table space
> 5. Creates SYSCAT, SYSFUN, and SYSSTAT schemas
> 6. Grants DBADM authority to the database creator
> 7. Grants selected database privileges to PUBLIC
> 8. Designed to default everything for you, but you can control the:
>   - Partition number
>   - Install path
>   - Database alias name
>   - Collating characteristics
>   - Codeset and territory
>   - Table space characteristics (types, sizes, containers
>   - Database configuration parameters (using feature called "autoconfigure" which we'll go through later)
>   - Many other things (see: Database CREATE_DATABASE_Command.txt)

4. Find the new database in the Control Center

Now let's create a database in a different instance: **DB2BKP**

5. Review and run these scripts:

> **Database02001.CMD** which executes ⇨ **Database02002.DDL**

- Remember, it takes about 90 seconds to finish... while this is running, see box below for differences between Oracle and DB2 create database issues

---

Note: In DB2, the following Oracle Concepts are handled differently:

1. PFILE

    - DB config parameters are in file SQLDBCON, but this is binary not editable directly

        ▪ Change with: update db cfg
        ▪ Or Control Center

    - DB2 uses a similar method to the new SPFILE concept introduced in Oracle 9i

        ▪ Oracle requires a PFILE to create a database though

    - Autoconfigure feature calculates and displays initial values for the buffer pool size, database configuration and database manager configuration parameters, with the option of applying these recommended values

    - If you create a database from a backup, you can get the exact db configuration parameter settings from the backup since backups keep db config information

2. Control File

    - File DB2RHIST.ASC history file contains history information about backups, restores, loading of tables, reorganization of tables, altering of a table space, and other changes to a database (has a backup copy)

    - File SQLSPCS.1 provides table space information (has a backup copy)

    - File SQLBP.1 provide buffer pool information (has a backup copy)

    - DB2TSCHNG.HIS file contains a history of table space changes at a log-file level. For each log file, DB2TSCHG.HIS contains information that helps to identify which table spaces are affected by the log file. Table space recovery uses information from this file to determine which log files to process during table space recovery.

    - The above are all automatically created and backed up for you during create database

---

6. Go to Control Center

7. Find instance db2bkp ⇨ databases ⇨ right click ⇨ Add ⇨ [Discover]
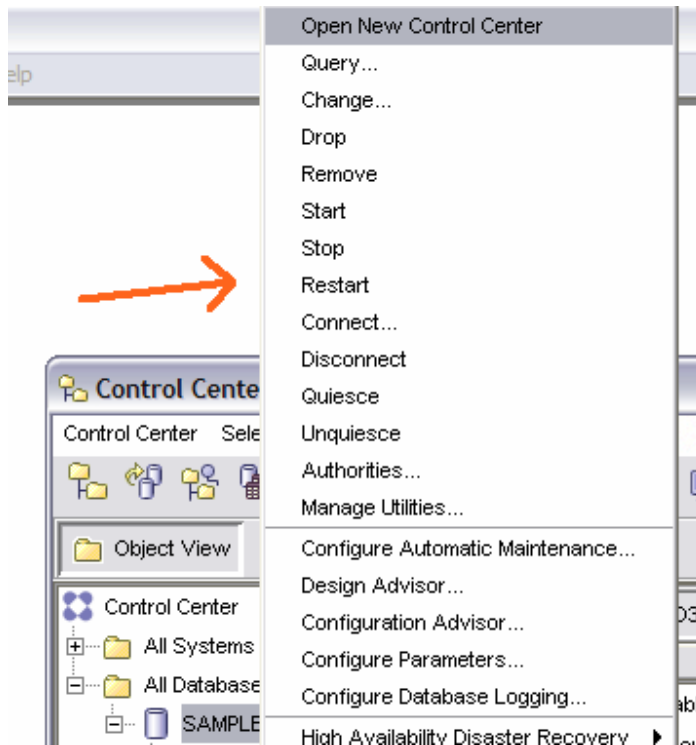
8. Find and select BACKDB ⇨ [OK] ⇨ [Apply]

D. EXPLORE THE CONTROL CENTER

Explore database level options in CC

⚠️ *Please make sure you do not execute any of these!*

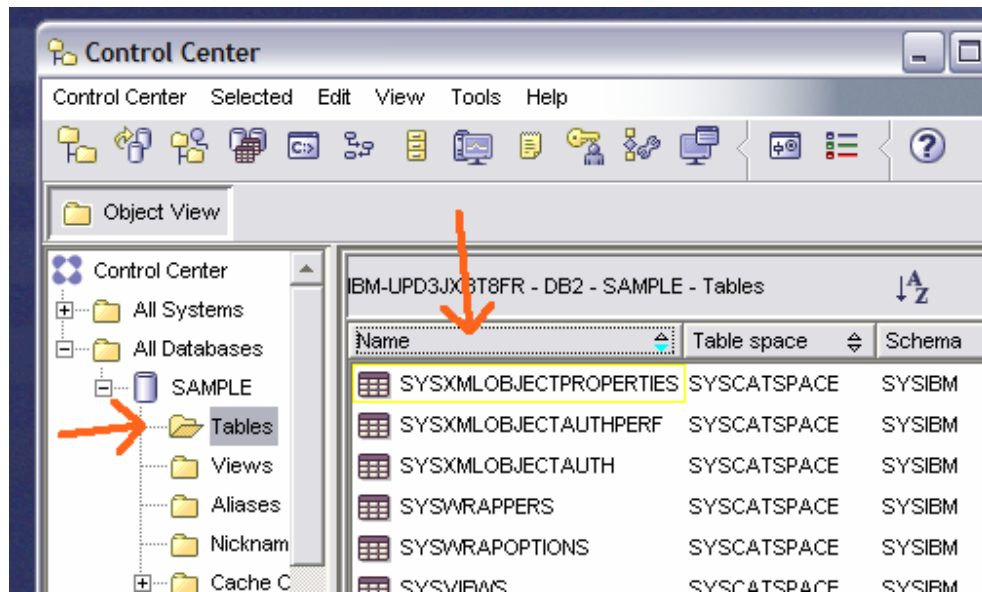1. Right click on database "**sample**" and review these:

Change	Changes database alias and comments
Drop	Drops (deletes) database: **CAUTION: IF YOU CHOOSE IT, DON'T EXECUTE!**
Remove	Uncatalogs database
Start/Stop	Starts and stops the *instance* (not the database)
Restart	Starts a database and ignores problem table spaces (special case usage)
Connect	Connect with another user id
Disconnect	Destroys connection
Quiesce	Force users off and puts database in restricted mode, for maintenance reasons for example. (The CLP command 'quiesce' actually has more nuances than this GUI option which simply forces all applications off.)
Unquiesce	Takes database out of restricted mode.

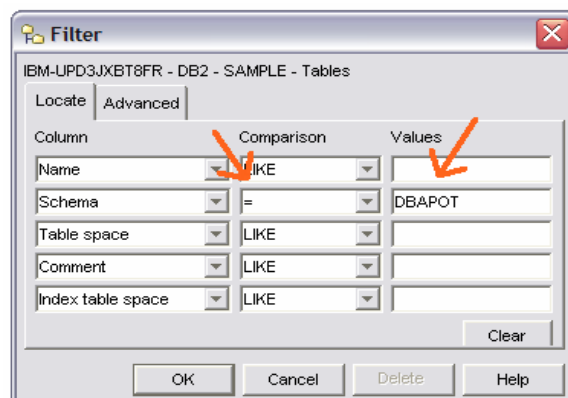2. We'll explore other database options later

Explore objects in CC

3. Object View Manipulation

- Expand database "sample" with [+]

- Select "Tables" ⇨ Click and hold any column heading and drag it where you want in the table view

- Single click any column heading to sort the view on it

- Make column arrangement permanent with "View" ⇨ "customize columns"

- Select "Tables" ⇨ Right click "filter" ⇨ "create"

  - Next to column "Schema" change column "Comparison" to "=" and column "Values" to your userid

  - Click [OK] to create this filter; notice how the view is restricted according to the filter

  - The Control Center uses this kind of filtering in lots of places as you will see later

4. Table Actions

- **REVIEW:**   Find table EMPLOYEE ⇨ click on it once and review bottom of screen to get a review of the that table's description

- **ALTER:**   Click on EMPLOYEE ⇨ right click ⇨ Alter (explore this option but don't change table permanently). Click [Cancel] to get out of this

- **DATA:**   Double click on ORG ⇨ [add a row] to ORG ⇨ [COMMIT] ⇨ [Close] ⇨ Double click again to see if data add was permanent.

- **RUNSTAT:**   Click and hold Shift to highlight DEPARTMENT and EMP_ACT tables ⇨ right click ⇨ runstats ⇨ leave everything as is and get to "Schedule" tab ⇨ [show command] ⇨ [OK]

- **ESTIMATE SIZE:**   Click on EMPLOYEE ⇨ right click ⇨ estimate size ⇨ Under "Rows" enter new figure like 10,000 and then [refresh]

- **COPY:**   Click on EMPLOYEE ⇨ right click ⇨ copy ⇨ pick instance **DB2BKP**, database **BACKDB** as a target. Use 'XXXXXX' as the target table name ⇨ [OK] and check results in the other instance.

    Click on table ⇨ Refresh (to reset view from this copy)

- **GENERATE DDL:**   Click on a table at the top of the view and hold [Shift] and click on a table at the bottom of the view to select many tables ⇨ Right click ⇨ generate DDL ⇨ [show command] ⇨ [generate]

5. Explore other objects

- Views
- Triggers
- Schemas
- Indexes
- Table spaces
- Buffer pools

E. EXTRA EXERCISES

Creating Objects in CC

1. Create a new buffer pool script by right clicking on "Buffer Pools" in the CC view ⇨ Create. Use the wizard to fill out the following:

   • Buffer Pool Name: BP32

   • Page size: 32K

   • Size in pages: 200

   • Non blocked (won't require sequential prefetching, leave them all non blocked)

   • Use extended storage (check this)

      ▪ If extended storage is enabled, pages that are being evicted from this buffer pool will be cached in extended storage which is enabled by setting the database configuration parameters NUM_ESTORE_SEGS and ESTORE_SEG_SIZE to non-zero values.

   • Create buffer pool immediately: select this

   • [Show SQL] and [Save] this script before you click [OK]

   • Your saved script should look like: **Database02004.DDL**

2. Create a new table space. Right click on Table Spaces and follow the wizard:

   • Name it TS32, comment: "32K table space" [Next]

   • Regular table space type   [Next]

   • Use buffer pool BP32 (you just created it!)   [Next]

   • Use database managed (DMS)   [Next]

   • [Add] one container for it which is:

      ▪ A file

      ▪ Size: 20 MB

      ▪ Look in C:\DB2\ directory

      ▪ File name: TS32K_CONTAINER1.DBF [Apply] [Cancel] [Next]

   • Use read/write extent size defaults   [Next]

   • Use high speed defaults   [Next]

   • Enable dropped table recovery   [Next]

   • Make sure you do a [show SQL] before you [Save] it. Then click [Finish]

   • Your saved script should look like: **Database02005.DDL**