











INTRODUCTION TO **IBM DB2 v.7 sp8**

For Microsoft Windows 2000 Professional
Revised Spring 2005

Table of Contents

1	INTRODUCTION	1
1.1	Remarks about this tutorial and last minute changes	1
1.2	Required files for the assignments	1
1.2.1	How to access the DB-SRV-1 server	1
1.3	Some Reading Guidelines	2
2	OVERVIEW OF THE DB2 ENVIRONMENT	3
1.1	Introduction	3
2.1.1	Memory aspects	3
1.1.1	Graphical User Interfaces (GUIs) versus Command Line Processors (CLPs)	3
2.2	DB2 Structures	4
2.2.1	 System	4
2.2.2	 Instance	5
2.2.3	 Database	6
2.2.4	 Tables and Table spaces	6
3	GETTING STARTED WITH IBM DB2	7
3.1	Starting the DB2 environment	7
3.2	How to stop the database manager and the server	7
3.3	When things go bad...	8
4	DB2 TOOLS	9
4.1	 The Command Center	9
4.1.1	Executing queries	10
4.2	 The Command Line Processor (CLP)	11
4.2.1	Starting CLP and changing between modes	12
4.2.2	Backing up the database	13
4.3	 The Control Center	14
4.3.1	Create a new local database	14
4.3.2	Create a new table	15
4.3.3	Setting constraints (primary and foreign keys)	18
4.3.4	Dropping a database	21
4.3.5	Dropping a table	21
4.3.6	Dropping a primary key.	22
4.3.7	Dropping a foreign key.	22

4.4	 The Information Center	23
4.4.1	How to search for information using Information Center	23
5	SQL QUERIES	24
5.1	Data Manipulation Language (DML SQL)	25
5.1.1	DELETE - to remove existing rows in tables.	25
5.1.2	INSERT - to add new rows in a table.	25
5.1.3	SELECT - to retrieve (fetch) data from existing rows.	25
5.1.4	UPDATE - to change data in existing rows.	26
5.2	Data Definition Language (DDL SQL)	26
5.2.1	CREATE - to make a new database, table, index, or view.	26
5.2.2	ALTER - to change an existing database, table, index or view	26
5.2.2	DROP - to destroy an existing database, table, index, or view.	27
5.3	Data Control Language (DCL SQL)	27
5.3.1	GRANT - to allow specified users to perform specified tasks.	27
5.3.2	REVOKE - to cancel previously granted or denied permissions.	27
5.4	Executing Queries in DB2 Command Center	27

1 Introduction

DB2 Universal Database™ version 7 from IBM is a marketplace leader in relational database technology. The complete product family ranges from servers to database management systems (DBMS) and includes different extenders that allow for complex handling of structures like text, imagery and sound. DB2 runs on virtually any computing environment and has its main focus on business applications, e-business and data warehousing.

The purpose of this document is to familiarize and present common tasks and concepts found in the DB2 environment. The aim is not to provide a complete and in-depth description of the whole environment but rather to point out key aspects that are helpful to students at the Department of Computer and System Sciences (DSV) at the University of Stockholm / Royal Institute of Technology (KTH) in Sweden when performing different course assignments. For a complete and detailed documentation of DB2 V.7, please visit the IBM website: <http://www-3.ibm.com/software/data/db2/os390/v7books.html>

1.1 Remarks about this tutorial and last minute changes

This tutorial's purpose is to make DB2 as easy, fun and comprehensible as possible. With this in mind we also know that this introduction is far from complete and that it only targets parts of the whole DBMS environment. Through experience we know that errors are going to be introduced by this text, ranging all from typos to misunderstandings, so therefore it's recommended to keep an open mind to what is presented. ***Please do not try to skip through the text since that increases the risk of misinterpretation dramatically. Try to read as much as possible!***

1.2 Required files for the assignments

Some of the assignments you will work on require that you have access to various kinds of files. All the required files are located on a file server called DB-SRV-1.

1.2.1 How to access the DB-SRV-1 server

If you happen to delete the files required for the assignments from your disk, they are also available at the DB-SRV-1 server. The files can be accessed at: <\\Db-srv-1\\StudKursInfo\\IS4 vt2005> (notice the space between "IS4" and "vt2005").

When you try to access the server a login prompt will appear. Type your personal DSV username (not db2admin), **together with the domain name** (@dsv.su.se), and your user password (for Windows) at the login prompt, as shown in the following example:



Click OK and you should now have access to the server and be able to copy the files you need to your disk if you wish.

1.3 Some Reading Guidelines



Actions: This icon symbolizes operations you are expected to perform by yourself. This can, for example, be typed commands or mouse clicks on user interface buttons.



Recommendations: This icon represents additional tips that are not required but could help speed things up as you go along.



Warnings: This symbol is the somewhat opposite of the recommendation symbol. These symbols point out things to avoid and well-known sources of errors.

Important text: This format marks extra important parts in the text that we want to emphasize because we consider it to be vital to the understanding of the text and to the progress of your work.

COMMANDS: This denotes text that is used as input to the DBMS. The DB2 commands are usually not case or location sensitive but are displayed in this introduction as illustrated below. The difference of case and prompt used in this text is only intended to ease the reading.



Answers: Represents an expected feedback from the system given as a reply to user input.

prompt: \>**KEYWORD parameter** (for example: c: \>**DB2 CONNECT TO myDatabase**
is treated the same as: m: \>**db2 connect to MYDATABASE**
and: d: \>**dB2 cOnNeCt to mYdAtAbaSE**)

2 Overview of the DB2 Environment

This chapter presents key concepts and data structures commonly used in the DB2 environment. Some DB2 user guidelines are also mentioned briefly.

1.1 Introduction

There are many ways of performing tasks in DB2, ranging from graphical wizards and step-by-step instructions to hard-core batch processing. As with everything there are benefits and downsides to whatever approach you choose. The most popular is probably a mix of it all in a way that you feel comfortable with. Maybe you choose to create a database by the help of a wizard and then use batch processing when it comes to insertion of data in your tables. Depending on your course's subject there might be some restrictions on what to use, but often it's really up to you to decide on how to do things! ***One rule of thumb though; you have a splendid opportunity of learning how to handle and understand a database manager, if you choose to rely on wizards and guides to do your work, you're not really learning how databases in general work.*** Rather you're learning how to handle IBM DB2 v. 7's user interfaces and that is something that is definitely changing each and every year. Come two or three versions later and you've probably wasted time taking this course.

Please do yourself a favor and try to find out what goes on "behind the scenes". Do it all the hard way (i.e. manually) a couple of times and only use guides when you fully understand what they do or just to get new ideas.

2.1.1 Memory aspects

DB2 user interfaces are written in Java and that equals large memory requirements. Sadly our school computers are not fully up to the task, which sometimes causes problems. Here are some issues to keep in mind in order to keep things working at their best:

- Avoid starting several DB2 tools from the start-menu. For example; if you already started *DB2 Control Center* and then want to start *DB2 Command Center*, do this from within your open *DB2 Control Center* toolbar menu and do not use the start-menu icon again. This is better since it doesn't demand as much memory.
- Try keeping executing programs at a minimum. Use simple text editing tools if possible.
- Feel free to create as many databases as you like but if you have heavy data stored (e.g. multimedia - sound, imagery or video), keep a watchful eye on your hard drives and free up space if necessary. Dropping sample or unused old databases is also recommended. (Be sure to use the *Drop* and not the *Remove* command in the *DB2 Control Center* when doing this).

1.1.1 Graphical User Interfaces (GUIs) versus Command Line Processors (CLPs)

The database manager is the core foundation of DB2. On top of this are a number of clients that enable tasks to be performed in the system. Some things are worth to mention when it comes to choosing between GUI and CLP client based actions. (See figure 1) Remember that almost everything you can do in one surrounding could be done in the other and vice versa, so don't get confused if you find several ways of doing the same thing. But first some general concerns about GUIs and CLPs:

A great thing with DB2 is its scripting possibilities. Scripts take little time to construct and in return you could recreate your whole database from nothing with just a few clicks of the mouse. Aim at creating as many scripts as possible or even try to create and populate your whole database by the means of one single text file. The benefits are numerous since you could delete and recreate the database at will and thus, at the same time, get less vulnerable for crashes or malignant deletions. During the process you'll develop an in-depth understanding of the commands used as well as the SQL syntax. There are ways of using scripts in both GUIs and CLPs and which of these clients you choose is depending on your own preference. Some are more familiar with text-based command while others like to point-and-click in a window environment. The importance is that you try to use scripts as much as possible.

Use GUIs as ideas and visual aids. If you utilize graphical user interfaces you can get explained SQL to most of the options available. These are good sources for optimization and development ideas. It is often also easier to overview the whole system from within a GUI.

CLPs are less memory demanding than using the GUI clients and are as a result often faster and not as likely to cause errors. This is important (as you can ask anyone who has had their computer's memory filled up and rebooted four times this day alone).

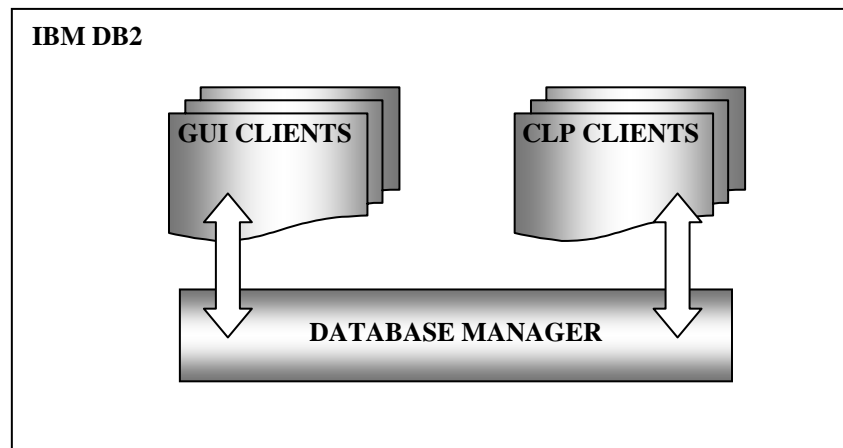


Figure: 1 – The DBMS works as a foundation for the interface clients

2.2 DB2 Structures

This section presents the different data structures that are commonly used in DB2. The composition of these objects is important in order to understand how the system is constructed and how the different part collaborates. Figure 2 is a graphical representation of the most common database objects that sums up this chapters brief introduction:

2.2.1 System

On the top of the hierarchy is the system. This shows all the actual DB2 installations available to you at this computer. DB2 administration manual refers to it as –“A logical name representing the computer with a DB2 installation”. By default you see your own locally installed DB2 system. This layer allows access to different systems over the network and connects DB2 installations over different physical machines. For information about the system, refer to the path:

file:///C:/SQLLIB/DOC/HTML/db2help/nlsyst01.htm#HDRSYST_WW_TOC

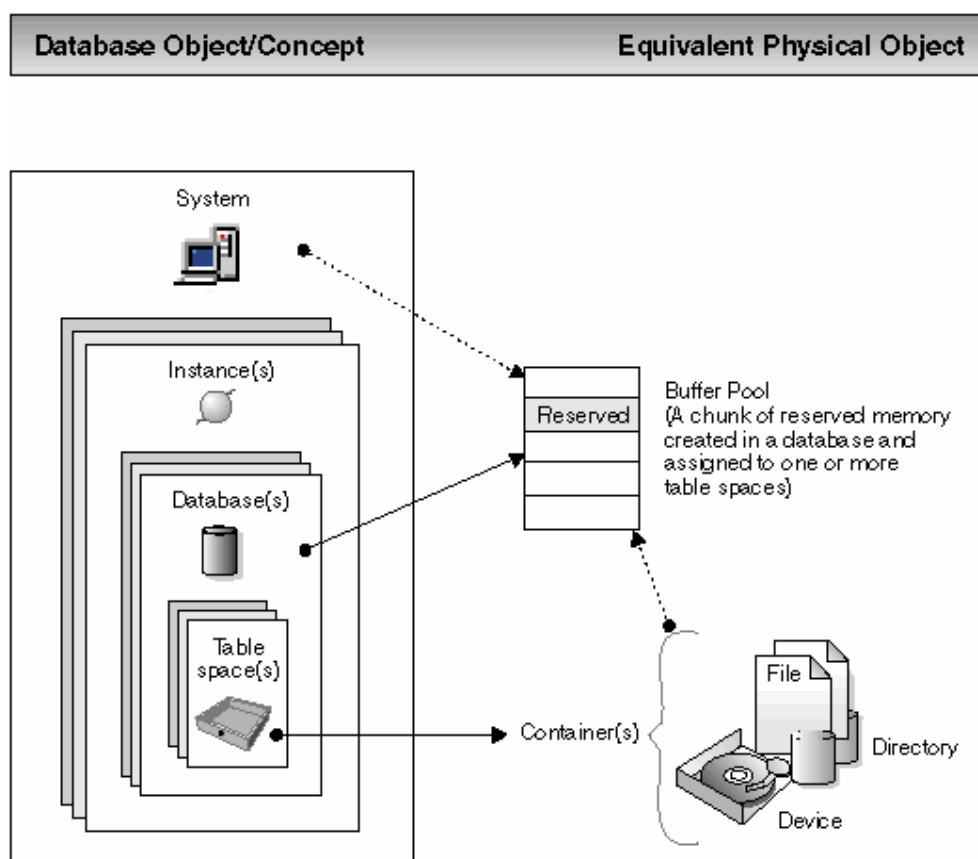


Figure: 2 – The major objects in the DB2 environment and how they relate.

2.2.2 Instance

The instance is the actual database manager that is installed on a computer running DB2. This is the central hub that all the clients connect to, meaning it is the DBMS who controls access, locks and performs all tasks that are appointed to the database(s) through the GUIs and CLPs. The DBMS is a computer program that manages data by providing the services of centralized control, data independence, and complex physical structures for efficient access, integrity, recovery, concurrency control, privacy, and security. A system could harbor any number of instances active at one and the same time.

It is the instance that is configured and started in chapter 3 - *Getting started with IBM DB2*. For more information about the instance, refer to: <file:///C:/SQLLIB/doc/html/db2help/index.htm#nlnst01>

2.2.3 Database

The database is a central concept and is often the first thing people associate with DBMSs. It contains any number of table spaces that contain the actual user data and database objects, and it represents one or several physical files or drives on the computer system. The database, as a container of pages of data, sets the physical size allocated in the system and manages main memory through the use of buffer pools. Since most database courses only covers single database access this is often the top container you will encounter. Use the database metaphor to manage your tables and optimize performance through table spaces and buffer pools. For more information about the database, refer to the *DB2 Information Center* path: <file:///C:/SQLLIB/DOC/HTML/db2help/nldbass01.htm>

2.2.4 Tables and Table spaces

A relational database presents data as a collection of tables. A table is a named database object consisting of a specific number of columns and some unordered rows that holds persistent data.

The data in the tables is logically related, and relationships can be defined between tables. Data can be viewed and manipulated based on mathematical principles and operations called relations, and it is accessed through *Structured Query Language* (SQL), a standardized language for defining and manipulating data in a relational database. (See chapter 5). A query is used in applications or by users to add, manipulate and retrieve data from a database.

Basically, tables are objects that help gather, group and link together information that are related. Along with the database the table is probably one of the most common database object you encounter when trying to learn how to handle a DBMS.

All table data are assigned to table spaces. A database is organized into parts called table spaces, which are physical allocations on the disc. When creating a table, you can decide to have certain objects such as indexes and large object (LOB) data kept separately from the rest of the table data. A table space can also be spread over one or more physical storage devices to optimize performance. For more information about the tables and table spaces, refer to the *DB2 Information Center* paths: <file:///C:/SQLLIB/DOC/HTML/db2d0/db2d0102.htm> and <file:///C:/SQLLIB/doc/html/db2d0/frame3.htm#db2d045>.

3 Getting started with IBM DB2

Basically, there are two major ways to issue commands in DB2. Either you use the *DB2 Command Window / DB2 Command Line Processor* tool or you take advantage of the DB2 graphical user interfaces like *DB2 Control Center* and *DB2 Command Center*. This section assumes that you use the *DB2 Command Line Processor* in *Command mode* (i.e. a *DB2 Command Window*) to type in the commands. (If the concept of different modes brings confusion, please refer to section 4.3 for an explanation on the concepts).

3.1 Starting the DB2 environment

Before we can start working with DB2, we need to start the complete DB2 environment that we are going to use during this course. This includes several servers, extenders and help services. To make things as easy as possible, all start commands have already been created and put in a batch-file on your start-menu. ***Please note that since all services are stopped when you log out from your account you'll need to use this startup procedure every time you log back on!***



Start all services by following the start-menu path: ***Start » Programs » DatabaseManagementSystems » IBM-Database-Systems » IBM DB2 » netStartDB2***



A good way to actually see what happens when issuing the start commands is to open a *DB2 Command Line Processor* in *Command mode* and type in ***c:\myprog\netstartdb2.bat***



Yet another good way of skipping the whole workaround with the long start-menu path is to use the *run* command: ***c:\myprog\netstartdb2.bat*** After the first time the command is found under the drop-down list which gives a speedier access to the batch-file.



It is easy to forget to start all services again and again. If you get mysterious errors, try closing all programs, stopping all services (the *netStopDB2* command found at the same place as the *netStartDB2* command) and then issue the *netStartDB2* command again. Quite often DB2 tend to freeze (and/or crash) and the safest way is to restart all services when this happens.

3.2 How to stop the database manager and the server

Constructed in a similar way as the start batch-file, *netStopDB2* has a number of commands that terminates all servers and services started by *netStartDB2*. This is very handy since the DB2 environment is constructed of several processes who are quite error prone on our multi user system. Sometimes dead locks (“freezes”) occur and you cannot use the system although you restart the “program” (i.e. often just the user interfaces). This is not enough because DB2 consists of several layers and if you get errors on the server layer, restarting the interfaces wouldn’t solve the problem. The safest way to ensure system integrity is to restart the whole DBMS when you get in to any kind of problem. To manually stop the DBMS, do the following:



Stop all services by following the start-menu path: **Start » Programs » DatabaseManagementSystems » IBM-Database-Systems » IBM DB2 » netStopDB2**



A good way to actually see what happens when issuing the stop command is to open a command prompt or *DB2 Command Line Processor* and type in:
c:\myprog\netstopdb2.bat



As with the *netStartDB2* command, speed up access by using the *run* command:
c:\myprog\netstopdb2.bat



If nothing works, not even the stop command, you might want to issue the command: **DB2 FORCE APPLICATION ALL** in a *Windows Command Prompt* or a *DB2 Command Window*, which forces every user and application off the system i.e., terminates any dead locks that might exist.

3.3 When things go bad...

In order for you to get the most out of this database experience you are appointed database administrator privileges on your local machine. This is good because it means that you are free to practice with what you are interested in and at the same time it is bad because you are also free to make radical mistakes. Sometimes things just don't work, without any obvious reason. That is why it is recommended to commit changes only small steps at a time.

Try to remember what you have done and backtrack, if possible, when things don't turn out as they are supposed to. Change and test, and then change and test again! Keep an open mind and even try to redo the problem in order to understand what is causing it. Be persistent and don't give up! To work out a problem often requires a lot of detective work so any clue is vital. The following pointers are a small attempt to make a checklist to work through when solving problems. The list is presented in linear order, starting with the easiest solutions. Skip parts that don't seem relevant to your problem:



If SQL-problem, check syntax or try another formulation of the query. Break the query down to smaller pieces so it is exactly clear what part/line/keyword is causing the error.



Close all running programs and try again



Use *netStopDB2* (or/and use the **FORCE APPLICATION ALL** command)



Start servers and services through *netStartDB2* and try again



Restart the computer (a reboot fixes memory problems) and try again



Search the course First Class conference for any similar problem, post if not present.



If working in DB2 GUI environment, try doing the same in DB2 CLP instead



If command not working in a *DB2 Command Line Processor*, try ordinary *Windows Command Prompt* instead with DB2 in front of command. (C:>\DB2 CONNECT TO myDB)



Check the space on the hard drive the database is placed on, remove something if full and try again.







Drop your whole database and recreate it with your script.



Search for any reference of the error message, error code or SQL state, if available, in *DB2 Information Center*

4 DB2 Tools

This chapter gives a short introduction to the most commonly used interfaces provided in the DB2 environment:

1.  The *Command Center* is a graphical client with which you can execute DB2 commands, SQL statements and work with command scripts.
2.  The *Command Window* and the *Command Line Processor (CLP)* are text-based clients used to execute DB2 commands, SQL statements, to create database command scripts or to remotely manage a database server.
3.  The *Control Center* is a graphical interface (GUI) client that will allow you to display all of your systems, databases, and database objects and perform administration tasks on them.
4.  In the *Information Center* you can find instructions, answers, and other information about DB2 Universal Database. Most often you will use it to find the syntax of a given SQL command.

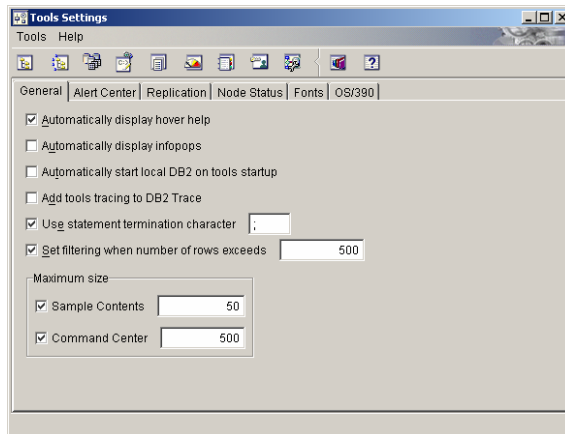
We recommend that you use either the *Command Center* or the *Command Window* and *Command Line Processor (CLP)* as it will be by using them that you will learn the SQL commands needed to create and manage an SQL database. By blindly using only the *Control Center* you will not learn the underlying SQL commands that the *Control Center* GUI interface hides from you.

4.1 The Command Center

Use the *DB2 Command Center* to execute DB2 commands and SQL statements, work with command scripts, and view a graphical representation of the access plan for explained SQL.



Pay special attention to which function (i.e. tab) you are using in *Command Center*. *Interactive mode* lets you execute **one** single SQL command or query at a time and saves the latest queries in a handy drop-down list. *Script mode* is used to execute several commands in sequence and provides for later saving all the commands in a script file by going to the menu *Script » Save Script As...*



Always check your *Statement Termination Character*. If selected, your query always has to end with the specified character. If not selected, every line is considered a query in *Script mode* but only the first line will be processed in *Interactive mode*. You find the *Statement Termination Character* under the menu-bar path: *Tools » Tool Settings* found in *Command Center*. Once in the *Tool Settings* window, the option is found under the *General* tab.



In some cases (often when creating triggers or stored procedures) it can be necessary to use a termination character other than ; since the same character may appear inside the trigger's or stored procedure's body. You can for example use the @ symbol.



It is also strongly recommended to change the *Command Center Options*, which could be found under the menu *Command Center » Options...* On the *Results* tab check the *Verbose* option so that every command is shown in the output and uncheck the option to *Automatically display results on the query results page* since, when selected, this makes saving (or rather copying) the results much harder.

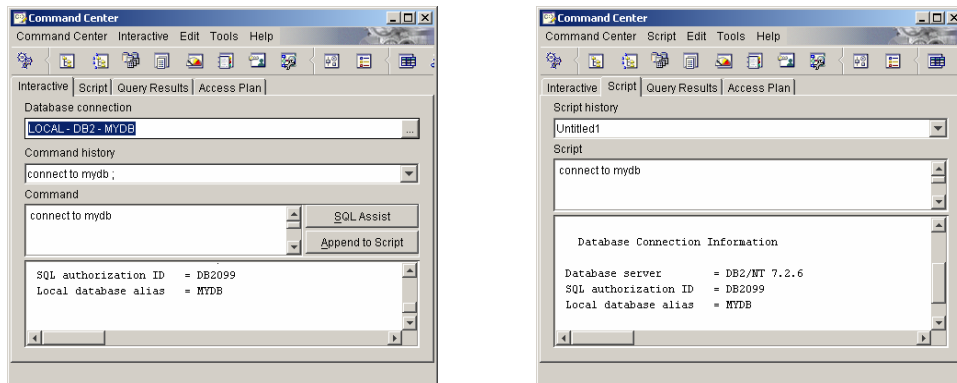
4.1.1 Executing queries



Open DB2 Command Center by using the start-menu path: ***Start » Programs » Database Management Systems » IBM-Database-Systems » IBM DB2 » Command Center*** or if you already have a DB2 GUI open, use the *Command Center* button on the toolbar instead. (Remember that queries could also be executed through a CLP. See chapter 4.3.1 for details on that matter).



If you want to execute a query on the database you have to establish a connection to it first. Select either *Interactive* or *Script mode* by selecting the corresponding tab in *Command Center*, and use the syntax: ***CONNECT TO <database-name>***. Database-name stands for the name of the database you want to connect to. It doesn't matter if you switch between modes after the connection has been made; the connection is the same for both of them.



In *Interactive mode* you'll see your connection displayed in the *Database connection* field. In *Script mode* you'll get no such feedback.



Type, import or paste (ctrl + v) the query you want to execute and press the cogwheel icon on the top left corner to execute it (or ctrl + enter).



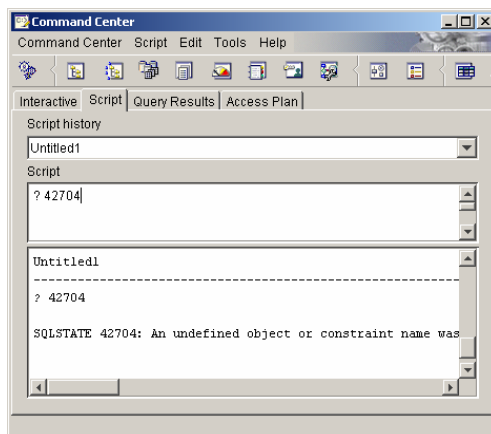
As long as the cogwheels are spinning the query is executing. If you want to interrupt the process, press the cogwheel icon again.



Depending on your settings and the query, you either see the execution result on the command log or on the query results page. If you want to change this behavior go to the menu *Command Center » Options* and in the *Results* tab set "*Automatically display query results on the query results page*".



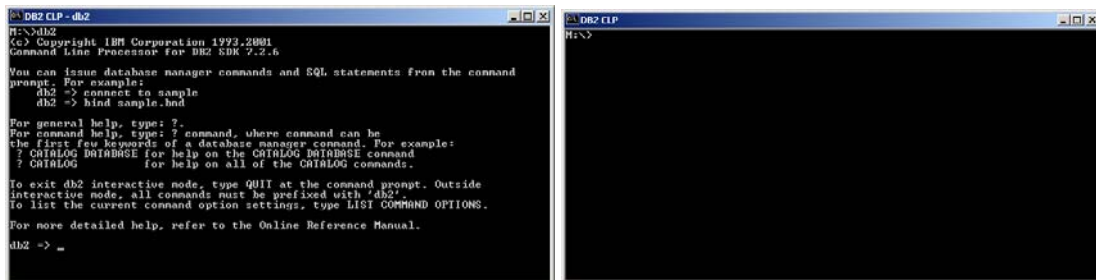
If you are uncertain about a specific command you can use the '?'-sign in front of the command in *DB2 Command Center* who will then present the command specific syntax. The same goes for SQL states, use for example: ? 42704 to get a textual explanation of that state. Handy when solving query problems.



4.2 The Command Line Processor (CLP)

The CLP is used to execute database utilities, SQL statements and online help. It offers a variety of command options, and can be started in: *Interactive input mode* (characterized by the *db2 =>* input prompt), *Command mode* (where each command must be prefixed by *DB2*) or in *Batch mode* (which uses the *-f* file input option). **Hence, DB2 Command Line Processor and DB2 Command Window is the same tool acting in different modes.**

When working with DB2 extenders for multimedia (IAV Extender) or XML data (XML Extender) you will also encounter command line processors specifically tailored to manage the functions provided by the extenders.



4.2.1 Starting CLP and changing between modes



Open the *DB2 Command Line Processor* by using the start-menu path: **Start » Programs » Database Management Systems » IBM-Database-Systems » IBM DB2 » Command Window** or simply use **Start » Run...** and type in `db2cmd`.



A window similar to *Windows Command Prompt* opens. With the tiny difference that in the top left corner it says *DB2 CLP* and not *C:\WINNT\system32\cmd.exe*.



You are now in *Command mode*, which is almost the same as a plain *Windows Command Prompt* with the DB2 environment initialized in the background. To switch to interactive mode just type `DB2` and the prompt switches to `db2=>`. The two modes are very alike with one major difference; you don't need to specify the *DB2* prefix in interactive mode. This means, to write: `SELECT * FROM myDB` in *Interactive mode* is exactly the same as writing: `DB2 SELECT * FROM myDB` in *Command mode*.



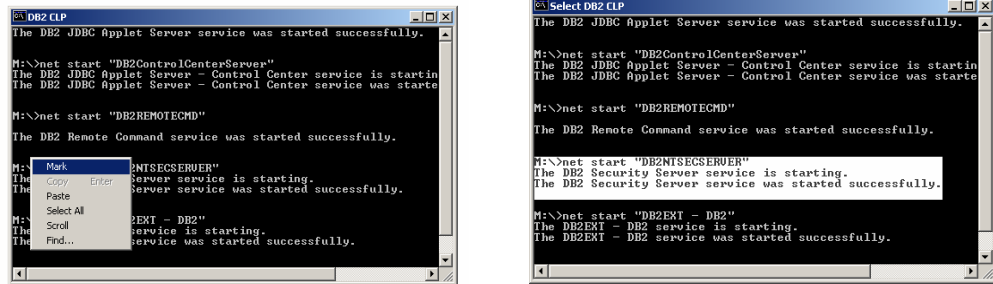
To switch back to *Command mode* from *Interactive mode*, just type `quit` and the prompt goes back to normal.



To switch to *Batch mode* initiate the *Interactive mode* with the command: `DB2 -f <filename>` where `<filename>` is the path and the name of the text file containing the SQL script. (e.g.: `DB2 -f m:\SQL\myscript.sql`).



A good thing is to right-click in the CLP window and to select *Mark* which gives the opportunity to select an area through click-and-drag with the mouse. When typing *Enter* all marked text is copied and could be reinserted by right-clicking and *Paste* back in the CLP window or through `ctrl + v` in most other Windows applications.



```

DB2 CLP
The DB2 JDBCC Applet Server service was started successfully.
M:\>net start "DB2ControlCenterServer"
The DB2 JDBCC Applet Server - Control Center service is starting.
The DB2 JDBCC Applet Server - Control Center service was started successfully.
M:\>net start "DB2REMOTECMD"
The DB2 Remote Command service was started successfully.
M:\>net start "DB2SECURITY"
The DB2 Security Server service is starting.
The DB2 Security Server service was started successfully.
M:\>net start "DB2EXT - DB2"
The DB2EXT - DB2 service is starting.
The DB2EXT - DB2 service was started successfully.

Select DB2 CLP
The DB2 JDBCC Applet Server service was started successfully.
M:\>net start "DB2ControlCenterServer"
The DB2 JDBCC Applet Server - Control Center service is starting.
The DB2 JDBCC Applet Server - Control Center service was started successfully.
M:\>net start "DB2REMOTECMD"
The DB2 Remote Command service was started successfully.
M:\>net start "DB2SECURITY"
The DB2 Security Server service is starting.
The DB2 Security Server service was started successfully.
M:\>net start "DB2EXT - DB2"
The DB2EXT - DB2 service is starting.
The DB2EXT - DB2 service was started successfully.

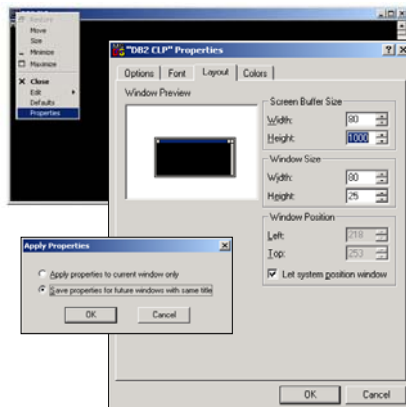
```



The *Information center* can be started directly from CLP by the command: **DB2 help**.



Often the CLPs are not capable of showing the whole answer on a single screen and without any scrollbars in the window it's hard to view the printouts. To solve this issue, right-click on the CLPs top left corner (i.e. the window's icon) and select *Properties* from the drop-down menu. Under the *Layout* tab in the window that opens, increase the *Height* value of the *Screen buffer size* to anything around the value 1000. If you like you could also change the screen color so you recognize the CLP easier in the future. Then press *Ok* to show the *Apply Properties* window. In this window, select the *Save properties for future windows of the same type* radio button and press *OK* to make these settings appear the next time you start the CLP again.



4.2.2 Backing up the database



Open the *DB2 Command Line Processor* by using the start-menu path: **Start » Programs » DatabaseManagementSystems » IBM-Database-Systems » IBM DB2 » Command Window** or simply use **Start » Run...** and then type in **db2cmd**.



Create a backup directory on D: by the command: **MD d:\DB2backup**



Disconnect access to the database through the command: **DB2 DISCONNECT ALL**



DB20000I The SQL DISCONNECT command completed successfully.



Terminate the CLP processes through the command: **DB2 TERMINATE**



DB20000I The TERMINATE command completed successfully.



Force all users and client applications off the database by the command: **DB2 FORCE APPLICATION ALL**



**DB20000I The FORCE APPLICATION command completed successfully.
B21024I This command is asynchronous and may not be effective immediately.**



Backup the database with the command: *DB2 BACKUP DB <database-name> TO <path>* where database-name is exchanged with the name of the database you want to backup. For example: **DB2 BACKUP DB mydb TO m:\DB2backup**



**Backup successful. The timestamp for this backup image is :
20040124143208 (stamp is different every time)**

4.3 **The Control Center**

The *DB2 Control Center* is the heart of the DB2 GUIs, and in this environment; you can display all of your systems, databases, and database objects and perform administration tasks on them. From this client, you can also open other centers and tools to help you optimize queries, jobs, and scripts, perform data warehousing tasks, create stored procedures, and work with DB2 commands.

As mentioned before we encourage students to avoid using *DB2 Control Center* as much as possible as by using a GUI all the needed SQL commands will be hidden from you. Pay attention to the *Show SQL* button that appears in most of the forms as by clicking on it you will be shown the SQL the GUI send to the server.

The following sub-sections are examples of different tasks that can be performed through this interface:

4.3.1 Create a new local database



Open *DB2 Command Center* by using the start-menu path: **Start » Programs » DatabaseManagementSystems » IBM-Database-Systems » IBM DB2 » Control Center** or if you already have a DB2 GUI open, use the *Control Center* button on the toolbar.



Double click your local *Systems* icon to show your local instances and then browse down to the *Databases* folder icon.

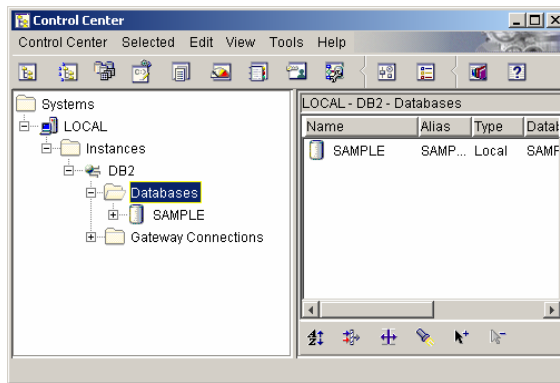


Figure: 3 – Finding the local databases in the Control Center



Right-click the *Databases* folder icon in order to bring up the sub-menu. Select *Create » Database Using Wizard....*

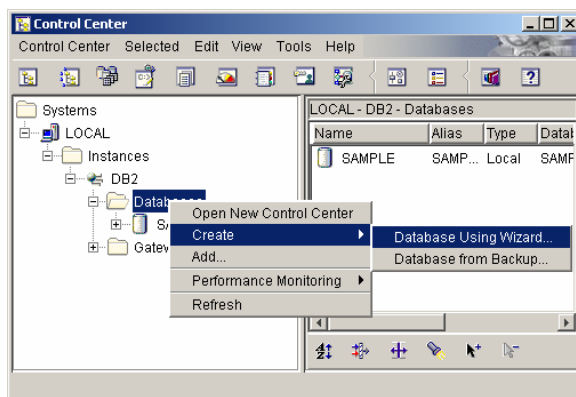


Figure: 4 – Starting the database wizard.



Fill in the eight character name of the database in the red bordered field but **avoid using names that include white space or non-English characters** like å, ä, ö. (It works fine but could cause problems, especially when writing SQL queries!) If you are content with the rest of the default setup, click *Finish* to create your database and return to *DB2 Control Center*.

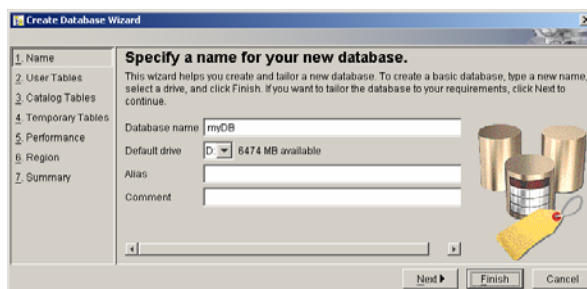


Figure: 5 – The database wizard in Control Center



Note that your newly created database is added under the *Databases* folder icon.

4.3.2 Create a new table



Double click your database from the left hand side tree list to view its content. From this view you have access to all parts of your database like tables, triggers, indexes and so on. Right-click on the *Tables* folder icon to bring up the sub-menu and select *Create » Table....*

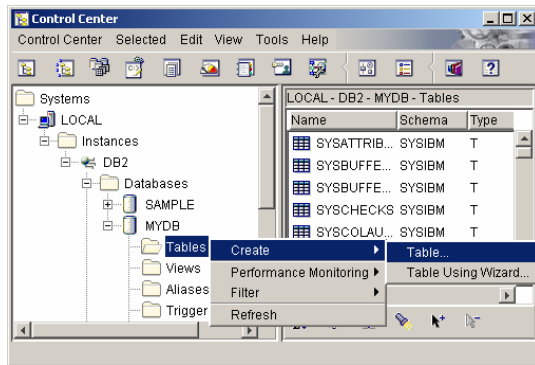


Figure: 6 – To create a table in Control Center



In the *Create Table* window; fill in the table name in the red bordered field. As said before for the name of the database, **avoid here too the use of names that include white space or non-English characters** like å, ä, ö. In this example we'll use *myPerson* as the first table. Then click the tab *Columns*.

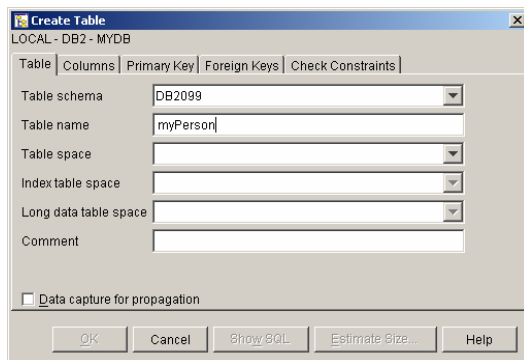


Figure: 7 – To specify the table name



On the *Columns* tab press the *Add...* button to add a new column through the *Add Column* window.

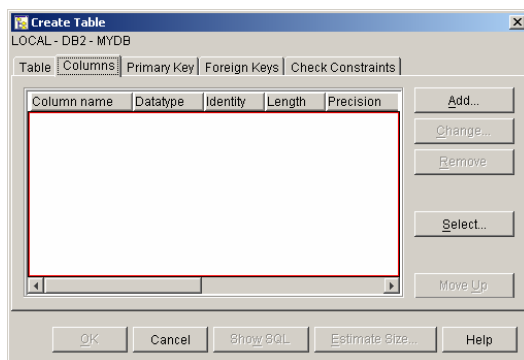


Figure: 8 – Adding columns to a table



In the *Add Column* window, type in the name of the new column that you want to insert into your table, specify the details of the column; the data type and its characteristics and if you want it to nullable or not. (Important when defining primary keys). When you are content, press *OK* to close this window, or if you want to add additional columns press *Apply* to be able to specify another column name. In this example we add the columns *pName* and *pDog*, both of the data type *character* with a default length of ten characters. Since we want the column *pName* to be the identifier and primary key of this table, the *nullable* option is unchecked to disallow any null values in this column.

The 'Add Column' dialog box shows the following settings for column 'pName':
 Column name: pName
 Datatype: CHARACTER
 Length: 10
 LOB unit: Bytes
 Precision: (empty)
 Scale: (empty)
 LOB option: ☐ Logged
☐ Bit data
☐ Compact
☐ Nullable

The 'Add Column' dialog box shows the following settings for column 'pDog':
 Column name: pDog
 Datatype: CHARACTER
 Length: 10
 LOB unit: Bytes
 Precision: (empty)
 Scale: (empty)
 LOB option: ☐ Logged
☐ Bit data
☐ Compact
☒ Nullable



In the *Create Table* window, check that the inserted values corresponds to the settings you wanted and then press *OK* to create your new table.

The 'Create Table' dialog box shows the following table structure:

Column name	Datatype	Identity	Length	Precision
PNAME	CHARAC...	No	10	
PDOG	CHARAC...	No	10	

Buttons: Add..., Change..., Remove, Select..., Move Up, OK, Cancel, Show SQL, Estimate Size..., Help.

Figure: 9 – The new columns specified



Note that your table is added to the folder *Tables* in *DB2 Control Center*. In this folder there are a lot of system-defined tables that hold meta-information about your database. An easy way to spot your own tables is to look at the schema and table spaces for each table since the system defined ones differ from your own ones.



A good way to get an understanding of the actions that lay beneath the GUIs is to use the *Show SQL* button. This shows the GUI generated textual commands that are sent to the DBMS. As you soon will see these are all just basic SQL commands, the very same that you could use in your own scripts. Understand what the syntax stands for and use these as building blocks when creating your own scripts.

4.3.3 Setting constraints (primary and foreign keys)



Start with creating the table *myPerson* as described in the previous section and then start creating an additional new table and name this *myDog*. In the *myDog* table, add a column *dName*, set the data type to *character* with the length *10*, and disallow any null values.

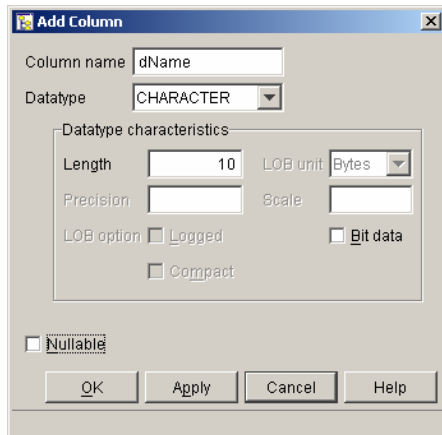


Figure: 10 – Specifying the column details



Still in the *Create Table* window, click on the *Primary Key* tab and then select the primary key constraint *dName* by moving it from the left to the right field. Label the constraint with a meaningful name (optional) and press *OK* to create the new *myDog* table and return to *DB2 Control Center*.

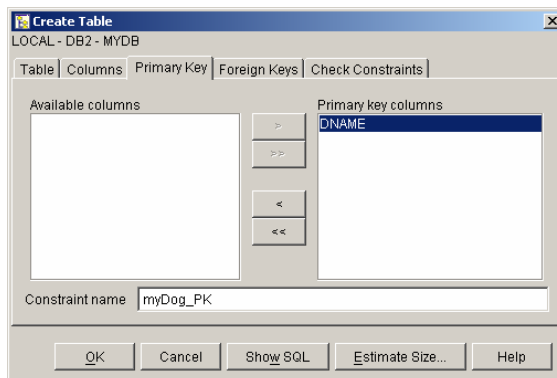


Figure: 11 – Setting the primary key



From the *DB2 Control Center*, select the *Tables* folder icon in the tree list on the left hand side and find the *myPerson* table, right-click it and select *Alter...*

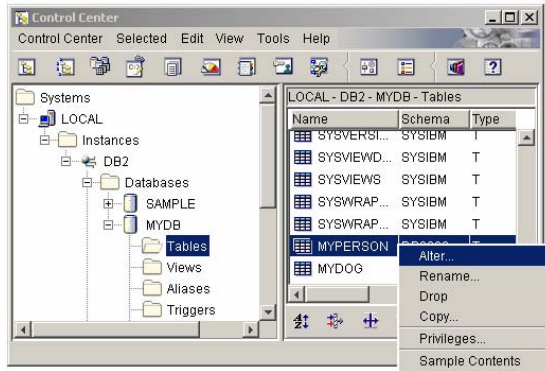


Figure: 12 – To alter a table in Control Center



In the *Alter Table* window that opens, select the tab *Primary Key* and select what columns that should be included in the primary key by moving them over from the left to the right hand side field. Feel free to name the constraint with a meaningful name so that you have an easier job identifying the primary key later. (If you do not specify a name, DB2 will generate a meaningless name for you). In this example we let the column *pName* constitute the primary key of the table *myPerson*.

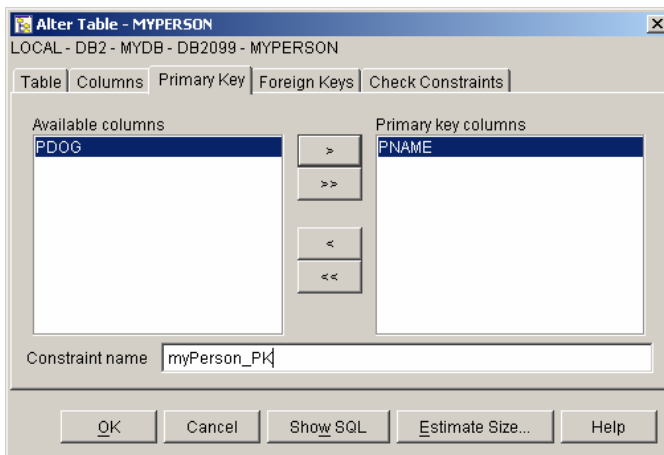


Figure: 13 – Selecting the primary key



To create a foreign key, select the *Foreign Key* tab in the *Alter Table* window and press *Add...* to bring up the *Add Foreign Key* window. In this window select the parent table in the upper section and mark the selected primary key you want as a determinant for the foreign key. In this example we choose *myDog.dName* to be the determinant of this foreign key relation

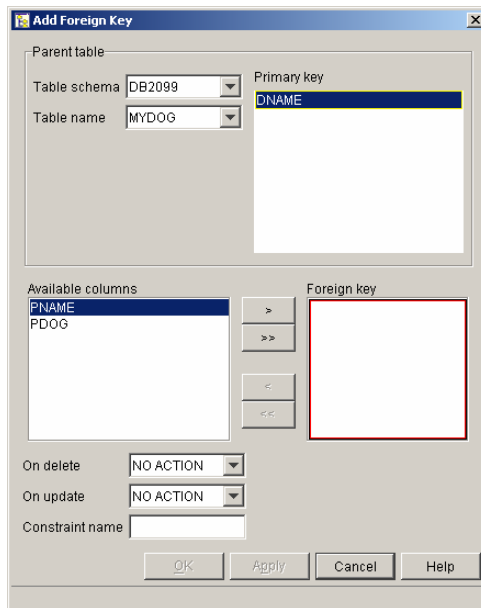


Figure: 14:1 – Specifying the determinant of a foreign key



In the lower part, select which column you want to link in the to the determinant by moving it from the *Available columns* field to the *Foreign key* field and optionally add a name for the constraint. In this example we bind the *dName* determinant in the table *myDog* with the *pDog* column in the table *myPerson*, stating that each person could be the owner of a dog. Furthermore we state that if a dog is deleted from the system, the owner's (*myPersons*) value of *pDog* should be updated to null (*ON DELETE SET NULL*).

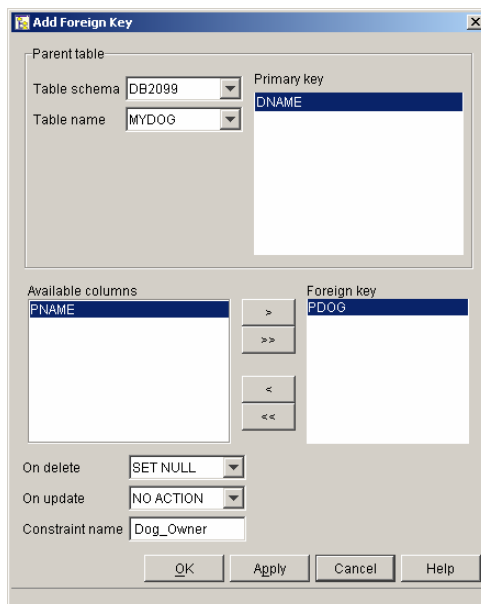


Figure: 13:2 – Specifying the dependent fields of a foreign key



After everything is set, press *OK* to return to the *Alter Table* window. And then *OK* again to return to *Control Center*, or *Explain SQL* to show the syntax sent to the database manager.

4.3.4 Dropping a database



In *DB2 Control Center* right-click the database you want to delete in the tree list on the left hand side and select *Drop*. This opens a window that ask you to verify your deletion. Make sure it is the correct database and press *OK*.

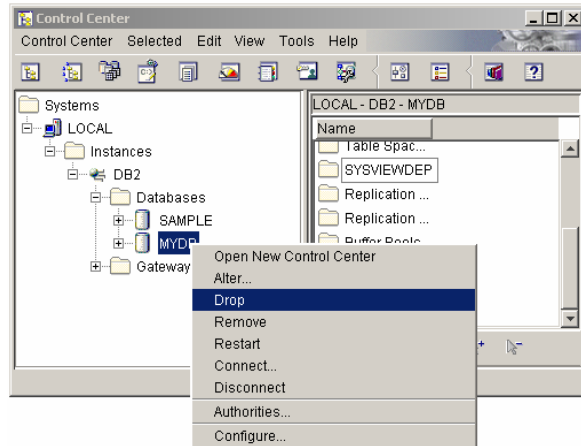


Figure: 15 – Dropping a database in Control Center



Note that the database is removed from the *Databases* folder in the tree list on the left hand side.



Do not use the *Remove* command since this only removes the database from the tree view and not from the database manager. Any database removed from *Control Center* could be retrieved through the *Add...* command.

4.3.5 Dropping a table



From the *Control Center* select the *Tables* folder icon in the tree list on the left hand side and then find and right-click the table and select *Drop*. This opens a window that asks you to verify what you want to delete. Make sure it is the correct table and press *OK*.

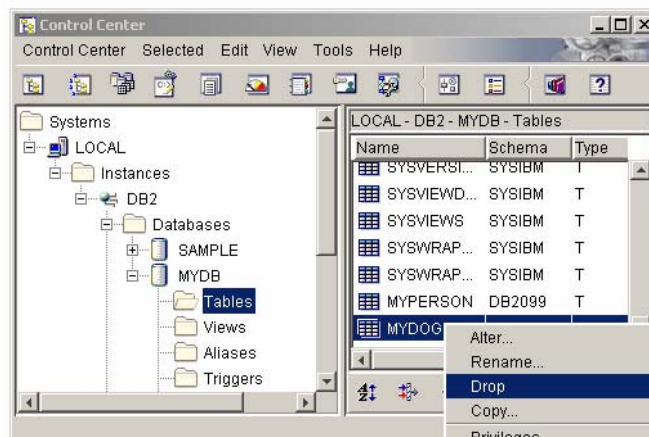


Figure: 16 – Dropping a table in Control Center



Note that the table is removed from the *Tables* folder.

4.3.6 Dropping a primary key.



From the *Control Center* select the *Indexes* folder icon in the tree list on the left hand side and then find and right-click the index you want to delete and select *Drop*. This opens a window that asks you to verify what you want to delete. Make sure it is the correct constraint and press *OK*.

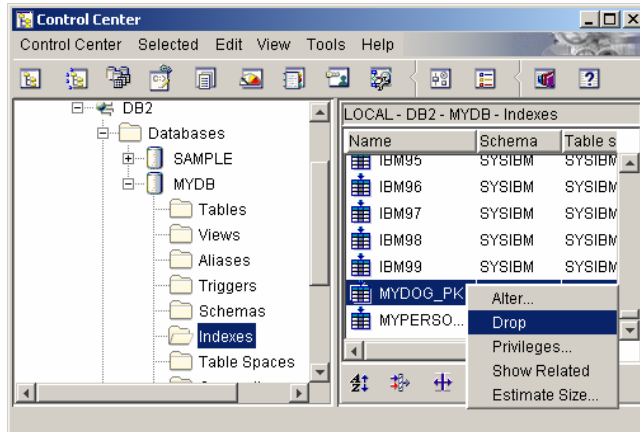


Figure: 17 – Dropping a primary key in Control Center

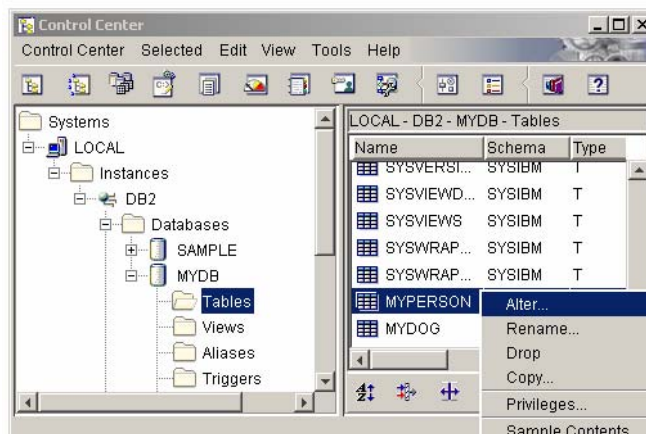


Hopefully you have named your constraints in an understandable way so that you could select them easily. If you've forgot to name your keys you'll have to go into each index by the *Alter...* command to see what columns it uses or look at the *Table name* section in *Control Center*.

4.3.7 Dropping a foreign key.



From the *Control Center* look for your linked table in the *Tables* folder icon in the tree list on the left hand side, then find and right-click the table and select *Alter...* to bring up the *Alter Table* window.



In the *Alter Table* window select the *Foreign Keys* tab. Then select the foreign key constraint you want to remove and press *Remove*. Commit the drop and return to *Control Center* by clicking *OK*

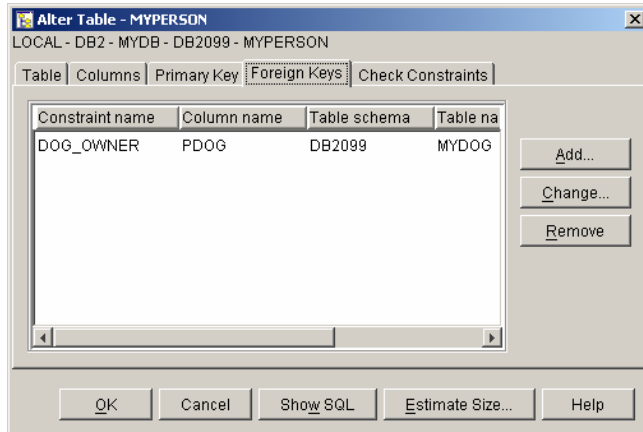


Figure: 18 – Dropping a foreign key in Control Center

4.4 The Information Center

Use the *Information Center* to find instructions, answers, and other information about DB2 Universal Database. It is divided into six groups (shown on the different tabs), that presents the same information structured in different ways:

Tasks - lists common tasks using the Control Center and other major administration and programming tasks.

Reference - addresses particular keywords or concepts from the SQL Reference and Command Reference etc.

Books - shows information from particular books and what books are available.

Troubleshooting - handles answers to common problems encountered while using DB2 like error messages, "how-to"-documents and so on.

Sample Programs - presents source code and featured programs.

Web - is about sources for DB2 information on the Web.

The Information Center is a Java application that launches a Web browser to display the information. Much of the documentation is in PDF format that could also be found at the URL: http://www-3.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/v7pubs.d2w/en_main

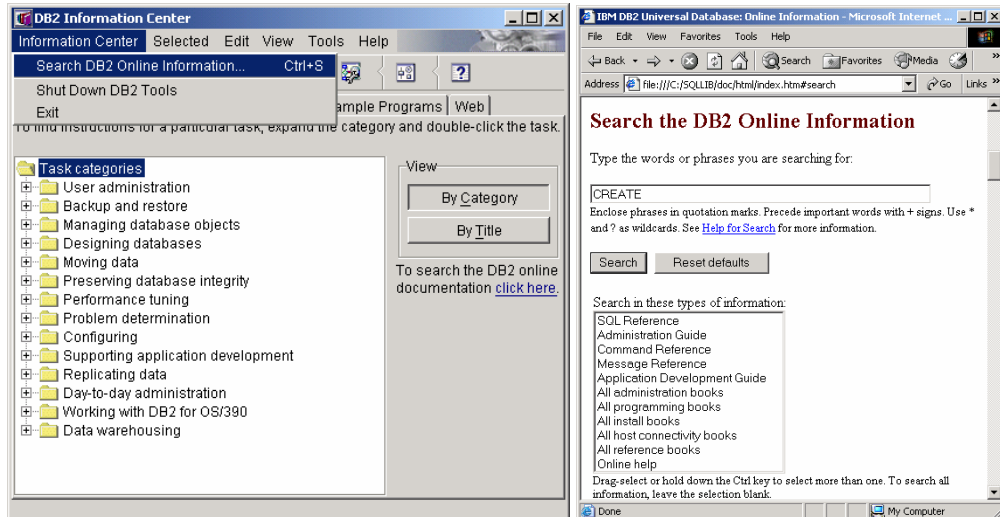
4.4.1 How to search for information using Information Center



Open *DB2 Information Center* by using the start-menu path: **Start » Programs » Database Management Systems » IBM-Database-Systems » IBM DB2 » Information » Information Center** or if you already have a DB2 GUI open, use the *Information Center* button on the toolbar instead. If you are using a DB2 CLP, type in the command `db2 help` if you are in *Command mode* or just `help` if you're in *Interactive mode*.



In the information center select: *Information Center » Search DB2 online information...* from the menu-bar to launch the Web browser.



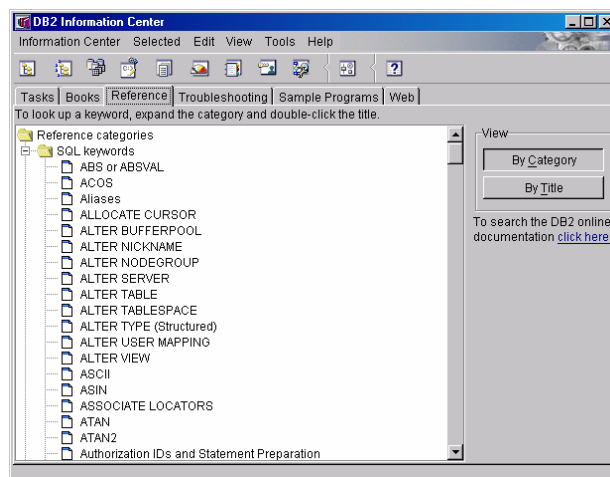
Make sure you have started the search service before launching the Web browser (i.e. the netDB2Start batch-file. See chapter 3.2 for details on that) or the search will fail and display a “*Page cannot be displayed*” response.



Type in the keywords of your search in the Web browser window and delimit your search by selecting sources to search from. Not specifying any source causes a full search of all the sources. Then press the *Search* button the get directed to the search results.



To find the syntax of a given SQL command you can also go to the *Reference* tab, open the *SQL keywords* folder and double-click on the command you are interested in.



5 SQL Queries

SQL is a cross platform language used to select, update, insert, or delete data in relational databases. SQL is also used to administer the RDBMS (relational database management system). SQL was developed in the 1970s by IBM and is supported by most of the commercial RDBMS producers including Oracle, Sybase, and Microsoft SQL Server.

However, each DBMS has its own variation of SQL. Between dialects of SQL, the basic access syntax does not vary much from the "official" ANSI and ISO version (a.k.a. SQL-89/SQL1 and SQL-92/SQL2).

This chapter provides a few short examples on how to query the DB2 database. For a full syntax definition of the examples and other queries please refer to the *DB2 SQL Reference* on-line or use the *DB2 Information Center* and search on keywords in the *SQL Reference* manual.

5.1 Data Manipulation Language (DML SQL)

These SQL statements are used to retrieve and manipulate data. This category encompasses the most fundamental commands including DELETE, INSERT, SELECT, and UPDATE. DML SQL statements have only minor differences between SQL variations and the commands include the following:

5.1.1 DELETE - to remove existing rows in tables.

Delete department (DEPTNO) 'D11' from the DEPARTMENT table.

```
DELETE FROM department
WHERE deptno = 'D11'
```

5.1.2 INSERT - to add new rows in a table.

Insert a new department with the following values into the DEPARTMENT table:

Department number (DEPTNO) is 'E31'

Department name (DEPTNAME) is 'ARCHITECTURE'

Managed by (MGRNO) a person with number '00390'

Reports to (ADMRDEPT) department 'E01'.

```
INSERT INTO department
VALUES ('e31', 'architecture', '00390', 'e01')
```

5.1.3 SELECT - to retrieve (fetch) data from existing rows.

List every department with the employee number and last name of the manager, including departments without a manager.

```
SELECT deptno, deptname, empno, lastname
FROM department LEFT OUTER JOIN employee
ON mgrno = empno
```

List the employee numbers (EMPNO) of all employees in the EMPLOYEE table whose department number (WORKDEPT) either begins with 'E' or who are assigned to projects in the EMP_ACT table whose project number (PROJNO) equals 'MA2100', 'MA2110', or 'MA2112'.

```
SELECT empno
FROM employee
WHERE workdept LIKE 'e%'
UNION
SELECT empno
FROM emp_act
WHERE PROJNO IN ('ma2100', 'ma2110', 'ma2112')
```

Using the *EMPLOYEE* table, select the department number (*WORKDEPT*) and maximum departmental salary (*SALARY*) for all departments whose maximum salary is less than the average salary in all other departments.

```
SELECT workdept, MAX (salary)
FROM employee emp_cor
GROUP BY workdept
HAVING MAX (salary) < (SELECT AVG (salary)
FROM EMPLOYEE
WHERE NOT workdept = emp_cor.workdept)
```

5.1.4 UPDATE - to change data in existing rows.

All the employees except for the manager of department (*WORKDEPT*) 'E21' have been temporarily reassigned. Indicate this by changing their job (*JOB*) to NULL and their pay (*SALARY*, *BONUS*, *COMM*) values to zero in the *EMPLOYEE* table.

```
UPDATE employee
SET job=NULL, salary=0, bonus=0, comm=0
WHERE workdept = 'E21' AND job <> 'manager'
```

5.2 Data Definition Language (DDL SQL)

These SQL statements define the structure of a database, including rows, columns, tables, indexes, and database specifics such as file locations. DDL SQL statements are more part of the DBMS and have large differences between the SQL variations. DDL SQL commands include the following:

5.2.1 CREATE - to make a new database, table, index, or view.

Create table TDEPT in the DEPARTX table space. DEPTNO, DEPTNAME, MGRNO, and ADMRDEPT are column names. CHAR means the column will contain character data. NOT NULL means that the column cannot contain a null value. VARCHAR means the column will contain varying-length character data. The primary key consists of the column DEPTNO.

```
CREATE TABLE tdept
  (deptno    CHAR(3)      NOT NULL,
   deptname  VARCHAR(36)  NOT NULL,
   mgrno     CHAR(6),
   admrdept  CHAR(3)      NOT NULL,
   PRIMARY KEY (deptno))
IN departx
```

1.1.2 ALTER – to change an existing database, table, index or view

Add a referential constraint to the *EQUIPMENT* table so that the owner (*EQUIP_OWNER*) must be a department number (*DEPTNO*) that is present in the *DEPARTMENT* table. *DEPTNO* is the primary key of the *DEPARTMENT* table. If a department is removed from the *DEPARTMENT* table, the owner (*EQUIP_OWNER*) values for all equipment owned by that department should become unassigned (or set to null). Give the constraint the name *DEPTQUIP*.

```
ALTER TABLE equipment
ADD CONSTRAINT deptquip
FOREIGN KEY (equip_owner)
REFERENCES department
ON DELETE SET NULL
```

5.2.2 DROP - to destroy an existing database, table, index, or view.

Drop the table *DEPARTMENT*.

```
DROP TABLE department
```

5.3 Data Control Language (DCL SQL)

These SQL statements control the security and permissions of the objects or parts of the database(s). DCL SQL statements are also more part of the DBMS and have large differences between the SQL variations. DCL SQL commands include the following:

5.3.1 GRANT - to allow specified users to perform specified tasks.

Grant the appropriate privileges on the *CALENDAR* table so that users *PHIL* and *CLAIRE* can read it and insert new entries into it. Do not allow them to change or remove any existing entries.

```
GRANT SELECT, INSERT ON calendar  
TO USER phil, USER claire
```

5.3.2 REVOKE - to cancel previously granted or denied permissions.

Revoke all privileges on table *EMPLOYEE* from users *PELLOW* and *MIKE* and from group *PLANNERS*.

```
REVOKE ALL  
ON employee  
FROM USER pellow, USER mike, GROUP planners
```

5.4 Executing Queries in DB2 Command Center

As we mentioned before in Section 4.1, pay special attention to the following settings in *Command Center*:

- Always check your *Statement Termination Character*. If selected, your query always has to end with the specified character. If not selected, every line is considered a query in *Script mode* but only the first line will be processed in *Interactive mode*. You find the *Statement Termination Character* under the menu-bar path: *Tools » Tool Settings* found in *Command Center*. Once in the *Tool Settings* window, the option is found under the *General* tab.
- In some cases (often when creating triggers or stored procedures) it can be necessary to use a termination character other than ";" since the same character may appear inside the trigger's or stored procedure's body. You can for example use the "@" symbol.