INTRODUCTION TO
IBM DB2 v.8.2 fp9a

For Microsoft Windows XP Professional
Revised January 2006

Table of Contents

1 Introduction

DB2 Universal Database™ version 8.2 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 version 8.2, please visit the IBM website: http://www-306.ibm.com/software/data/db2/udb/support/manualsv8.html

1.1 Remarks about this tutorial and last minute changes

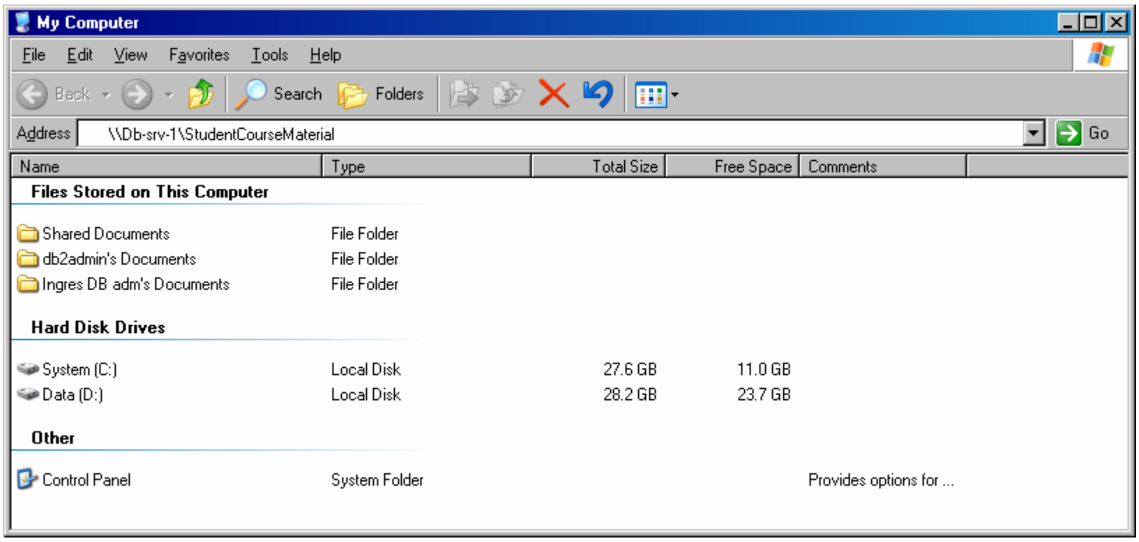
This tutorial's purpose is to make DB2 as easy, fun and comprehensible as possible. With this in mind we also know that this introduction is far from complete and that it only targets parts of the whole DBMS environment. Through experience we know that errors are going to be introduced by this text, 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 of different courses may require that you have access to various kinds of files. All the required files are located on a file server called DB-SRV-1.

Please note that the DB2 introduction itself does not require any additional files!

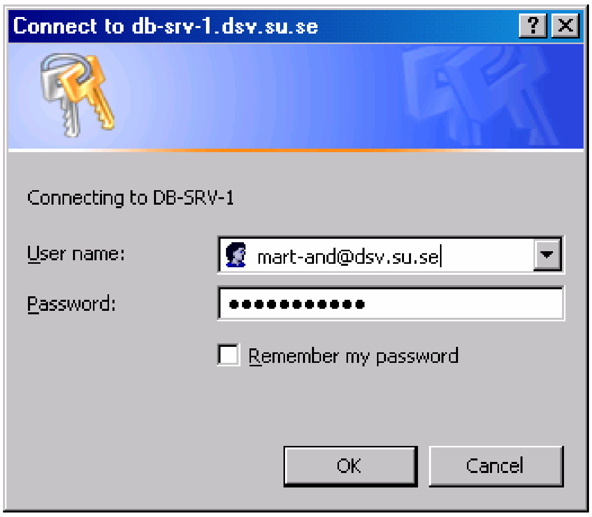
To access the file server DB-SRV-1 (from within DSV's network), open *My Computer* (or *Windows Explorer*) and type in: **`\\Db-srv-1\StudentCourseMaterial`** in the *Address* field, as shown in the figure below:



Address: \\Db-srv-1\StudentCourseMaterial

Name	Type	Total Size	Free Space	Comments
Files Stored on This Computer				
Shared Documents	File Folder			
db2admin's Documents	File Folder			
Ingres DB adm's Documents	File Folder			
Hard Disk Drives				
System (C:)	Local Disk	27.6 GB	11.0 GB	
Data (D:)	Local Disk	28.2 GB	23.7 GB	
Other				
Control Panel	System Folder			Provides options for ...

When you try to access the server for the first time, 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:



Connect to db-srv-1.dsv.su.se

Connecting to DB-SRV-1

User name: mart-and@dsv.su.se

Password: ●●●●●●●●●●●●

□ Remember my password

Click OK to connect to the server. When you connect to *\\Db-srv-1\StudentCourseMaterial*, open the folder ***with your course code and the current term***, e.g. *x62 ht2005* or *IS4 vt2006* (*ht* stands for the autumn term, and *vt* for the spring term). Now you are ready to copy files required for your course assignments.

You can also access the course files on the web by entering: **`http://db-srv-1.dsv.su.se/`** in the *Address* field of your web browser. This can be useful when you are trying to access the server from outside DSV's network.

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 as illustrated below.

for example:	`c:\>`**`DB2 CONNECT TO myDatabase`**
is treated the same as:	`m:\>`**`db2 connect to MYDATABASE`**
and:	`d:\>`**`dB2 cOnNeCt tO mYdAtAbAsE`**)

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

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.

2.1 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 with 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, but 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. 8's user interfaces and that is something that is definitely changing each and every year.

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. 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 Editor,* do this from within the *DB2 Control Center* window (by selecting *Command Editor* from the *Tools* menu or by clicking on the *Command Editor* icon in the toolbar). 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).

2.1.2 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.

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 the GUIs and are as a result often faster and not as likely to cause errors.

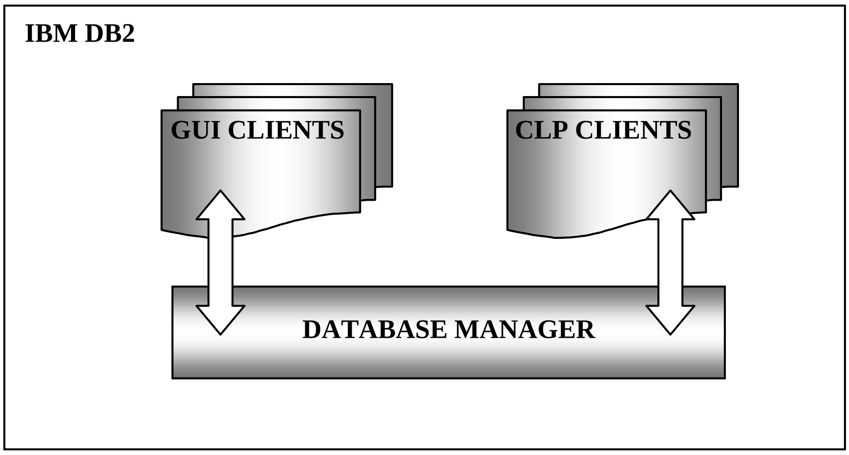


Figure 1 The DBMS works as a foundation for the interface clients

The different GUIs and CLPs are presented and explained later in this introduction.

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 chapter's brief introduction:

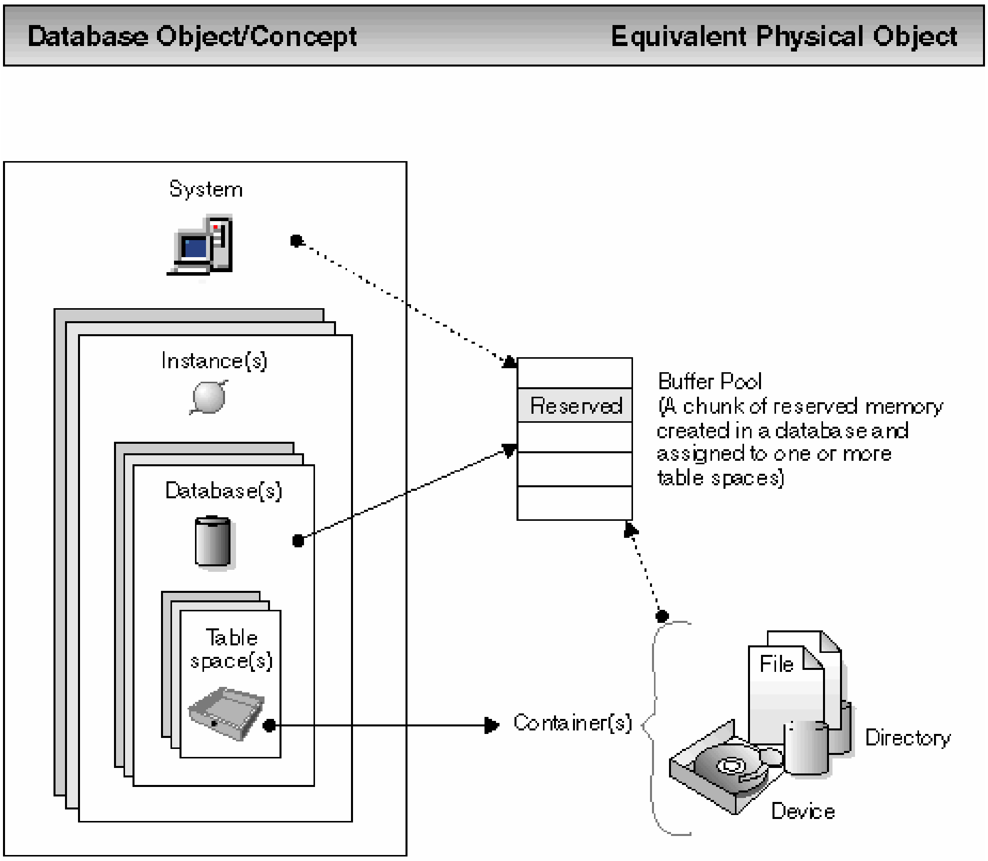

Database Object/Concept
Equivalent Physical Object
System
Instance(s)
Database(s)
Table space(s)
Reserved
Buffer Pool (A chunk of reserved memory created in a database and assigned to one or more table spaces)
Container(s)
File
Directory
Device

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

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.

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 the 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, which is configured and started in chapter 3 - *Getting started with IBM DB2.*

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. Since most database courses only cover single database access this is often the top container you will encounter. Use the database metaphor to manage your tables and other database objects, as well as optimize performance through table spaces and buffer pools.

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 hold 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, which can be performed 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 data that is related. Along with the database, tables are probably the most common database objects 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 a 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 systems, instances, databases, tables, and table spaces, refer to DB2 Information Center, chapters “Designing/Database systems” and “Administering/ Database systems” and

3 Getting started with IBM DB2

Basically, there are two major ways to issue commands in DB2. You can either use the text-based clients *DB2 Command Window* and *DB2 Command Line Processor* or you can take advantage of the graphical user interfaces like *DB2 Control Center* and *DB2 Command Editor*. In this chapter, we recommend you to use the *DB2 Command Window* to type in the commands. The *Command Window* can be opened from the start-menu: ***Start » Programs » Databases » IBM DB2 » Command Line Tools » Command Window*** (adjust the size of the window if it is too large for your screen resolution)**.** More about the different DB2 tools will be explained in the next chapter.

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 into 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 » Databases » IBM DB2 » netStartDB2***

👍 A good way to actually see what happens when issuing the start commands is to open a *DB2 Command Window* 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`** in the start-menu. 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. Sometimes DB2 may 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 terminate 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 encounter any kind of problem. To manually stop the DBMS, do the following:

⌨ Stop all services by following the start-menu path: ***Start » Programs » Databases » IBM DB2 » netStopDB2***

- To see what happens when issuing the stop command, open a *DB2 Command Window* and type in: **`c:\myprog\netstopdb2.bat`**
- As with the *netStartDB2* command, speed up access by using the *Run* command **`c:\myprog\netstopdb2.bat`** in the start-menu.
- If nothing works, not even the stop command, you might want to issue the command: **`DB2 FORCE APPLICATION ALL`** in 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 granted 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 **`DB2 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 Command Window* instead.
- 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*, or in discussion forums and the Internet.

4 DB2 Tools

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

1. The *Command Editor* 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 Editor* or the *Command Window*, because by using them 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 graphical interface of the *Control Center* hides from you.

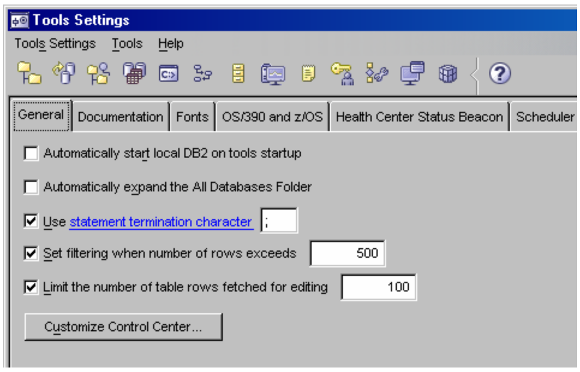
4.1 The Command Editor

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

4.1.1 Setting up the Command Editor

Before we begin to use the *Command Editor*, we should examine some of its settings. Open *DB2 Command Editor* by using the start-menu: ***Start » Programs » Databases » IBM DB2 » Command Line Tools » Command Editor***.

Check your *Statement Termination Character*. This character is used to separate one SQL statement from another in the *Command Editor*. If selected, your query has to end with the specified character. If not selected, every line is considered a query. To specify the *Statement Termination Character*, open the *Tool Settings* window (from the *Tools* menu), select the *Use statement termination character* check box on the *General* tab, and type the character that you want to use in the entry field. The default character is the semicolon (;).



□ Automatically start local DB2 on tools startup

□ Automatically expand the All Databases Folder

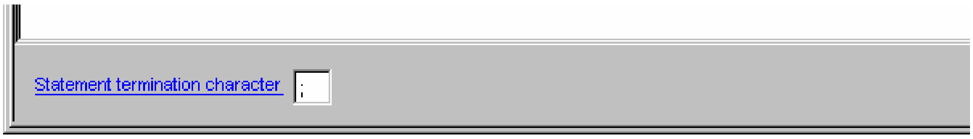
☑ Use statement termination character ;

☑ Set filtering when number of rows exceeds 500

☑ Limit the number of table rows fetched for editing 100

Figure 3 The General tab in the Tool Settings

☠ You can temporarily change the *Statement Termination Character* directly in the *Command Editor* window. Type the character that you want to use in the *Statement termination character* field at the bottom of the *Command Editor* window (see the next figure). Note that the termination character in this field applies only to the current *Command Editor* session and does not affect any other *Command Editor* sessions that might be open! Also, if you want to unselect the *Statement Termination Character*, you must do it in the *Tool Settings* window and then re-open the *Command Editor*!



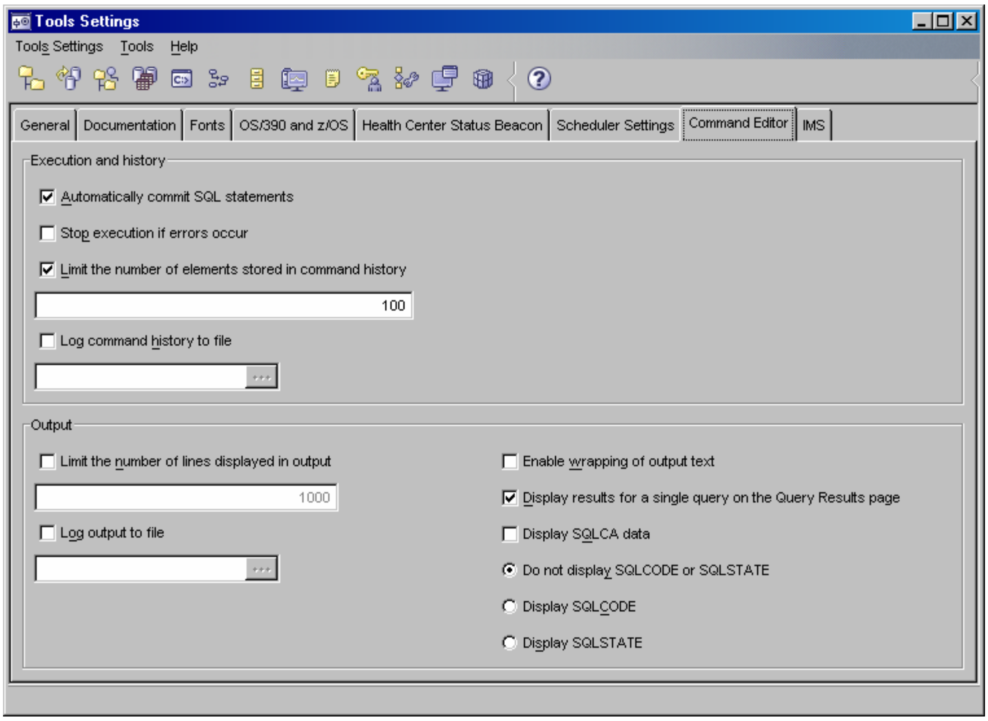
Statement termination character ;

Figure 4 The Statement termination character field in the Command Editor

☠ If you want to split your statement over several lines, the *Statement Termination Character* must be selected, in order to override the new line character!

☠ 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 set up where you want to view the query results. It is easier to save the query results from the *Commands* tab, rather than *Query Results*. Open the *Command Editor* tab in the *Tool Settings* window and unselect the *Display results for a single query on the Query Results page* check box:



Execution and history

☑ Automatically commit SQL statements

□ Stop execution if errors occur

☑ Limit the number of elements stored in command history

100

□ Log command history to file

Output

□ Limit the number of lines displayed in output

1000

□ Log output to file

□ Enable wrapping of output text

☑ Display results for a single query on the Query Results page

□ Display SQLCA data

● Do not display SQLCODE or SQLSTATE

○ Display SQLCODE

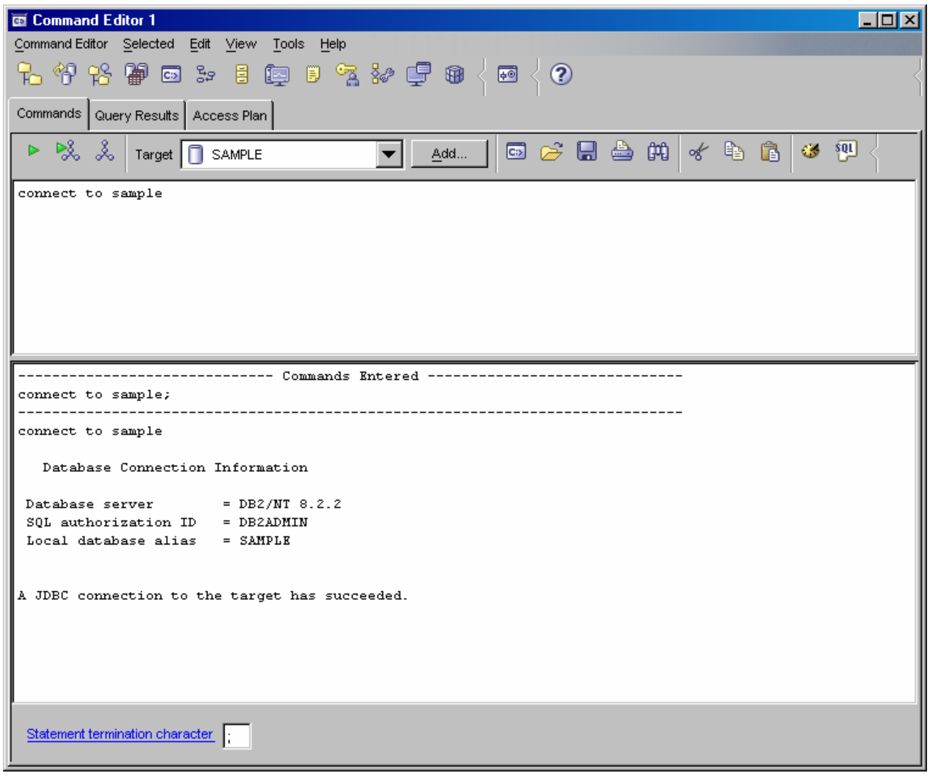
○ Display SQLSTATE

Figure 5 The Command Editor tab in the Tool Settings

4.1.2 Executing queries

Open *DB2 Command Editor* by using the start-menu path: ***Start » Programs » Databases » IBM DB2 » Command Line Tools » Command Editor*** or if you already have a *DB2 Control Center* open, use the *Command Editor* button on the toolbar instead. (Remember that queries could also be executed through a CLP. See section 4.2 for details on that matter).

If you want to execute a query on the database you have to establish a connection to it first. Use the syntax: *CONNECT TO <database-name>*. Database-name stands for the name of the database you want to connect to. Type **`connect to sample`** and press *Ctrl+Enter*. You will now see your connection information displayed in the output area of the *Command Editor* window:



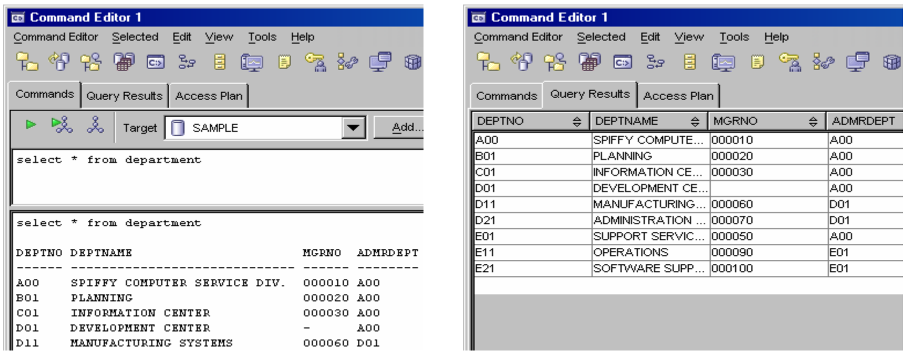
```
connect to sample
```

```
------------------------------ Commands Entered ------------------------------
connect to sample;
------------------------------------------------------------------------------
connect to sample

   Database Connection Information

 Database server        = DB2/NT 8.2.2
 SQL authorization ID   = DB2ADMIN
 Local database alias   = SAMPLE


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

Figure 6 Connection information in the Command Editor

Type, import or paste (*Ctrl+V*) the query you want to execute (for example, **`select * from department`**) and use the *Execute* button ▶ to the left on the toolbar (or press *Ctrl+Enter*).

Depending on your tool settings, you can view the execution results of a single query either in the output area of the *Commands* tab or in the *Query Results* tab:


Command Editor 1
Command Editor Selected Edit View Tools Help
Commands Query Results Access Plan
Target SAMPLE
Add...

```
select * from department
```

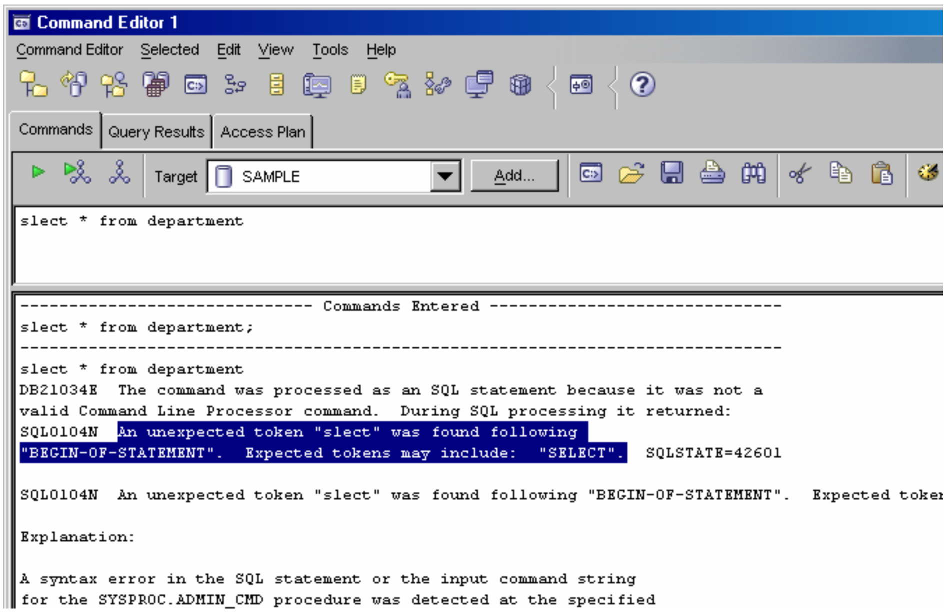
```
select * from department

DEPTNO DEPTNAME                     MGRNO  ADMRDEPT
------ ---------------------------- ------ --------
A00    SPIFFY COMPUTER SERVICE DIV. 000010 A00
B01    PLANNING                     000020 A00
C01    INFORMATION CENTER           000030 A00
D01    DEVELOPMENT CENTER           -      A00
D11    MANUFACTURING SYSTEMS        000060 D01
```

DEPTNO	DEPTNAME	MGRNO	ADMRDEPT
A00	SPIFFY COMPUTE...	000010	A00
B01	PLANNING	000020	A00
C01	INFORMATION CE...	000030	A00
D01	DEVELOPMENT CE...		A00
D11	MANUFACTURING...	000060	D01
D21	ADMINISTRATION ...	000070	D01
E01	SUPPORT SERVIC...	000050	A00
E11	OPERATIONS	000090	E01
E21	SOFTWARE SUPP...	000100	E01

Figure 7 The Commands and Query Results tabs in the Command Editor

If you want to change this behavior, open the *Tool Settings* window (use the *Tool Settings* button to the right on the *Control Editor* toolbar), choose the *Command Editor* tab in the *Tool Settings* window, and select or unselect the *Display results for a single query on the Query Results page* check box.

If your query contains errors, the *Command Editor* will return an appropriate error description. For example, if you try to execute a faulty query: **`slect * from department`**, the *Command Editor* will display an error message which identifies the problem and suggests a solution:

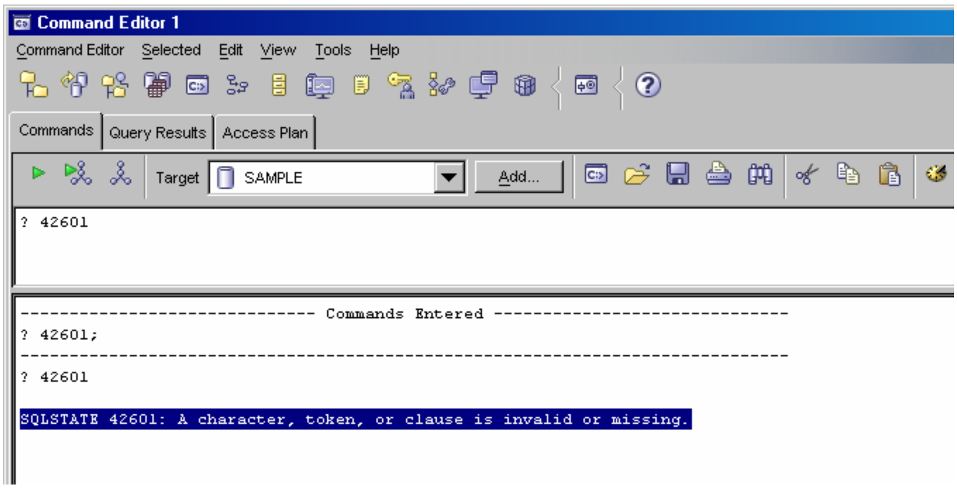

Command Editor 1
Command Editor Selected Edit View Tools Help
Commands Query Results Access Plan
Target SAMPLE
Add...

```
slect * from department
```

```
------------------------------ Commands Entered ------------------------------
slect * from department;
------------------------------------------------------------------------------

slect * from department
DB21034E  The command was processed as an SQL statement because it was not a
valid Command Line Processor command.  During SQL processing it returned:
SQL0104N  An unexpected token "slect" was found following
"BEGIN-OF-STATEMENT".  Expected tokens may include:  "SELECT".  SQLSTATE=42601

SQL0104N  An unexpected token "slect" was found following "BEGIN-OF-STATEMENT".  Expected toke

Explanation:

A syntax error in the SQL statement or the input command string
for the SYSPROC.ADMIN_CMD procedure was detected at the specified
```

Figure 8 An error message in the Command Editor

If the error description is vague, you can search for the meaning of the SQL state code, included in the error description. To display information about different SQL states, use the syntax: *? <SQL state>*. For example, if the SQL state is 42601 (as in the figure above), type: **`? 42601`** to get the definition of this state:



```
? 42601
```

```
------------------------------ Commands Entered ------------------------------
? 42601;
------------------------------------------------------------------------------
? 42601

SQLSTATE 42601: A character, token, or clause is invalid or missing.
```

Figure 9 The definition of an SQL state

You can also use this syntax to get information about other DB2 commands. For example, to find more about the *create database* command, you should enter: **`? create database`**. To get the complete list of DB2 commands, use **`?`** only.

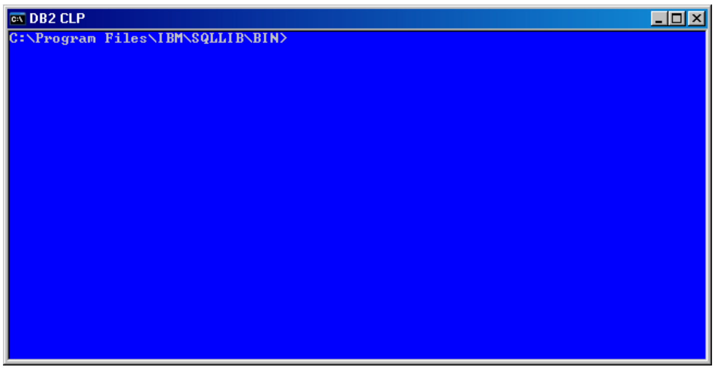
4.2 The Command Line Processor (CLP)

The CLP is used to execute database utilities, SQL statements and online help. It offers a variety of command options. The *DB2 Command Window* and *DB2 Command Line Processor* tools, mentioned earlier, are in fact two different modes of the same application (command mode and interactive input mode, respectively). The main difference between the two modes is that in the *Command Window* each DB2 SQL command should be prefixed by **`DB2`**. The *Command Window* mode can also be used to issue certain DB2 administrative commands. Another difference is that the *Command Line Processor* has the *db2 =>* input prompt.

4.2.1 Starting CLP and changing between the two modes

Open the *DB2 Command Window* by clicking: ***Start » Programs » Databases » IBM DB2 » Command Line Tools » Command Window*** in the start-menu. (You can also open *DB2 Command Window* by using ***Start » Run…*** and typing **`db2cmd`** or by executing the **`db2cmd`** command in a command prompt.)

A window similar to a command prompt opens. With a small difference that it has a blue background and in the top left corner it says *DB2 CLP* and not *Command Prompt*. You are now in the *Command Window* (or command mode):

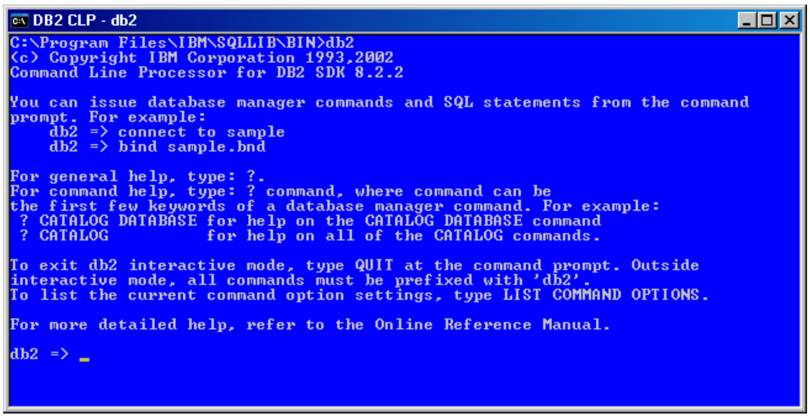


```
C:\Program Files\IBM\SQLLIB\BIN>
```

Figure 10 The DB2 Command Window (command mode)

Next, you can switch to the *DB2 Command Line Processor* (or interactive input mode). Type **`db2`** and press *Enter*. (If you want to open the *DB2 Command Line Processor* from the start-menu, click: ***Start » Programs » Databases » IBM DB2 » Command Line Tools » Command Line Processor***)

The window displays a short message about the *Command Line Processor*, and the prompt changes to *db2=>*. You are now in the *Command Line Processor*:



```
C:\Program Files\IBM\SQLLIB\BIN>db2
(c) Copyright IBM Corporation 1993,2002
Command Line Processor for DB2 SDK 8.2.2

You can issue database manager commands and SQL statements from the command
prompt. For example:
    db2 => connect to sample
    db2 => bind sample.bnd

For general help, type: ?.
For command help, type: ? command, where command can be
the first few keywords of a database manager command. For example:
 ? CATALOG DATABASE for help on the CATALOG DATABASE command
 ? CATALOG          for help on all of the CATALOG commands.

To exit db2 interactive mode, type QUIT at the command prompt. Outside
interactive mode, all commands must be prefixed with 'db2'.
To list the current command option settings, type LIST COMMAND OPTIONS.

For more detailed help, refer to the Online Reference Manual.

db2 => _
```

Figure 11 The DB2 Command Line Processor (interactive input mode)

The two modes are very alike with one major difference: you don't need to specify the **`DB2`** prefix in the interactive mode. This means, to write: **`SELECT * FROM department`** in the interactive mode is exactly the same as writing: **`DB2 SELECT * FROM department`** in the command mode.

To switch back to the command mode *(Command Window)* from the interactive mode, just type **`quit`** and the prompt goes back to normal.

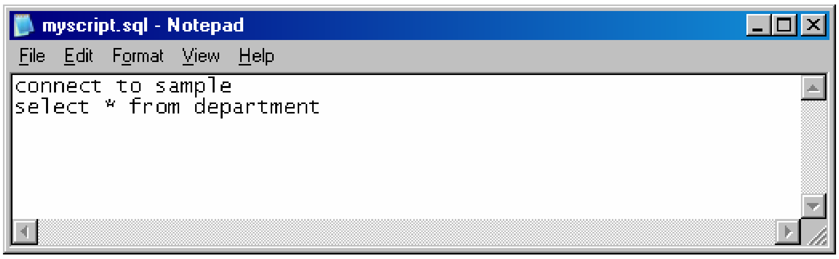
If you want to copy text from the *Command Window* or *Command Line Processor* window, mark the text area which you want to copy and press *Enter*. The copied text can be reinserted by right-clicking on any CLP window or through *Ctrl+V* command in most other Window applications.

The *Information Center* can be started from the *Command Line Processor* with the **`help`** Command.

4.2.2 CLP in a batch mode

You can execute several commands at once in *DB2 Command Window* by using the syntax: *DB2 –f <filename>* where *<filename>* is the path and the name of the text file containing the SQL script. It is called to run the *Command Line Processor* in the batch mode. In the example below, we write a short SQL script and execute it in the *Command Window.*

First, we have to create a simple SQL script. Open the *Notepad* or any other text editor of your choice. Type in two SQL statements: **`connect to sample`** and **`select * from department`**. Each statement should be on a separate line:

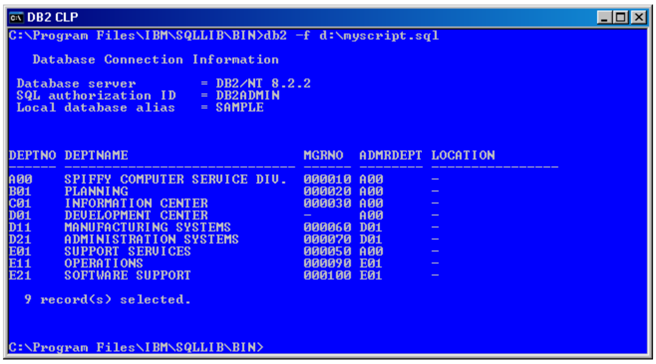


```
connect to sample
select * from department
```

Figure 12 Creating an SQL script in the Notepad

Save the file as **`myscript.sql`** on *D:*

Open *DB2 Command Window*. Type: **`db2 -f d:\myscript.sql`** and press *Enter*. The Command Window will execute the SQL script:



```
C:\Program Files\IBM\SQLLIB\BIN>db2 -f d:\myscript.sql

   Database Connection Information

 Database server        = DB2/NT 8.2.2
 SQL authorization ID   = DB2ADMIN
 Local database alias   = SAMPLE


DEPTNO DEPTNAME                      MGRNO  ADMRDEPT LOCATION
------ ----------------------------- ------ -------- ----------------
A00    SPIFFY COMPUTER SERVICE DIV.  000010 A00      -
B01    PLANNING                      000020 A00      -
C01    INFORMATION CENTER            000030 A00      -
D01    DEVELOPMENT CENTER            -      A00      -
D11    MANUFACTURING SYSTEMS         000060 D01      -
D21    ADMINISTRATION SYSTEMS        000070 D01      -
E01    SUPPORT SERVICES              000050 A00      -
E11    OPERATIONS                    000090 E01      -
E21    SOFTWARE SUPPORT              000100 E01      -

  9 record(s) selected.


C:\Program Files\IBM\SQLLIB\BIN>
```

Figure 13 Executing an SQL script in the DB2 Command Window

4.2.3 Backing up a database

Open a *DB2 Command Window*.

Create a backup directory on D: with the command: **`MD d:\db2backup`**

Disconnect access to the database with the command: **`DB2 DISCONNECT ALL`**

```
DB20000I  The SQL DISCONNECT command completed successfully.
```

Terminate the CLP processes with the command: **`DB2 TERMINATE`**

```
DB20000I  The TERMINATE command completed successfully.
```

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

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

Use the syntax: *DB2 BACKUP DB <database-name> TO <path>* to backup the database. *<Database-name>* is the name of the database you want to backup. For example: **`DB2 BACKUP DB sample TO d:\db2backup`**

`Backup successful. The timestamp for this backup image is : 20050902193425` (stamp is different every time)

4.3 The Control Center

The *DB2 Control Center* is the heart of the DB2 GUI. 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 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, because by using a GUI all the needed SQL commands will be hidden from you. We recommend therefore using the *Show SQL* button that appears in some of the forms (for example, the *Alter Table* window). The *Show SQL* button displays the SQL statements which are sent to the server. Pay also attention to the *DB2* messages which confirm the execution of your commands and include the executed SQL statements.

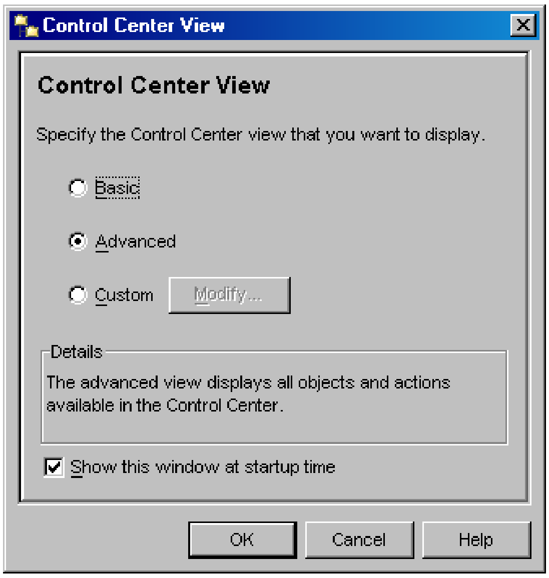
The following sub-sections give a quick introduction into the *DB2 Control Center* and include examples of different tasks that can be performed through this interface:

4.3.1 Open the Control Center

Open *DB2 Control Center* by using the start-menu path: ***Start » Programs » Databases » IBM DB2 » General Administration Tools » Control Center*** or if you already have a DB2 GUI open, use the *Control Center* button on the toolbar.

Another way to open *DB2 Control Center* is to use the green *DB2* button on the *Windows Taskbar* (normally in the right bottom corner of the desktop). Click on this button and choose *DB2 Control Center* from the menu.

When you open the *Control Center* for the first time, it will display the *Control Center View* window:



Control Center View

Control Center View

Specify the Control Center view that you want to display.

○ Basic

● Advanced

○ Custom

Modify...

Details

The advanced view displays all objects and actions available in the Control Center.

☑ Show this window at startup time

OK Cancel Help

Figure 14 The Control Center View window

In this window you can choose between different ways to view the *Control Center*. The *Basic* view provides you with the core *DB2* functions and does not include the database objects that are at a higher level than the database in the *DB2* hierarchy (systems and instances). The *Advanced* view provides you with all the folders, folder objects, and actions available in the *Control Center*. The *Custom* view allows you to tailor the *Control Center* to your needs, i.e. you can select the folders, folder objects, and object actions that you want to appear in your *Control Center* view. We recommend you to use the *Advanced* view (the default option) because it helps you to get more accustomed with the *Control Center*.

If you want to skip this window next time you open the *Control Center*, unselect the *Show this window at startup time* check-box. (You can always return to this window by clicking ***Tools » Customize Control Center*** in the menu bar.)

Press *OK* to confirm the changes and close the *Control Center View* window.

Now you can begin to work on different tasks in the *Control Center*:

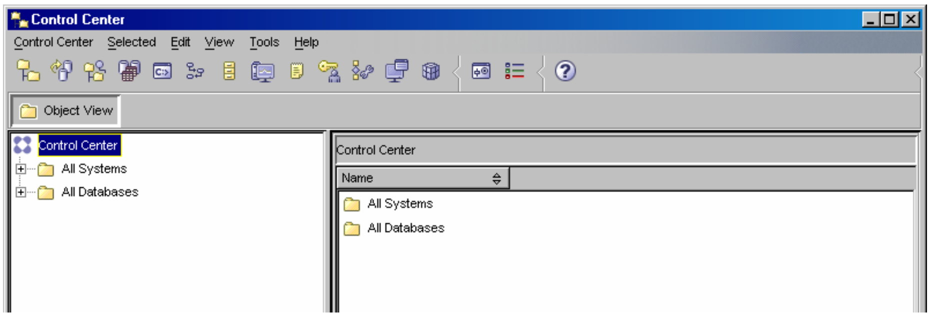
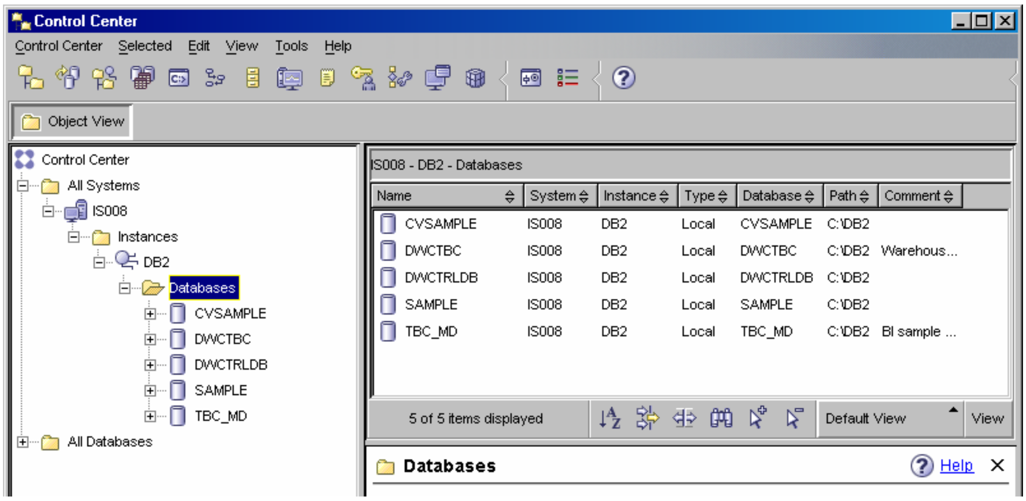


Figure 15 The Control Center (the default view)

4.3.2 Create a new database

Double-click on the *All Systems* icon 📁 to show your local instances and then browse down to the *Databases* folder icon:



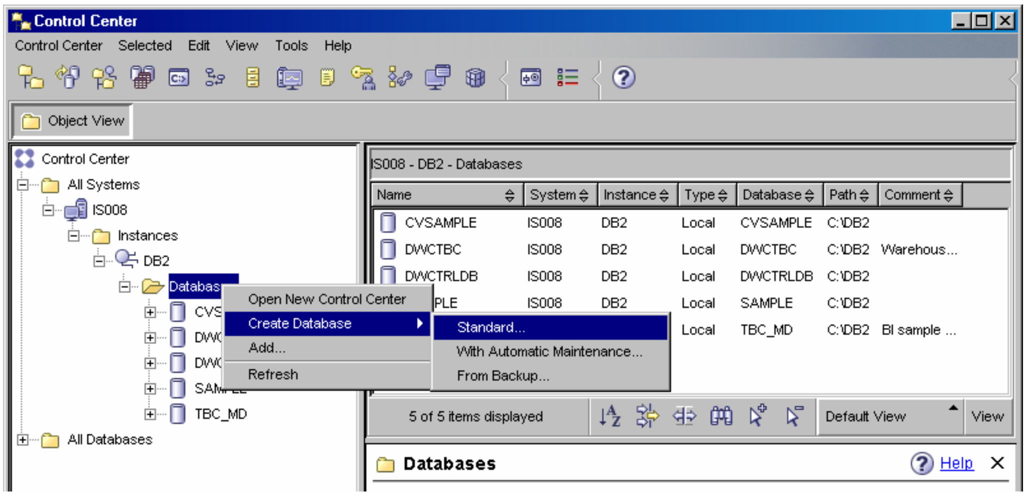
IS008 - DB2 - Databases

Name	System	Instance	Type	Database	Path	Comment
CVSAMPLE	IS008	DB2	Local	CVSAMPLE	C:\DB2	
DWCTBC	IS008	DB2	Local	DWCTBC	C:\DB2	Warehous...
DWCTRLDB	IS008	DB2	Local	DWCTRLDB	C:\DB2	
SAMPLE	IS008	DB2	Local	SAMPLE	C:\DB2	
TBC_MD	IS008	DB2	Local	TBC_MD	C:\DB2	BI sample ...

5 of 5 items displayed

Figure 16 Finding the local databases in the Control Center

Right-click the *Databases* folder icon 📂 and select ***Create Database* » *Standard…*** from the menu:



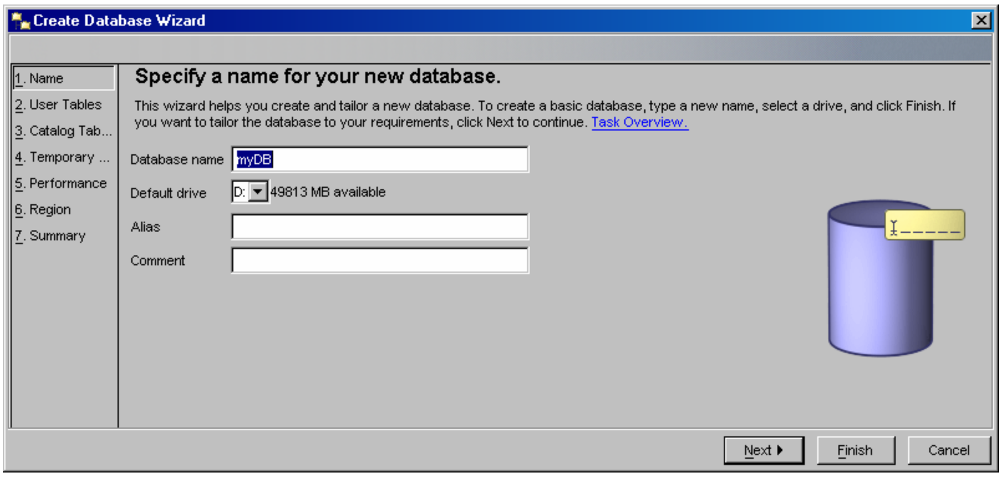
IS008 - DB2 - Databases

Name	System	Instance	Type	Database	Path	Comment
CVSAMPLE	IS008	DB2	Local	CVSAMPLE	C:\DB2	
DWCTBC	IS008	DB2	Local	DWCTBC	C:\DB2	Warehous…
DWCTRLDB	IS008	DB2	Local	DWCTRLDB	C:\DB2	
PLE	IS008	DB2	Local	SAMPLE	C:\DB2	
			Local	TBC_MD	C:\DB2	BI sample …

5 of 5 items displayed

Figure 17 Starting the database wizard

Fill in the database name in the *Database name* field, for example: **`myDB`**, and select *D:* as your default drive:



Specify a name for your new database.

This wizard helps you create and tailor a new database. To create a basic database, type a new name, select a drive, and click Finish. If you want to tailor the database to your requirements, click Next to continue. Task Overview.

Database name: myDB

Default drive: D: 49813 MB available

Alias:

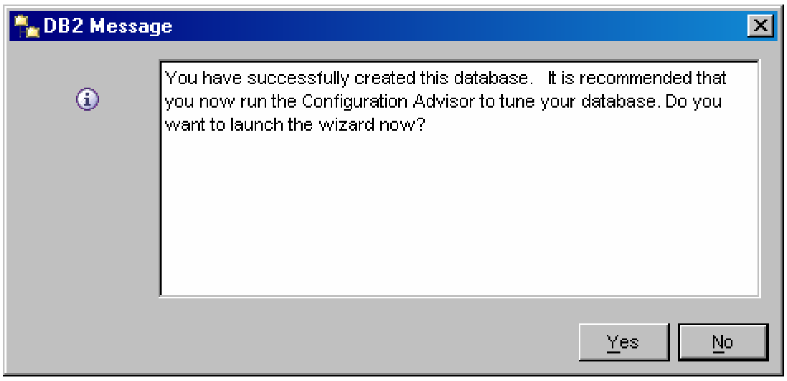
Comment:

Figure 18 The Create Database wizard in the Control Center

The database name should not contain more than eight characters. You should also 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 the main *DB2 Control Center* window.

When the database is successfully created, the *Control Center* will recommend to run the *Configuration Advisor* in order to tune the new database. You may skip this step for now. Click *No* to return to the *Control Center*:



You have successfully created this database. It is recommended that you now run the Configuration Advisor to tune your database. Do you want to launch the wizard now?

Figure 19 The DB2 message about the Configuration Advisor

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

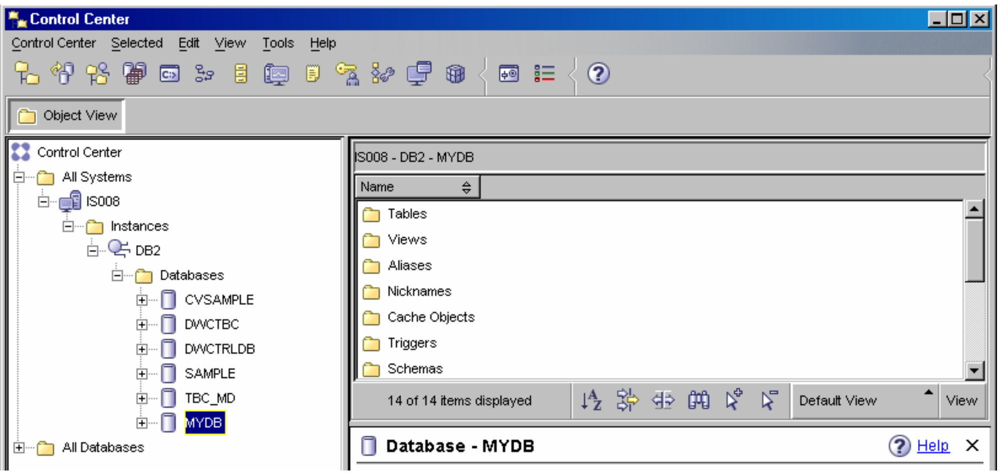
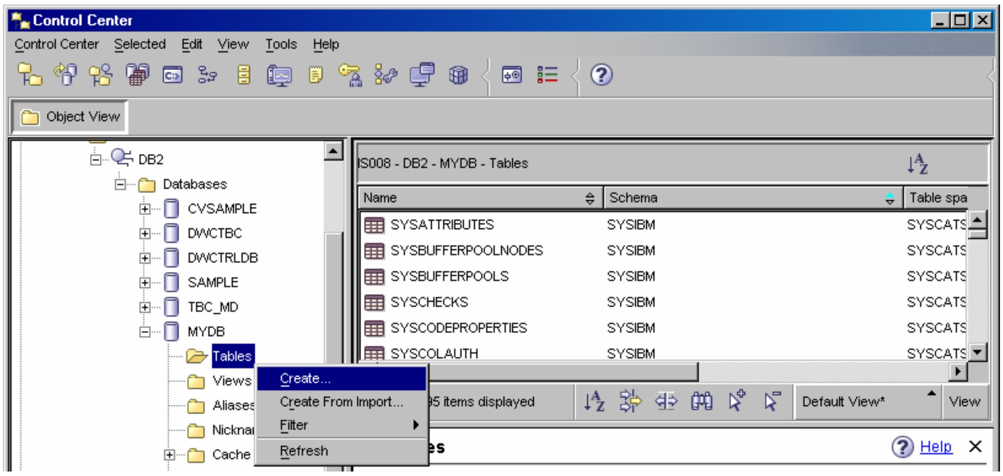


Figure 20 The new database is added to the list of databases

4.3.3 Create a new table

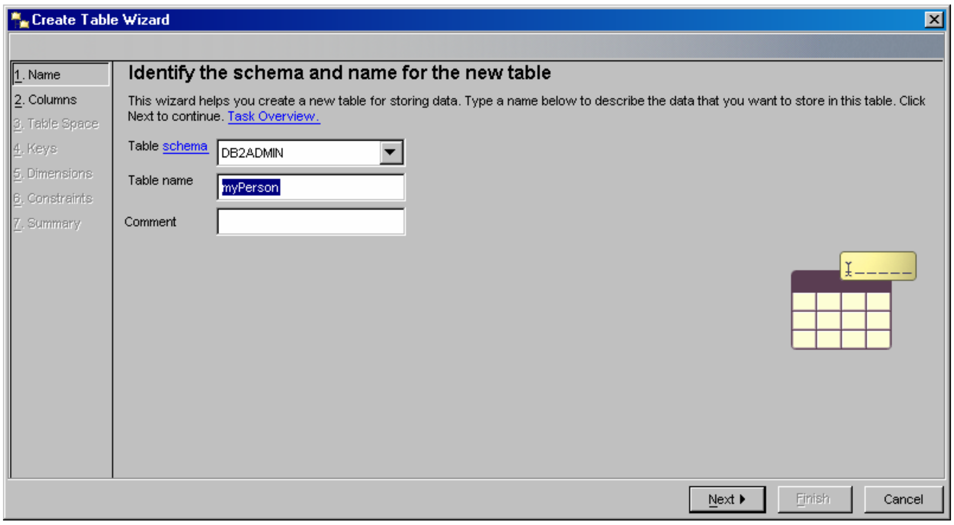
Double-click on your database name in the tree to the left to view its content. From this view you have access to all parts of your database like tables, triggers, indexes and so on. Right-click on the *Tables* folder to bring up the sub-menu and select *Create…*



Name	Schema	Table spa
SYSATTRIBUTES	SYSIBM	SYSCATS
SYSBUFFERPOOLNODES	SYSIBM	SYSCATS
SYSBUFFERPOOLS	SYSIBM	SYSCATS
SYSCHECKS	SYSIBM	SYSCATS
SYSCODEPROPERTIES	SYSIBM	SYSCATS
SYSCOLAUTH	SYSIBM	SYSCATS

Figure 21 Creating a table in the Control Center

In the *Create Table Wizard* window, select *DB2ADMIN* in the *Table schema* menu and fill in the table name (e.g. **`myPerson`**) in the *Table name* field:



Identify the schema and name for the new table

This wizard helps you create a new table for storing data. Type a name below to describe the data that you want to store in this table. Click Next to continue. Task Overview.

Table schema: DB2ADMIN

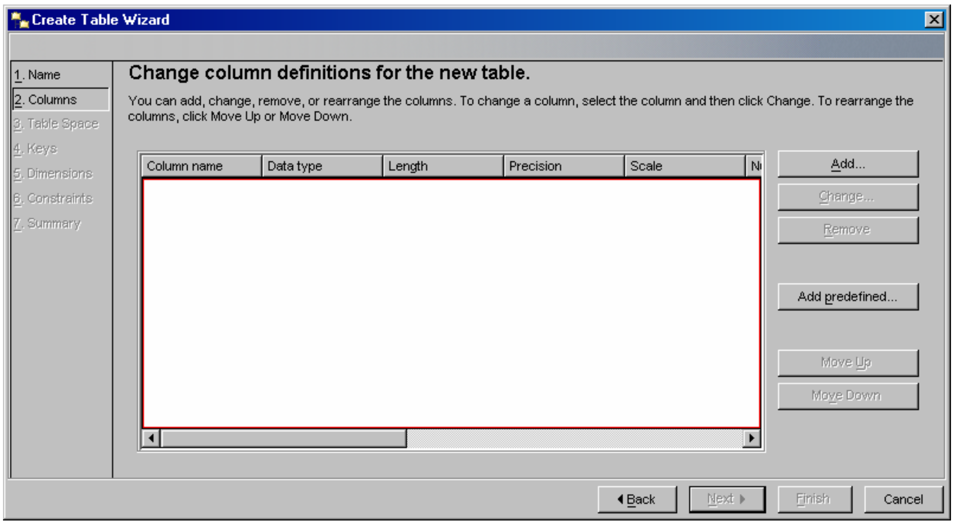
Table name: myPerson

Comment:

Figure 22 The Create Table wizard

As mentioned earlier 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.

Next, select the *Columns* page (click *Columns* to the left):



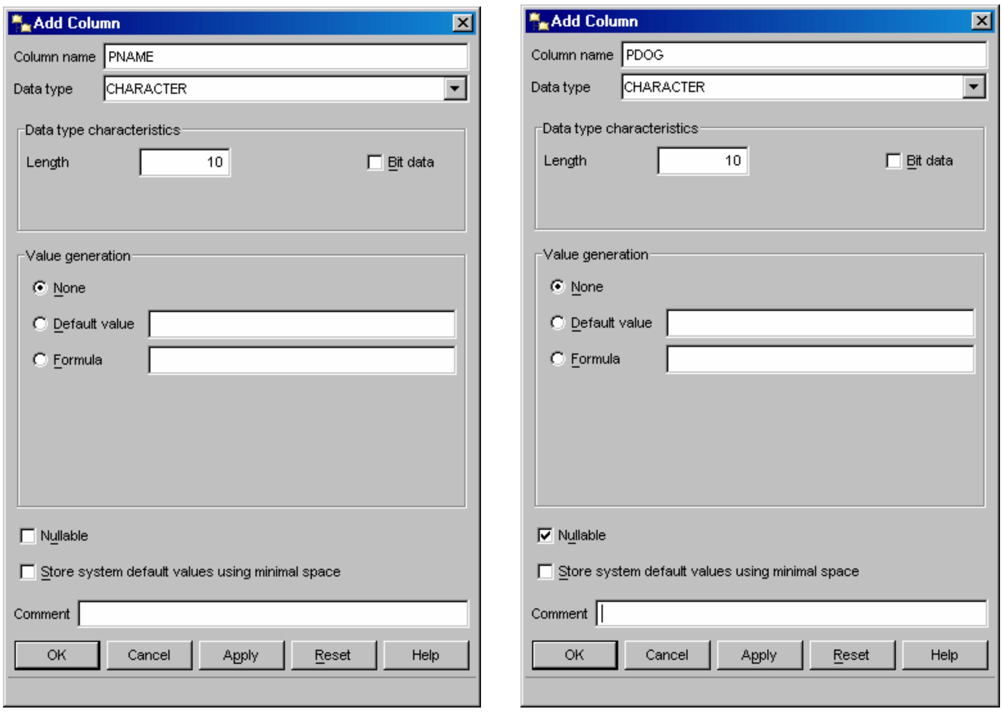
Change column definitions for the new table.

You can add, change, remove, or rearrange the columns. To change a column, select the column and then click Change. To rearrange the columns, click Move Up or Move Down.

Column name	Data type	Length	Precision	Scale	N

Figure 23 The Columns page in the Create Table Wizard

Press the *Add…* button to add a new column. In the *Add Column* window (see the next figure), type in the name of the new column that you want to insert into your table and specify the details of the column: the data type, its characteristics, and if you want it to be 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.

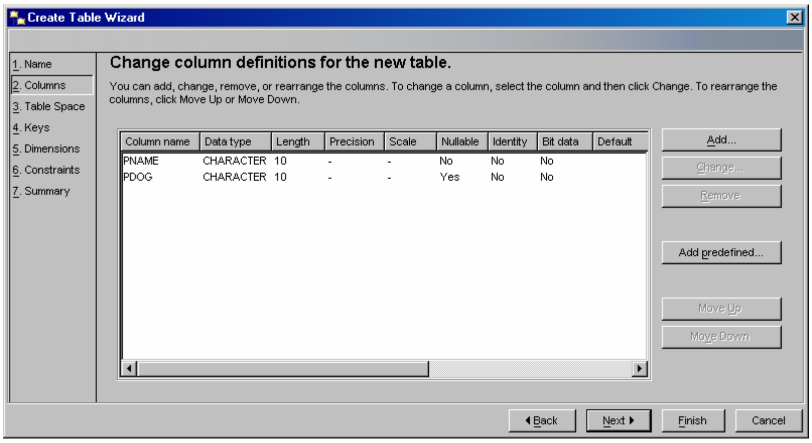


Column name: PNAME
Data type: CHARACTER
Length: 10
□ Bit data
● None
○ Default value
○ Formula
□ Nullable
□ Store system default values using minimal space
Comment:

Column name: PDOG
Data type: CHARACTER
Length: 10
□ Bit data
● None
○ Default value
○ Formula
☑ Nullable
□ Store system default values using minimal space
Comment:

Figure 24 The column details in the Add Column window

In the *Create Table Wizard* window, check that the inserted values correspond to the settings you wanted and then press *Finish* to create your new table:



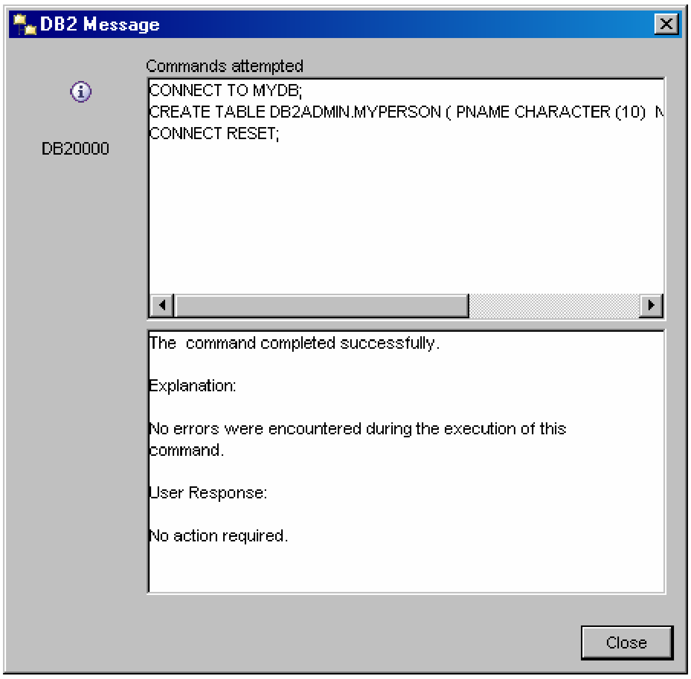
Change column definitions for the new table.

You can add, change, remove, or rearrange the columns. To change a column, select the column and then click Change. To rearrange the columns, click Move Up or Move Down.

Column name	Data type	Length	Precision	Scale	Nullable	Identity	Bit data	Default
PNAME	CHARACTER	10	-	-	No	No	No	
PDOG	CHARACTER	10	-	-	Yes	No	No	

Figure 25 The new columns specified

When the table is successfully created, the *Control Center* will display the DB2 message with the results of executed SQL commands:



```
CONNECT TO MYDB;
CREATE TABLE DB2ADMIN.MYPERSON ( PNAME CHARACTER (10)  N
CONNECT RESET;
```

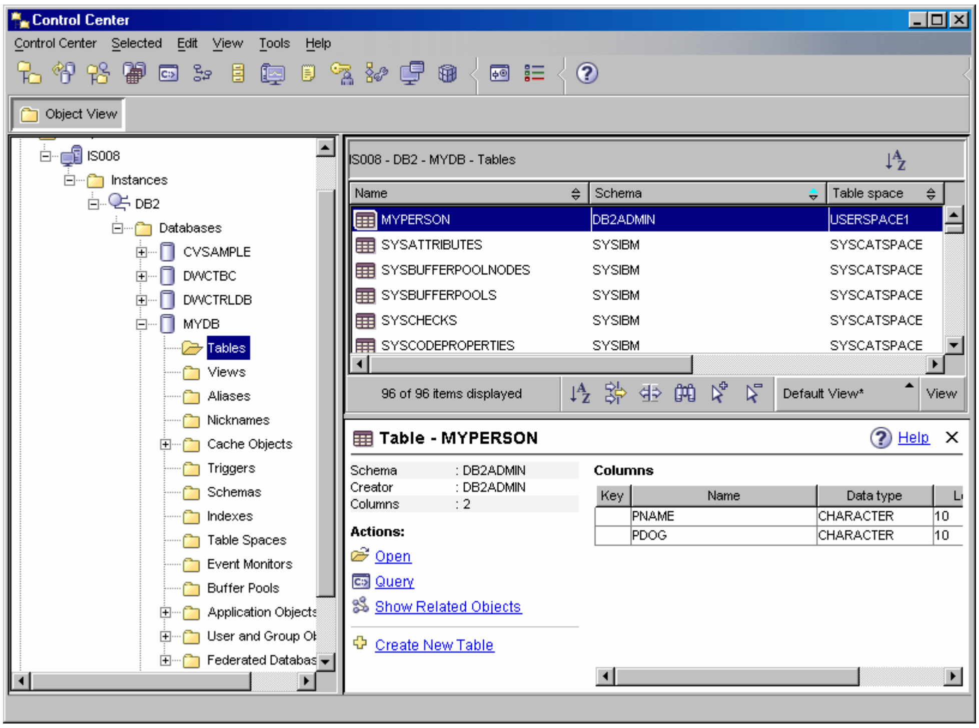
```
The  command completed successfully.

Explanation:

No errors were encountered during the execution of this
command.

User Response:

No action required.
```

Figure 26 The table is successfully created

Read through the SQL statements included in this message. It is a good way to get an understanding of the actions that lie beneath the *Control Center*. As you can see, these are 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.

Press *Close* when you are done.

Note that your table is added to the *Tables* folder. The content of this folder is displayed in the *Contents pane* in the upper right area of the window (see the figure below). In the *Tables* folder there are numerous 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:



Name	Schema	Table space
MYPERSON	DB2ADMIN	USERSPACE1
SYSATTRIBUTES	SYSIBM	SYSCATSPACE
SYSBUFFERPOOLNODES	SYSIBM	SYSCATSPACE
SYSBUFFERPOOLS	SYSIBM	SYSCATSPACE
SYSCHECKS	SYSIBM	SYSCATSPACE
SYSCODEPROPERTIES	SYSIBM	SYSCATSPACE

Table - MYPERSON

Schema : DB2ADMIN
Creator : DB2ADMIN
Columns : 2

Actions:

- Open
- Query
- Show Related Objects
- Create New Table

Columns

Key	Name	Data type	L
	PNAME	CHARACTER	10
	PDOG	CHARACTER	10

Figure 27 The myPerson table is added to the myDB database

Now that you have created your first table, we can look more closely at the *Control Center* window and its components. It has three main areas: *Object tree* (the area to the left), *Contents pane* (the upper right area), and *Object Details pane* (the lower right area). Between the *Contents pane* and the *Object Details pane*, there is the *Contents pane toolbar* .

The *Object tree* displays folders and folder objects. Selecting an item in the *Object tree* displays related objects, actions, and information in the *Contents pane* and the *Object Details pane*. Right-clicking an item displays a pop-up menu listing all the actions that you can perform on that item. In the figure above, the *Object tree* displays all databases, including the *myDB* database.

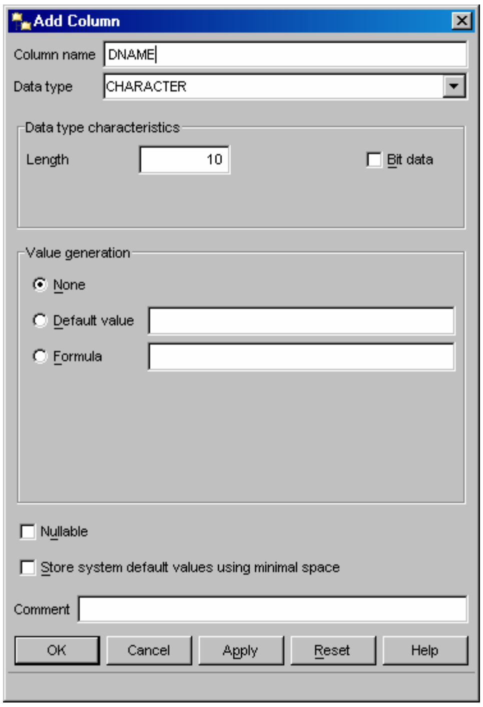
The *Contents pane* displays the contents of the folder or folder object selected in the *Object tree*. Selecting an item in the *Contents pane* displays its associated actions and information in the *Object Details pane*. In the example above, the *Contents pane* displays all tables in the *myDB* database (because the *Tables* folder is selected in the *Object tree*).

The *Object Details pane* displays information on the folder or folder object that you have selected in the *Object tree* or *Contents pane*. In this case, the *Object Details pane* displays information related to the *myPerson* table, since this table is selected in the *Contents pane* above. For instance, it shows the two new columns of this table – *pName* and *pDog*.

For more help on the *DB2 Control Center*, please refer to the *DB2 Information Center* (click on the *Information Center* button ? on the toolbar and select ***Product overviews » Tools for administration and application development » Administering » Control Center*** in the *Table of Contents* to the left).

4.3.4 Setting constraints (primary and foreign keys)

Start with creating a new table as described in the previous section and name this table *myDog*. Select *DB2ADMIN* as the table schema. In the *myDog* table, add a column *dName*, set the data type to *character* with the length *10*, and disallow any null values:



Column name: DNAME

Data type: CHARACTER

Data type characteristics

Length: 10 □ Bit data

Value generation

● None

○ Default value

○ Formula

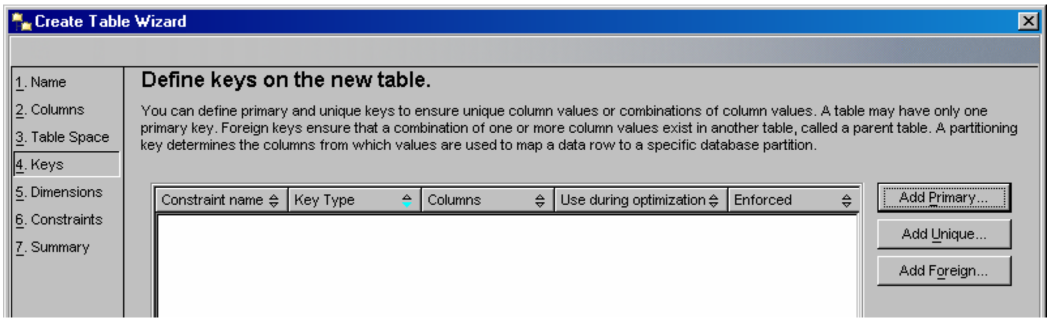
□ Nullable

□ Store system default values using minimal space

Comment:

Figure 28 Specifying the column details

In the *Create Table Wizard* window, select the *Keys* page (click *Keys* in the menu to the left):



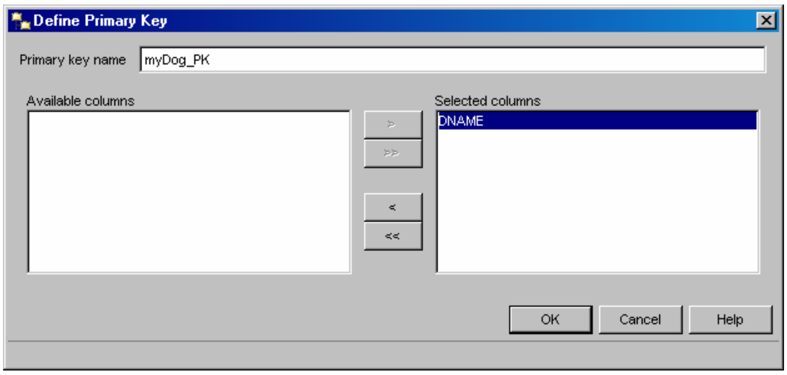
Define keys on the new table.

You can define primary and unique keys to ensure unique column values or combinations of column values. A table may have only one primary key. Foreign keys ensure that a combination of one or more column values exist in another table, called a parent table. A partitioning key determines the columns from which values are used to map a data row to a specific database partition.

Constraint name	Key Type	Columns	Use during optimization	Enforced

Figure 29 The Keys page in the Create Table Wizard

Press the *Add Primary…* button to add a primary key. In the *Define Primary Key* window, select the primary key column *dName* by moving it from the left to the right field. Label the constraint with a meaningful name (optional) and press *OK* to confirm the changes:



Primary key name: myDog_PK

Selected columns: DNAME

Figure 30 Setting the primary key

In the *Create Table Wizard* window, press *Finish* to create the new *myDog* table. When this command is completed, the *Control Center* will display the DB2 message with the executed SQL statements and their results. Compare these queries with your recent commands in the *Create Table Wizard*. Press *Close* to return to the *Control Center*.

From the *DB2 Control Center*, select the *Tables* folder in the *Object tree* to the left and locate the *myPerson* table in the *Contents pane* to the right. Right-click on the table name and select *Alter…*from the menu:

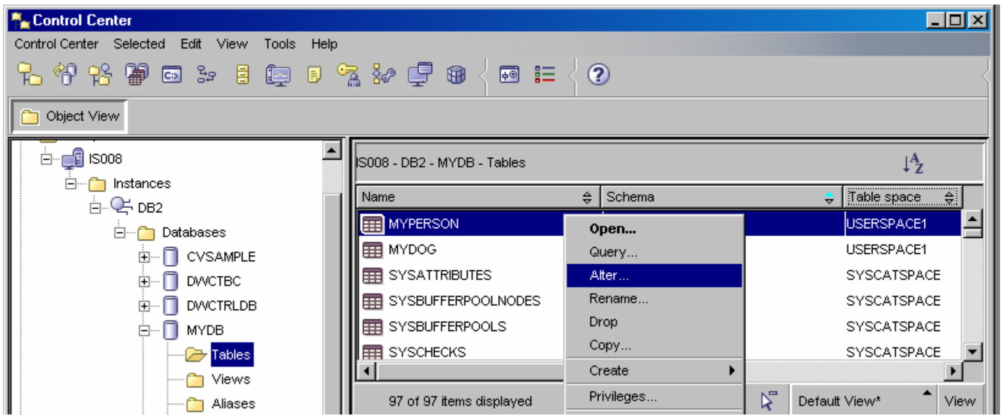


Figure 31 Altering a table in the Control Center

In the *Alter Table* window that opens, select the *Keys* tab:

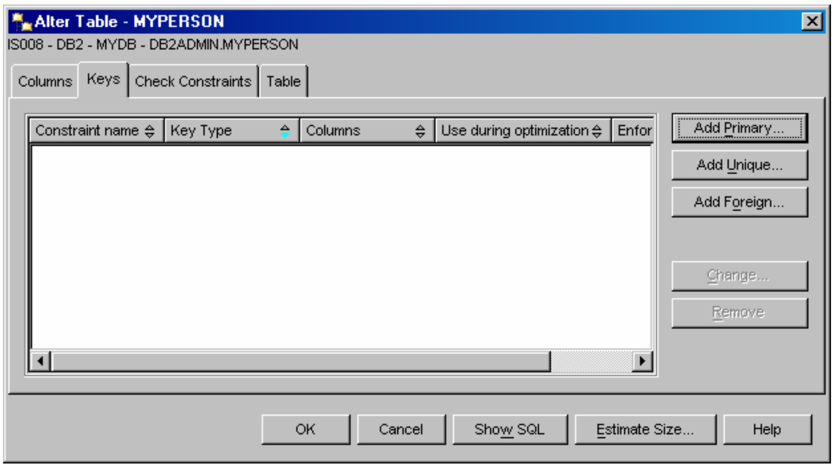
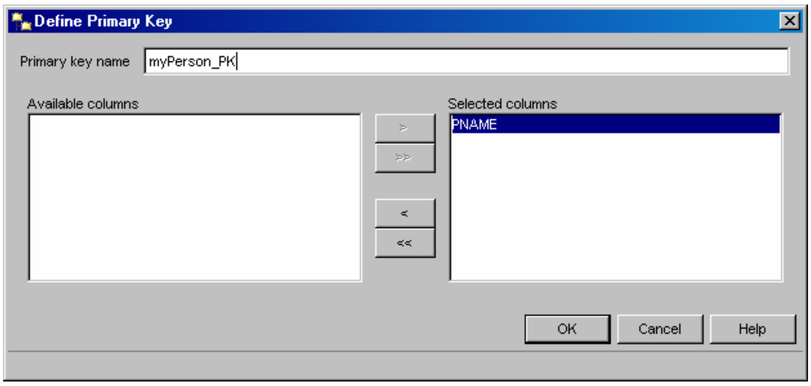


Figure 32 The Keys tab in the Alter Table window

Press the *Add Primary…* button to add a primary key. In the *Define Primary Key* window select the primary key column by moving it over from the left to the right field. Feel free to name the PK 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 name meaningless to you.) In this example we let the column *pName* constitute the primary key of the table *myPerson*. Press *OK* when you are ready:



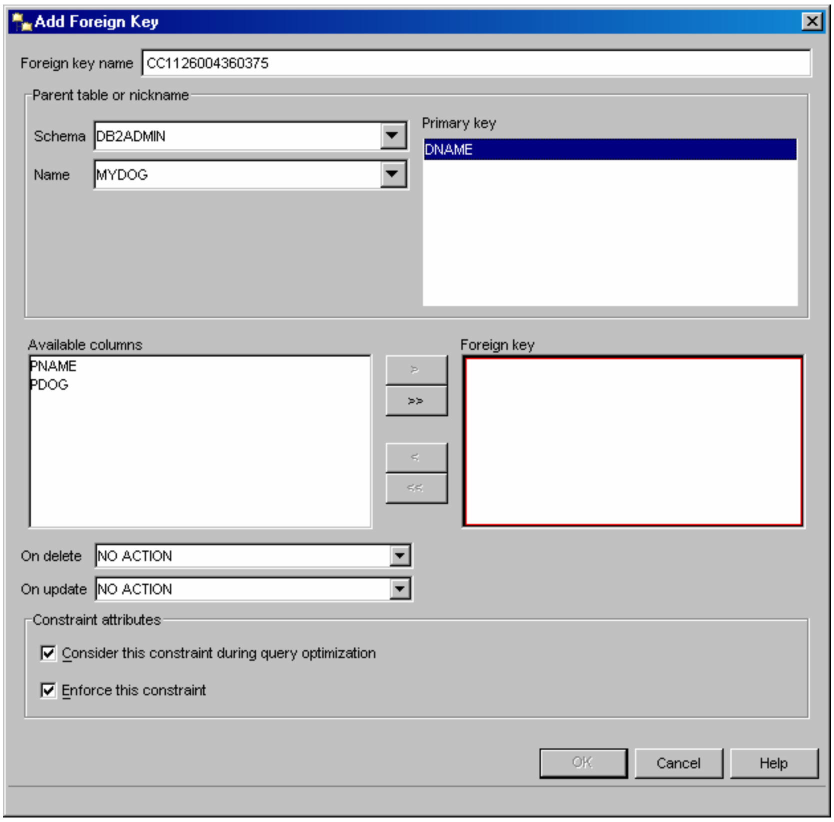
Primary key name: myPerson_PK

Available columns:

Selected columns: PNAME

Figure 33 Selecting the primary key

To create a foreign key, press *Add Foreign…*in the *Keys* tab, which opens 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:



Foreign key name: CC1126004360375

Parent table or nickname

Schema: DB2ADMIN

Name: MYDOG

Primary key: DNAME

Available columns: PNAME
PDOG

Foreign key:

On delete: NO ACTION

On update: NO ACTION

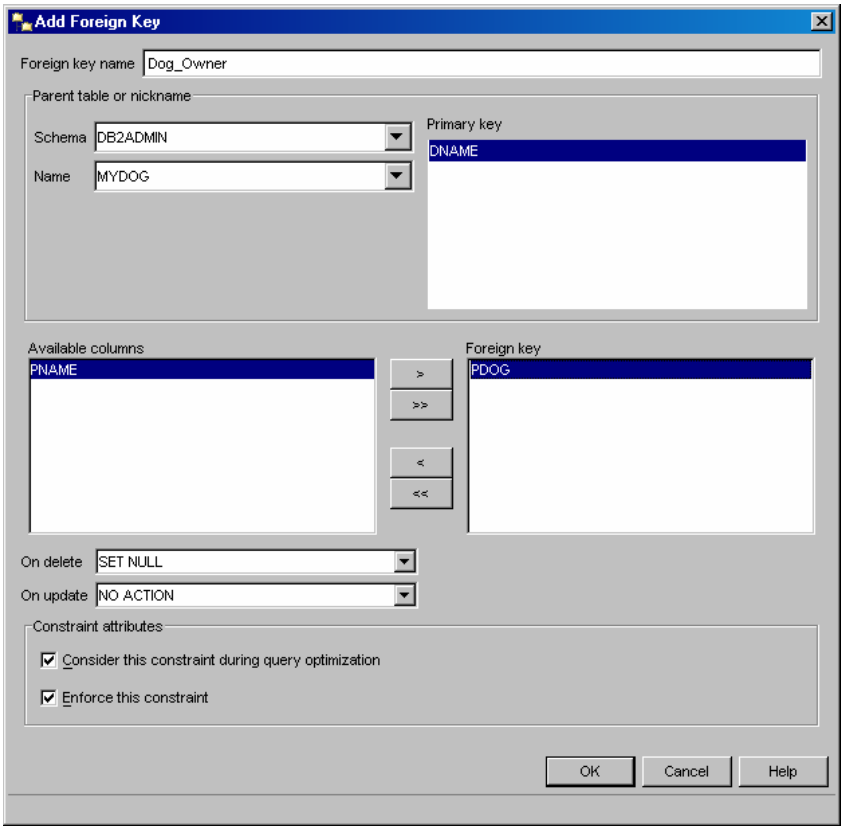
Constraint attributes

☑ Consider this constraint during query optimization

☑ Enforce this constraint

Figure 34 Specifying the determinant of a foreign key

In the lower part, select which column you want to link 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*). Finally, we name the constraint *Dog_Owner*:



Add Foreign Key

Foreign key name: Dog_Owner

Parent table or nickname

Schema: DB2ADMIN

Name: MYDOG

Primary key: DNAME

Available columns: PNAME

Foreign key: PDOG

On delete: SET NULL

On update: NO ACTION

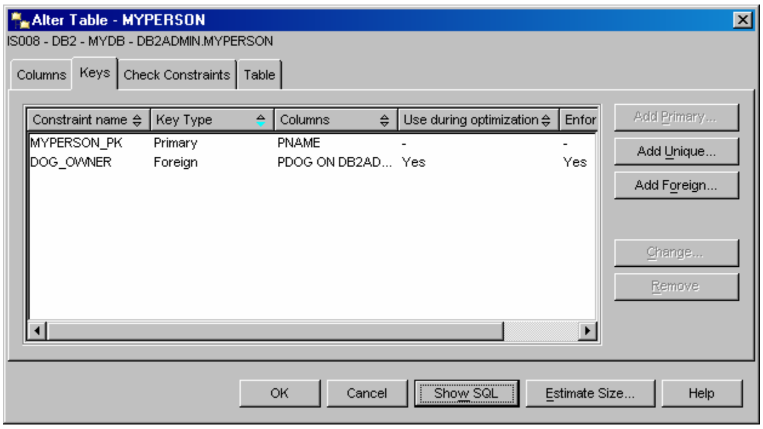
Constraint attributes

☑ Consider this constraint during query optimization

☑ Enforce this constraint

Figure 35 Specifying the dependent field of a foreign key

After everything is set, press *OK* to return to the *Alter Table* window where you can see the two new constraints:



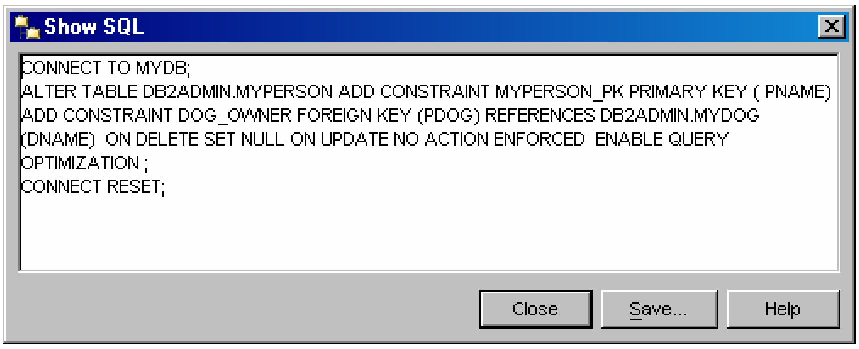
Alter Table - MYPERSON

IS008 - DB2 - MYDB - DB2ADMIN.MYPERSON

Constraint name	Key Type	Columns	Use during optimization	Enfor
MYPERSON_PK	Primary	PNAME	-	-
DOG_OWNER	Foreign	PDOG ON DB2AD...	Yes	Yes

Figure 36 New constraints in the Keys tab

To display the SQL statements equivalent to your commands, press the *Show SQL* button. Examine the *ALTER TABLE* statement and its parameters in the *Show SQL* window, and compare it with your commands. Press *Close* to return to the *Alter Table* window:



```
CONNECT TO MYDB;
ALTER TABLE DB2ADMIN.MYPERSON ADD CONSTRAINT MYPERSON_PK PRIMARY KEY ( PNAME)
ADD CONSTRAINT DOG_OWNER FOREIGN KEY (PDOG) REFERENCES DB2ADMIN.MYDOG
(DNAME)  ON DELETE SET NULL ON UPDATE NO ACTION ENFORCED  ENABLE QUERY
OPTIMIZATION ;
CONNECT RESET;
```

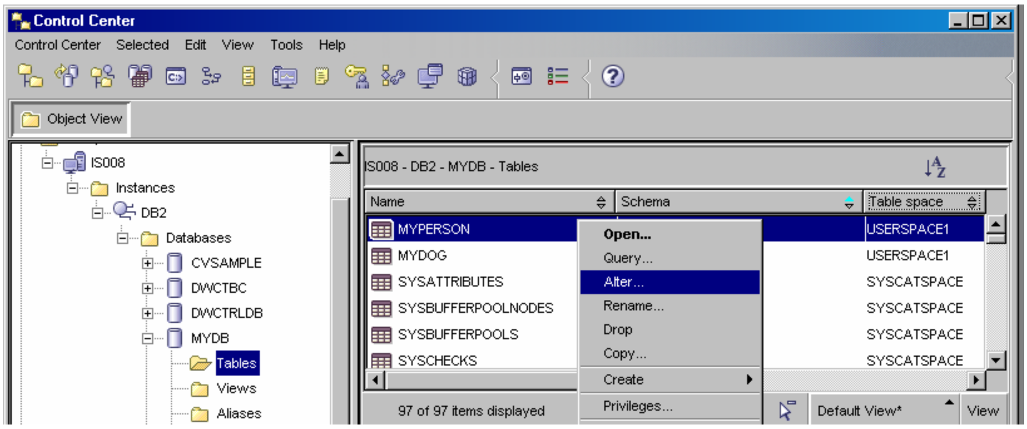
Figure 37 SQL statements in the Show SQL window

In the *Alter Table* window, press *OK* to close the *Alter Table* window and to execute the commands. When the commands are completed, the *Control Center* will display a new DB2 message with the execution results. Close the message and return to the *Control Center*.

4.3.5 Dropping a foreign key

In the following four sections we will look at how to drop database objects from the *Control Center*. We will drop a foreign key, a primary key, a table, and a database.

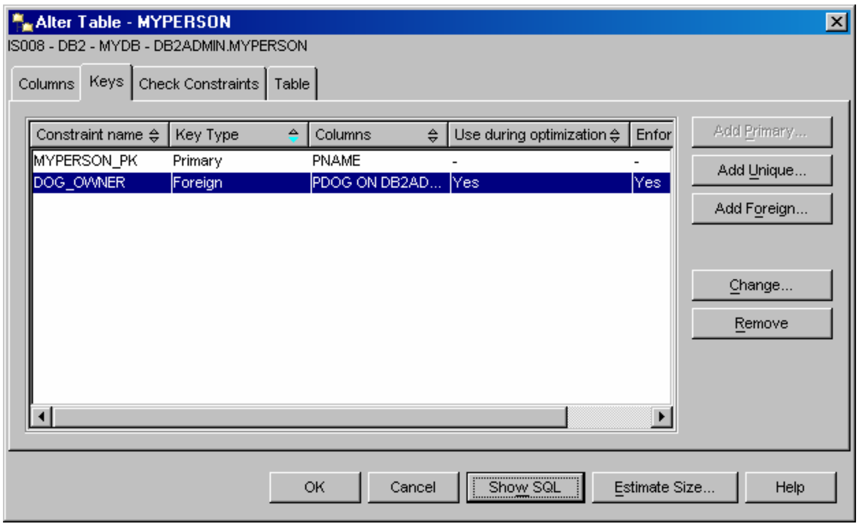
Select the *Tables* folder in the *Object tree* to the left and locate the table which contains a foreign key (*myPerson*) in the *Contents pane* to the right. Right-click on the table name and select *Alter…* from the menu:



Name	Schema	Table space
MYPERSON		USERSPACE1
MYDOG		USERSPACE1
SYSATTRIBUTES		SYSCATSPACE
SYSBUFFERPOOLNODES		SYSCATSPACE
SYSBUFFERPOOLS		SYSCATSPACE
SYSCHECKS		SYSCATSPACE

Figure 38 Altering a table in the Control Center

In the *Alter Table* window, open the *Keys* tab. Select the foreign key constraint you want to remove (*Dog_Owner*) and press *Remove*. The constraint is removed from the list in the *Keys* tab. Press *OK* to commit the drop, i.e. to send the *DROP FOREIGN KEY* command to the server.



Alter Table - MYPERSON

IS008 - DB2 - MYDB - DB2ADMIN.MYPERSON

Constraint name	Key Type	Columns	Use during optimization	Enfor
MYPERSON_PK	Primary	PNAME	-	-
DOG_OWNER	Foreign	PDOG ON DB2AD...	Yes	Yes

Figure 39 Dropping a foreign key in the Control Center

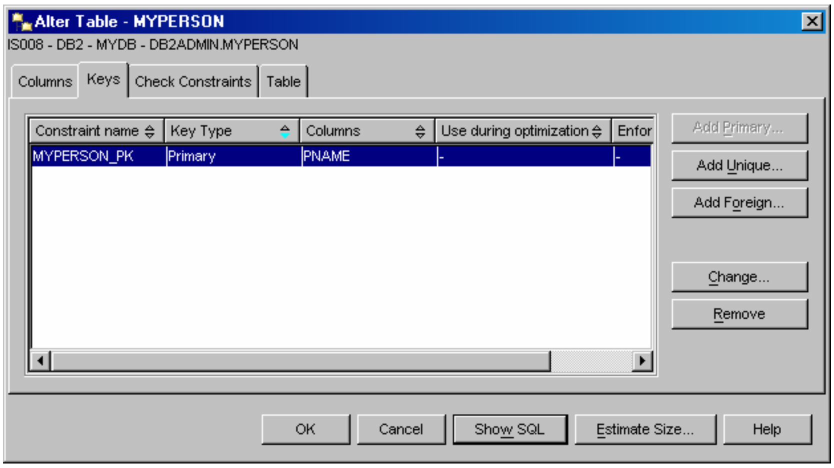
The *Control Center* displays a DB2 message with the executed SQL statements and their results. Review the statements and close the message window to return to the *Control Center*.

Hopefully you have named your constraints in an understandable way so that you could select them easily. If you've forgotten to name your keys, you will have to open each constraint by using the *Change…* command to see which columns it uses.

If you remove the wrong constraint from the *Keys* tab by mistake, you can still undo this command by pressing the *Cancel* button in the *Alter Table* window. However, this command is valid only until you confirm the removal of the constraint, i.e. before you press *OK*.

4.3.6 Dropping a primary key

A primary key can be dropped similarly to a foreign key. Open the *Alter Table* window for the *myPerson* table, choose the *Keys* tab, and select the primary key constraint you want to drop (in this case, *myPerson_PK*):



Alter Table - MYPERSON

IS008 - DB2 - MYDB - DB2ADMIN.MYPERSON

Constraint name	Key Type	Columns	Use during optimization	Enfor
MYPERSON_PK	Primary	PNAME	-	-

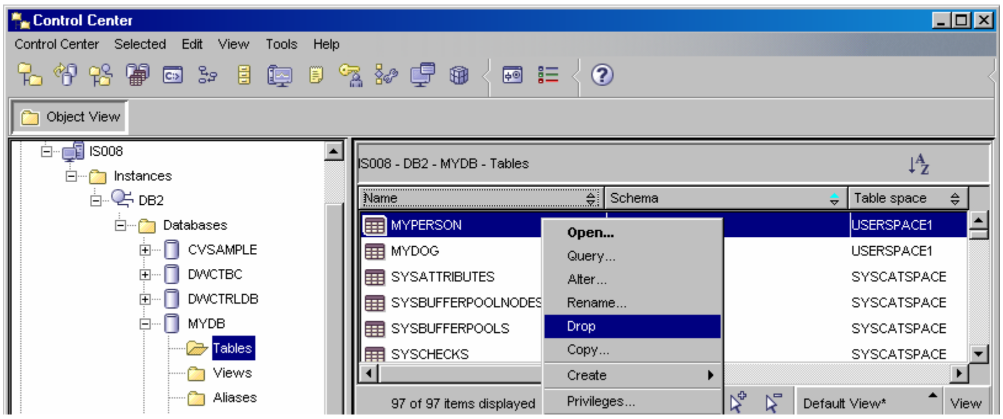
Figure 40 Dropping a primary key in the Control Center

Press the *Remove* button. The constraint is removed from the constraint list in the *Keys* tab. Click *OK* to confirm the drop of the primary key.

The *Control Center* displays the executed SQL statements in the DB2 message. Review the statements and return to the *Control Center* window.

4.3.7 Dropping a table

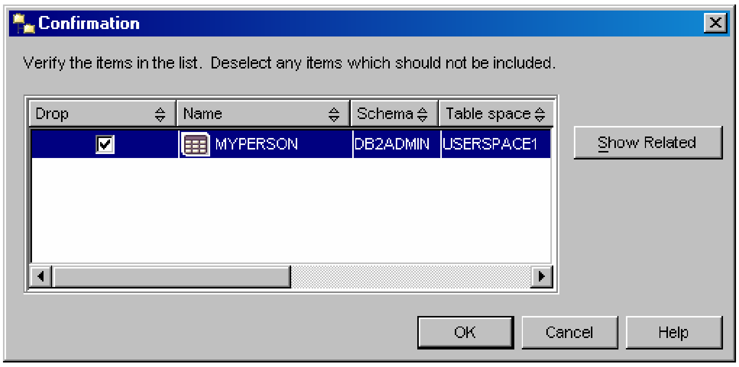
Select the *Tables* folder in the *Object tree* and find the table you want to delete (*myPerson*) in the *Contents pane*. Right-click on the table and select *Drop*:



Name	Schema	Table space
MYPERSON		USERSPACE1
MYDOG		USERSPACE1
SYSATTRIBUTES		SYSCATSPACE
SYSBUFFERPOOLNODES		SYSCATSPACE
SYSBUFFERPOOLS		SYSCATSPACE
SYSCHECKS		SYSCATSPACE

Figure 41 Dropping a table in the Control Center

This opens the *Confirmation* window that asks you to verify what you want to delete. Make sure it is the correct table and press *OK*:



Verify the items in the list. Deselect any items which should not be included.

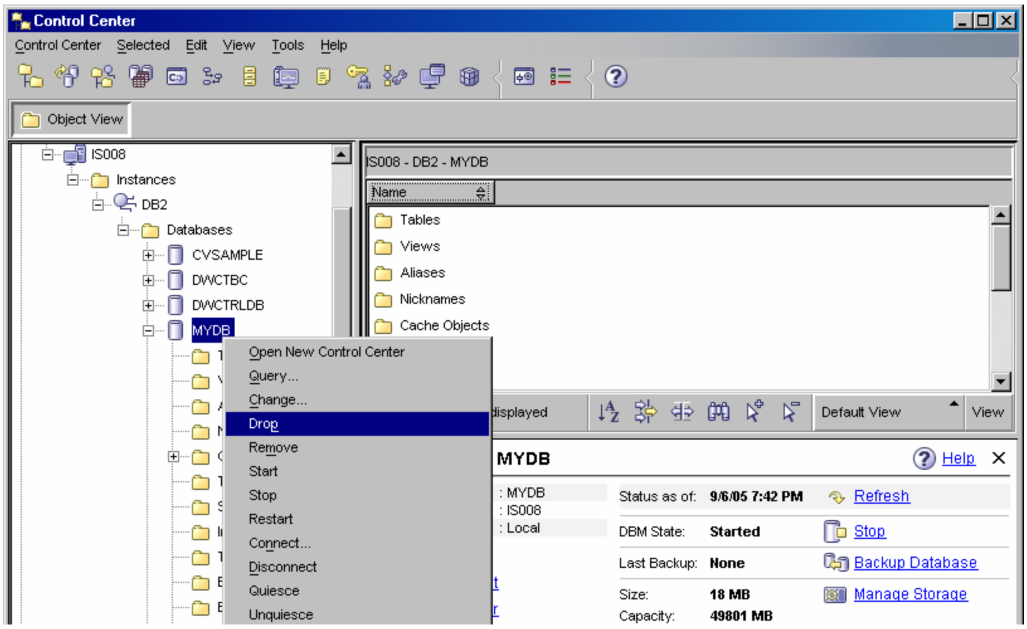
Drop	Name	Schema	Table space
☑	MYPERSON	DB2ADMIN	USERSPACE1

Figure 42 The Confirmation window to delete a table

Note that the *myPerson* table is removed from the table list in the *Contents pane*.

4.3.8 Dropping a database

Right-click on the database you want to delete (in this case, *myDB*) in the *Object tree* and select *Drop* from the menu:

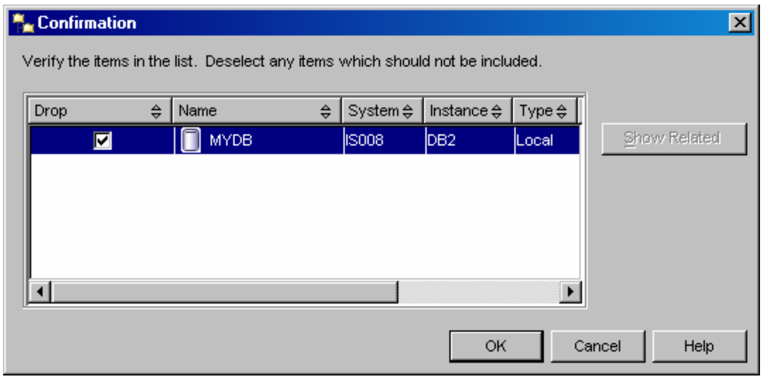


: MYDB
: IS008
: Local

Status as of:	9/6/05 7:42 PM
DBM State:	Started
Last Backup:	None
Size:	18 MB
Capacity:	49801 MB

Figure 43 Dropping a database in Control Center

This opens the *Confirmation* window which asks you to verify your deletion. Make sure it is the correct database and press *OK*:



Verify the items in the list. Deselect any items which should not be included.

Drop	Name	System	Instance	Type
☑	MYDB	IS008	DB2	Local

Figure 44 The Confirmation window to delete a database

Note that the database is removed from the *Databases* folder in the *Object tree*:

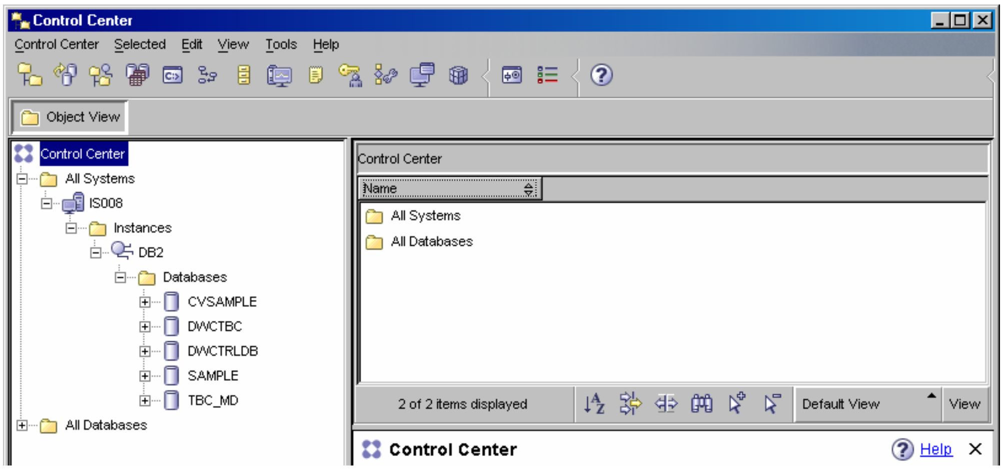


Figure 45 The database myDB is removed from the Databases folder

Do not use the *Remove* command in the pull-down menu since this only removes the database from the *Object tree* view and not from the database manager. Any database removed from the *Control Center* could be retrieved through the *Add…* command.

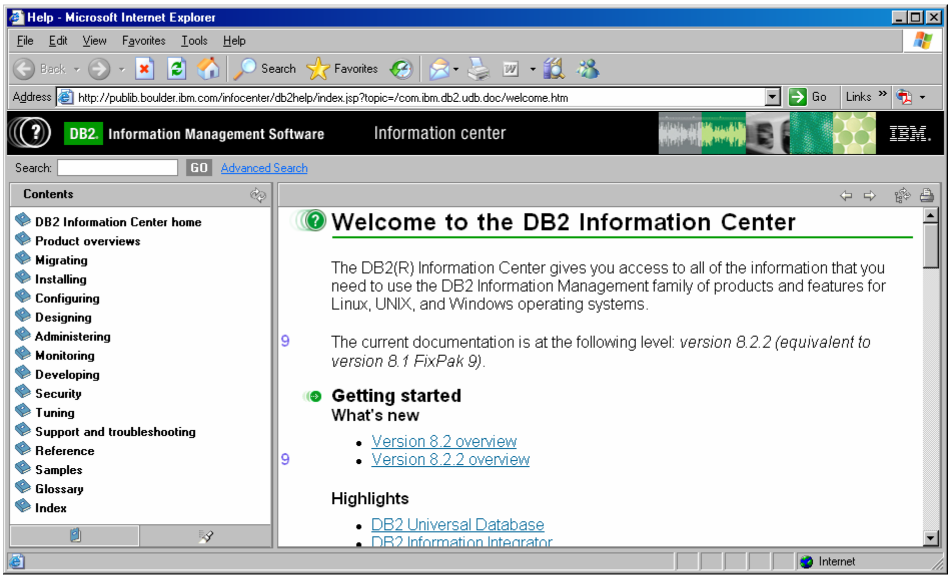
4.4 The Information Center

The *DB2 Information Center* gives you access to all of the information you need to take full advantage of *DB2 Universal Database* and other products of the DB2 family. The DB2 *Information Center* also contains information on major DB2 features and components including replication, data warehousing, and the DB2 extenders.

4.4.1 How to locate topics in the Information Center

Open *DB2 Information Center* by using the start-menu path: ***Start » Programs » Databases » IBM DB2 » Information » Information Center*** or if you have a *DB2 GUI* open, use the *Information Center* button ? on the toolbar. If you are using the *DB2 Command Window*, type in the command **`db2 help`**.

The *DB2 Information Center* will open in your default Web browser:


Help - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://publib.boulder.ibm.com/infocenter/db2help/index.jsp?topic=/com.ibm.db2.udb.doc/welcome.htm
Go Links
DB2. Information Management Software
Information center
IBM.
Search: GO Advanced Search
Contents
Internet

DB2 Information Center home
Product overviews
Migrating
Installing
Configuring
Designing
Administering
Monitoring
Developing
Security
Tuning
Support and troubleshooting
Reference
Samples
Glossary
Index

Welcome to the DB2 Information Center

The DB2(R) Information Center gives you access to all of the information that you need to use the DB2 Information Management family of products and features for Linux, UNIX, and Windows operating systems.

9 The current documentation is at the following level: *version 8.2.2 (equivalent to version 8.1 FixPak 9)*.

Getting started
What's new

- Version 8.2 overview
- 9 Version 8.2.2 overview

Highlights

- DB2 Universal Database
- DB2 Information Integrator

Figure 46 The DB2 Information Center

The traditional way to locate topics in the DB2 documentation is to use the *Table of Contents* in the left frame of the website. Click on any of the main topics in the *Table of Contents* to expand the sub-menus, and browse down in the sub-menus until you find the preferred topic.

The *Table of Contents* is organized primarily by the kind of tasks you may want to perform, but also includes *Product overviews*, *Reference*, *Glossary*, and *Index.*

The *Product overviews* describe the relationship between the available products in the DB2 family, the features offered by each of those products, and up to date release information for each of these products.

The task-based categories such as *Installing*, *Administering*, and *Developing* include topics that enable you to quickly complete tasks and develop a deeper understanding of the background information for completing those tasks.

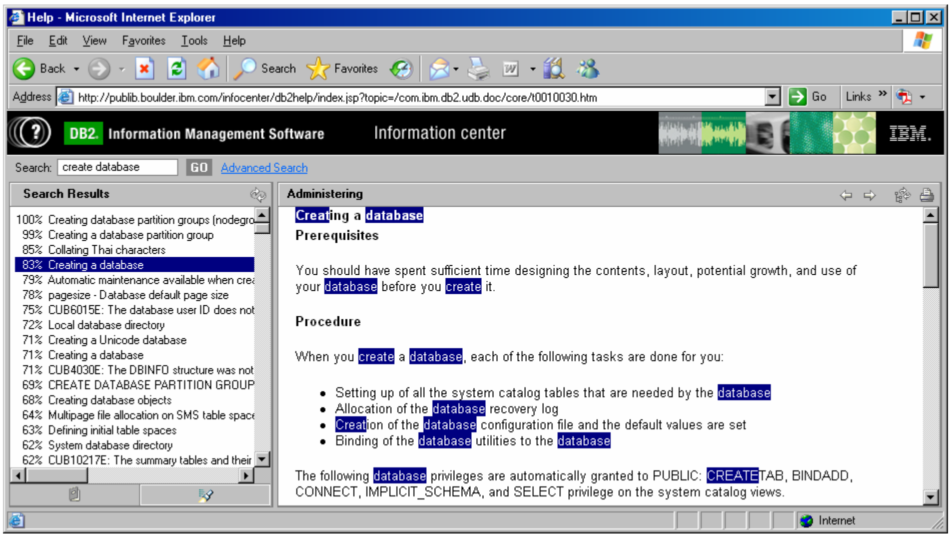
The *Reference* topics provide detailed information about a subject, including statement and command syntax, message help, and configuration parameters.

In the *Glossary*, you can look up definitions of terms used in the *DB2* documentation. The *Index* provides an alternative way to access all of the documentation.

4.4.2 How to search information in the Information Center

You can also search all of the topics in the *DB2 Information Center* by entering a search term (one or more keywords) in the *Search* text field. For example, if you need information about how to create a database, type **`create database`** in the *Search* field and press *Enter*. Next, look through the *Search Results* menu to the left and select the most preferred topic (e.g. *Creating a database*).

The *DB2 Information Center* will display the contents of the selected topic in the frame to the right. The topic text will be marked with the highlighted keywords:


Help - Microsoft Internet Explorer
File
Edit
View
Favorites
Tools
Help
Back
Search
Favorites
Address
http://publib.boulder.ibm.com/infocenter/db2help/index.jsp?topic=/com.ibm.db2.udb.doc/core/t0010030.htm
Go
Links
DB2. Information Management Software
Information center
IBM.
Search:
create database
GO
Advanced Search
Search Results
Administering
Internet

100% Creating database partition groups (nodegro
99% Creating a database partition group
85% Collating Thai characters
83% Creating a database
79% Automatic maintenance available when crea
78% pagesize - Database default page size
75% CUB6015E: The database user ID does not
72% Local database directory
71% Creating a Unicode database
71% Creating a database
71% CUB4030E: The DBINFO structure was not
69% CREATE DATABASE PARTITION GROUP
68% Creating database objects
64% Multipage file allocation on SMS table space
63% Defining initial table spaces
62% System database directory
62% CUB10217E: The summary tables and their

Creating a database

Prerequisites

You should have spent sufficient time designing the contents, layout, potential growth, and use of your database before you create it.

Procedure

When you create a database, each of the following tasks are done for you:

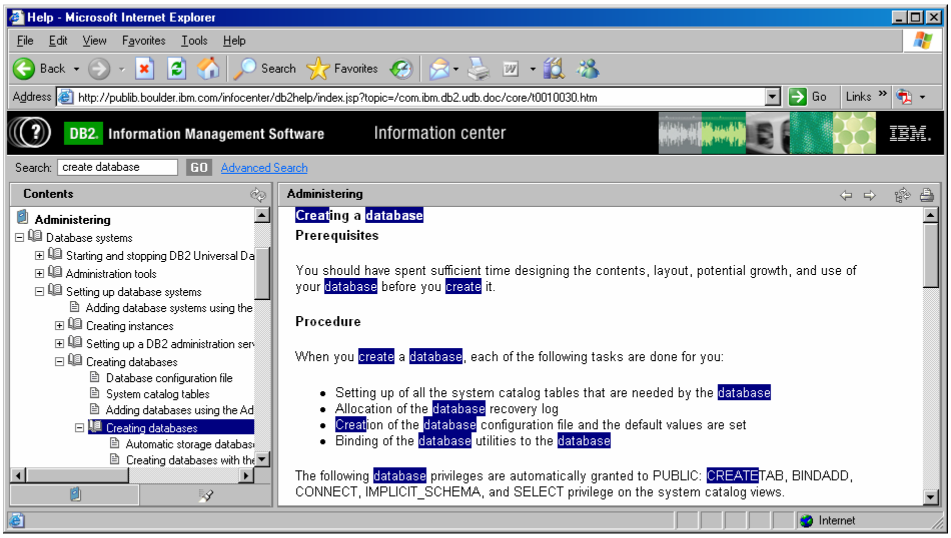
- Setting up of all the system catalog tables that are needed by the database
- Allocation of the database recovery log
- Creation of the database configuration file and the default values are set
- Binding of the database utilities to the database

The following database privileges are automatically granted to PUBLIC: CREATETAB, BINDADD, CONNECT, IMPLICIT_SCHEMA, and SELECT privilege on the system catalog views.

Figure 47 The search results in the DB2 Information Center

If you want to see where the current topic fits into the *Table of Contents*, click the *Show in Table of Contents* button on the toolbar above the right frame. This feature is very helpful if you have arrived at a topic from the search results or followed through several links of the related topics.

Click this button for the current topic *Creating a database*. The *DB2 Information Center* will display the refreshed *Table of Contents* with the current topic in its menu:


Help - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
http://publib.boulder.ibm.com/infocenter/db2help/index.jsp?topic=/com.ibm.db2.udb.doc/core/t0010030.htm
Go
Links
DB2. Information Management Software
Information center
IBM.
Search:
create database
GO
Advanced Search
Contents
Administering
Internet

- Administering
  - Database systems
    - Starting and stopping DB2 Universal Da
    - Administration tools
    - Setting up database systems
      - Adding database systems using the
      - Creating instances
      - Setting up a DB2 administration ser
      - Creating databases
        - Database configuration file
        - System catalog tables
        - Adding databases using the Ad
        - Creating databases
          - Automatic storage databas
          - Creating databases with the

Creating a database

Prerequisites

You should have spent sufficient time designing the contents, layout, potential growth, and use of your database before you create it.

Procedure

When you create a database, each of the following tasks are done for you:

- Setting up of all the system catalog tables that are needed by the database
- Allocation of the database recovery log
- Creation of the database configuration file and the default values are set
- Binding of the database utilities to the database

The following database privileges are automatically granted to PUBLIC: CREATETAB, BINDADD, CONNECT, IMPLICIT_SCHEMA, and SELECT privilege on the system catalog views.

Figure 48 The refreshed Table of Contents with the current topic

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 SQL queries please refer to the *DB2 Information Center* (sections ***Queries*** and ***Statements*** in the menu ***Reference » SQL***).

To execute the queries in the examples below, open the *DB2 Command Editor* and connect to the *SAMPLE* database (use the command: **`connect to sample`**).

5.1 Data Manipulation Language (DML SQL)

The DML SQL statements are used to retrieve and manipulate data from a database. This category encompasses the most fundamental commands, including SELECT, INSERT, UPDATE, and DELETE.

5.1.1 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)
```

5.1.2 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(deptno, deptname, mgrno, admrdept)
VALUES ('E31', 'ARCHITECTURE', '00390', 'E01')
```

5.1.3 UPDATE - to change data in existing rows

The Architecture department has changed its name to ‘Maintenance’, is assigned a new manager with the employee number ‘00400’, and should report to the department ‘E21’ instead of ‘E01’. Update the *DEPARTMENT* table with the above changes.

```
UPDATE department
SET deptname = 'MAINTENANCE', mgrno = '00400', admrdept = 'E21'
WHERE deptno = 'E31'
```

5.1.4 DELETE - to remove existing rows in tables

Delete the Maintenance department (*DEPTNO* = 'E31') from the *DEPARTMENT* table.

```
DELETE FROM department
WHERE deptno = 'E31'
```

5.2 Data Definition Language (DDL SQL)

The DDL SQL statements define the structure of a database, including rows, columns, tables, indexes, and database specifics such as file locations. The DDL SQL commands include the CREATE, ALTER, and DROP statements.

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

Create table *DEPT2* in the *USERSPACE1* 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 dept2
      (deptno    CHAR(3)      NOT NULL,
      deptname   VARCHAR(36)  NOT NULL,
      mgrno      CHAR(6),
      admrdept   CHAR(3)      NOT NULL,
      PRIMARY KEY (deptno))
IN userspace1
```

Similarly, create table *EMPLOYEE2* with columns *EMPNO*, *FIRSTNAME*, *LASTNAME*, and *WORKDEPT*. The primary key should be the column *EMPNO*.

```
CREATE TABLE employee2
      (empno     CHAR(6)      NOT NULL,
      firstname  VARCHAR(12)  NOT NULL,
      lastname   CHAR(15)     NOT NULL,
      workdept   CHAR(3)      NOT NULL,
      PRIMARY KEY (empno))
IN userspace1
```

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

Add a referential constraint to the *DEPT2* table so that the department manager *(MGRNO)* must be an employee number *(EMPNO)* that is present in the *EMPLOYEE2* table. *EMPNO* is the primary key of the *EMPLOYEE* table. If an employee is removed from the *EMPLOYEE* table, the department manager *(MGRNO)* values for all departments managed by this employee should become unassigned (or set to *NULL*). The constraint should be called *DEPTMGR*.

```
ALTER TABLE dept2
ADD CONSTRAINT deptmgr
FOREIGN KEY (mgrno)
REFERENCES employee2
ON DELETE SET NULL
```

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

Drop the table *DEPT2*.

```
DROP TABLE dept2
```

Drop the table *EMPLOYEE2*.

```
DROP TABLE employee2
```

5.3 Data Control Language (DCL SQL)

The DCL SQL statements control the security and permissions of the objects or parts of the database. The DCL SQL commands include the GRANT and REVOKE statements.

5.3.1 GRANT - to allow specified users to perform specified tasks

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

```
GRANT SELECT, INSERT
ON employee
TO USER phil, USER claire, GROUP planners
```

5.3.2 REVOKE - to cancel previously granted or denied permissions

Revoke all privileges on the *EMPLOYEE* table from the users *PHIL* and *CLAIRE* and from the group *PLANNERS*.

```
REVOKE ALL
ON employee
FROM USER phil, USER claire, GROUP planners
```