This tutorial shows you the basic steps and requirements to create and connect to a database in DB2® 10.1. This tutorial also introduces you to the objects that make up a DB2 database, as well as how to create and manipulate them. This tutorial is third of a series of six "DB2 10.1 fundamentals certification exam 610 prep" tutorials that will help prepare you for the DB2 10.1 fundamentals certification exam 610.

View more content in this series

Before you start

About this series
Thinking about obtaining the IBM Certified Database Associate — DB2 10.1 Fundamentals certification? If so, you've come to the right place. This series of DB2 10.1 certification preparation tutorials is designed to cover all the topics you need to know before you sit down to take the DB2 10.1 fundamentals certification exam (Exam 610). Even if you're not planning to seek certification right away, the information presented in this series can help you learn about many of the new features and functionality available in DB2 10 for z/OS® and DB2 10.1 for Linux®, UNIX®, and Windows®.

Don't see the tutorial you're looking for yet? You can review the DB2 9 tutorials in the DB2 9 Fundamentals certification 730 prep series.

About this tutorial
This is Part 3 of a series of six tutorials that will help you prepare for the DB2 10.1 Fundamentals Certification (Exam 000-610). This tutorial primarily covers the objectives in Section 3 of the test.

This tutorial provides an overview of the primary objects that exist in a DB2 database. Then, primarily using IBM Data Studio, you are given step-by-step instructions to create, connect to, and drop a database. Finally, you will use Data Studio to create, update, and drop database objects.

The installation of DB2 and IBM Data Studio is not covered here. We strongly recommend that you download and install IBM DB2 Express-C and IBM Data Studio 3.1.1. Installing DB2 and Data Studio and following along will help you understand many of the concepts that are tested on the certification exam.
Objectives
After completing this tutorial, you should be able to:

- Identify the database objects that exist in a DB2 database, including their basic characteristics and properties
- Use IBM Data Studio to:
  - Create, connect to, and drop a database
  - Create and modify objects that exist within the database

Prerequisites
Anyone with general knowledge of relational databases will be able to complete the objectives of this tutorial.

Since DB2 and IBM Data Studio are not required to complete this tutorial, the process of installing these products is not covered here. It is recommended that you download both (see Related topics).

System requirements
You need neither a copy of DB2 nor IBM Data Studio to complete this tutorial. Once again, it is recommended that you download and install both products, as you will get more out of this tutorial if you follow along each step, See Related topics to download both products.

Introduction to DB2 databases and database objects
Basic DB2 organization
Before showing you how to create, connect to, and alter objects in a database, you should have a general understanding of the organization and terminology DB2 uses. Similar to file systems that use a hierarchy of folders to organize files, DB2 uses a hierarchy of objects to organize data.

At the top of this hierarchy is the server or computer on which DB2 is loaded. Below this is the database instance level. An instance controls the databases under it and manages system resources. There must be at least one instance on the server, but there can be more. Each instance acts as its own, independent DB2 installation and controls access to the databases in it.

The database is our third level in the progression of database objects, and it is on this level you will start learning basic commands and the creation process. A database is a collection of data. DB2, being a relational database, contains tables that have a relationship to one another and use these relationships to organize data while keeping duplication of that data to a minimum.

All of the objects that follow can be considered part of the fourth level in our database hierarchy, as they are all parts of a DB2 database.

Objects in a database
The syntax for CREATE statements of the database objects is not covered in this tutorial, but you should become familiar with the syntax for the exam. More detail will be given about many of these objects (for example: tables, indices, views) in other tutorials, and the full CREATE statements are
fully detailed in the DB2 10.1 Information Center. Direct links to each object’s `CREATE` statement are in the Related topics section of this tutorial. Also, by going through the examples in "Creating, Altering, and Dropping Objects in Data Studio," you can generate your own `CREATE` statements through a hands-on process.

### Tablespace

A *tablespace* is a storage structure used by DB2 to group data and database objects so they may be stored on the system. A tablespace is used as a layer between the database and the container objects that hold the actual data. A tablespace can contain more than one table. When a database is first created, DB2 automatically creates the SYSCATSPACE, USERSPACE1, and TEMPSPACE1 tablespaces. When a table space is created, it must be associated with a buffer pool.

### Buffer pool

As mentioned, every tablespace in DB2 must be assigned a buffer pool. A *buffer pool* caches table and index information read from the disk. When a tablespace is created, it must be associated with a buffer pool, and the page size of the buffer pool must match that of the tablespace. When a database is created, DB2 creates the IBMDEFAULTBP buffer pool, and all the default table spaces use it.

### Schema

A *schema* is a logical grouping of database objects. It also is used for name qualifiers, which help naturally organize database objects by their name. The convention for this is `<SCHEMA-NAME>`.`<OBJECT-NAME>`. This also allows more than on object to have the same object name as long as each is in a different schema.

For example, if you sold hardware and software, you could have a HARDWARE and SOFTWARE schema. In each of them, you could have an ORDERS table to differentiate between hardware and software orders (HARDWARE.ORDERS and SOFTWARE.ORDERS).

If a schema is not qualified, or explicitly stated when the object is created, the schema name is the user ID of the user who created the object. If I am logged in as `db2user` and create a table named TEMP1, the full name of that table would be `DB2USER.TEMP1`.

### Table

*Tables* are where your data is stored, and because of this, they are the heart of DB2. A table consists of rows and columns. Each row is called a *record*, and the data at intersection of a row and a column is called a *value*. As the data may need to be described, access, and organized in different ways, there are many different types of tables. Tables will be covered in more detail in other tutorials in this series, but the types of tables are:

- Base tables — These tables hold persistent data, or data you want to keep throughout multiple connections/disconnections from the database. This group includes:
  - Regular tables — The most frequently used type of table; considered to be general purpose.
• Clustered tables — While not covered in detail, these tables have their data clustered together to provide access benefits, primarily in very large database environments. The types of clustered tables are:
  • Insert Time Clustering (ITC) Tables
  • Range Clustering Tables (RCT)
  • Multidimensional Clustering (MDC) Tables
• Temporal tables — A new feature of DB2 10.1:
  • System-period temporal tables — This type of temporal table uses an associated history table to keep historical versions of each row of the main table updated or deleted. The two tables use SYSTEM_TIME columns to determine when the row is current. Then a row is changed, the original row and the updated SYSTEM_TIMEs are inserted into the history table. The database generates both these columns and maintains the history table.
  • Application-period temporal tables — These store the stores data based on the time that the data is valid, using two BUSINESS_TIME period columns. The range provided by these columns indicates when the row is valid. Since there isn't a history table, all event periods, past, present, and future, are denoted in separate records of the same table, effective dates and their associated data are maintained in a single table.
  • Bitemporal tables — This type of temporal table combines the aspects of both application- and system-period temporal tables: the history table and the valid time period.
• Temporary tables — As the name implies, these are used for temporary work:
  • Declared — These are only good for one database session. When your database connection is terminated, the table and all of the data in it is dropped.
  • Created — Unlike Declared Temporary Tables, these do remain after each session, although any data in it does not.
• Materialized query tables — These are created by a `SELECT` statement and are used to improve the performance of queries.

Index

In most cases, there is no guarantee that the data in your tables is in any order because when data is added to a table, it is appended to the end of the table. This means that in most cases, every row in the table would need to be scanned to see if it contained the required information. When tables become very large, this can dramatically affect the amount of time it takes to access your data.

Like the index of a book organizes the topics in the book and points you to a page to assist you in finding the information that you want faster, an index in DB2 improves the database’s performance in retrieving data. While a book’s index organizes data in a book alphabetically by topic, as the DBA, you get to decide what DB2 uses to organize data. Primarily, you would choose one or multiple columns, but indices can even be used to point to XML data or blocks of data.

You can always see what indices are enabled on your database by finding the metadata in the system catalog. Indices will be covered in more detail in later tutorials in this series.
Views
A view provides an efficient way of examining data in any combination of one or more tables and/or one or more views. Views can be used like tables and have columns and rows, but they do not require any permanent storage.

A view uses a SELECT SQL statement that combines these sources (more in later tutorials). Since users of databases tend to have to draw data from multiple tables on a constant basis, a view is an efficient and simple method of data retrieval. Also, some in tables may contain sensitive data — employee salaries, for example. By creating a view with a SELECT statement that excludes that column, the data can remain secure. Views will be covered in more detail in later tutorials in this series.

Triggers
A trigger contains an action or set of actions executed when an insert, update, or delete operation occurs on a specified table, also called a triggering event. There are three primary types of triggers:

• Before — The trigger's actions occur just before the triggering event takes place.
• After — The trigger's actions occur immediately after the triggering event takes place.
• Instead of — Used to define insert, update, or delete actions performed against complex views.

Sequences
Sequences are used to automatically generate numeric values independently from all other database objects. They are frequently used to generate unique values. Consecutive values are retrieved using the NEXT VALUE or PREVIOUS VALUE expressions. If many numbers will need to be generated in rapid succession, they can be cached to help expedite I/O operations.

Alias
Like the name sounds, an alias is another name the database relates to a sequence, table, view, or another alias. They are normally used to reference an object without having to specify the schema name, but are useful for a shorter, simple, or colloquial term for the object.

Creating a database
Preparing to create a database
In this section, you'll learn the basic steps involved in creating and connecting to a database. This tutorial will show you how to do so using IBM Data Studio 3.1.1 and will assume you have just installed it and are using it for the first time. DB2 also allows you to create and manage database objects from a command line. If you want to use the command line to interact with DB2, we will show you the commands that Data Studio generates along each step of the way.

After DB2 is installed, you can easily access the DB2 command window by opening a Windows command prompt and typing db2cmd. In Linux, you may run the commands that follow in a terminal. If you open and use the command line in this manner, the commands below must be preceded by db2, a space, then the command shown.
Before you can create a database, you must have either SYSADM or SYSCTRL authority. For more information about these authorities, please refer to Part 2 of this series (see Related topics). If you have just installed a copy of DB2 to go through this tutorial, you already will have SYSADM authority.

IBM Data Studio is an Eclipse-based environment. If you have used Eclipse or other Eclipse-based environments, you should feel comfortable. In preparation for creating a database, open IBM Data Studio. When it opens, you will see a window asking you to select a workspace.

**Figure 1. Select workspace window**

This location identifies the default location of all the files you create in Data Studio. Also, it creates a .metadata folder containing information about all your preferences and your database connections. If you relaunch Data Studio after using it for a while and select a different location for your workspace, you will no longer see any database connections.

When you click **Ok**, Data Studio should open in the Database Administration Perspective. This can be confirmed by looking at the tab in the upper-right corner of the window, below the menu bar. Looking around the window, there are many panes along the left and bottom. In Data Studio, these are called views. Primarily, you will use the Administration Explorer view, which will be on the left side of the window as well as the Properties and SQL Results view, which are tabs on the pane that is across most of the bottom of the window.
Perspectives in Data Studio provide groups of tools and other functions based on the tasks you are trying to accomplish. In our case, it is Database Administration. For more detailed information about using Data Studio and its features, please see the Data Studio Information Roadmap provided in the Resources section.

Creating a database

To create a database, locate the Administration Explorer view. At the top, middle of that view, there is the New button. Click on the drop-down arrow on the right side of it and select New Database.

The new database window will open, asking you to specify the instance where the database will be located. Assuming you are running DB2 on your computer, the default values for the host name and port should be localhost and 50000, respectively. The Instance name, username, and password were set when DB2 was installed. The default instance name in Windows is DB2, while in Linux/UNIX it is db2inst1. You should test the connection prior to clicking Finish to be sure that you entered the information correctly.
In the Administration Explorer view, the hostname of the instance will appear as a server under the "All Databases" folder. Also, a new tab will open in your main work area to enter information about your new database.

The very least you must do to create a new database is enter the name you would like to call the database. For this example, enter MYNEWDB.

To see the command Data Studio issues, click Preview Command next to the Run button. This will expand the Command section of the tab. If you have been following along, this section should have the following command:
CREATE DATABASE MYNEWDB AUTOMATIC STORAGE YES;

This also is the command you can issue in the command line to create a database named MYNEWDB. Any changes you make prior to running the command will automatically update this section. For example: If you were to change the default buffer pool and tablespace page size option from its default of 4 KB to 8 KB, you will see the command change to: CREATE DATABASE MYNEWDB AUTOMATIC STORAGE YES PAGESIZE 8 K;.

There are many more options available for creating database. To see all these options, see the CREATE DATABASE command page of the DB2 10.1 Information Center.

To create the database, click Run.

When the command runs, the SQL Results view on the pane at the bottom of the window should be visible. Assuming all information was entered correctly, the status column should have a blue checkmark and say Success. Also, the database name should now be visible in the Administration Explorer view to the right of the window.

**Figure 6. Success in the SQL Results View**

![Success in the SQL Results View](image)

**Figure 7. The newly created MYNEWDB database**

![The newly created MYNEWDB database](image)

Congratulations! You just created a database in DB2 10.1. To create more, you may continue to alter the information under the New Database tab and click the Run button again. Each database must have its own unique name.

**Connecting to and dropping databases**

Now that you have created a database, we will see the process to connect to an existing database. For our example, we will connect to the Sample database that is created when DB2 is installed.

**Connecting to a database**

If this is the first time you are using Data Studio, the Sample database should not be in the Administration Explorer view. As you did with creating a new database, click on the drop-down arrow to the right of the New button. This time, select New Connection to a database. This will bring up the New Connection window.
On the left side of the window, you can see the many types of databases to which Data Studio can connect. This is because DB2 Connect is now packaged with IBM Data Studio.

Select **DB2 for Linux, UNIX, and Windows** from the list. Enter **SAMPLE** in the database field and similarly to when you created a database in the previous section, the default values for the host name and port should be localhost and 50000, respectively. Be sure to test your connection after you enter your username and password. If you do not wish to enter your password again, you may have Data Studio save it. After all the database connection information is entered and your ping to test the connection succeeds, click Finish. Just like when we created the MYNEWDB database, the SAMPLE database should appear in the Administration Explorer view.

**Figure 8. New Database Connection Window**

If the Sample database is already in the Administration Explorer view, all you need to do is double-click on the database. If you haven't saved your username and password, enter that and you will be connected.

To confirm that you are connected, you should see a list of folders coming from the database in the Administration Explorer view. These folders are the names of objects that make up DB2 databases. If you click on one, it will open a list of all objects of that type in the main work area. This list is called the List Editor.
Figure 9. Connected to the SAMPLE database

This is one of the few circumstances where Data Studio does not generate the command issued for those that use the command line. If connecting to the sample database, the command is CONNECT TO SAMPLE USER &lt;USERNAME&gt; USING &lt;PASSWORD&gt;;.

Entering your password this way will cause your password to be printed in the command-line window. This is an unadvisable practice for obvious security reasons. However, if you omit the USING clause, DB2 will immediately ask you to provide your password on the following line. No characters that are entered will show on the screen, and the cursor will not move until you press Enter.

CONNECT TO SAMPLE USER &lt;USERNAME&gt;;

If you are logged in to the operating system as the user, you may truncate the command even further to simply CONNECT TO SAMPLE;.

Disconnecting from a database

Disconnecting from a database is easy. For Data Studio, all you need to do is right-click on the database in the Administration Explorer view and select Disconnect. For the command line, simply enter TERMINATE;.
Of course, if you were to close the window, it will also sever your connection to the database.

**Dropping a database**

We will no longer be using the MYNEWDB database in any further examples, so we will now delete it. In Data Studio, before you can drop the database, you must first connect to it, then right-click on the database in the Administration Explorer view and select drop.

**Figure 11. Drop option**

This will bring up a tab in the main work area that will look similar to the one used to create the database. Unlike that window, there are not options that you may adjust. If you preview the command again, you should see:

```sql
CONNECT TO MYNEWDB; UNQUIESCE DATABASE; QUIESCE DATABASE IMMEDIATE FORCE CONNECTIONS; UNQUIESCE DATABASE;
CONNECT RESET; DROP DATABASE MYNEWDB;
```

The point of the `QUIESCE` commands is to ensure that no other users are currently connected to the database. Since we have just created this database, we know that no one else is using it. If this is the case, in the command line, you only need to enter `DROP DATABASE MYNEWDB;`.

After you click **Run** and the command processes, the MYNEWDB database should no longer be visible in the Administration Explorer view. The MYNEWDB was removed completely from the disk.

**Creating, altering, and dropping objects in Data Studio**

Now that you know how to create, connect to, and drop a database, and have an overview of the types of objects in DB2, you'll learn how to use Data Studio to interact with and create those objects. Once again, for the detailed creation commands to enter into the command line, use the links to the Information Center in the **Related topics** section.
In this latest version of Data Studio, instead of deploying each change to database objects individually and immediately, now Data Studio utilizes change plans, which will let you deploy all changes at one time.

**Creating database objects**

In this example, we will create a new schema in the Sample database called TEMP. Once you connect to the Sample database, left-click on the Schema folder in the Administration Explorer view, which will load the list of all schemas in the List Editor. At the top of the List Editor, there is a green plus sign (+) that, when clicked, creates a new object with default values and opens the Properties view at the bottom of the screen with editable values. You may instead right-click on the Schema folder in the Administration Explorer view and select Create Schema.

**Figure 12. Create Button in the List Editor**

**Figure 13. Creating an object from the Administration Explorer View**

For our example, a new schema will appear with the default name of Schema2 and to the left of that name in the List Editor will be a delta symbol, denoting an undeployed change. Change the name to TEMP in the Properties view. Until these changes are deployed, you may reselect the undeployed object and continue to alter it. You will not see the List Editor reflect the name change until you click on it in the List Editor or click on the schema folder in the Administration Explorer view.

**Figure 14. The new schema in the List Editor**
Figure 15. Editing the schema name in the Properties view

To deploy this change to the database, you may click on the Change Plans folder in the Administration Explorer view, right-click on the change plan that is still in Pending status, then select Review and Deploy. Alternatively, you may click the Review and Deploy Changes button no matter what objects you may be reviewing in the List Editor. This button is third from the right on the second level of buttons at the top right of the List Editor.

Figure 16. Reviewing the Change Plan in the List Editor

Figure 17. The Review and Deploy Button

Before deploying changes to the database, Data Studio will open the Review and Deploy window, which will show you the exact command that it is about to issue to the database. This is the same command that you would issue if you created the schema using the command line.
While our example created a schema, you should know that the same process can be used to create any database objects mentioned in Introduction to Databases and Database Objects, with the differences being in the options you would need to select and alter in the Properties View.

### Altering and dropping database objects

Altering and dropping database objects is done in a similar format. Start by viewing the object in the Administration Explorer view. Right-click on the object you want to alter or drop and select the appropriate option. If you select Alter, the values in the Properties view become editable, just as they were when creating a new object. When all your changes are complete, you may then deploy them as you did when creating an object.

### Altering table data

Data Studio can also be used to insert, update, and delete data from tables. To do so, click on the Tables folder in the Administration Explorer view and find the table you would like to alter in the list editor. Right-click on that table and select **Edit Data**. This will open the table data editor in another tab, showing the data is in the table’s rows and columns. Individual cells may be updated, and whole rows may be created or deleted.

### Figure 19. The Commit button

As with altering to database objects, these changes do not take effect until committed. The button to commit your changes is in a similar same location to the deploy changes button from the List Editor, third from the right, at the top-right of the table data editor’s tab. If you would not like to commit the changes you have made, you may click the refresh button, which is the right-most button of the group in the top-right corner of the tab.
Conclusion

You have now learned how to use IBM Data Studio to create and connect to databases. You also learned how to use it to create and modify objects that exist within the database. Most actions you do in Data Studio can generate the command that can be saved for use in a script or entered in the command line, even though using Data Studio can be much simpler than typing in commands. Additionally, you have been introduced to the major objects that exist in a DB2 database as well as their general function.

The DB2 10.1 Information Center is most comprehensive source for DB2 documentation and should contain answers to most questions you might have, including all commands and descriptions for all command options. Also, developerWorks has many other very useful DB2 articles and tutorials that are designed for those that are new to databases.
Related topics

- Read the Preparation Guide for DB2 10.1 Fundamentals Exam 610 to learn in-depth information about each of the concepts presented in this tutorial. This guide is a compilation of topics from the DB2 10.1 Information Center documentation.
- Use an RSS feed to request notification for the upcoming tutorials in this series. (Find out more about RSS feeds of developerWorks content.)
- The DB2 10.1 Information Center contains in-depth information on all aspects and features on the DB2 10.1 family of products. The pages contain the CREATE commands for the database objects in this tutorial follow. If you have not reviewed commands in this format before, be sure to review How to read the syntax diagrams in DB2 Information Center:
  - CREATE DATABASE
  - CREATE TABLESPACE
  - CREATE BUFFERPOOL
  - CREATE SCHEMA
  - CREATE TABLE
  - CREATE INDEX
  - CREATE VIEW
  - CREATE TRIGGER
  - CREATE SEQUENCE
  - CREATE ALIAS
- Get an overview of Test 000-610: DB2 10.1 Fundamentals, especially the objectives, and find more resources to prepare for the exam.
- Find more information about other IBM Certification Tests, including Test 000-611: DB2 10.1 DBA for Linux, UNIX, and Windows.
- Learn more in-depth DB2 information with the DB2 basics series, especially if you want more information about Tablesspaces and bufferpools.
- Some excellent information can be found in the DB2 9 Fundamentals certification 730 prep series, especially Part 3: Accessing DB2 data, which relates to topics covered in this tutorial, although some information has been deprecated in DB2 10.1, such as the Control Center.
- Learn more about IBM Data Studio on developerWorks with the IBM Data Studio 3.1.1 Information Roadmap.
- Get the latest edition of IBM DB2 Express-C free of charge.
- Download a trial version of IBM DB2 for Linux, UNIX, and Windows.
- Get IBM Data Studio for free and start creating and managing databases right away.
- Now you can use DB2 for free. Download DB2 Express-C, a no-charge version of DB2 Express Edition for the community that offers the same core data features as DB2 Express Edition and provides a solid base to build and deploy applications.

© Copyright IBM Corporation 2012
Trademarks
(www.ibm.com/developerworks/ibm/trademarks/)