Load Sample Database

**Summary**: in this tutorial, you will learn how to create a new database in SQL Server and execute the script to load the sample database.

First, you need to download the following zip file if you have not done so:

[Download SQL Server Sample Database](https://www.sqlservertutorial.net/wp-content/uploads/SQL-Server-Sample-Database.zip)

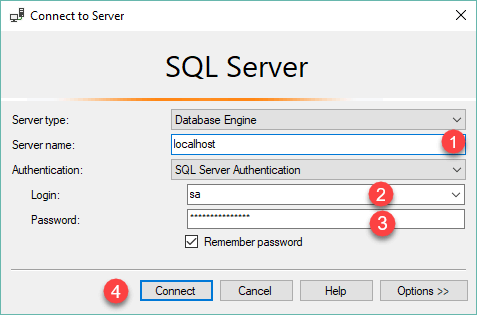
Second, uncompress the zip file, you will see three SQL script files:

* BikeStores Sample Database - create objects.sql – this file is for creating database objects including schemas and tables.
* BikeStores Sample Database - load data.sql – this file is for inserting data into the tables
* BikeStores Sample Database - drop all objects.sql – this file is for removing the tables and their schemas from the sample database. It is useful when you want to refresh the sample database.

Third, let’s create a database, create the schemas and tables, and load the sample data.

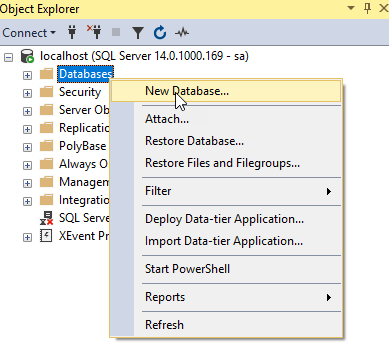
Step 1

Connect to the SQL Server by (1) choosing the server name, (2) enter the user and (3) password and (4) click the **Connect** button.



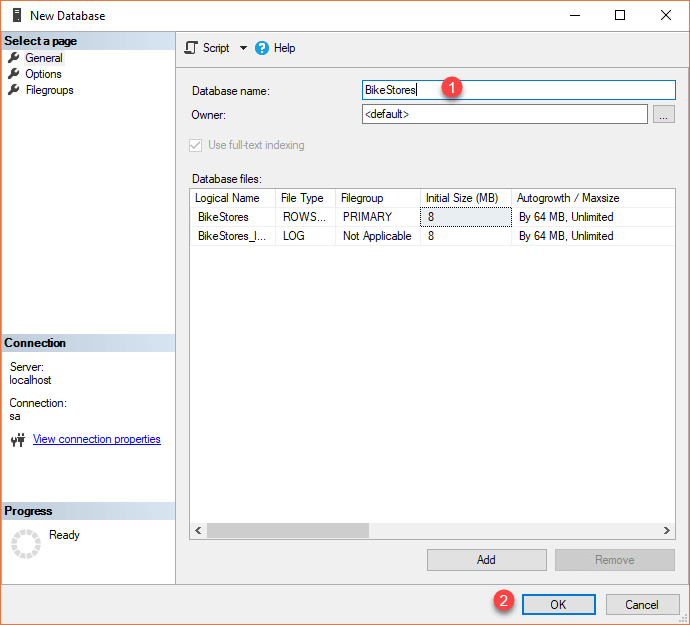
Step 2

Right-click the **Databases** node in the **Object Explorer** and select the **New Database…** menu item



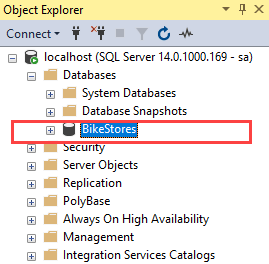
Step 3

(1) Enter the **Database name**as BikeStores and (2) click the **OK** button to create the new database.



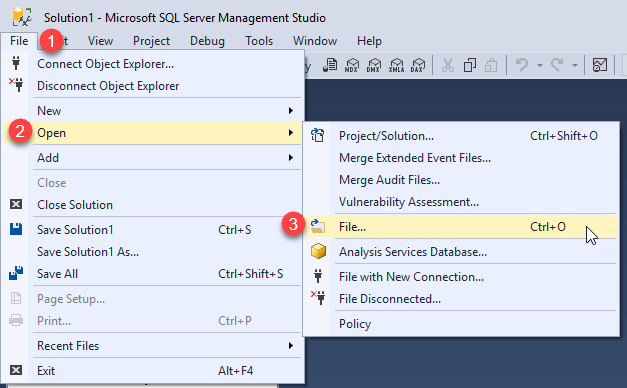
Step 4

If everything is fine, you will see the database **BikeStores** appears under Databases node as shown in the screenshot below:



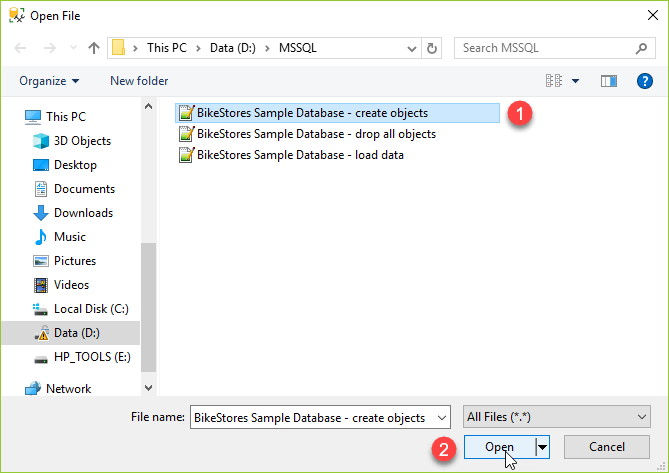
Step 5

From the File menu, choose Open > File… menu item to open a script file.



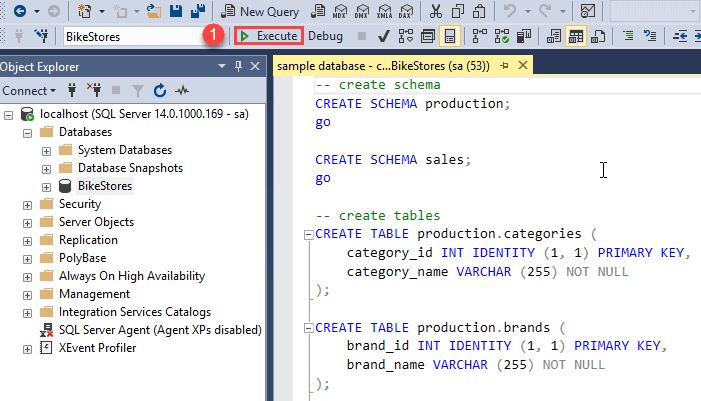
Step 6

Select the **BikeStores Sample Database – create**objects.sql file and click the Open button

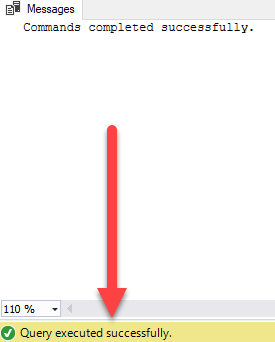


Step 7

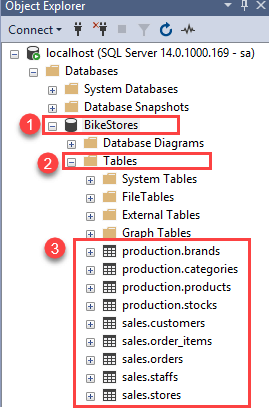
Click the **Execute** button to execute the SQL script.



You should see the following result indicated that the query executed successfully.

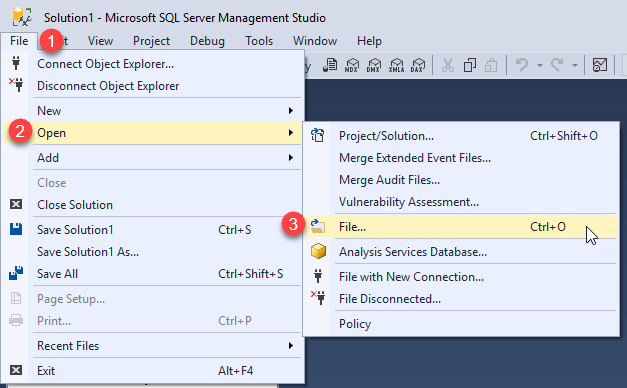


If you expand the **BikeStores > Tables**, you will see the schemas and their tables are created as shown below:



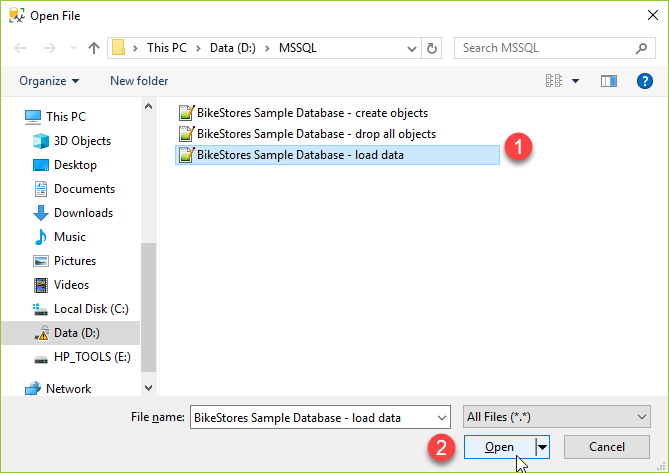
Step 8

Open the file for loading data into the tables.



Step 9

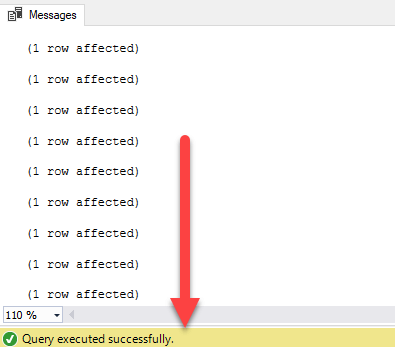
Choose the **BikeStores Sample Database – load data.sql** file and click the Open button.



Step 10

Click the **Execute** button to load data into the tables.

You should see the following message indicating that all the statements in the script were executed successfully.



# SQL Server Views

When you use the [SELECT](https://www.sqlservertutorial.net/sql-server-basics/sql-server-select/) statement to query data from one or more tables, you get a result set.

For example, the following statement returns the product name, brand, and list price of all products from the products and brands tables:

SELECT

product\_name,

brand\_name,

list\_price

FROM

production.products p

INNER JOIN production.brands b

ON b.brand\_id = p.brand\_id;

Code language: SQL (Structured Query Language) (sql)

Next time, if you want to get the same result set, you can save this query into a text file, open it, and execute it again.

SQL Server provides a better way to save this query in the database catalog through a view.

A view is a named query stored in the database catalog that allows you to refer to it later.

So the query above can be stored as a view using the [CREATE VIEW](https://www.sqlservertutorial.net/sql-server-views/sql-server-create-view/) statement as follows:

CREATE VIEW sales.product\_info

AS

SELECT

product\_name,

brand\_name,

list\_price

FROM

production.products p

INNER JOIN production.brands b

ON b.brand\_id = p.brand\_id;

Code language: SQL (Structured Query Language) (sql)

Later, you can reference to the view in the SELECT statement like a table as follows:

SELECT \* FROM sales.product\_info;

Code language: SQL (Structured Query Language) (sql)

When receiving this query, SQL Server executes the following query:

SELECT

\*

FROM (

SELECT

product\_name,

brand\_name,

list\_price

FROM

production.products p

INNER JOIN production.brands b

ON b.brand\_id = p.brand\_id;

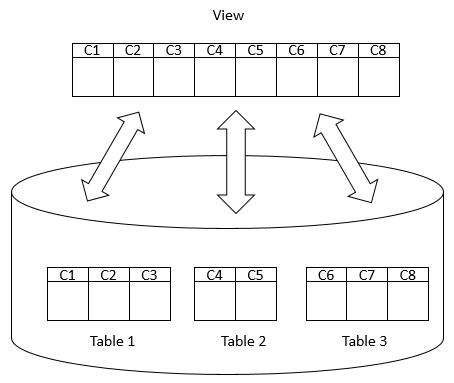
);

Code language: SQL (Structured Query Language) (sql)

By definition, views do not store data except for [indexed views](https://www.sqlservertutorial.net/sql-server-views/sql-server-indexed-view/).

A view may consist of columns from multiple tables using joins or just a subset of columns of a single table. This makes views useful for abstracting or hiding complex queries.

The following picture illustrates a view that includes columns from multiple tables:



## Advantages of views

Generally speaking, views provide the following advantages:

### Security

You can restrict users to access directly to a table and allow them to access a subset of data via views.

For example, you can allow users to access customer name, phone, email via a view but restrict them to access the bank account and other sensitive information.

### Simplicity

A relational database may have many tables with complex relationships e.g., one-to-one and one-to-many that make it difficult to navigate.

However, you can simplify the complex queries with joins and conditions using a set of views.