

## Introduction to MySQL

- ♣ Installing MySQL and starting MySQL instance.
- ♣ History of MySQL
- ♣ Components of MySQL -DML,DDL, DCL, DQL
- ♣ Data types in MySQL
- ♣ Creating databases and show databases

### 🔊 Introduction:

MySQL is an open-source relational database management system (RDBMS). It is the most popular database system used with PHP. MySQL is developed, distributed, and supported by Oracle Corporation.

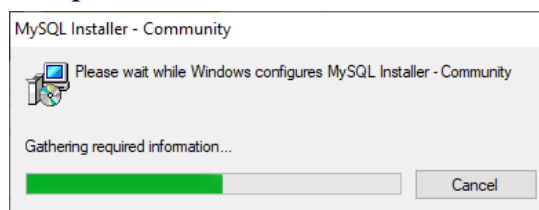
- The data in a MySQL database are stored in tables which consists of columns and rows.
- MySQL is a database system that runs on a server.
- MySQL is ideal for both small and large applications.
- MySQL is very fast, reliable, and easy to use database system. It uses standard SQL
- MySQL compiles on a number of platform

### 🔊 Installing MySQL on Windows

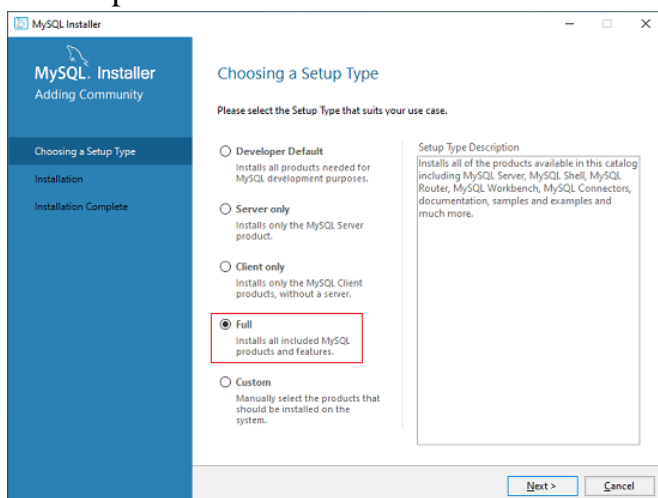
MySQL is one of the most popular relational database management software that is widely used in today's industry. It provides multi-user access support with various storage engines.

- MySQL Setup Software
- Microsoft .NET Framework 4.5.2
- Microsoft Visual C++ Redistributable for Visual Studio 2019
- RAM 4 GB

**Step 1:** After downloading the setup, double click the MSI **installer .exe file**. It will give the following screen:

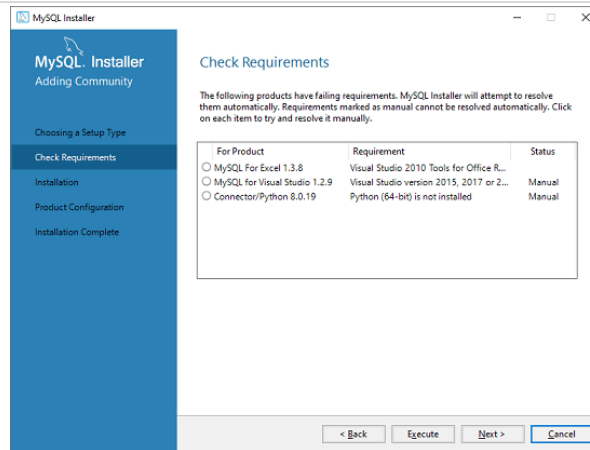


**Step 2:** In the next wizard, choose the **Setup Type**. There are several types available, and you need to choose the appropriate option to install MySQL product and **features**. Here, we are going to select the **Full** option and click on the Next button.

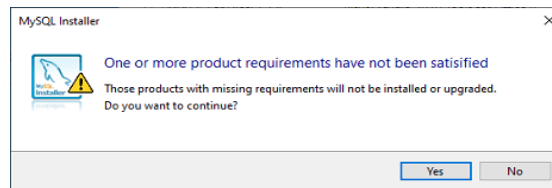


This option will install the following things: MySQL Server, MySQL Shell, MySQL Router, **MySQL Workbench**, MySQL Connectors, documentation, samples and examples, and many more.

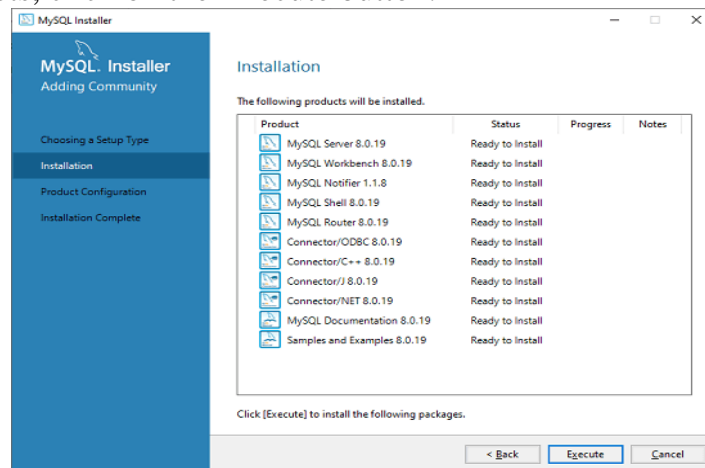
**Step 3:** Once we click on the Next button, it may give information about some features that may fail to install on your system due to a lack of requirements. We can resolve them by clicking on the **Execute** button that will install all requirements automatically or can skip them. Now, click on the Next button.



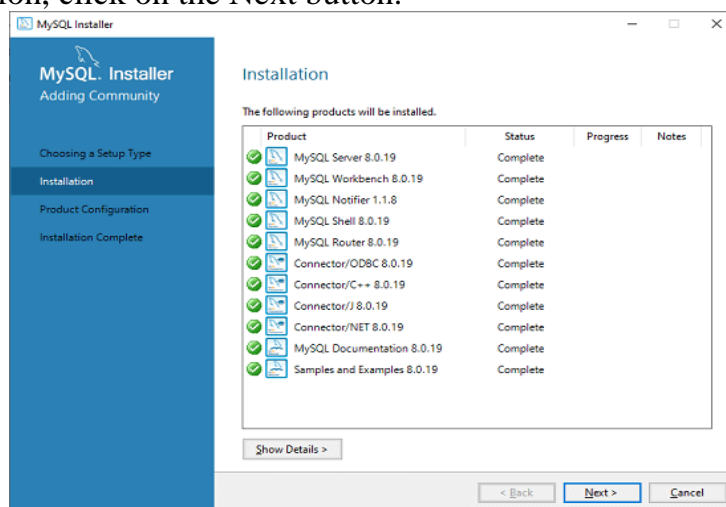
**Step 4:** In the next wizard, we will see a dialog box that asks for our confirmation of a few products not getting installed. Here, we have to click on the **Yes** button.



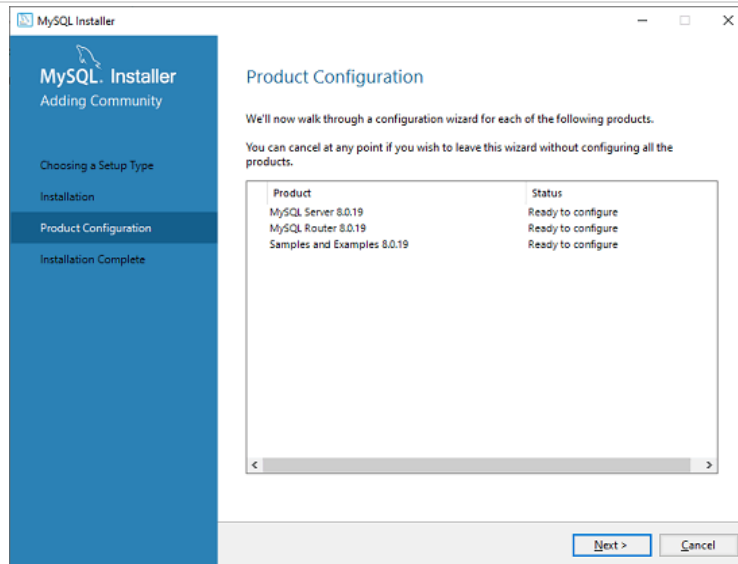
After clicking on the Yes button, we will see the list of the products which are going to be installed. So, if we need all products, click on the **Execute** button.



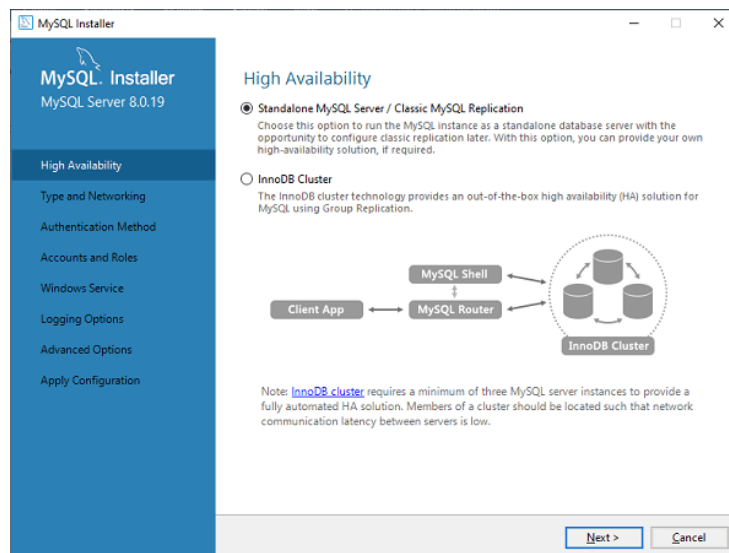
**Step 5:** Once we click on the Execute button, it will download and install all the products. After completing the installation, click on the **Next** button.



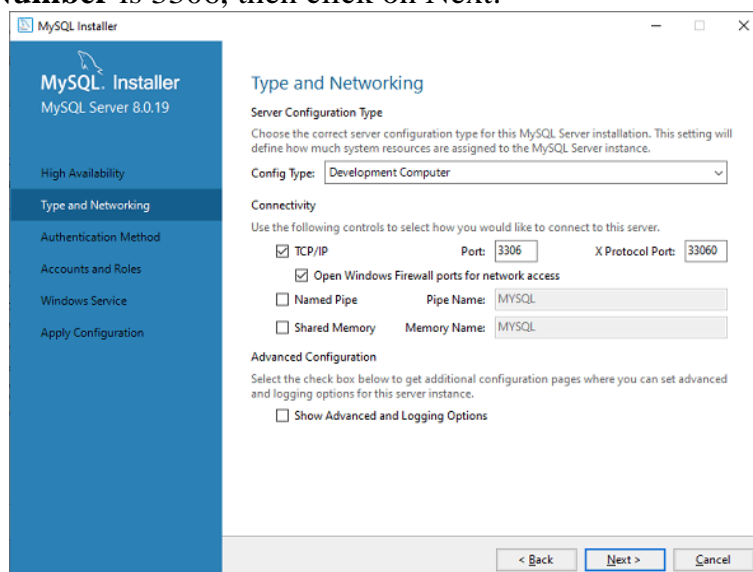
**Step 6:** In the next wizard, we need to configure the MySQL Server and Router. Here, I am not going to configure the Router because there is no need to use it with MySQL. We are going to show you how to configure the server only. Now, click on the **Next** button.



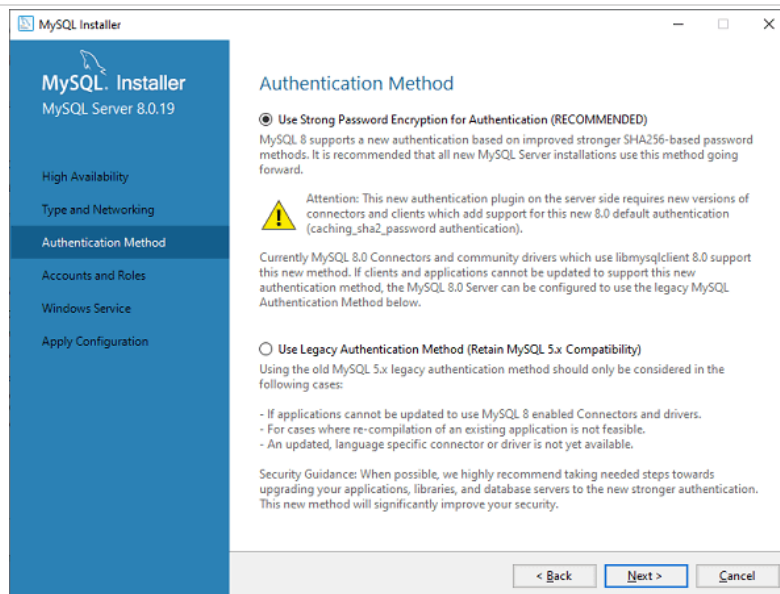
**Step 7:** As soon as you will click on the Next button, you can see the screen below. Here, we have to configure the MySQL Server. Now, choose the Standalone MySQL Server/Classic MySQL Replication option and click on Next. Here, you can also choose the InnoDB Cluster based on your needs.



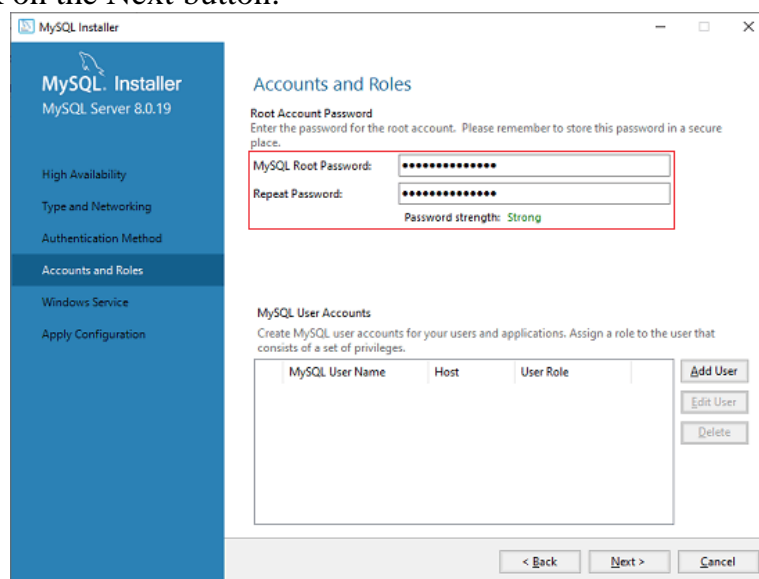
**Step 8:** In the next screen, the system will ask you to choose the Config Type and other connectivity options. Here, we are going to select the **Config Type** as 'Development Machine' and Connectivity as **TCP/IP**, and **Port Number** is 3306, then click on Next.



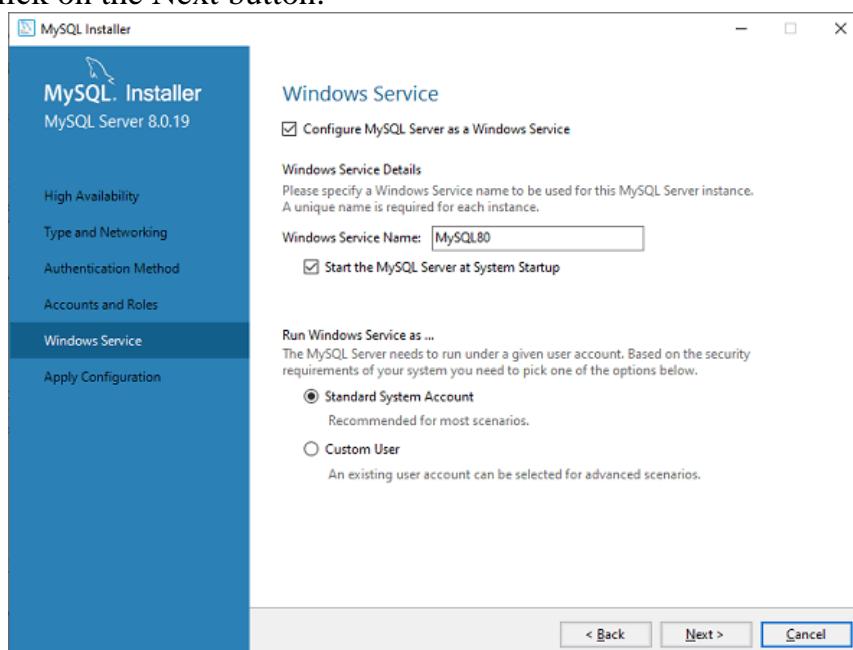
**Step 9:** Now, select the Authentication Method and click on Next. Here, I am going to select the first option.



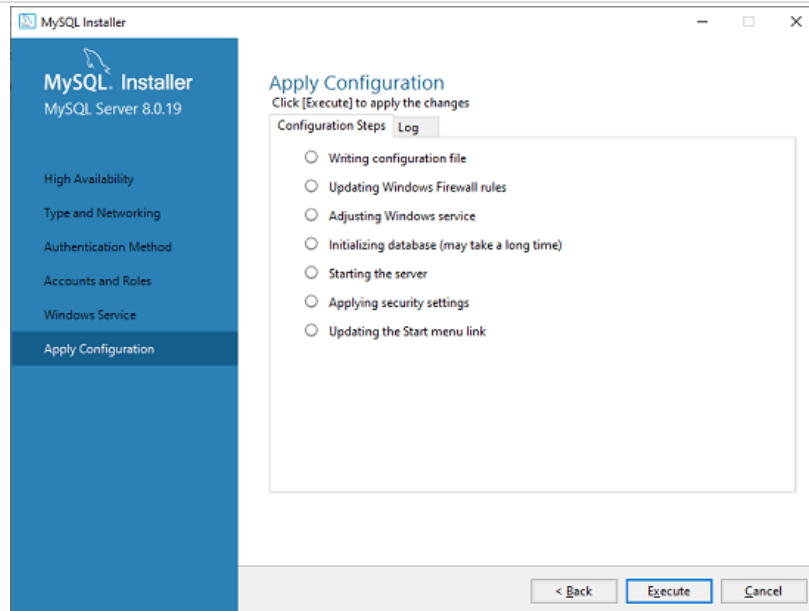
**Step 10:** The next screen will ask you to mention the MySQL Root Password. After filling the password details, click on the Next button.



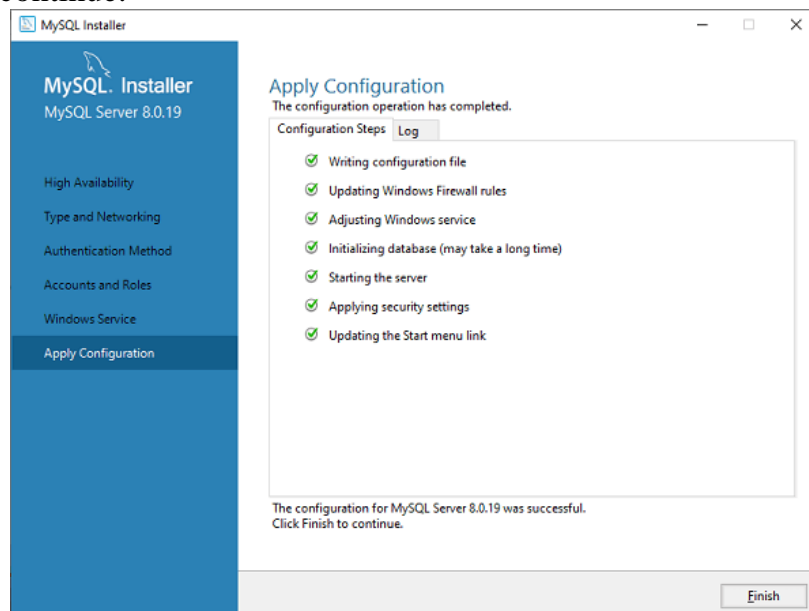
**Step 11:** The next screen will ask you to configure the Windows Service to start the server. Keep the default setup and click on the Next button.



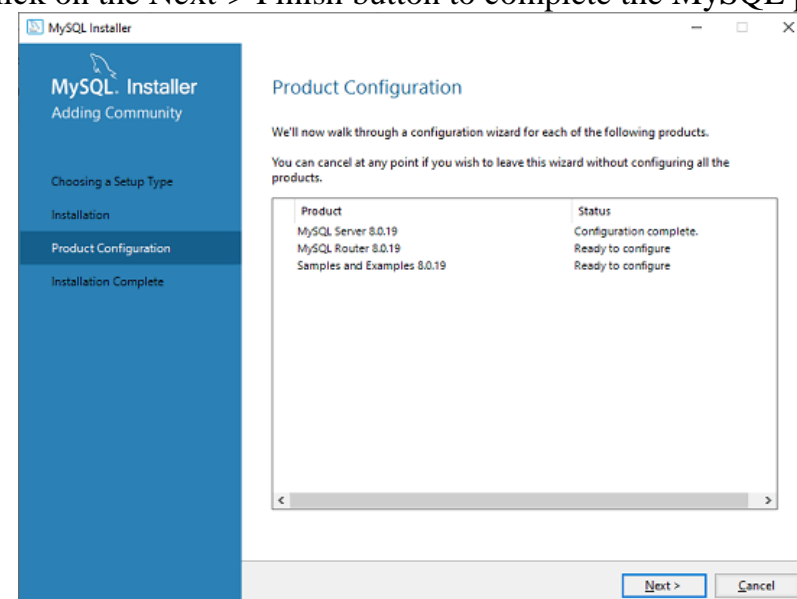
**Step 12:** In the next wizard, the system will ask you to apply the Server Configuration. If you agree with this configuration, click on the Execute button.



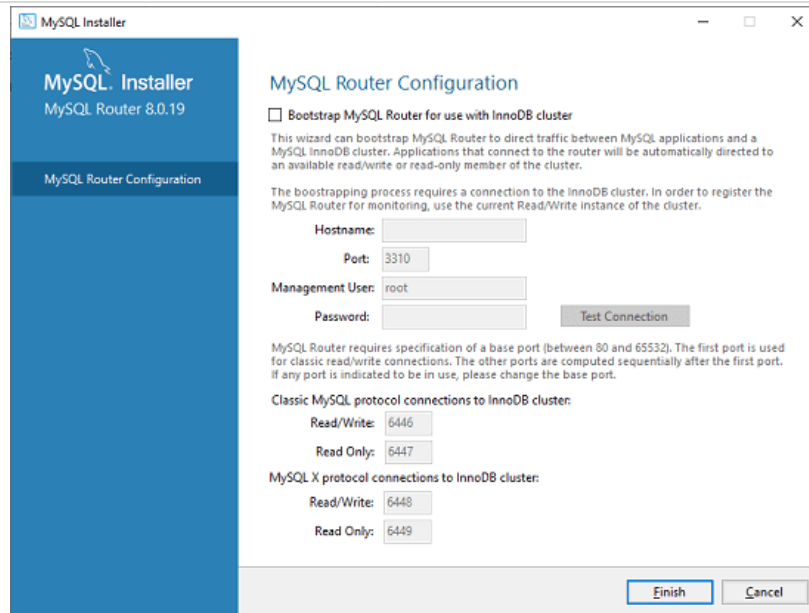
**Step 13:** Once the configuration has completed, you will get the screen below. Now, click on the **Finish** button to continue.



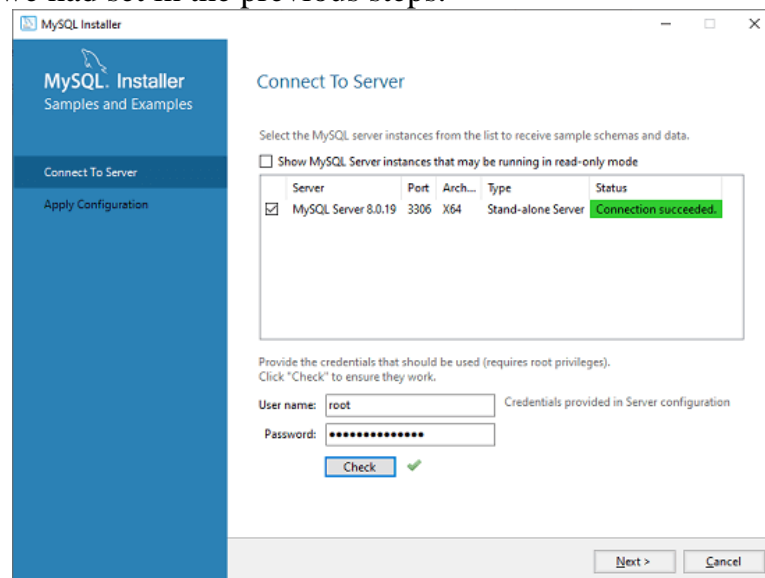
**Step 14:** In the next screen, you can see that the Product Configuration is completed. Keep the default setting and click on the Next-> Finish button to complete the MySQL package installation.



**Step 15:** In the next wizard, we can choose to configure the Router. So click on Next->Finish and then click the Next button.

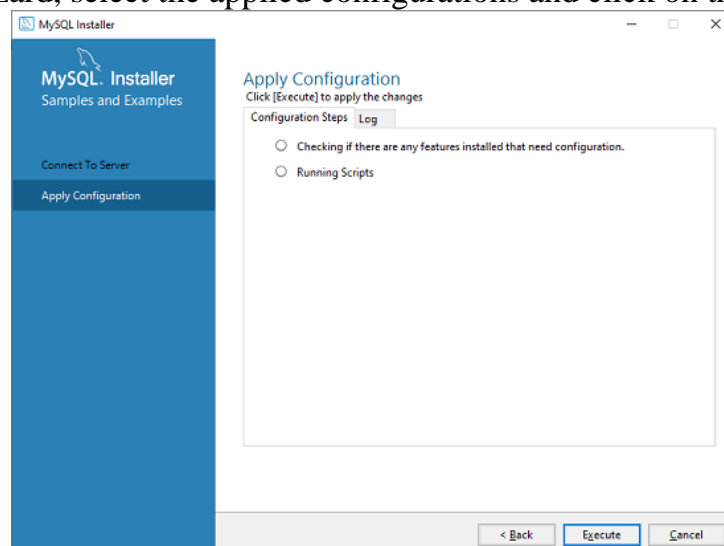


**Step 16:** In the next wizard, we will see the Connect to Server option. Here, we have to mention the root password, which we had set in the previous steps.

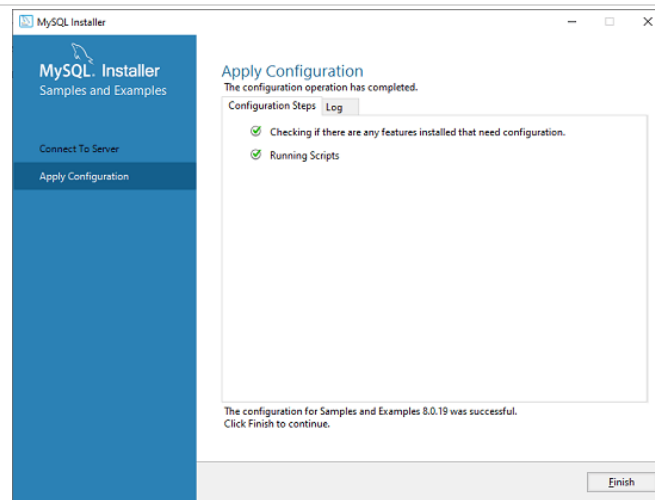


In this screen, it is also required to check about the connection is successful or not by clicking on the Check button. If the connection is successful, click on the Execute button. Now, the configuration is complete, click on Next.

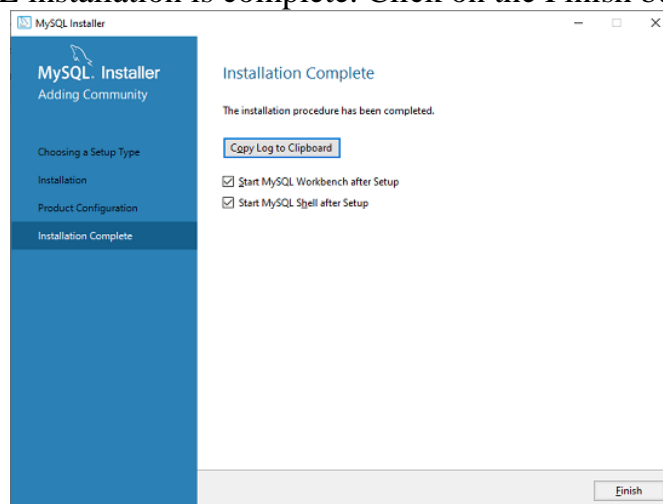
**Step 17:** In the next wizard, select the applied configurations and click on the Execute button.



**Step 18:** After completing the above step, we will get the following screen. Here, click on the Finish button.



**Step 19:** Now, the MySQL installation is complete. Click on the Finish button.



### Starting MySQL instance

Once MySQL has been successfully installed, the base tables have been initialized, and the server has been started, you can verify its working via some simple tests.

Open your **MySQL Command Line Client**; it should have appeared with a **mysql> prompt**. If you have set any password, write your password here. Now, you are connected to the MySQL server, and you can execute all the SQL command at **mysql> prompt** as follows:

```

MySQL 8.0 Command Line Client
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 14
Server version: 8.0.19 MySQL Community Server - GPL

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql      |
| performance_schema |
| sakila     |
| sys        |
| world      |
+-----+
6 rows in set (0.24 sec)

mysql>

```

### History of MySQL

MySQL is an open source SQL (structured query language) database management system. It is a system that helps store and manage data efficiently.

- Unireg, which is the code base of MySQL, was started in **1981**.
- MySQL was founded in **1995** in Sweden.
- In **2000**, MySQL went open source, so it could be accessed and used by all.
- In the year **2001**, Marten Mickos was elected as the CEO of MySQL.
- In the year **2002**, MySQL launched its headquarters in USA, in addition to Sweden headquarters.



- In **2003**, MySQL entered into a partnership with SAP, and many features were developed in MySQL keeping SAP in mind.
- In **2005**, MySQL launched MySQL Network. Oracle purchased Innobase, which is the backend of MySQL's InnoDB storage.
- In **2008**, MySQL was acquired by Sun Microsystems.
- In **2009**, Sun Microsystems and Oracle entered into a definitive agreement under which Oracle acquired Sun Microsystems.
- The mSQL database system was designed to connect the data that is stored in tables using customized fast low level (ISAM) routines.
- ISAM refers to indexed sequential access method, which is a file management system. It is a technique that helps access records in the tables sequentially, i.e in the same order in which the records were entered into the table, or randomly with the help of an index. Every index can be used to define a different order for the records in the table.
- ISAM was originally developed by IBM before the development of VSAM (Virtual Storage Access Method) and relational databases.
- When mSQL was tested to see its efficiency, it was observed that mSQL was not quick enough or flexible enough for the requirements in hand.
- MySQL was initially created for personal usage from mSQL based on the low-level language ISAM.
- MySQL has been named after co-founder Monty Widenius's daughter- My. The logo, a dolphin is known as 'Sakila'.

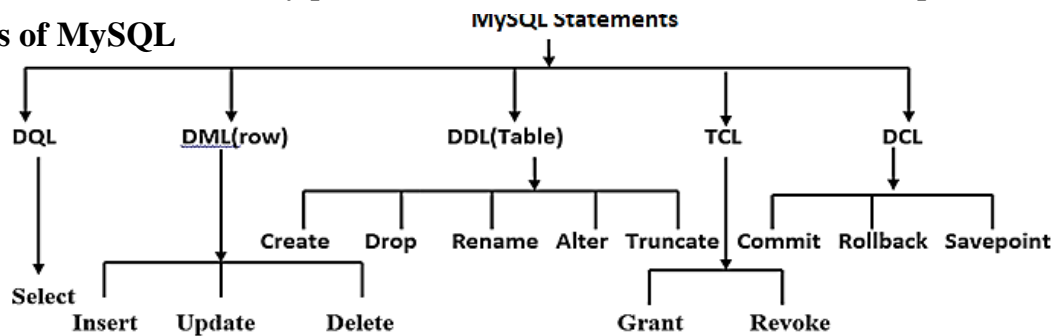
### 🔊 MySQL Features

- **Easy to use:** MySQL is easy to use. We have only basic knowledge of SQL.
- **Secure Data Protection:** Only authorized user's access databases. Passwords are encrypted.
- **Client/ Server Architecture:** MySQL follows the working of a client/server architecture. There is a database server (MySQL) and randomly many clients (application programs), which communicate with the server; that is, they can query data, save changes, etc.
- **Open Source and Free:** MySQL is free to use so that we can download it from MySQL official website without any cost.
- **High Scalable and Flexible:** MySQL supports multi-threading that makes it easily scalable. You can run deeply embedded applications and create data warehouses holding a huge amount of data up to as much as 50 million rows or more. The default file size limit is about 4 GB.
- **Speed:** MySQL is one of the very fast database languages.
- **Compatible on many operating systems:** MySQL is compatible to run on many operating systems like Windows, Linux etc. MySQL also provides a facility that the clients can run on the same computer as the server or on another computer.
- **Allows roll-back:** MySQL allows transactions to be rolled back, commit, and crash recovery.
- **Memory efficiency:** Its efficiency is high because it has a very low memory leakage problem.
- **High Performance:** MySQL is faster, more reliable, and cheaper because of its unique storage engine architecture. It provides very high-performance results in comparison to other databases without losing an essential functionality of the software. It has fast loading utilities because of the different cache memory.
- **High Productivity:** MySQL uses Triggers, Stored procedures, and views that allow the developer to give higher productivity.
- **Platform Independent:** It can download, install, and execute on most of the available OS.
- **Partitioning:** improves the performance and provides fast management of the large database.
- **GUI Support:** MySQL provides a unified visual database GUI tool "**MySQL Workbench**" to work with database architects, developers, and Database Administrators. [MySQL Workbench](#) provides SQL development, data modeling, data migration, and comprehensive administration tools for server configuration, user administration, backup, and many more. MySQL has a fully GUI supports from MySQL Server version 5.6 and higher.



- **Dual Password Support:** MySQL version 8.0 provides support for dual passwords-one is the current password, and secondary password allows us to transition to the new password.

## 🔊 Components of MySQL



### ➤ DDL: Data Definition Language

All DDL commands are auto-committed. It saves all the changes permanently in database.

- DDL statements are used for creating, modifying and dropping structure of database objects like table, view, database etc.
- It also defines indexes (keys), specifies links between tables, and forces constraints between tables.

| Command         | Description   |
|-----------------|---|
| <b>create</b>   | Creates a database object like new table, a view of a table etc.                  |
| <b>alter</b>    | Alters or modifies structure of an existing database object like table, view etc. |
| <b>truncate</b> | remove all records from a table, including all spaces allocated for the records   |
| <b>drop</b>     | Deletes an entire database object like table, view etc.                           |
| <b>rename</b>   | Rename a database object like table, view etc.                                    |

### ➤ DML: Data Manipulation Language

DML statements used for storing, retrieving, modifying, & deleting data within the database. DML commands are not auto-committed. Changes are not permanent to database, they can be rolled back.

| Command       | Description  |
|---------------|--|
| <b>insert</b> | insert data (new row) into a table                                 |
| <b>update</b> | updates existing data(row) within a table                          |
| <b>delete</b> | deletes all records from a table, the space for the records remain |

### ➤ TCL: Transaction Control Language

TCL statements are used to manage the changes made by DML statements. It allows statements to be grouped together into logical transactions. These commands can cancel changes made by other commands by rolling back to original state. It can also make changes permanent.

| Command          | Description         |
|------------------|---------------------|
| <b>commit</b>    | to permanently save |
| <b>rollback</b>  | to undo change      |
| <b>savepoint</b> | to save temporarily |

### ➤ DCL : Data Control Language

DCL statements control access of data and database. It assign permission to user for use database object. It can also take back (revert) the permission. It allows central administration of user privileges and access.

| Command       | Description  |
|---------------|--|
| <b>grant</b>  | gives user's access privileges to database                             |
| <b>revoke</b> | withdraw access rights given with GRANT command (take back permission) |

### ➤ DQL: Data Query Language

DQL allows getting data from the database. When SELECT is fired against a table or tables, the result is compiled into a temporary table.

| Command       | Description                             |
|---------------|---|
| <b>select</b> | retrieve records from one or more table |

## ☛ Data types in MySQL

A Data Type specifies a particular type of data, like integer, floating points, Boolean, etc. It also identifies the possible values for that type, the operations that can be performed on that type, and the way the values of that type are stored. In MySQL, each database table has many columns and contains specific data types for each column.

We can determine the data type in MySQL with the following characteristics:

- ✓ The type of values (fixed or variable) it represents.
- ✓ The storage space it takes is based on whether the values are a fixed-length or variable length.
- ✓ Its values can be indexed or not.
- ✓ How MySQL performs a comparison of values of a particular data type.

MySQL supports number of different data types that can be broken into the following categories: numeric, date and time, string types, spatial types, and **JSON** data types.

### ➤ Numeric Data Type

These data types can include the exact numeric data types like integer, decimal, numeric, etc. as well as the approximate numeric data types like float, real, and double precision. It also supports bit datatype to store bit values. In MySQL, numeric data types are categories into two types, either signed or unsigned except for bit data type.

| Data Type    | Description   |
|--------------|---|
| int          | INT is used for storing exact numbers. It is a normal-sized integer that can be signed or unsigned. We can specify a width of up to 11 digits. It requires 4 bytes for storage. There are five INT types- TINYINT, INT, SMALLINT, MEDIUMINT, and BIGINT (the range of TINYINT is the smallest and of BIGINT is the maximum).                                      |
| float(m,d)   | The length of digits is specified using <i>the m</i> parameter. The number of digits after the decimal point is specified in the <i>d</i> parameter. Decimal precision can go to 24 places for a float type. It requires 2 bytes for storage. Ex: FLOAT (3,2) can store a number 3.12, here the size of the number is 3 and there are 2 digits after the decimal. |
| double(m,d)  | The length of digits is specified using <i>the m</i> parameter. The number of digits after the decimal point is specified in the <i>d</i> parameter. Decimal precision can go to 53 places for a double. Real is a substitute for double. It requires 8 bytes for storage. Ex: 502.00232  |
| decimal(m,d) | DECIMAL is used for columns that preserve exact precision e.g., money data in accounting systems. The length of digits is specified using <i>the m</i> parameter. The number of digits after the decimal point is specified in the <i>d</i> parameter. Numeric is a synonym for decimal. Ex: 702. 23  |
| bit(m)       | BIT is used to store bit values i.e a number containing 0's and 1's only. The number of bits per value is specified using the <i>size</i> parameter. The <i>size</i> parameter can hold a value from 1 to 64. The default value for <i>size</i> is 1. Ex: 1101  |
| boolean      | It is used only for the true and false condition. It considered numeric value 1 as true and 0 as false.   |

### ➤ Date and Time Data Type:

This data type is used to represent values such as date, time, datetime, timestamp, and year. Each type contains values, including zero. When we insert the invalid value, MySQL cannot represent it, and then zero value is used.

| Data Type   | Description  |
|-------------|--|
| year[(2 4)] | YEAR can be used for storing a year in a four-digit format. Values are allowed in the four-digit format: 1901 to 2155, and 0000. MySQL converts Year values 2 digits to 4 digits. It takes 1 byte for storage. Ex: '70' to '99' to 1970 to 1999. |
| date        | Managing dates. It can store the date in the YYYY-MM-DD format. Dates within the range from '1000-01-01' to '9999-12-31' can be stored. It takes 3 bytes for storage. Example: 2020-10-24  |
| time        | TIME is used for storing the time of a day. The time can be stored in the hh:mm:ss format only. You can enter time within the range from '-838:59:59' to '838:59:59'. It takes 3 bytes plus fractional seconds for storage. Example: 09:00:00    |

|              |   |
|--------------|---|
| datetime     | It contains a combination of date and time. The format for storing data and time is YYYY-MM-DD hh:mm:ss. Dates within the range from '1000-01-01 00:00:00' to '9999-12-31 23:59:59' can be stored. You can add DEFAULT and ON UPDATE in the column definition to get automatic initialization and updating to the current date and time. Values range from '1000-01-01 00:00:00' to '9999-12-31 23:59:59'. It takes 5 bytes plus fractional seconds for storage. Example: 2020-10-24 11:55:40 |
| timestamp(m) | Values range from '1970-01-01 00:00:01' UTC to '2038-01-19 03:14:07' TC. It takes 4 bytes plus fractional seconds for storage. Example: 2020-10-24 11:55:40   |

### ➤ String Data Types:

The string data type is used to hold plain text and binary data like files, images, etc. MySQL can perform searching and comparison of string value based on the pattern matching such as LIKE operator, Regular Expressions, etc.

| data type     | Description   |
|---------------|---|
| char(size)    | It can hold a fixed-length string. Maximum size of 255 characters.  |
| varchar(size) | Stands for VARIABLE CHARACTER. It can hold a variable-length string alphabets, number, or special characters). The range of characters can be between 0 and 65,535.   |
| text(size)    | TEXT holds up to 65,535 bytes or 64KB of data. The TEXT is useful for storing long-form text strings such as articles, blogs, etc. It has more features than CHAR and VARCHAR. It can hold from 1 byte to 4 GB of data. |
| binary(size)  | Similar to CHAR(), but stores fixed-length binary byte strings. It does not contain any character set.  |

### ➤ Binary Large Object Data Types (BLOB):

BLOB is a **B**inary **L**arge **O**bject that can hold a variable amount of data. It can store binary data such as images, multimedia, and PDF files.

| data type  | Description                                 |
|------------|---|
| blob(size) | Maximum size of 65,535 bytes.               |
| longblob   | Maximum size of 4gb or 4,294,967,295 bytes. |

### ➤ Spatial Data Types

It is a special kind of data type which is used to hold various geometrical and geographical values. It corresponds to OpenGIS classes. The following table shows all spatial types that support in MySQL

| Data Types         | Description   |
|--------------------|---|
| geometry           | It is a point or aggregate of points that can hold spatial values of any type that has a location.  |
| point              | A point in geometry represents a single location. It stores the values of X, Y coordinates.   |
| polygon            | It is a planar surface that represents multisided geometry. It can be defined by zero or more interior boundary and only one exterior boundary. |
| linestring         | It is a curve that has one or more point values. If it contains only two points, it always represents Line.                                     |
| geometrycollection | It is a kind of geometry that has a collection of zero or more geometry values.   |
| multilinestring    | It is a multi-curve geometry that has a collection of linestring values.  |
| multipoint         | It is a collection of multiple point elements. Here, the point cannot be connected or ordered in any way.                                       |
| multipolygon       | It is a multisurface object that represents a collection of multiple polygon elements. It is a type of two-dimensional geometry.                |

### ➤ JSON Data Type

MySQL provides support for native JSON data type from the version v5.7.8. This data type allows us to store and access the JSON document quickly and efficiently. The JSON data type has the following advantages over storing JSON-format strings in a string column:

1. It provides automatic validation of JSON documents. If we stored invalid documents in JSON columns, it would produce an error.
2. It provides an optimal storage format.

## 🔊 Creating databases and show databases

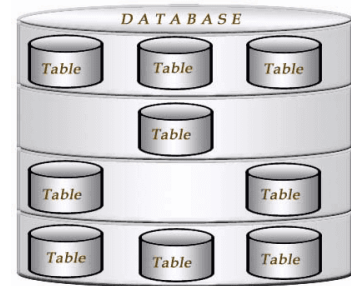
A database is used to store the collection of records in an organized form. It allows us to hold the data into tables, rows, columns, and indexes to find the relevant information frequently. We can access and manage the records through the database very easily.

**MySQL** implements a database as a directory that stores all files in the form of a table. It allows us to create a database mainly in **two ways**:

1. MySQL Command Line Client
2. MySQL Workbench

**Syntax: create database** [if not exists] database\_name;

**Ex: create database** employee;



### ➤ Selecting Database

SELECT Database is used in MySQL to select a particular database to work with. This query is used when multiple databases are available with MySQL Server. You can use SQL command **USE** to select a particular database.

**Syntax: use database name;**

**Ex: use** employee;

### ➤ Show/List Databases

SHOW statement displays a list of currently existing databases on the server.

**Syntax: Show databases;**

**Ex:**

```
MySQL 8.0 Command Line Client
mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| employee |
| information_schema |
| myemployee |
| mysql |
| mysqltestdb |
| mystudentdb |
| mytestdb_copy |
| performance_schema |
| sakila |
| sys |
| testdb |
| world |
+-----+
```

### ➤ Drop Database

We can drop/delete/remove a MySQL database quickly with the MySQL DROP DATABASE command. It will delete the database along with all the tables, indexes, and constraints permanently. Therefore, we should have to be very careful while removing the database in MySQL because we will lose all the data available in the database. If the database is not available in the MySQL server, the DROP DATABASE statement throws an error.

**Syntax: drop database** [if not exists] database\_name;

**Ex: drop database** employee;

### ➤ MySQL show tables statement

MySQL 'show tables' statement displays a list of the tables in the database in use. If there is no table in the database, it returns empty rows.

**Syntax: Show tables;**

### ➤ COPY Database

MySQL copy or clone database is a feature that allows us to create a duplicate copy of an existing database, including the table structure, indexes, constraints, default values, etc. Making a duplicate copy of an original database into a new database is very useful when accidentally our database is lost or failure. The most common use of making a duplicate copy of the database is for data backups. It is also useful when planning the major changes to the structure of the original database.

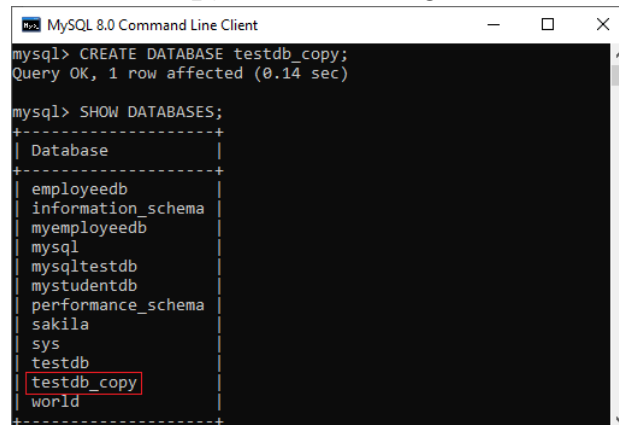
In **MySQL**, making the clone of an original database is a **three-step process**:

- 1) First, the original database records are dumped (copied) to a temporary file that holds the SQL commands for reinserting the data into the new database.
- 2) Second, it is required to create a new database.
- 3) Finally, the [SQL](#) file is processed, and the data will be copied into the new database.

We need to follow these steps to copy a database to another database:

1. First, use the **CREATE DATABASE** statement to create a new database.
2. Second, store the data to an **SQL file**. We can give any name to this file with a **.sql** extension.
3. Third, export all the database objects along with its data to copy using the **mysqldump** tool and then import this file into the new database.

**Example:** we will copy the **testdb** database to **testdb\_copy** database using the following steps-



```

mysql> CREATE DATABASE testdb_copy;
Query OK, 1 row affected (0.14 sec)

mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| employeedb |
| information_schema |
| myemployeedb |
| mysql |
| mysqltestdb |
| mystudentdb |
| performance_schema |
| sakila |
| sys |
| testdb |
| testdb_copy |
| world |
+-----+

```

**create database** testdb\_copy;  
**show databases;**

Now, open a DOS terminal window to access the MySQL server on the command line. For example, if we have installed the MySQL in the **C folder**, copy the following folder and paste it in our DOS command. Then, press the **Enter** key.

**C:\Users\ecs> CD C:\Program Files\MySQL\MySQL Server 8.0\bin**

In the next step, we need to use the **mysqldump** tool to copy the database objects and data into the SQL file. Suppose we want to dump (copy) the database objects and data of the testdb into an SQL file located at **D:\Database\_backup** folder.

**mysqldump -u root -p testdb > D:\Database\_backup\testdb.sql**

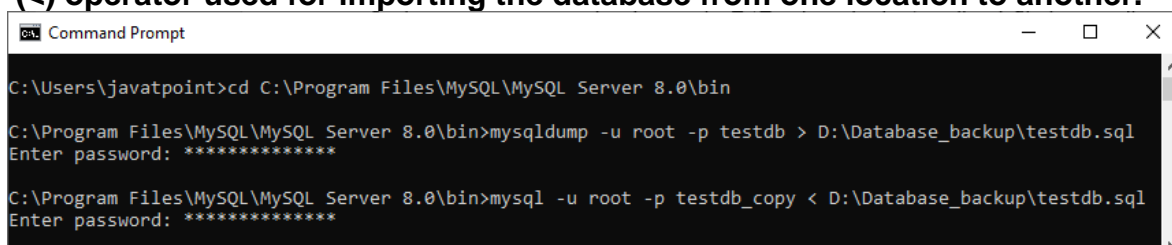
**Enter password: \*\*\*\*\***

The above statement exports the database objects and data of the testdb database to **D:\Database\_backup\testdb.sql** (**>**) operator used for exporting the database from one location to another. In the next step, we need to import the **D:\Database\_backup\testdb.sql** file into testdb\_copy database.

**mysql -u root -p testdb\_copy < D:\Database\_backup\testdb.sql**

**Enter password: \*\*\*\*\***

**(<) operator used for importing the database from one location to another.**



```

C:\Users\javatpoint>cd C:\Program Files\MySQL\MySQL Server 8.0\bin

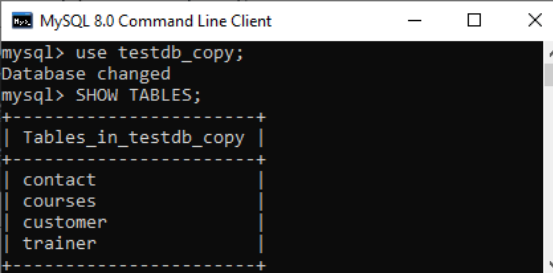
C:\Program Files\MySQL\MySQL Server 8.0\bin>mysqldump -u root -p testdb > D:\Database_backup\testdb.sql
Enter password: *****

C:\Program Files\MySQL\MySQL Server 8.0\bin>mysql -u root -p testdb_copy < D:\Database_backup\testdb.sql
Enter password: *****

```

Finally, we can verify whether the above operation is successful or not by using the **SHOW TABLES** command.

**SHOW TABLES;**



```
mysql> use testdb_copy;
Database changed
mysql> SHOW TABLES;
+-----+
| Tables_in_testdb_copy |
+-----+
| contact                |
| courses                |
| customer               |
| trainer                |
+-----+
```

all the objects and data from the testdb database to testdb\_copy database have successfully copied.