

MySQL Installation and Usage on Windows 10

Author: Sri Adilakshmi M

Table of Contents

1.	Overview:	4
2.	Install MySQL Server:	5
3.	Set up Environment Variables:	20
4.	Verify MySQL:	23
5.	Connect MySQL Server:	23
6.	Run MySQL Commands:	25
6.1.	Create Database:	25
6.2.	Select Database:	25
6.3.	Create Table:	26
6.4.	Show Columns:	26
6.5.	Insert Data:	27
6.6.	Select Data:	27
6.7.	Drop Table:	28
6.8.	Drop Database:	28
7.	MySQL Server CLI Utilities:	29
7.1.	mysql:	29
7.2.	mysqlshow:	35
7.3.	mysqladmin:	37
7.4.	mysqldump:	40
7.5.	mysqlimport:	47
7.6.	mysqlcheck:	51
8.	MySQL Shell:	54
8.1.	Verify MySQL Shell Installation:	54
8.2.	Start MySQL Shell:	55
8.3.	MySQL Shell Commands:	56
8.4.	Connect to MySQL Instance:	58
8.5.	Run SQL Commands:	58
8.6.	Run Python Code:	59
8.7.	Run JavaScript Code:	61

8.8. MySQL Shell Utilities:	62
8.8.1. Dump Utilities:	62
8.8.2. Load Utility:	66
8.8.3. Table Export Utility:	67
8.8.4. Parallel Table Import Utility:	68
8.8.5. JSON Import Utility:	69
8.8.6. Copy Utilities:	70
8.8.7. Upgrade Checker Utility:	71
8.8.8. Diagnostic Utilities:	71
9. MySQL Workbench:	73
9.1. Create Database:	76
9.2. Create Table:	78
9.3. Insert Data:	85
9.4. Select Data:	89
9.5. Drop Table:	90
9.6. Drop Database:	92
9.7. Export Table:	93
9.8. Import Table:	96
9.9. Export Database:	101
9.10. Import Database:	104
9.11. Create User:	107

This document outlines the steps needed to install MySQL and access various MySQL utilities on Windows Operating system.

1. Overview:

MySQL is the most popular open-source relational database management system (RDBMS) that uses Structured Query Language (SQL) for managing and manipulating data. It is known for its speed, reliability, scalability and ease of use, widely used in global services and major web applications like Facebook, Netflix, Uber, etc.

MySQL is named after co-founder Monty Widenius's daughter, **My** and initially developed by the Swedish company named **MySQL AB** which was bought by **Sun Microsystems**. Later, in 2010, when Oracle Corporation acquired Sun Microsystems, the MySQL is being sponsored by **Oracle** since then.

MySQL works on a client-server architecture such that a **client** (a web application or command line tool) sends a request to the **MySQL server** which processes the request, optimizes the query, interacts with the storage engine to read data from disk and generates result after which the server sends the results back to the client application.

MySQL is a popular choice due to its key features:

- MySQL is free and open source allowing modification of MySQL software under GPL (General Public License).
- It uses a standard programming language for database called **SQL**, allowing users to work on data effectively.
- It is written in C and C++ language and runs on multiple operating systems including Windows, Linux, and macOS, and works with various programming languages such as Java, PHP, Perl, Python, etc.
- It provides transactional and non-transactional storage engines like InnoDB (default), MyISAM for flexible use.
- When using the InnoDB storage engine, it supports full ACID (Atomicity, Consistency, Isolation, Durability) properties to ensure data integrity, reliability and accurate data processing.
- It handles high traffic and supports large databases (*more than 50 million records in a table with default file size limit of 4 GB which can be extended to 8 million TB, supporting 64 indexes per table with each index consists of up to 16 columns*) using features like partitioning, replication and clustering to ensure high performance and availability.
- It includes several client programs including **MySQL Workbench** GUI, and command-line utilities like mysql, mysqlsh, mysqladmin, mysqldump, mysqlimport, mysqlshow, mysqlcheck, etc.

Common use cases of MySQL include:

- **Web Applications:** MySQL is widely used as backend in many websites and online services to manage user data, content and session information.
- **Content Management Systems (CMS):** MySQL is a core component of the popular LAMP (Linux, Apache, MySQL, PHP/Python/Perl) software stack, forming the backend for content management systems like WordPress, Joomla, and Drupal.
- **E-commerce & Social Platforms:** Large-scale transactional systems and social networks rely on MySQL for managing product catalogs, customer details, user profiles, content, and financial data.
- **Data Analytics & Reporting:** Organizations use MySQL to store and analyze large datasets to gain business insights.

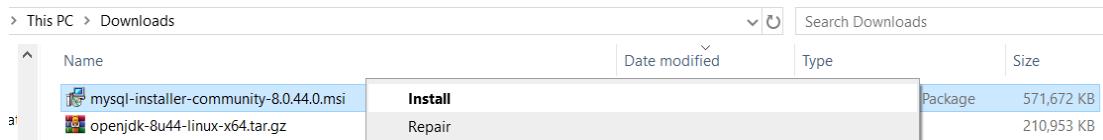
Considering the heavy usage of MySQL, many cloud providers offer managed services of MySQL such as **Amazon RDS** (offered by Amazon Web Services), **Google Cloud SQL** (offered by Google Cloud Platform), **Azure Database** (offered by Microsoft cloud), **MySQL Heatwave** (offered by Oracle cloud) etc. to simply database management by automating patching, backups and scaling.

2. Install MySQL Server:

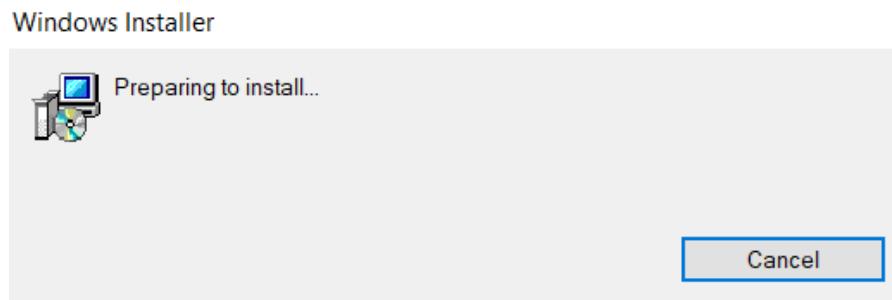
Go to [MySQL Installer Downloads](https://dev.mysql.com/downloads/installer/) website, select the latest version (*at the time of this document writing, the latest version was 8.0.44*) and click on **Download** button to download the MSI Installer community file which gets saved into your **Downloads** folder.

The screenshot shows the MySQL Community Downloads page. At the top, there's a navigation bar with links for 'MySQL Community Downloads' and 'MySQL Installer'. Below that is a search bar and a 'General Availability (GA) Releases' tab, which is currently selected. The main content area displays the 'MySQL Installer 8.0.44' page. It includes a note about MySQL 8.0 being the final series. There are dropdown menus for 'Select Version' (set to 8.0.44) and 'Select Operating System' (set to Microsoft Windows). Two download options are listed: 'Windows (x86, 32-bit), MSI Installer' and 'Windows (x86, 32-bit), MSI Installer'. The second option is highlighted with a red border. Both entries show the version (8.0.44), file size (2.1M or 558.3M), and a 'Download' button. Below the download buttons is a note about verifying package integrity using MD5 checksums and GnuPG signatures.

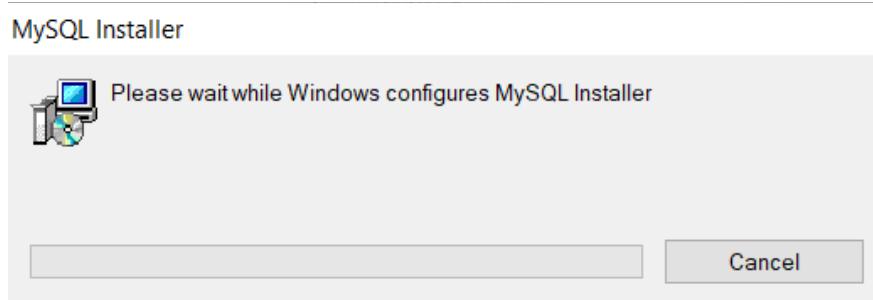
After the file is downloaded, go to your **Downloads** folder and right click on mysql-installer-community-* .msi and select **Install** option.



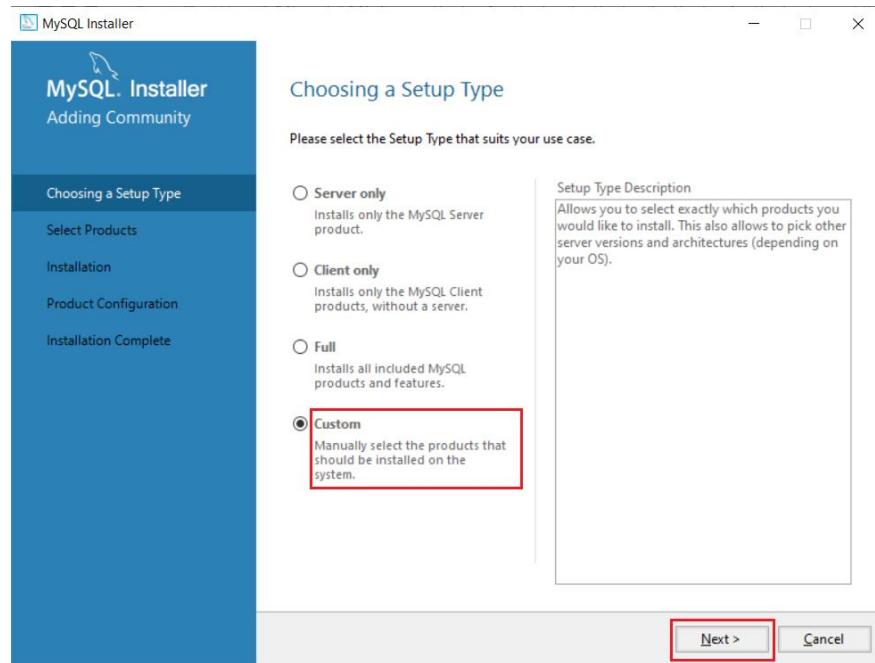
It starts with preparing for installation.



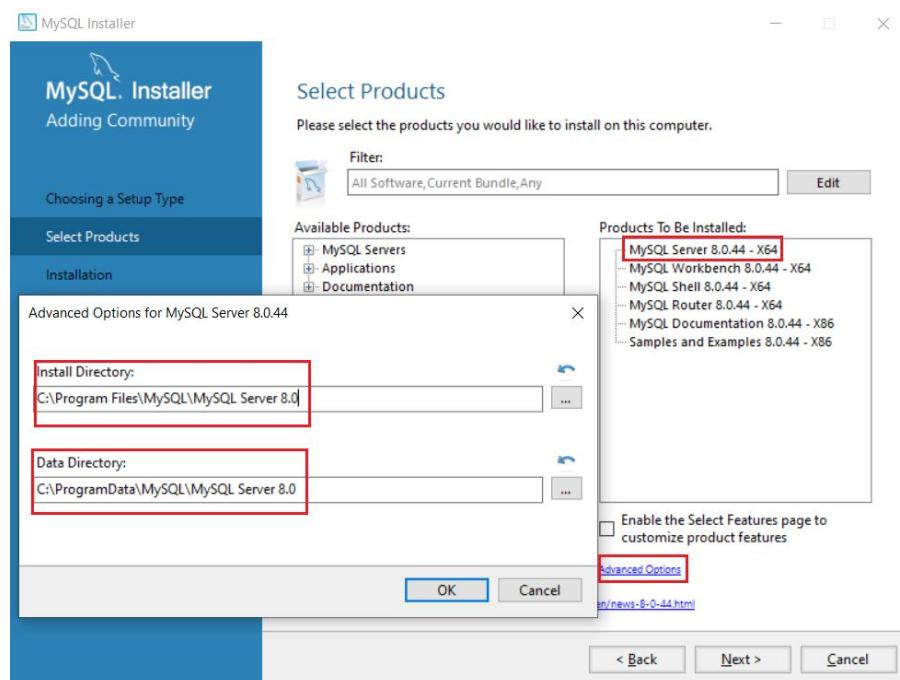
Then it configures MySQL Installer to open up.



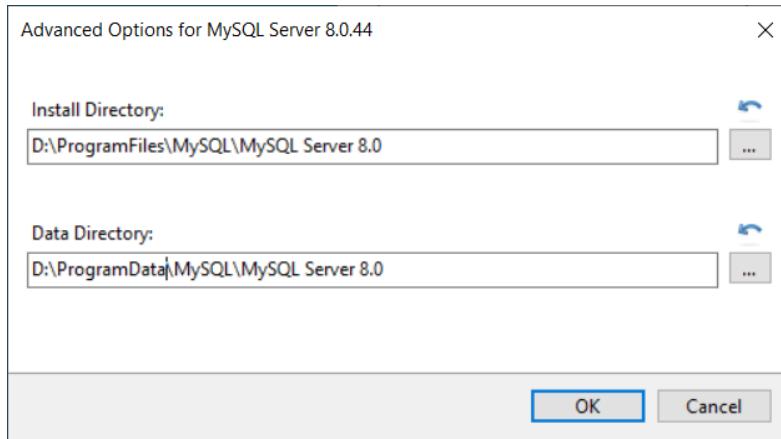
On the **MySQL Installer** wizard, choose the **Setup Type** as **Custom** to install all MySQL products. Then press **Next** to continue.



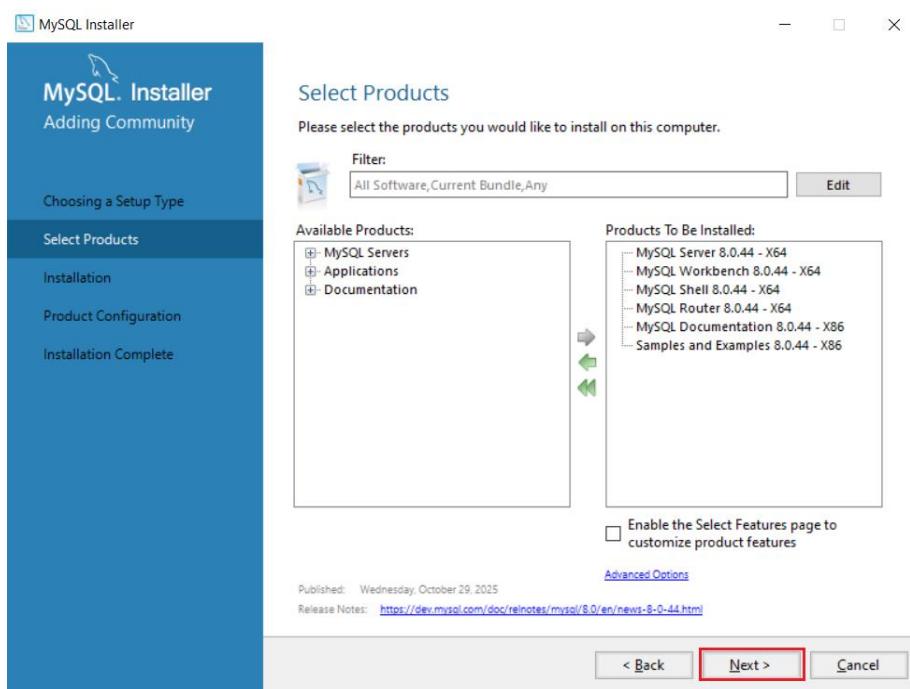
On the **Select Products** tab, it displays the list of products like MySQL Server, MySQL Workbench, MySQL Shell, etc. to be installed. Select **MySQL Server** product and click on **Advanced options** where you see the default **Install Directory** pointing to C:\Program Files\MySQL\MySQL Server 8.0 and **Data Directory** pointing to C:\ProgramData\MySQL\MySQL Server 8.0.



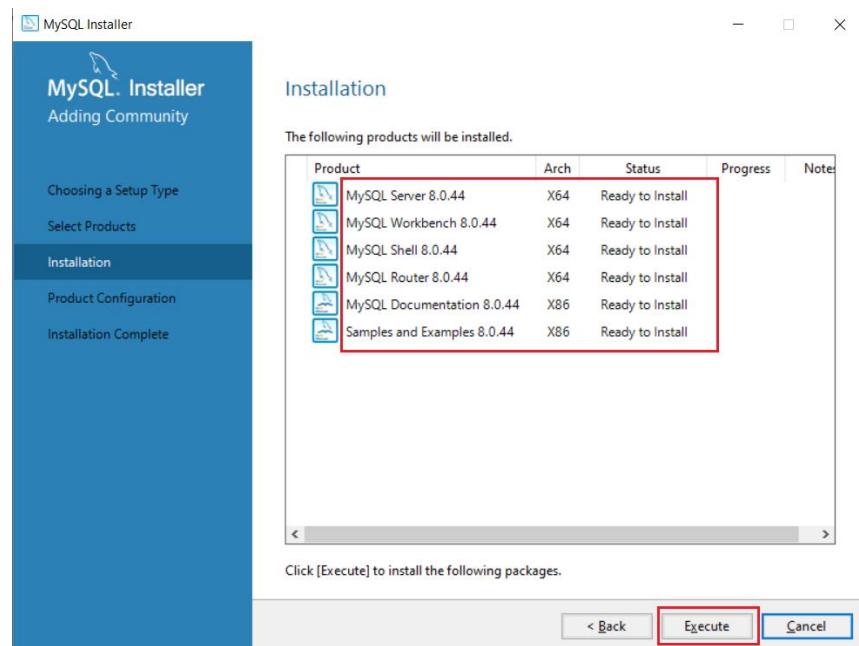
Change the default paths if needed. Here, I am changing from C:\Program Files to D:\ProgramFiles and C:\ProgramData to D:\ProgramData path to install in D drive instead of C drive.



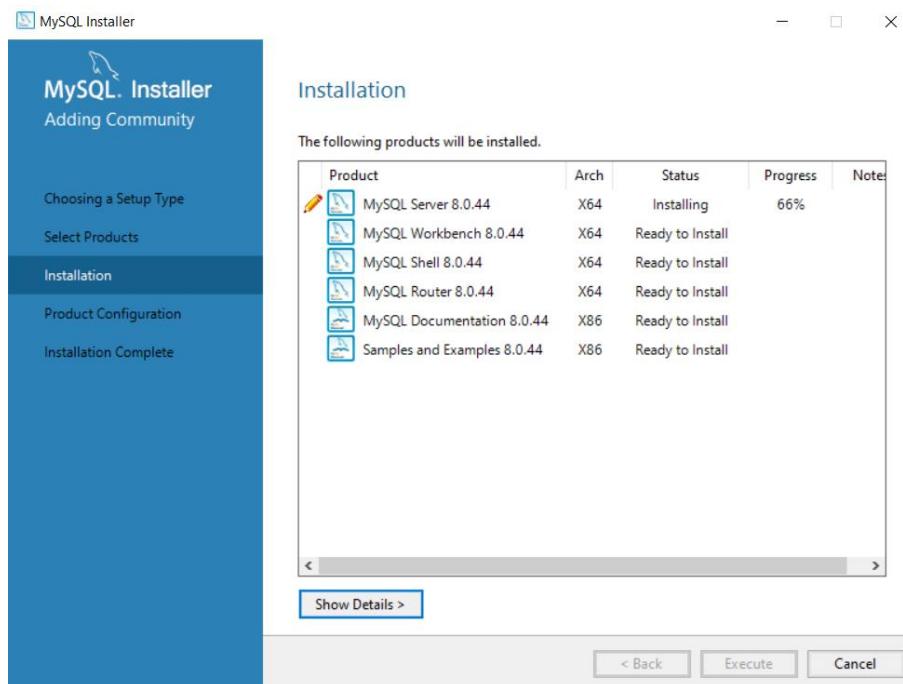
Similarly, select other products like MySQL Workbench, MySQL Shell, MySQL Router, MySQL Documentation, Samples and Examples, and click on **Advanced Options** to change the install directory if needed. Then, press on **Next** to continue.



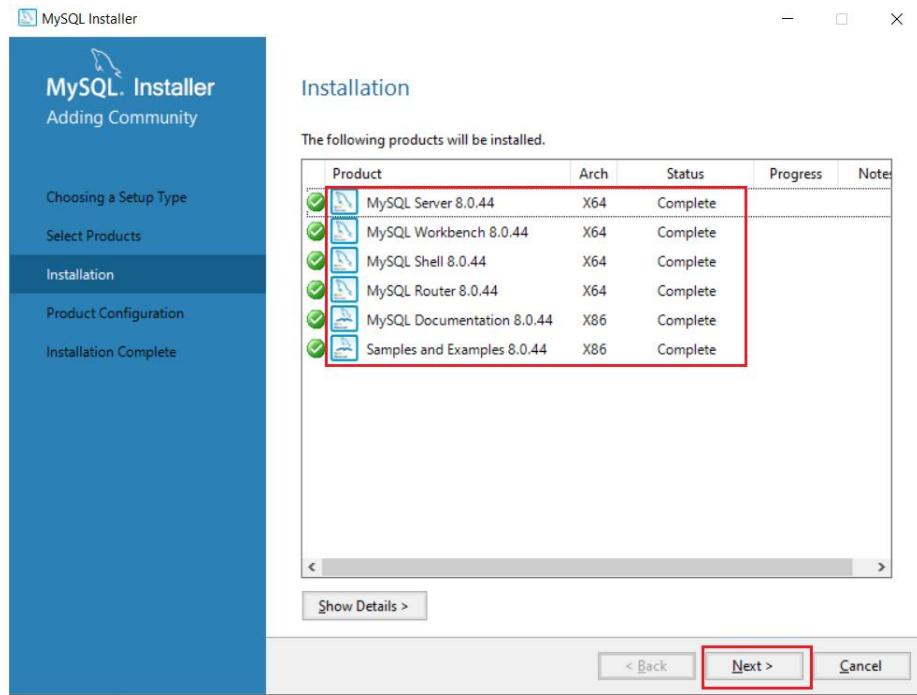
On the **Installation** tab, it displays the list of products to be installed along with their status (*which is set to Ready to install initially*). Press **Execute** to start the installation.



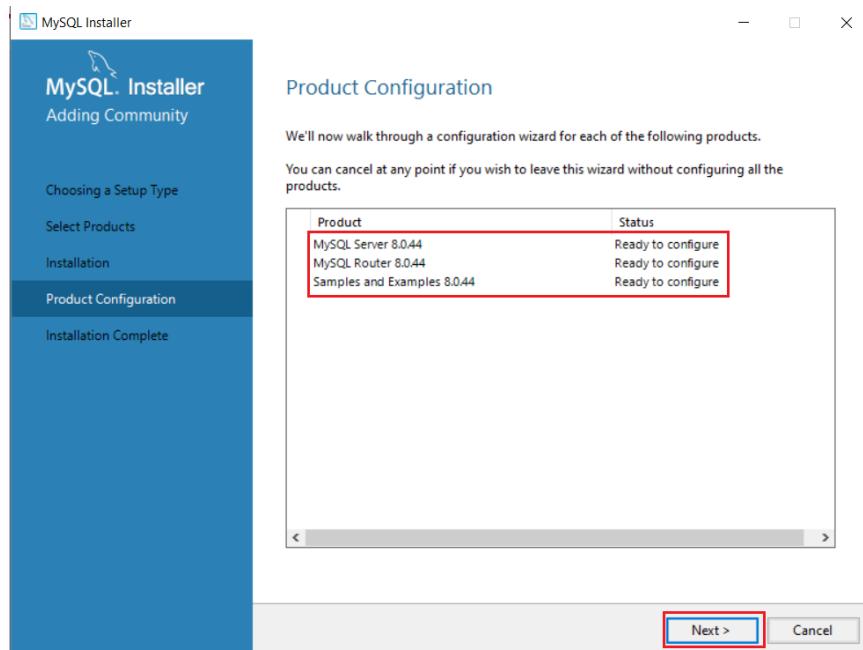
It begins with the installation of **MySQL Server**.



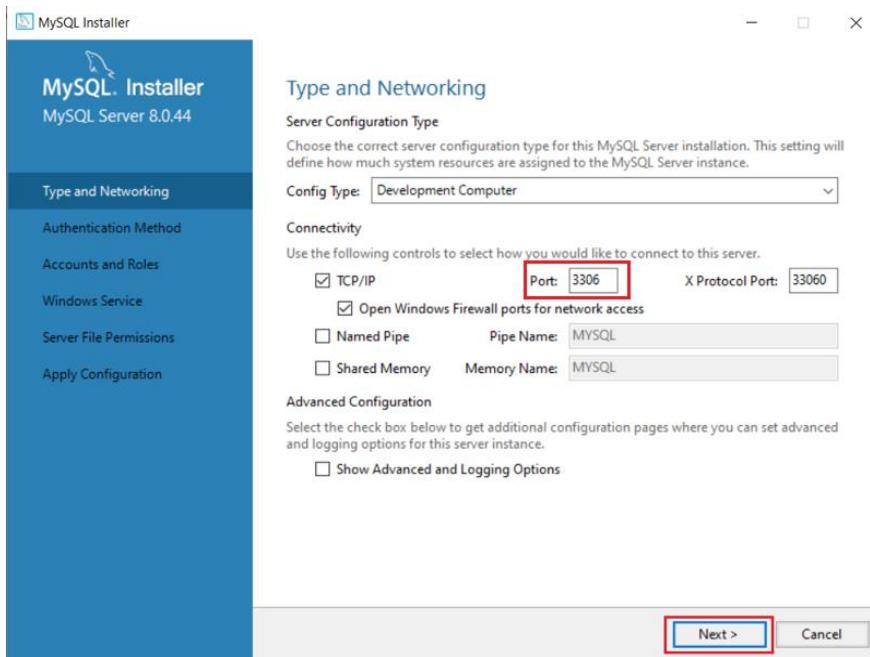
In few minutes, status of all products changes to **Complete**. Press **Next** to continue.



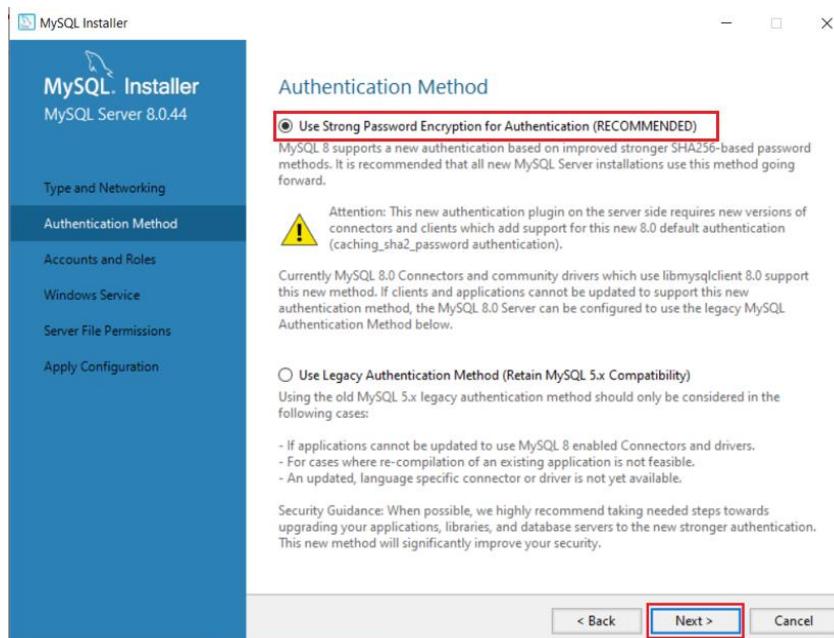
On the **Product Configuration** tab, it displays the list of products to be configured along with their status (*which is set to Ready to configure initially*). Press **Next** to continue with MySQL Server configuration.



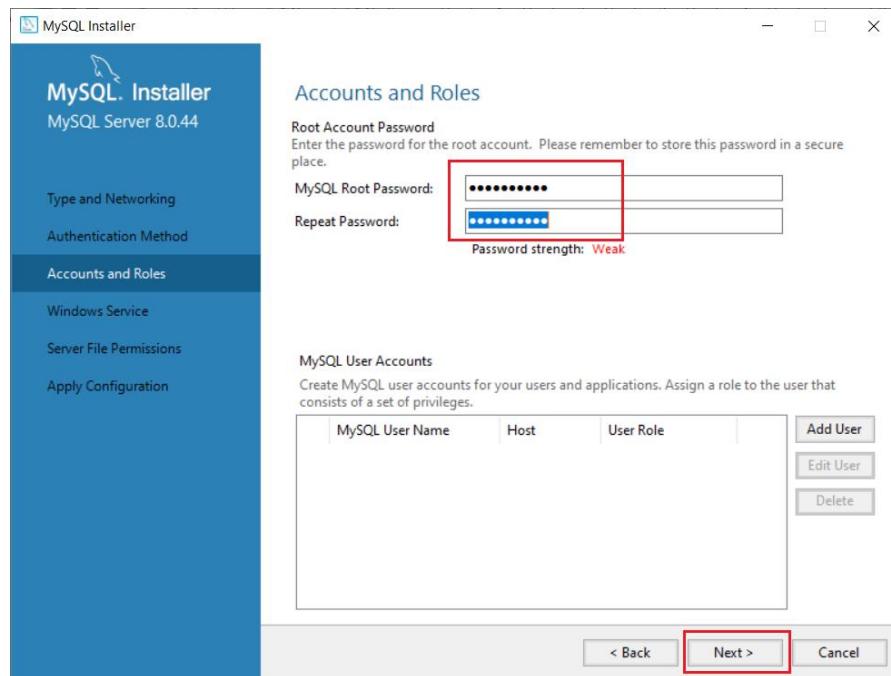
On the **Type and Networking** tab, it displays the network port for each configuration type (the default port for MySQL server is **3306**). Accept the default values or change port number as needed and press **Next** to continue.



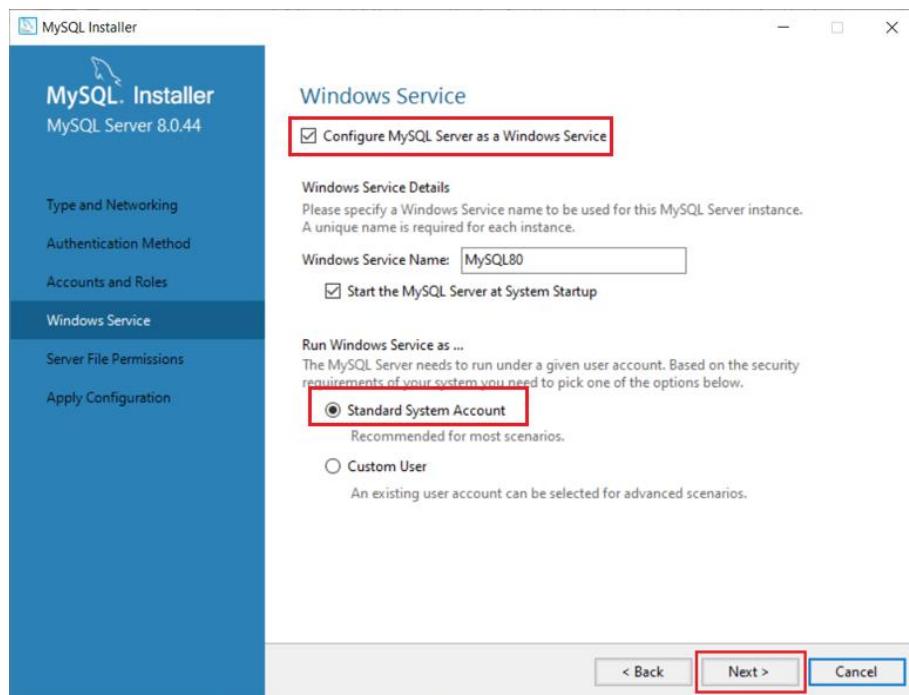
On the **Authentication Method** tab, chose the type of authentication or accept the default method which is **Use Strong Password Encryption for Authentication** and press **Next** to continue.



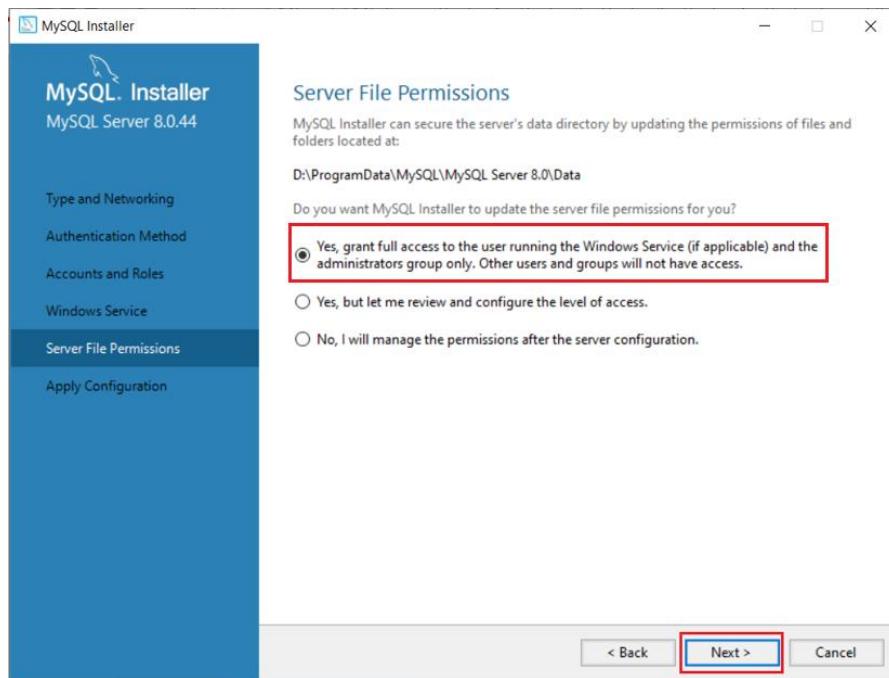
On the **Accounts and Roles** tab, provide **MySQL Root Password**. If needed, provide MySQL user accounts to be created at the time of installation (for now leave this) and press **Next** to continue.



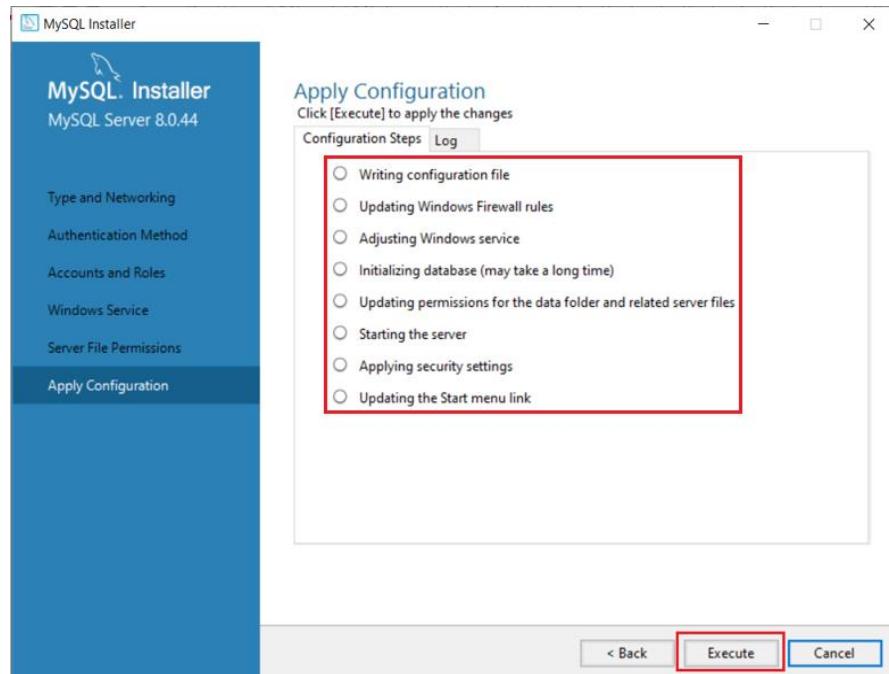
On the **Windows Service** tab, choose the option to **Configure MySQL Server as a Windows Service** and press **Next** to continue.



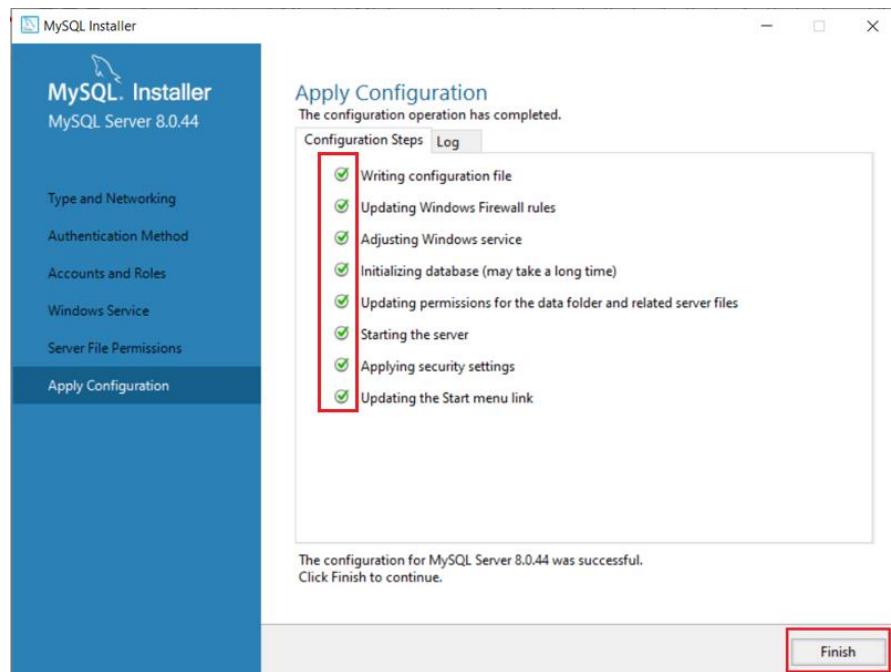
On the **Server File Permissions** tab, select the permissions type or access the default option which is **grant full access to the user running the Windows service** and press **Next** to continue.



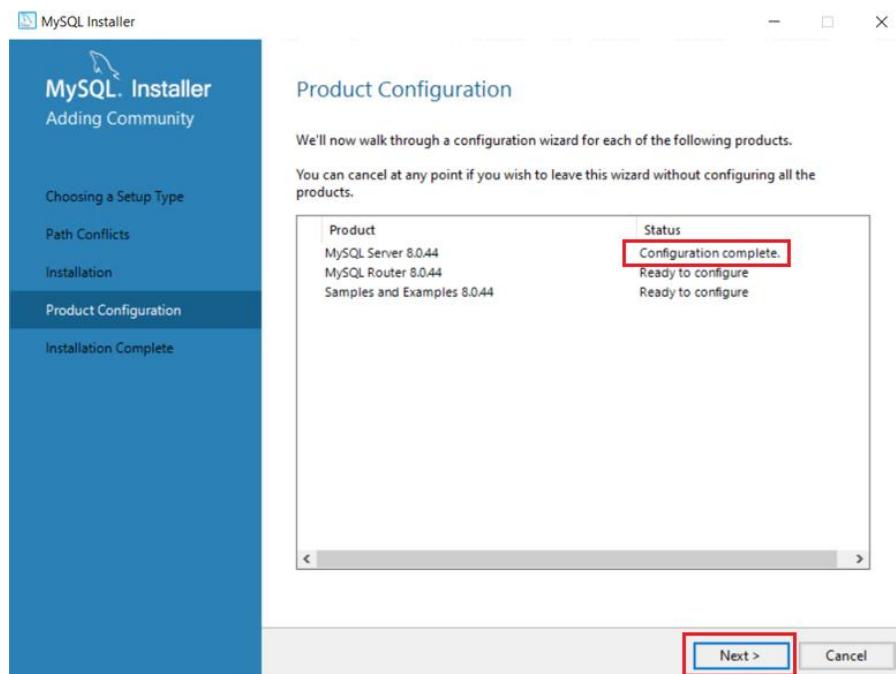
On the **Apply Configuration** tab, review the list of configuration steps that it follows and press **Execute** to continue.



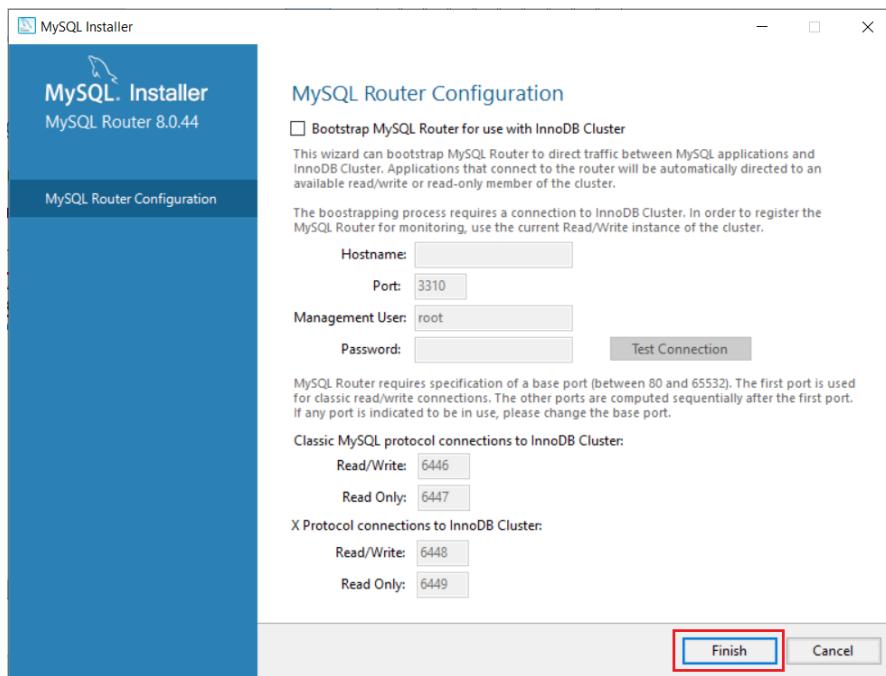
In few minutes, the status of all configuration steps changes to green which indicates the successful configuration. Click **Finish** to continue.



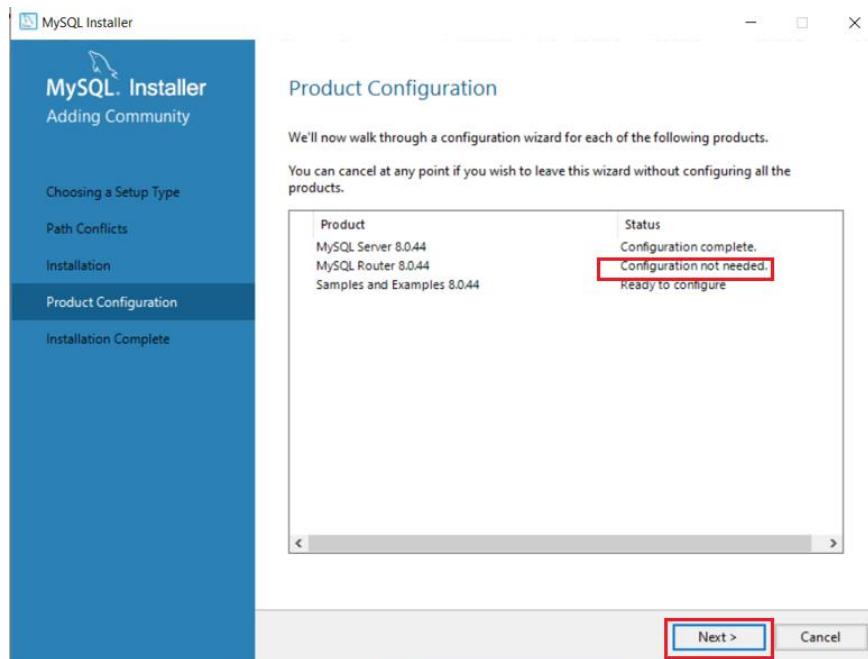
It navigates back to **Production Configuration** tab where you see the status of **MySQL Server** changes to **Configuration complete**. Click **Next** to continue with **MySQL Router** configuration.



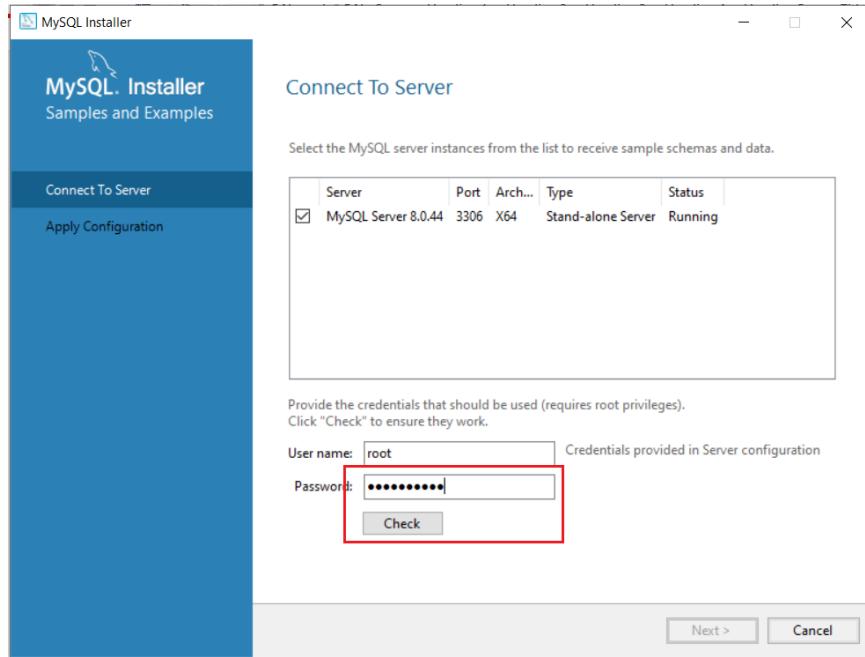
On the **MySQL Router Configuration** tab, it displays the default configuration to connect to InnoDB. Click **Finish** to continue.



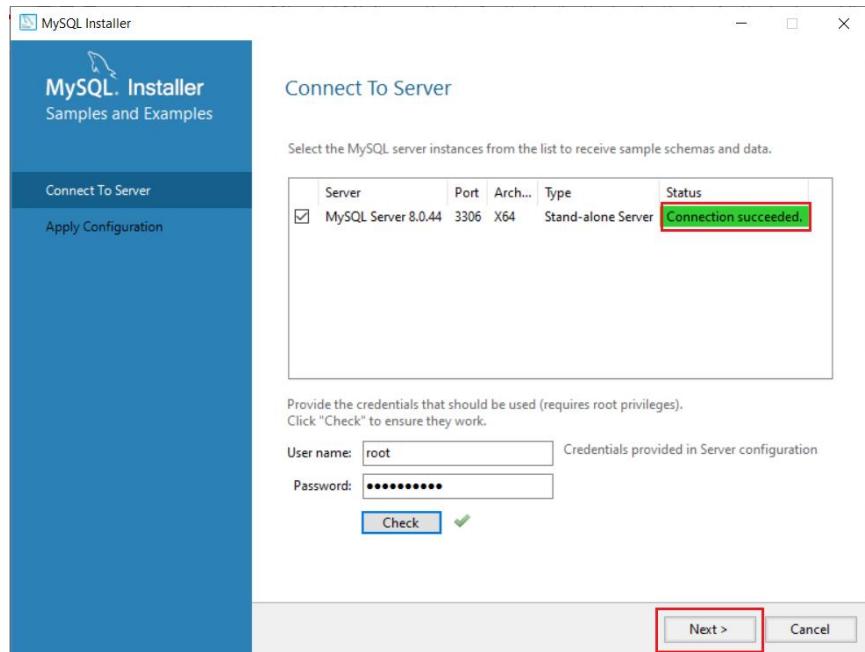
It navigates back to **Production Configuration** tab where you see the status of **MySQL Router** changes to **Configuration not needed**. Click **Next** to continue with **Samples and Examples** configuration.



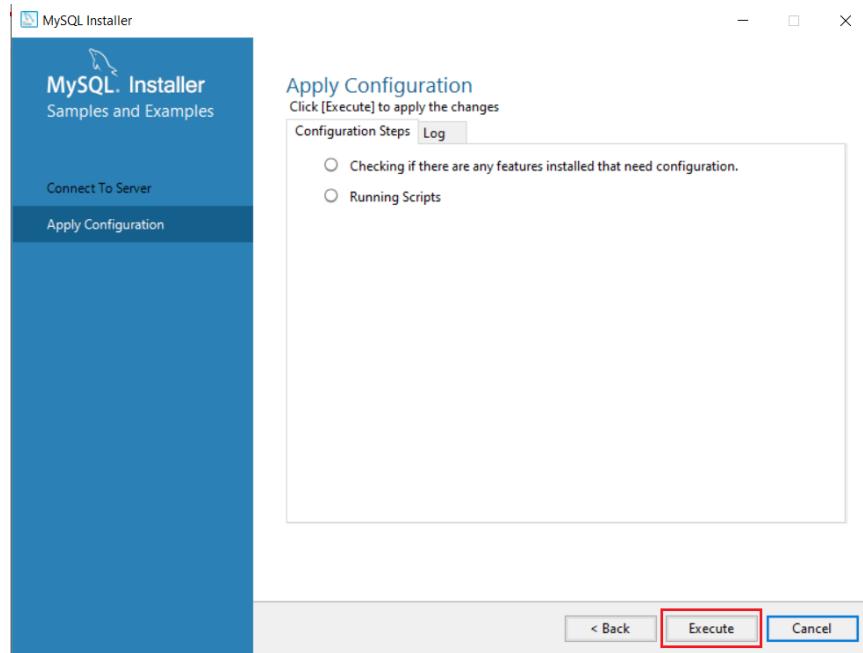
On the **Connect To Server** tab, provide the root password and click on **Check** button to verify the connectivity.



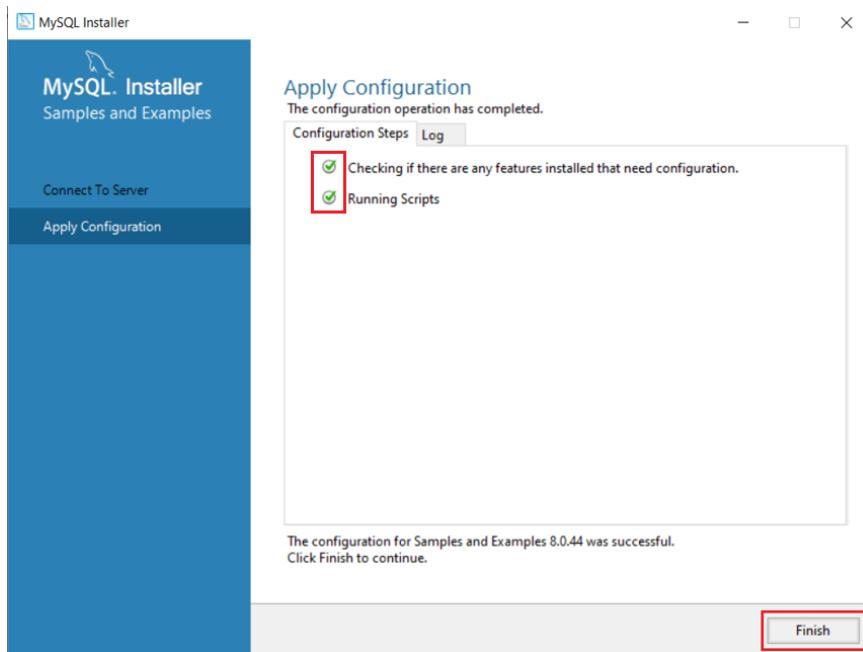
Once the connection is successful, click **Next** to continue.



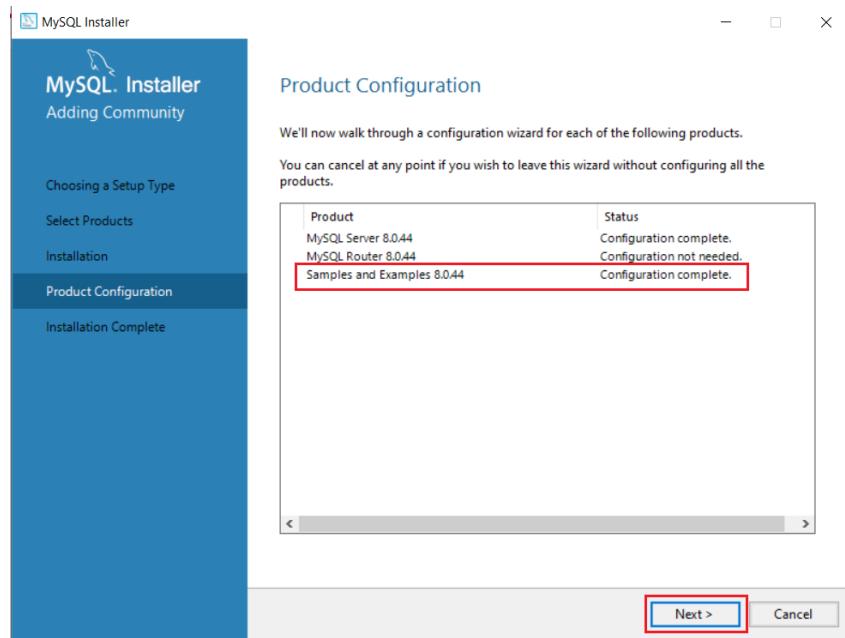
On the **Apply Configuration** tab, review the configuration steps and click on **Execute**.



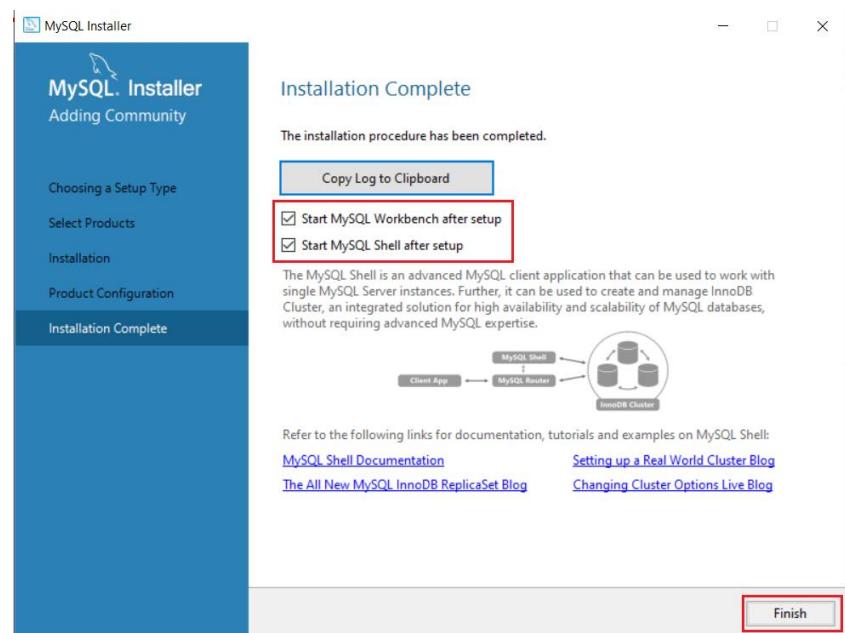
In few minutes, the status of all configuration steps changes to green which indicates the successful configuration. Click **Finish** to continue.



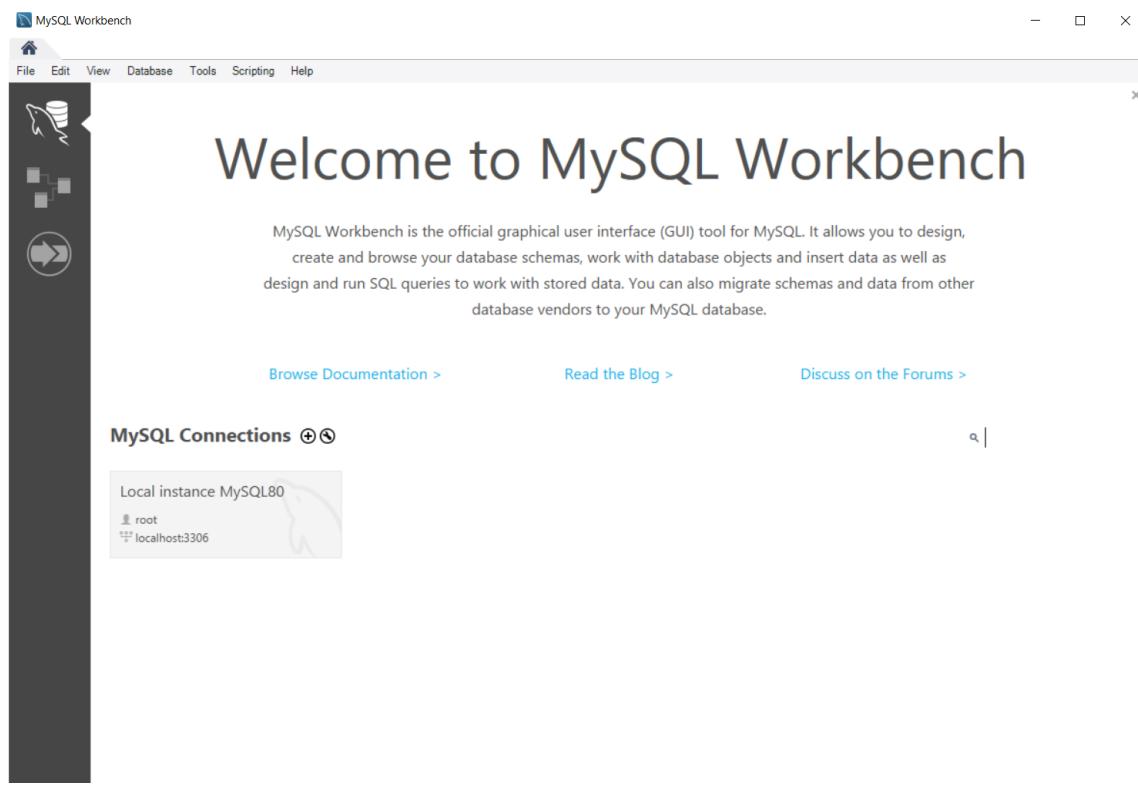
It navigates back to **Production Configuration** tab where you see the status of **Samples and Examples** changes to **Configuration complete**. Click **Next** to continue.



Then, it displays **Installation Complete** message. By default, it starts **MySQL Workbench** and **MySQL Shell** after setup. Click **Finish** to exit the wizard.



Once finished, it opens up **MySQL Workbench** and **MySQL Shell** applications parallelly

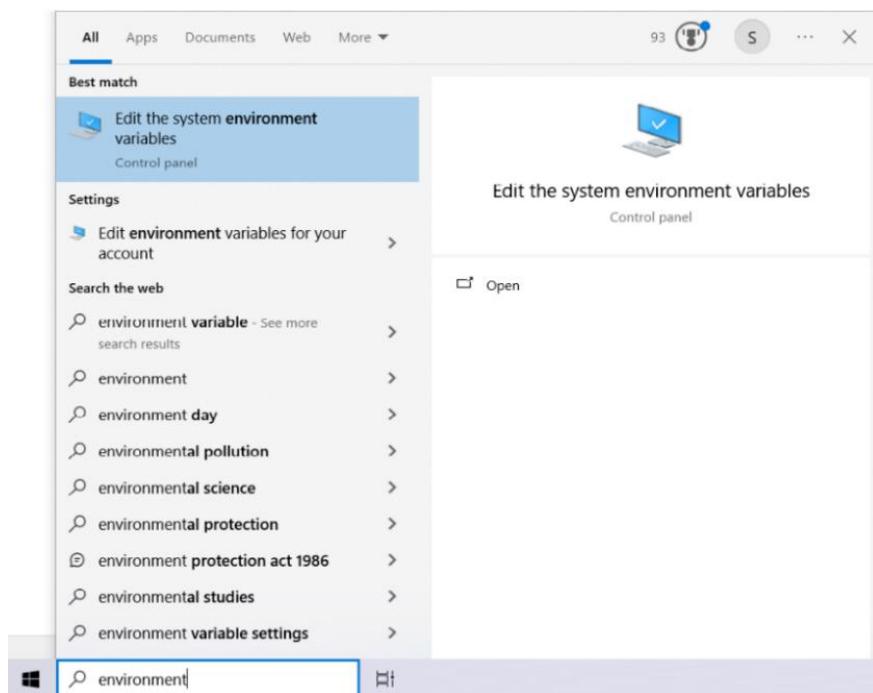
A screenshot of the MySQL Shell terminal window. The title bar says "D:\ProgramFiles\MySQL\MySQL Shell 8.0\bin\mysqlsh.exe". The MySQL Shell version is 8.0.44. The window displays the copyright notice from Oracle, the help command prompt ("Type '\help' or '\?' for help; '\quit' to exit."), and the MySQL and JS command line indicators.

3. Set up Environment Variables:

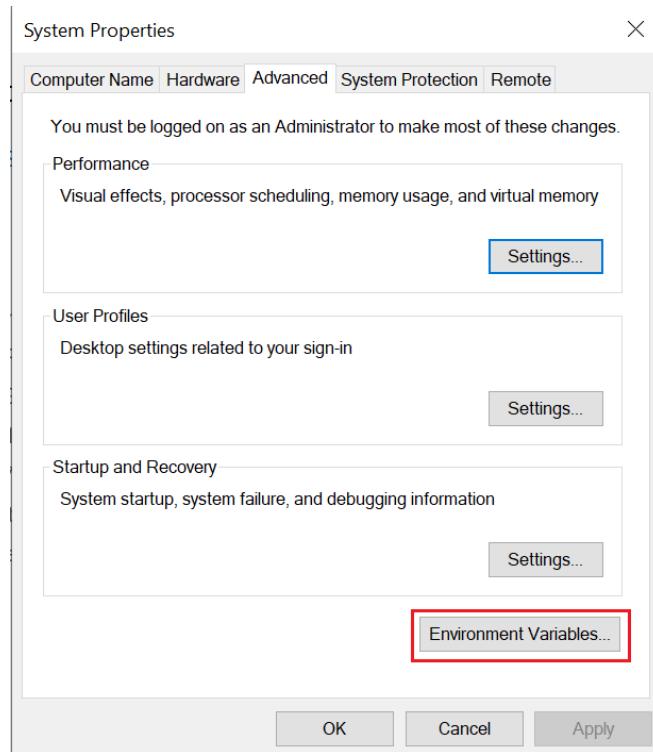
After installing MySQL Server, we should configure the PATH environment variable defining MySQL installation path. This variable must be configured under **User environment variables** or **System environment variables** depending on MySQL configuration needed **for a single user** or **for multiple users**.

Here, we will add User environment variables since we are configuring MySQL for a single user. If you would like to configure it for multiple users, then define System environment variables.

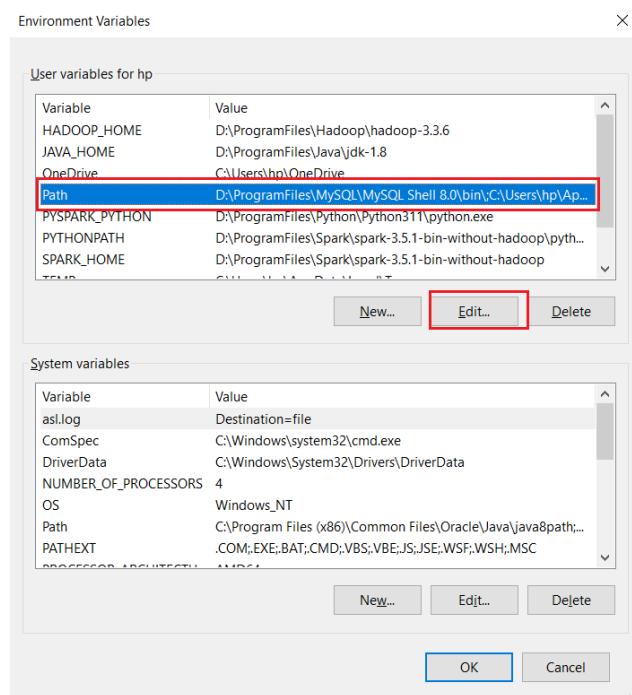
In the Windows search bar, start typing “environment variables” and select the first match which opens up **System Properties** dialog.



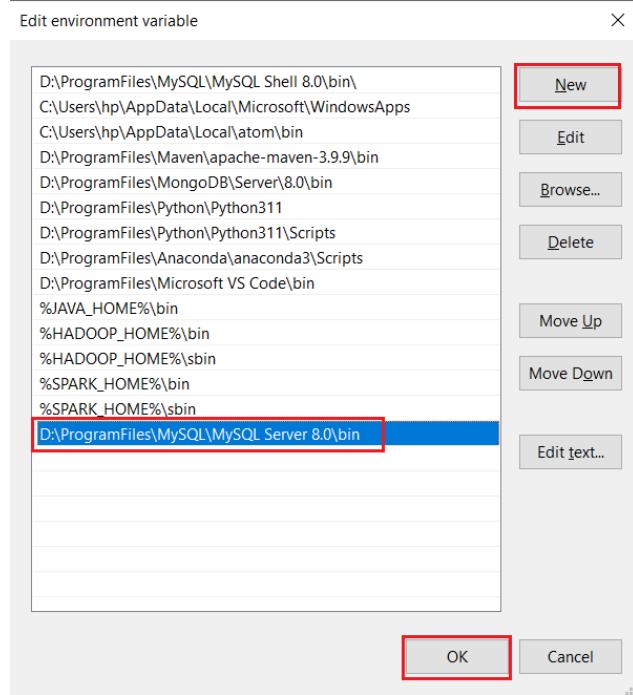
On the **System Properties** window, press **Environment Variables** button.



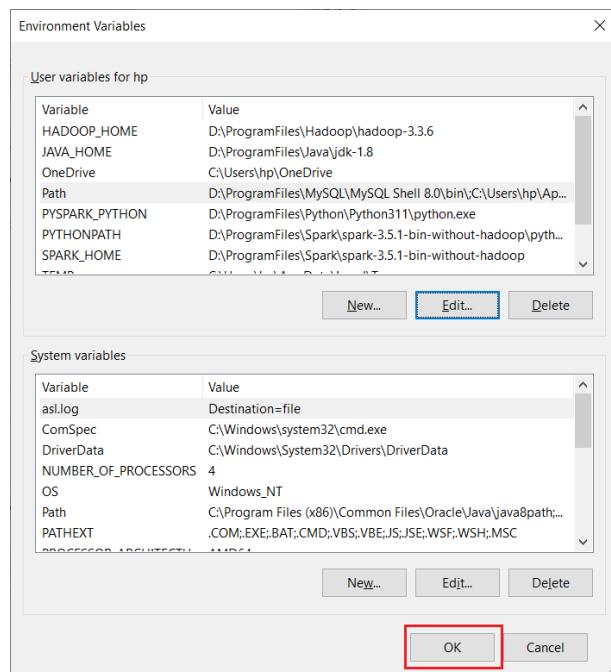
On the **Environment Variables** dialog, select **PATH** variable under **User variables** and press **Edit** button.



Press **New** and add your MySQL Server installation path (*in my case it is D:\ProgramFiles\MySQL\MySQL Server 8.0\bin*) and press **OK**.



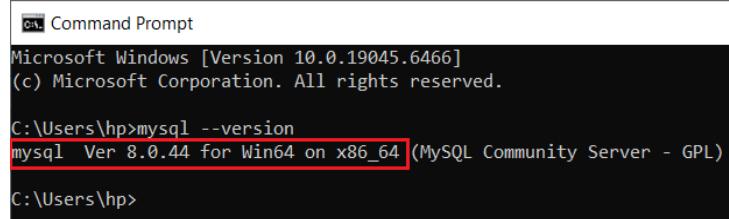
Press **OK** to apply environment variable changes and close window.



4. Verify MySQL:

Open new **Command Prompt** and run the following command to get the installed MySQL version:

```
mysql --version
```



The screenshot shows a Windows Command Prompt window titled "Command Prompt". The output of the command "mysql --version" is displayed, showing the MySQL version as "Ver 8.0.44 for Win64 on x86_64 (MySQL Community Server - GPL)". The entire line of output is highlighted with a red box.

```
Microsoft Windows [Version 10.0.19045.6466]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hp>mysql --version
mysql Ver 8.0.44 for Win64 on x86_64 (MySQL Community Server - GPL)

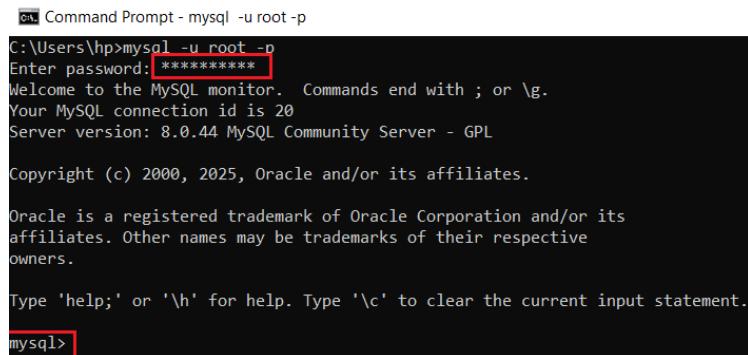
C:\Users\hp>
```

Here, it displays the mysql version 8.0.44 version that was installed on the Windows system.

5. Connect MySQL Server:

Run the following command to connect to MySQL server and provide the root password (*given at the time of installation*) when prompted:

```
mysql -u root -p
```



The screenshot shows a Windows Command Prompt window titled "Command Prompt - mysql -u root -p". It starts with the command "mysql -u root -p" followed by the prompt "Enter password: ****". The password is masked with asterisks. The MySQL monitor then displays its welcome message, connection ID, server version, copyright information, and a note about Oracle trademarks. Finally, it shows the standard help and clearing statements, ending with the MySQL prompt "mysql>".

```
C:\Users\hp>mysql -u root -p
Enter password: ****
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 20
Server version: 8.0.44 MySQL Community Server - GPL

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

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>
```

Once it is successfully connected, it displays `mysql>` prompt where all SQL commands can be executed.

For example, run the following command to get the current MySQL version:

```
select version();
```

```
mysql> select version();
+-----+
| version() |
+-----+
| 8.0.44   |
+-----+
1 row in set (0.00 sec)

mysql>
```

Run the following command to get the current user that has connected to MySQL:

```
select user();
```

```
mysql> select user();
+-----+
| user()      |
+-----+
| root@localhost |
+-----+
1 row in set (0.00 sec)

mysql>
```

Run the following command to display list of databases available in the current MySQL server:

```
show databases;
```

```
mysql> show databases;
+-----+
| Database      |
+-----+
| information_schema |
| mysql          |
| performance_schema |
| sakila         |
| sys            |
| world          |
+-----+
6 rows in set (0.96 sec)

mysql>
```

As you can see, it displays the default databases like `information_schema`, `mysql`, `performance_schema`, `sakila`, `sys` and `world`. Here, `sakila` and `world` are sample databases created at the time of installation.

6. Run MySQL Commands:

Follow the [official MySQL documentation](#) to understand the syntax for the SQL statements supported by MySQL.

Let us execute some SQL commands to create database, table and insert some data into it.

6.1. Create Database:

To create a database, MySQL follows specific syntax as below:

```
CREATE DATABASE database_name;
```

Run the following command to create a database named test:

```
create database test;
```

```
mysql> create database test;
Query OK, 1 row affected (0.37 sec)

mysql>
```

6.2. Select Database:

To select a database, MySQL follows specific syntax as below:

```
USE database_name;
```

Run the following command to use test database:

```
use test;
```

```
mysql> use test;
Database changed
mysql>
```

Use the following to verify the current database name:

```
select database();
```

```
mysql> select database();
+-----+
| database() |
+-----+
| test      |
+-----+
1 row in set (0.00 sec)

mysql>
```

6.3. Create Table:

To create a table in a database, MySQL follows specific syntax as below:

```
CREATE TABLE table_name (column_name1 data_type
<optional_constraint>, column_name2 data_type <optional_constraint>);
```

Run the following command to create a table named `employee` with columns like `employee_id`, `first_name`, `last_name`, `email`, `phone_number`, etc.:

```
create table employee (employee_id int not null, first_name
varchar(100), last_name varchar(100), email varchar(100),
phone_number varchar(20), hire_date date, salary int, manager_id int,
department_id int);
```

```
mysql> create table employee (employee_id int not null, first_name varchar(100), last_name varchar(100), email varchar(100), phone_number varchar(20), hire_date date, salary int, manager_id int, department_id int);
Query OK, 0 rows affected (0.96 sec)

mysql>
```

6.4. Show Columns:

Use the following command to see the list of columns along with their datatypes and constraints in the `employee` table:

```
show columns from employee;
```

```
mysql> show columns from employee;
+-----+-----+-----+-----+-----+
| Field | Type  | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| employee_id | int   | NO  |  | NULL    |  |
| first_name  | varchar(100) | YES |  | NULL    |  |
| last_name   | varchar(100) | YES |  | NULL    |  |
| email        | varchar(100) | YES |  | NULL    |  |
| phone_number | varchar(20)  | YES |  | NULL    |  |
| hire_date    | date   | YES |  | NULL    |  |
| salary       | int    | YES |  | NULL    |  |
| manager_id   | int    | YES |  | NULL    |  |
| department_id | int   | YES |  | NULL    |  |
+-----+-----+-----+-----+-----+
9 rows in set (0.10 sec)

mysql>
```

6.5. Insert Data:

To insert data into a table, MySQL follows specific syntax as below:

```
INSERT INTO TABLE table_name (column_name1, column_name2) VALUES  
(value1, value2);
```

Run the following command to insert 4 records into employee table:

```
insert into employee (employee_id, first_name, last_name, email,  
phone_number, hire_date, salary, manager_id, department_id)  
values(100, 'Steven', 'King', 'SKING@gmail.com', '515.123.4567',  
'2023-06-17', 24000, 0, 20);  
insert into employee (employee_id, first_name, last_name, email,  
phone_number, hire_date, salary, manager_id, department_id)  
values(101, 'Neena', 'Kochhar', 'NKOCHHAR@gmail.com', '515.123.4568',  
'2025-09-21', 17000, 100, 10);  
insert into employee (employee_id, first_name, last_name, email,  
phone_number, hire_date, salary, manager_id, department_id)  
values(102, 'Lex', 'De Haan', 'LDEHAAN@yahoo.com', '515.123.4569',  
'2021-01-13', 17000, 100, 10);  
insert into employee (employee_id, first_name, last_name, email,  
phone_number, hire_date, salary, manager_id, department_id)  
values(103, 'Alexander', 'Hunold', 'AHUNOLD@orbit.com',  
'590.423.4567', '2022-01-03', 9000, 102, 30);
```

```
mysql> insert into employee (employee_id, first_name, last_name, email, phone_number, hire_date, salary, manager_id, department_id) values(100, 'Steven', 'King', 'SKING@gmail.com', '515.123.4567', '2023-06-17', 24000, 0, 20);  
Query OK, 1 row affected (0.11 sec)  
  
mysql> insert into employee (employee_id, first_name, last_name, email, phone_number, hire_date, salary, manager_id, department_id) values(101, 'Neena', 'Kochhar', 'NKOCHHAR@gmail.com', '515.123.4568', '2025-09-21', 17000, 100, 10);  
Query OK, 1 row affected (0.09 sec)  
  
mysql> insert into employee (employee_id, first_name, last_name, email, phone_number, hire_date, salary, manager_id, department_id) values(102, 'Lex', 'De Haan', 'LDEHAAN@yahoo.com', '515.123.4569', '2021-01-13', 17000, 100, 10);  
Query OK, 1 row affected (0.09 sec)  
  
mysql> insert into employee (employee_id, first_name, last_name, email, phone_number, hire_date, salary, manager_id, department_id) values(103, 'Alexander', 'Hunold', 'AHUNOLD@orbit.com', '590.423.4567', '2022-01-03', 9000, 102, 30);  
Query OK, 1 row affected (0.12 sec)  
  
mysql>
```

6.6. Select Data:

To select data from a table, MySQL follows specific syntax as below:

```
SELECT column_names FROM table_name WHERE <condition>;
```

Run the following command to select department 10 employees list:

```
select * from employee where department_id=10;
```

```
mysql> select * from employee where department_id=10;
+-----+-----+-----+-----+-----+-----+-----+-----+
| employee_id | first_name | last_name | email | phone_number | hire_date | salary | manager_id | department_id |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 101 | Neena | Kochhar | NKOCHHAR@gmail.com | 515.123.4568 | 2025-09-21 | 17000 | 100 | 10 |
| 102 | Lex | De Haan | LDEHAAN@yahoo.com | 515.123.4569 | 2021-01-13 | 17000 | 100 | 10 |
+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

6.7. Drop Table:

To drop a specific table, MySQL follows specific syntax as below:

```
DROP TABLE table_name;
```

Run the following command to drop employee table:

```
drop table employee;
```

```
mysql> drop table employee;
Query OK, 0 rows affected (0.48 sec)

mysql>
```

6.8. Drop Database:

To drop a specific database, MySQL follows specific syntax as below:

```
DROP DATABASE database_name;
```

Run the following command to drop test database:

```
drop database test;
```

```
mysql> drop database test;
Query OK, 0 rows affected (0.32 sec)

mysql>
```

Run the following commands to verify if the current test database has been removed:

```
select database();
```

```
mysql> select database();
+-----+
| database() |
+-----+
| NULL      |
+-----+
1 row in set (0.00 sec)

mysql>
```

7. MySQL Server CLI Utilities:

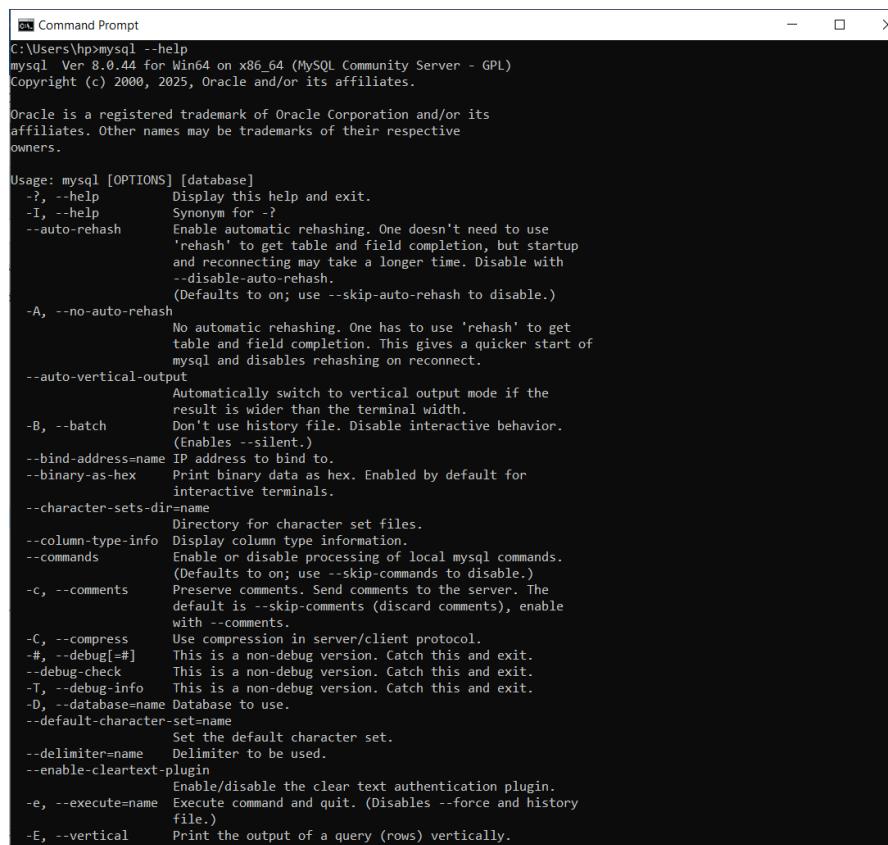
MySQL provides a bunch of command line utilities to perform various activities.

7.1. mysql:

`mysql` is a simple SQL shell with input line editing capabilities and it works in both interactive and noninteractive mode. `mysql` client is used to execute SQL commands.

`mysql` supports various options which can be specified in the command line. Use the following command to get the list of options supported by `mysql`.

```
mysql --help
```



The screenshot shows a Windows Command Prompt window with the title 'Command Prompt'. The command entered is 'C:\Users\hp>mysql --help'. The output displays the usage information for the MySQL command-line client, including various options and their descriptions. Key options include -?, -I, --auto-rehash, -A, --auto-vertical-output, -B, --bind-address-name, --binary-as-hex, --character-sets-dir, --column-type-info, --comments, --compress, -#, --debug, --debug-check, --debug-info, -D, --database-name, --default-character-set, --delimiter, --enable-cleartext-plugin, -e, --execute, and -E, --vertical.

```
C:\Users\hp>mysql --help
mysql Ver 8.0.44 for Win64 on x86_64 (MySQL Community Server - GPL)
Copyright (c) 2000, 2025, Oracle and/or its affiliates.

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

Usage: mysql [OPTIONS] [database]
  -?, --help            Display this help and exit.
  -I, --help            Synonym for -?
  --auto-rehash         Enable automatic rehashing. One doesn't need to use
                       'rehash' to get table and field completion, but startup
                       and reconnecting may take a longer time. Disable with
                       --disable-auto-rehash.
                       (Defaults to on; use --skip-auto-rehash to disable.)
  -A, --no-auto-rehash    No automatic rehashing. One has to use 'rehash' to get
                         table and field completion. This gives a quicker start of
                         mysql and disables rehashing on reconnect.
  --auto-vertical-output   Automatically switch to vertical output mode if the
                         result is wider than the terminal width.
  -B, --batch            Don't use history file. Disable interactive behavior.
                         (Enables --silent.)
  --bind-address=name    IP address to bind to.
  --binary-as-hex        Print binary data as hex. Enabled by default for
                         interactive terminals.
  --character-sets-dir=name
                         Directory for character set files.
  --column-type-info     Display column type information.
  --commands             Enable or disable processing of local mysql commands.
                         (Defaults to on; use --skip-commands to disable.)
  -c, --comments          Preserve comments. Send comments to the server. The
                         default is --skip-comments (discard comments), enable
                         with --comments.
  -C, --compress          Use compression in server/client protocol.
  -#, --debug[=#]          This is a non-debug version. Catch this and exit.
  --debug-check           This is a non-debug version. Catch this and exit.
  -T, --debug-info         This is a non-debug version. Catch this and exit.
  -D, --database=name     Database to use.
  --default-character-set
                         Set the default character set.
  --delimiter=name        Delimiter to be used.
  --enable-cleartext-plugin
                         Enable/disable the clear text authentication plugin.
  -e, --execute=name      Execute command and quit. (Disables --force and history
                         file.)
  -E, --vertical          Print the output of a query (rows) vertically.
```

For example, use the following commands to get the current `mysql` client version:

```
mysql -V
```

or

```
mysql --version
```

```
C:\Users\hp>mysql -V
mysql Ver 8.0.44 for Win64 on x86_64 (MySQL Community Server - GPL)

C:\Users\hp>
```

Use the following command to connect to specific database like sakila directly and enter root user password when prompted:

```
mysql -u root -p sakila
```

```
Windows Command Prompt - mysql -u root -p sakila
C:\Users\hp>mysql -u root -p sakila
Enter password: *****
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 40
Server version: 8.0.44 MySQL Community Server - GPL

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

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 tables;
+-----+
| Tables_in_sakila |
+-----+
| actor
| actor_info
| address
| category
| city
| country
| customer
| customer_list
| film
| film_actor
| film_category
| film_list
| film_text
| inventory
| language
| nicer_but_slower_film_list
| payment
| rental
| sales_by_film_category
| sales_by_store
| staff
| staff_list
| store
+-----+
23 rows in set (0.01 sec)

mysql>
```

mysql provides many built-in commands to perform tasks like connecting to a MySQL instance, sending command to MySQL server, executing a script file, etc.

Run the following command in mysql> prompt to get the list of those commands:

```
help
```

```

mysql> help

For information about MySQL products and services, visit:
  http://www.mysql.com/
For developer information, including the MySQL Reference Manual, visit:
  http://dev.mysql.com/
To buy MySQL Enterprise support, training, or other products, visit:
  https://shop.mysql.com/

List of all MySQL commands:
Note that all text commands must be first on line and end with ';'
?          (\?) Synonym for `help'.
clear      (\c) Clear the current input statement.
connect   (\r) Reconnect to the server. Optional arguments are db and host.
delimiter (\d) Set statement delimiter.
ego        (\G) Send command to mysql server, display result vertically.
exit      (\q) Exit mysql. Same as quit.
go         (\g) Send command to mysql server.
help       (\h) Display this help.
notee     (\t) Don't write into outfile.
print     (\p) Print current command.
prompt    (\R) Change your mysql prompt.
quit      (\q) Quit mysql.
rehash    (\#) Rebuild completion hash.
source   (\.) Execute an SQL script file. Takes a file name as an argument.
status   (\s) Get status information from the server.
system   (\!) Execute a system shell command, if enabled
tee      (\T) Set outfile [to_outfile]. Append everything into given outfile.
use      (\u) Use another database. Takes database name as argument.
charset  (\c) Switch to another charset. Might be needed for processing binlog with multi-byte charsets.
warnings (\W) Show warnings after every statement.
nowarning (\w) Don't show warnings after every statement.
resetconnection(\x) Clean session context.
query_attributes Sets string parameters (name1 value1 name2 value2 ...) for the next query to pick up.
ssl_session_data_print Serializes the current SSL session data to stdout or file

For server side help, type 'help contents'

mysql>

```

Following are some common commands being used:

- **? or \? or \h or help or \help:** Displays general help about mysql CLI.
- **\q or quit or exit:** Exits mysql CLI.
- **\c or clear:** Clears current input statement.
- **\r or connect:** Reconnects to the same MySQL instance.
- **\p or print:** Prints current input statement without executing it.
- **\R or prompt:** Changes the mysql prompt.
- **\g or go:** Sends command to the MySQL instance.
- **\s or status:** Displays the status information of MySQL server.
- **\. or source:** Executes script file using the active language.
- **! or system:** Executes the specified shell command.
- **\u or use:** Allows to specify schema to use.

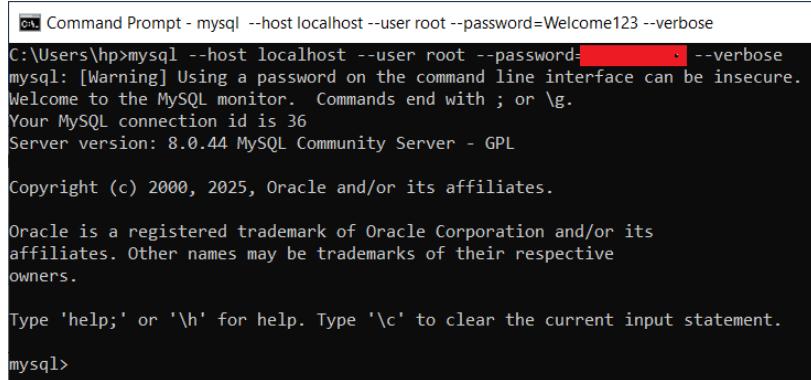
Use the following commands to connect to local MySQL instance with `root` user and password (*replace <root_password> with your root password given at the time of installation*) in verbose (*which displays SQL command in output*) mode.

```
mysql -h localhost -u root -p=<root_password> -v
```

or

```
mysql --host localhost --user root --password=<root_password> --verbose
```

Note that `-v` option can be given multiple times to produce more and more output such as `-v -v -v` produces table output format even in batch mode.



```
C:\ Command Prompt - mysql --host localhost --user root --password=Welcome123 --verbose
C:\Users\hp>mysql --host localhost --user root --password=[REDACTED] --verbose
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 36
Server version: 8.0.44 MySQL Community Server - GPL

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

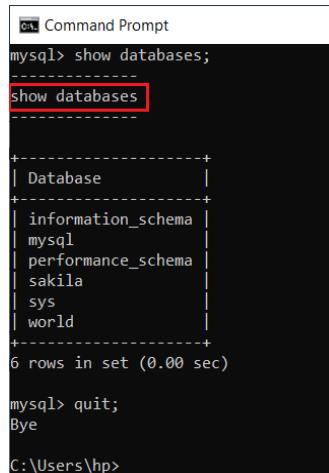
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql>
```

Once connected in verbose mode, `mysql` displays what the command does.

For example, run `show databases` command and it displays the command in the output:



```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sakila |
| sys |
| world |
+-----+
6 rows in set (0.00 sec)

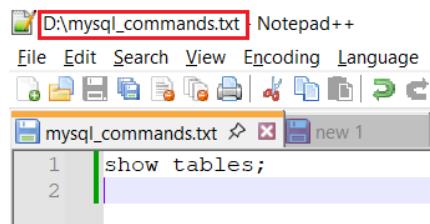
mysql> quit;
Bye

C:\Users\hp>
```

`mysql` tool also allows to execute SQL commands from the text file.

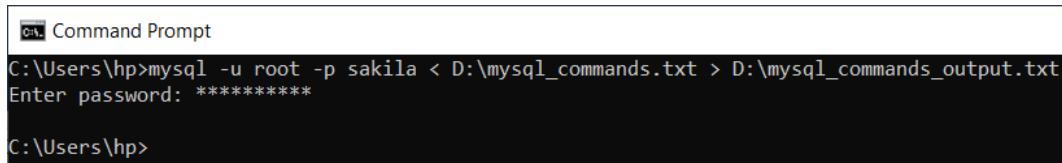
First, create a text file named `mysql_commands.txt` in D drive and enter the following commands in the file:

```
show tables;
```



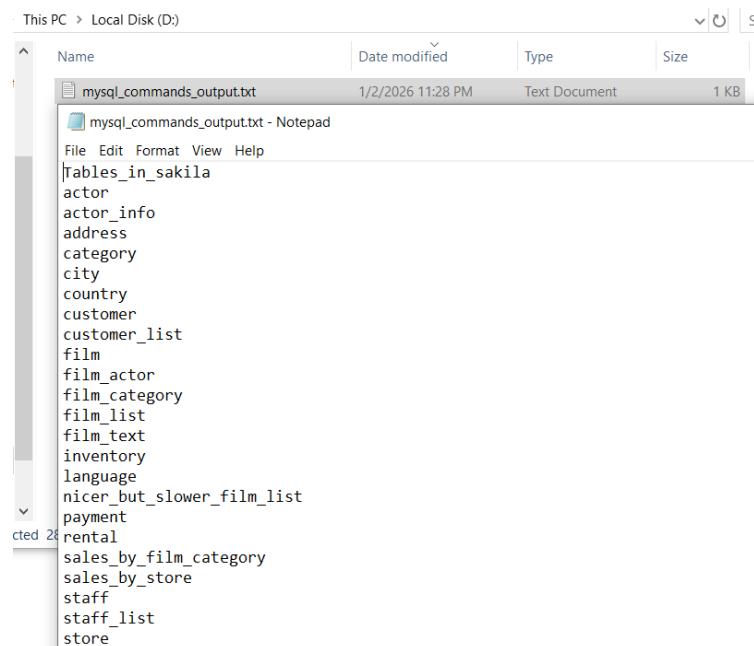
Now, run the following command to connect to `sakila` database and execute `mysql_commands.txt` file and save the output into output file named `mysql_commands_output.txt`:

```
mysql -u root -p sakila < D:\mysql_commands.txt >  
D:\mysql_commands_output.txt
```



The screenshot shows a Windows Command Prompt window titled "Command Prompt". The command entered is `C:\Users\hp>mysql -u root -p sakila < D:\mysql_commands.txt > D:\mysql_commands_output.txt`. A password prompt follows: "Enter password: *****". The command is then executed, and the prompt returns to `C:\Users\hp>`.

Open `D:\mysql_commands_output.txt` file where you can see the list of tables available in `sakila`.



SQL file can also be executed after connecting to MySQL server.

First, run the following command to connect to `sakila` database. Enter `root` user password when prompted:

```
mysql -u root -p sakila
```

```
 Command Prompt - mysql -u root -p sakila
C:\Users\hp>mysql -u root -p sakila
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 48
Server version: 8.0.44 MySQL Community Server - GPL

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

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>
```

Then run the following SQL command to execute D:\mysql_commands.txt file:

```
source D:\mysql_commands.txt
```

or

```
\. D:\mysql_commands.txt
```

```
 Command Prompt - mysql -u root -p sakila
mysql> source D:\mysql_commands.txt
+-----+
| Tables_in_sakila |
+-----+
| actor           |
| actor_info      |
| address         |
| category        |
| city            |
| country          |
| customer        |
| customer_list   |
| film            |
| film_actor      |
| film_category   |
| film_list       |
| film_text       |
| inventory       |
| language         |
| nicer_but_slower_film_list |
| payment          |
| rental           |
| sales_by_film_category |
| sales_by_store   |
| staff            |
| staff_list      |
| store            |
+-----+
23 rows in set (0.41 sec)

mysql> \. D:\mysql_commands.txt
+-----+
| Tables_in_sakila |
+-----+
| actor           |
| actor_info      |
| address         |
| category        |
| city            |
| country          |
| customer        |
| customer_list   |
| film            |
| film_actor      |
| film_category   |
| film_list       |
| film_text       |
| inventory       |
| language         |
+-----+
```

To export a specific table data into a file, use the following query:

```
SELECT * FROM customer INTO OUTFILE 'D:\sakila_customers.txt' FIELDS
TERMINATED BY ',' ENCLOSED BY '\"' LINES TERMINATED BY '\r\n';
```

```
Command Prompt - mysql -u root -p sakila
mysql> SELECT * FROM customer INTO OUTFILE 'D:\sakila_customers.txt' FIELDS TERMINATED BY ',' ENCLOSED BY '\"' LINES TERMINATED BY '\r\n';
ERROR 1290 (HY000): The MySQL server is running with the --secure-file-priv option so it cannot execute this statement
mysql>
```

To import data from a file to a specific table, use the following command:

```
LOAD DATA LOCAL INFILE 'D:\sakila_customers.txt' INTO TABLE customers_dump FIELDS TERMINATED BY ',' LINES TERMINATED BY '\r\n';
```

7.2. mysqlshow:

`mysqlshow` client can be used to quickly check the existence of databases, tables, columns and indexes.

`mysqlshow` supports various options which can be specified in the command line. Use the following command to get the list of options supported by `mysqlshow` and commands to be executed:

```
mysqlshow --help
```

```
Command Prompt
C:\Users\hp\mysqlshow --help
mysqlshow Ver 8.0.44 for Win64 on x86_64 (MySQL Community Server - GPL)
Copyright (c) 2000, 2025, Oracle and/or its affiliates.

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

Shows the structure of a MySQL database (databases, tables, and columns).

Usage: mysqlshow [OPTIONS] [database [table [column]]]

If last argument contains a shell or SQL wildcard (*, ?, % or _) then only
what's matched by the wildcard is shown.
If no database is given then all matching databases are shown.
If no table is given, then all matching tables in database are shown.
If no column is given, then all matching columns and column types in table
are shown.

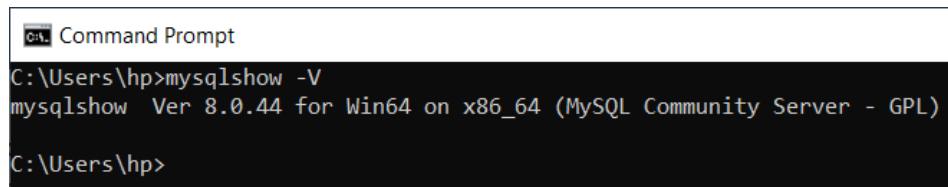
Default options are read from the following files in the given order:
C:\Windows\my.ini C:\Windows\my.cnf C:\my.ini D:\ProgramFiles\MySQL\MySQL Server 8.0\my.cnf
The following groups are read: mysqlshow client
The following options may be given as the first argument:
--print-defaults      Print the program argument list and exit.
--no-defaults        Don't read default options from any option file,
                    except for login file.
--defaults-file=#   Only read default options from the given file #.
--defaults-extra-file=# Read this file after the global files are read.
--defaults-group-suffix#
                    Also read groups with concat(group, suffix)
--login-path#        Read this path from the login file.
--bind-address=name  IP address to bind to.
-c, --character-set-name
                    Directory for character set files.
--default-character-set-name
                    Set the default character set.
--count              Show number of rows per table (may be slow for non-MyISAM
                    tables).
-C, --compress       Use compression in server/client protocol.
#, --debug=name     Output debug log. Often this is 'd:t,o, filename'.
--debug-check        Check memory and open file usage at exit.
--debug-info         Print some debug info at exit.
--default-auth=name Default authentication client-side plugin to use.
--enable-cleartext-plugin
                    Enable/disable the clear text authentication plugin.
-, --help            Display this help and exit.
-h, --host=name     Connect to host.
-i, --status         Shows a lot of extra information about each table.
```

For example, use the following commands to get the current `mysqlshow` client version:

```
mysqlshow -V
```

or

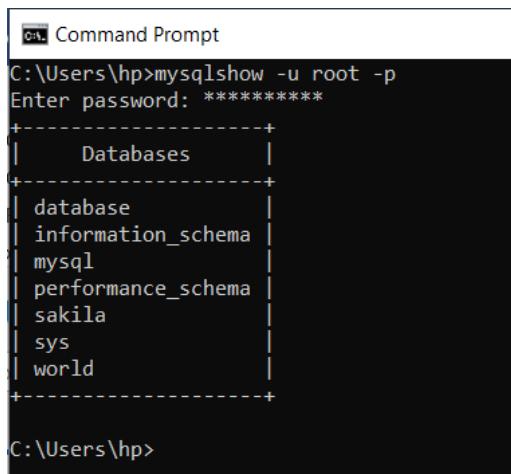
```
mysqlshow --version
```



```
Command Prompt
C:\Users\hp>mysqlshow -V
mysqlshow Ver 8.0.44 for Win64 on x86_64 (MySQL Community Server - GPL)
C:\Users\hp>
```

Use the following command to get the list of databases. Enter `root` user password when prompted:

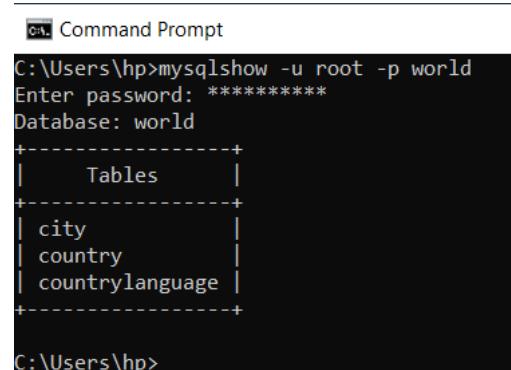
```
mysqlshow -u root -p
```



```
Command Prompt
C:\Users\hp>mysqlshow -u root -p
Enter password: *****
+-----+
|   Databases   |
+-----+
| database      |
| information_schema |
| mysql          |
| performance_schema |
| sakila         |
| sys            |
| world          |
+-----+
C:\Users\hp>
```

Use the following command to get the list of tables in `test` database. Enter `root` user password when prompted:

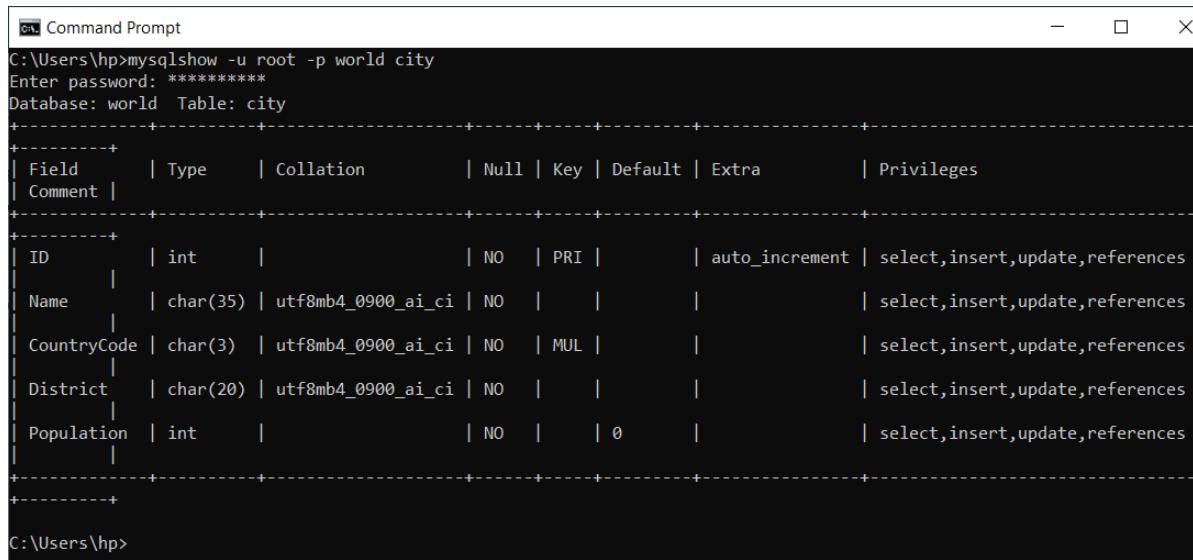
```
mysqlshow -u root -p world
```



```
Command Prompt
C:\Users\hp>mysqlshow -u root -p world
Enter password: *****
Database: world
+-----+
|   Tables    |
+-----+
| city        |
| country     |
| countrylanguage |
+-----+
C:\Users\hp>
```

Use the following command to get the list of columns in `city` table in `world` database. Enter `root` user password when prompted:

```
mysqlshow -u root -p world city
```



Field	Type	Collation	Null	Key	Default	Extra	Privileges
ID	int		NO	PRI		auto_increment	select,insert,update,references
Name	char(35)	utf8mb4_0900_ai_ci	NO				select,insert,update,references
CountryCode	char(3)	utf8mb4_0900_ai_ci	NO	MUL			select,insert,update,references
District	char(20)	utf8mb4_0900_ai_ci	NO				select,insert,update,references
Population	int		NO		0		select,insert,update,references

7.3. mysqladmin:

`mysqladmin` client can be used to perform various administrative activities such as check current server status, verify configuration, create and drop databases, stop server, etc.

`mysqladmin` supports various options which can be specified in the command line. Use the following command to get the list of options supported by `mysqladmin` and commands to be executed:

```
mysqladmin --help
```

```

C:\Users\hp\mysqladmin --help
mysqladmin Ver 8.0.44 for Win64 on x86_64 (MySQL Community Server - GPL)
Copyright (c) 2000, 2025, Oracle and/or its affiliates.

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

Administration program for the mysqld daemon.
Usage: mysqladmin [OPTIONS] command command....
--bind-address=name IP address to bind to.
-c, --count=# Number of iterations to make. This works with -i
                (--sleep) only.
-#, --debug[=#] This is a non-debug version. Catch this and exit.
--debug-check This is a non-debug version. Catch this and exit.
--debug-info This is a non-debug version. Catch this and exit.
-f, --force Don't ask for confirmation on drop database; with
                multiple commands, continue even if an error occurs.
-C, --compress Use compression in server/client protocol.
--character-sets-dir=name Directory for character set files.
--default-character-set=name Set the default character set.
-?, --help Display this help and exit.
-h, --host=name Connect to host.
-b, --no-beep Turn off beep on error.
-p, --password[=name]
                Password to use when connecting to server. If password is
                not given it's asked from the tty.
-, --password1[=name]
                Password for first factor authentication plugin.
-, --password2[=name]
                Password for second factor authentication plugin.
-, --password3[=name]
                Password for third factor authentication plugin.
-W, --pipe Use named pipes to connect to server.
-P, --port=# Port number to use for connection or 0 for default to, in
                order of preference, my.cnf, $MYSQL_TCP_PORT,
                /etc/services, built-in default (3306).
--protocol=name The protocol to use for connection (tcp, socket, pipe,
                memory).
-r, --relative Show difference between current and previous values when
                used with -i. Currently only works with extended-status.
--shared-memory-base-name=name Base name of shared memory.
-s, --silent Silently exit if one can't connect to server.
-S, --socket=name The socket file to use for connection.
-i, --sleep=# Execute commands repeatedly with a sleep between.
--ssl-mode=name SSL connection mode.

shutdown-timeout          3600
plugin-dir                (No default value)
default-auth               (No default value)
enable-clearTEXT-plugin    FALSE
show-warnings              FALSE
compression-algorithms    (No default value)
zstd-compression-level     3

Default options are read from the following files in the given order:
C:\Windows\my.ini C:\Windows\my.cnf C:\my.ini C:\my.cnf D:\ProgramFiles\MySQL\MySQL Server 8.0\my.ini D:\ProgramFiles\MySQL\MySQL Server 8.0\my.cnf
The following groups are read: mysqladmin client
The following options may be given as the first argument:
--print-defaults      Print the program argument list and exit.
--no-defaults         Don't read default options from any option file,
                    except for login file.
--defaults-file=#    Only read default options from the given file #.
--defaults-extra-file=# Read this file after the global files are read.
--defaults-group-suffix=# Also read groups with concat(group, suffix)
--login-path=#        Read this path from the login file.

Where command is one or more of: (Commands may be shortened)
create database name   Create a new database
debug                  Instruct server to write debug information to log
drop database name    Delete a database and all its tables
extended status        Gives an extended status message from the server
flush hosts            Flush all cached hosts
flush logs             Flush all logs
flush status           Clear status variables
flush tables           Flush all tables
flush threads          Flush the thread cache
flush privileges       Reload grant tables (same as reload)
kill id,id,...         Kill mysql threads
password [new-password] Change old password to new-password in current format
ping                  Check if mysqld is alive
processlist            Show list of active threads in server
reload                Reload grant tables
refresh               Flush all tables and close and open logfiles
shutdown              Take server down
status                Gives a short status message from the server
start-replica          Start replication
start-slave            Deprecated: use start-replica instead
stop-replica           Stop replication
stop-slave             Deprecated: use stop-replica instead
variables             Prints variables available
version               Get version info from server

C:\Users\hp>

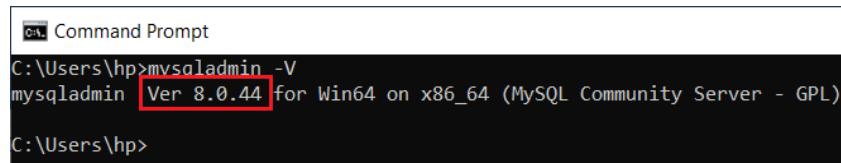
```

For example, use the following commands to get the current mysqladmin client version:

```
mysqladmin -V
```

or

```
mysqladmin --version
```

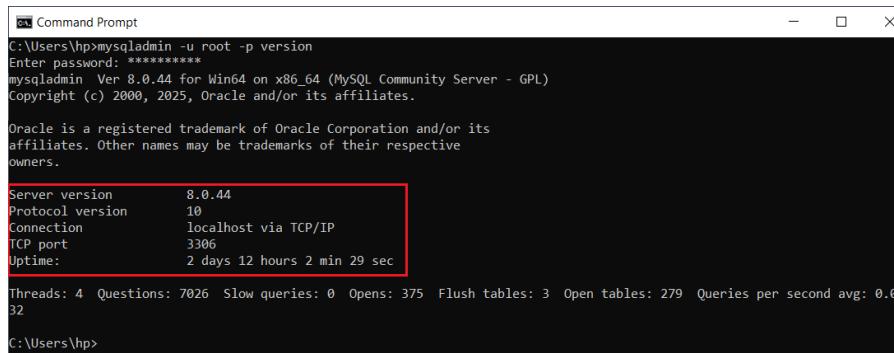


```
Command Prompt
C:\Users\hp>mysqladmin --version
mysqladmin Ver 8.0.44 for Win64 on x86_64 (MySQL Community Server - GPL)

C:\Users\hp>
```

Use the following command to connect to MySQL server and get the server version and uptime. Enter `root` user password when prompted:

```
mysqladmin -u root -p version
```



```
Command Prompt
C:\Users\hp>mysqladmin -u root -p version
Enter password: *****
mysqladmin Ver 8.0.44 for Win64 on x86_64 (MySQL Community Server - GPL)
Copyright (c) 2000, 2025, Oracle and/or its affiliates.

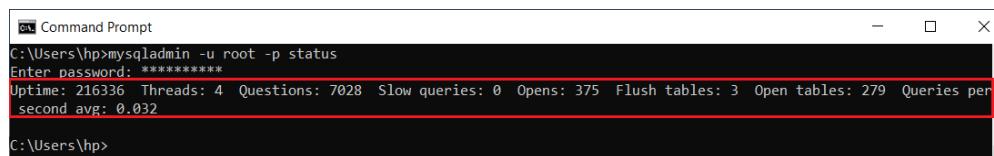
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Server version      8.0.44
Protocol version   10
Connection          localhost via TCP/IP
TCP port            3306
Uptime:             2 days 12 hours 2 min 29 sec
Threads: 4 Questions: 7026 Slow queries: 0 Opens: 375 Flush tables: 3 Open tables: 279 Queries per second avg: 0.0
32

C:\Users\hp>
```

Use the following command to get the short status of MySQL server. Enter `root` user password when prompted:

```
mysqladmin -u root -p status
```

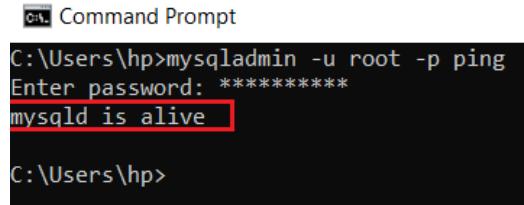


```
Command Prompt
C:\Users\hp>mysqladmin -u root -p status
Enter password: *****
Uptime: 216336 Threads: 4 Questions: 7028 Slow queries: 0 Opens: 375 Flush tables: 3 Open tables: 279 Queries per
second avg: 0.032

C:\Users\hp>
```

Use the following command to verify if `mysqld` (MySQL server) is alive. Enter `root` user password when prompted:

```
mysqladmin -u root -p status
```

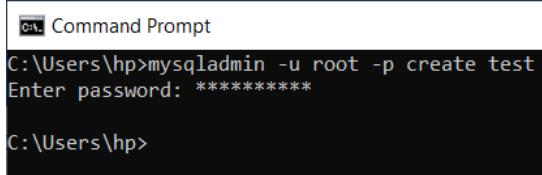


```
C:\Users\hp>mysqladmin -u root -p ping
Enter password: *****
mysqld is alive

C:\Users\hp>
```

Use the following command to create database named test. Enter root user password when prompted:

```
mysqladmin -u root -p create test
```



```
C:\Users\hp>mysqladmin -u root -p create test
Enter password: *****
C:\Users\hp>
```

7.4. mysqldump:

`mysqldump` client can help to perform logical backups by producing a set of SQL commands that can be executed to reproduce original database and its data. It is also used to generate table output in CSV, XML or any delimited text file.

`mysqldump` supports various options which can be specified in the command line. Use the following command to get the list of options supported by `mysqldump` and commands to be executed:

```
mysqldump --help
```

```
C:\Users\hp>mysqldump --help
mysqldump Ver 8.0.44 for Win64 on x86_64 (MySQL Community Server - GPL)
Copyright (c) 2000, 2025, Oracle and/or its affiliates.

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

Dumping structure and contents of MySQL databases and tables.
Usage: mysqldump [OPTIONS] database [tables]
OR      mysqldump [OPTIONS] --databases [OPTIONS] DB1 [DB2 DB3...]
OR      mysqldump [OPTIONS] --all-databases [OPTIONS]

Default options are read from the following files in the given order:
C:\Windows\my.ini C:\Windows\my.cnf C:\my.ini C:\my.cnf D:\ProgramFiles\MySQL\MySQL Server 8.0\my.ini D:\ProgramFiles\My
SQL\MySQL Server 8.0\my.cnf
The following groups are read: mysqlclient
The following options may be given as the first argument:
--print-defaults      Print the program argument list and exit.
--no-defaults        Don't read default options from any option file,
                    except for login file.
--defaults-file=#   Only read default options from the given file #.
--defaults-extra-file=# Read this file after the global files are read.
--defaults-group-suffix=#
                    Also read groups with concat(group, suffix)
--login-path#        Read this path from the login file.
-A, --all-databases Dump all the databases. This will be same as --databases
                    with all databases selected.
-Y, --all-tablespaces
                    Dump all the tablespaces.
-y, --no-tablespaces
                    Do not dump any tablespace information.
--add-drop-database Add a DROP DATABASE before each create.
--add-drop-table   Add a DROP TABLE before each create.
--add-drop-trigger (Defaults to on; use --skip-add-drop-table to disable.)
--add-locks         Add locks around INSERT statements.
                    (Defaults to on; use --skip-add-locks to disable.)
--allow-keywords   Allow creation of column names that are keywords.
--apply-replica-statements
                    Adds 'STOP SLAVE' prior to 'CHANGE MASTER' and 'START
                    SLAVE' to bottom of dump.
--apply-slave-statements
                    This option is deprecated and will be removed in a future
                    version. Use apply-replica-statements instead.
--bind-address=name IP address to bind to.
--character-sets-dir=name
                    Directory for character set files.
--column-statistics Add an ANALYZE TABLE statement to regenerate any existing
```

For example, use the following commands to get the current mysqldump client version:

```
mysqldump -V
```

or

```
mysqldump --version
```

```
C:\Users\hp>mysqldump -V
mysqldump Ver 8.0.44 for Win64 on x86_64 (MySQL Community Server - GPL)
C:\Users\hp>
```

Use the following command to export all databases to
D:\mysqldump_all_databases.txt file in SQL format. Enter root user password
when prompted:

```
mysqldump -u root -p --all-databases > D:/mysqldump_all_databases.txt
```

```

C:\ Command Prompt
C:\Users\hp>mysqldump -u root -p --all-databases > D:/mysqldump_all_databases.txt
Enter password: *****
C:\Users\hp>

```

Open D:\mysqldump_all_databases.txt file where you can see the create and insert scripts for all databases.

```

D:\mysqldump_all_databases.txt - Notepad+
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window 2
myqldump_all_databases.txt (1 page) (new 2) (new 3)
-- MySQL dump 10.13 Distrib 8.0.44, for Win64 (x86_64)
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42

```

Use the following command to export the all tables in world database to D:\mysqldump_world.txt file in SQL format. Enter root user password when prompted:

```
mysqldump -u root -p world > D:/mysqldump_world.txt
```

```

C:\ Command Prompt
C:\Users\hp>mysqldump -u root -p world > D:/mysqldump_world.txt
Enter password: *****
C:\Users\hp>

```

Open D:\mysqldump_world.txt file where you can see the create and insert scripts for all tables in world database.

```

D:\mysqldump_world.txt - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
mysqldump_world.txt new 1 new 2 new 3
1 -- MySQL dump 10.13 Distrib 8.0.44, for Win64 (x86_64)
2 --
3 -- Host: localhost      Database: world
4 --
5 -- Server version     8.0.44
6
7 /*!40101 SET @OLD_CHARACTER_SET_CLIENT =@@CHARACTER_SET_CLIENT */;
8 /*!40101 SET @OLD_CHARACTER_SET_RESULTS =@@CHARACTER_SET_RESULTS */;
9 /*!40101 SET @OLD_COLLATION_CONNECTION =@@COLLATION_CONNECTION */;
10 /*!50503 SET NAMES utf8mb4 */;
11 /*!40103 SET @OLD_TIME_ZONE =@@TIME_ZONE */;
12 /*!40103 SET TIME_ZONE='+00:00' */;
13 /*!40014 SET @OLD_UNIQUE_CHECKS =@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
14 /*!40014 SET @OLD_FOREIGN_KEY_CHECKS =@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
15 /*!40101 SET @OLD_SQL_MODE =@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
16 /*!40111 SET @OLD_SQL_NOTES =@SQL_NOTES, SQL_NOTES=0 */;
17
18 --
19 -- Table structure for table `city`
20 --
21
22 DROP TABLE IF EXISTS `city`;
23 /*!40101 SET @saved_cs_client      = @@character_set_client */;
24 /*!50503 SET character_set_client = utf8mb4 */;
25 CREATE TABLE `city` (
26   `ID` int NOT NULL AUTO_INCREMENT,
27   `Name` char(35) NOT NULL DEFAULT '',
28   `CountryCode` char(3) NOT NULL DEFAULT '',
29   `District` char(20) NOT NULL DEFAULT '',
30   `Population` int NOT NULL DEFAULT '0',
31   PRIMARY KEY (`ID`),
32   KEY `CountryCode` (`CountryCode`),
33   CONSTRAINT `city_ibfk_1` FOREIGN KEY (`CountryCode`) REFERENCES `country` (`Code`)
34 ) ENGINE=InnoDB AUTO_INCREMENT=4080 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
35 /*!40101 SET character_set_client = @saved_cs_client */;
36
37 --
38 -- Dumping data for table `city`
39 --
40
41 LOCK TABLES `city` WRITE;
42 /*!40000 ALTER TABLE `city` DISABLE KEYS */;

```

Use the following command to export only country table in world database to D:/mysqldump_world_countries.txt file in SQL format. Enter root user password when prompted:

```
mysqldump -u root -p world country > D:/mysqldump_world_countries.txt
```

```

C:\Command Prompt
C:\Users\hp>
C:\Users\hp>mysqldump -u root -p world country > D:/mysqldump_world_countries.txt
Enter password: *****
C:\Users\hp>

```

Open D:\mysqldump_world_countries.txt file where you can see the create and insert scripts for the country table.

```

D:\mysqldump_world_countries.txt - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
new 1 new 2 new 3
mysqldump_world_countries.txt
1 -- MySQL dump 10.13 Distrib 8.0.44, for Win64 (x86_64)
2 --
3 -- Host: localhost      Database: world
4 --
5 -- Server version      8.0.44
6
7 /*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
8 /*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
9 /*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
10 /*!40103 SET NAMES utf8mb4 */;
11 /*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
12 /*!40103 SET TIME_ZONE='+00:00' */;
13 /*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
14 /*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
15 /*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
16 /*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;
17
18 --
19 -- Table structure for table `country`
20 --
21
22 DROP TABLE IF EXISTS `country`;
23 /*!40101 SET @saved_cs_client      = @@character_set_client */;
24 /*!40101 SET character_set_client = utf8mb4 */;
25 CREATE TABLE `country` (
26   `Code` char(3) NOT NULL DEFAULT '',
27   `Name` char(52) NOT NULL DEFAULT '',
28   `Continent` enum('Asia','Europe','North America','Africa','Oceania','Antarctica','South America') NOT NULL DEFAULT 'Asia',
29   `Region` char(26) NOT NULL DEFAULT '',
30   `SurfaceArea` decimal(10,2) NOT NULL DEFAULT '0.00',
31   `IndepYear` smallint DEFAULT NULL,
32   `Population` int NOT NULL DEFAULT '0',
33   `LifeExpectancy` decimal(3,1) DEFAULT NULL,
34   `GNP` decimal(10,2) DEFAULT NULL,
35   `GNPold` decimal(10,2) DEFAULT NULL,
36   `LocalName` char(45) NOT NULL DEFAULT '',
37   `GovernmentForm` char(45) NOT NULL DEFAULT '',
38   `HeadofState` char(60) DEFAULT NULL,
39   `Capital` int DEFAULT NULL,
40   `Code2` char(2) NOT NULL DEFAULT '',
41   PRIMARY KEY (`Code`)
42 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

```

Use the following command to export city and country tables in world database to D:\mysqldump_world_city_countries.txt file in SQL format. Enter root user password when prompted:

```
mysqldump -u root -p world city country >
D:/mysqldump_world_city_countries.txt
```

Command Prompt

```
C:\Users\hp>mysqldump -u root -p world city country > D:/mysqldump_world_city_countries.txt
Enter password: *****
C:\Users\hp>
```

Open D:\mysqldump_world_city_countries.txt file where you can see the create and insert scripts for the city and country tables.

```

D:\mysqldump_world_city_countries.txt - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window 
new 1 new 2 new 3
mysqldump_world_city_countries.txt new 1 new 2 new 3
1 -- MySQL dump 10.13 Distrib 8.0.44, for Win64 (x86_64)
2 --
3 -- Host: localhost      Database: world
4 --
5 -- Server version     8.0.44
6
7 /*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
8 /*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
9 /*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
10 /*!50503 SET NAMES utf8mb4 */;
11 /*!40103 SET @OLD_TIME_ZONE=@TIME_ZONE */;
12 /*!40103 SET TIME_ZONE='+00:00' */;
13 /*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
14 /*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
15 /*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
16 /*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;
17
18 --
19 -- Table structure for table `city`
20 --
21
22 DROP TABLE IF EXISTS `city`;
23 /*!40101 SET @saved_cs_client      = @@character_set_client */;
24 /*!50503 SET character_set_client = utf8mb4 */;
25 CREATE TABLE `city` (
26   `ID` int NOT NULL AUTO_INCREMENT,
27   `Name` char(35) NOT NULL DEFAULT '',
28   `CountryCode` char(3) NOT NULL DEFAULT '',
29   `District` char(20) NOT NULL DEFAULT '',
30   `Population` int NOT NULL DEFAULT '0',
31   PRIMARY KEY (`ID`),
32   KEY `CountryCode` (`CountryCode`),
33   CONSTRAINT `city_ibfk_1` FOREIGN KEY (`CountryCode`) REFERENCES `country` (`Code`)
34 ) ENGINE=InnoDB AUTO_INCREMENT=4080 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
35 /*!40101 SET character_set_client = @saved_cs_client */;
36
37 --
38 -- Dumping data for table `city`
39 --
40
41 LOCK TABLES `city` WRITE;

```

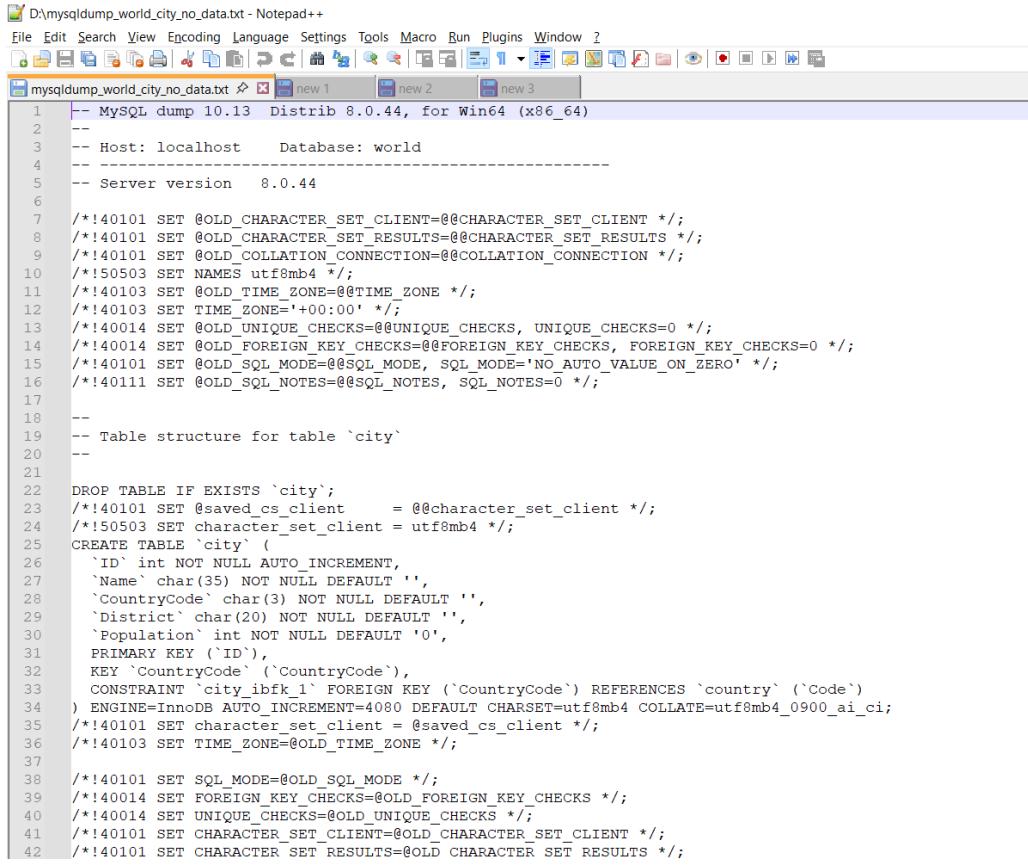
Use the following command to export `city` table in `world` database with `CREATE` script and no `INSERT` scripts. Enter `root` password when prompted:

```
mysqldump -u root -p world --no-data city >
D:/mysqldump_world_city_no_data.txt
```

```

Command Prompt
C:\Users\hp>mysqldump -u root -p world --no-data city > D:/mysqldump_world_city_no_data.txt
Enter password: *****
C:\Users\hp>
```

Open `D:\mysqldump_world_city_no_data.txt` file where you can see the create script for the `city` table.



```

D:\mysqldump_world_city_no_data.txt - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window 2
D:\mysqldump_world_city_no_data.txt new 1 new 2 new 3
1 -- MySQL dump 10.13 Distrib 8.0.44, for Win64 (x86_64)
2 --
3 -- Host: localhost      Database: world
4 --
5 -- Server version     8.0.44
6
7 /*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
8 /*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
9 /*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
10 /*!50503 SET NAMES utf8mb4 */;
11 /*!40103 SET @OLD_TIME_ZONE=@TIME_ZONE */;
12 /*!40103 SET TIME_ZONE='+00:00' */;
13 /*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
14 /*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
15 /*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
16 /*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;
17
18 --
19 -- Table structure for table `city`
20 --
21
22 DROP TABLE IF EXISTS `city`;
23 /*!40101 SET @saved_cs_client      = @@character_set_client */;
24 /*!50503 SET character_set_client = utf8mb4 */;
25 CREATE TABLE `city` (
26   `ID` int NOT NULL AUTO_INCREMENT,
27   `Name` char(35) NOT NULL DEFAULT '',
28   `CountryCode` char(3) NOT NULL DEFAULT '',
29   `District` char(20) NOT NULL DEFAULT '',
30   `Population` int NOT NULL DEFAULT '0',
31   PRIMARY KEY (`ID`),
32   KEY `CountryCode` (`CountryCode`),
33   CONSTRAINT `city_ibfk_1` FOREIGN KEY (`CountryCode`) REFERENCES `country` (`Code`)
34 ) ENGINE=InnoDB AUTO_INCREMENT=4080 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
35 /*!40101 SET character_set_client = @saved_cs_client */;
36 /*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;
37
38 /*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
39 /*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
40 /*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
41 /*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
42 /*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;

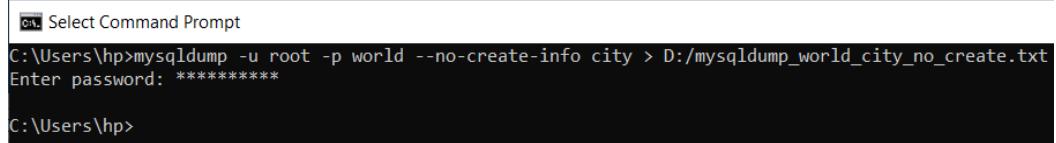
```

Use the following command to export `city` table in `world` database with only `INSERT` scripts.
Enter `root` user password when prompted:

```

mysqldump -u root -p world --no-create-info city >
D:/mysqldump_world_city_no_create.txt

```



```

Select Command Prompt
C:\Users\hp>mysqldump -u root -p world --no-create-info city > D:/mysqldump_world_city_no_create.txt
Enter password: *****
C:\Users\hp>

```

Open `D:/mysqldump_world_city_no_create.txt` file where you can see the insert scripts for the `city` table.

```

1 # D:\mysqldump.world_01.sql_create.txt - Notepad++
2 -- MySQL Dump 10.13 Distrib 8.0.44, for Win64 (x86_64)
3 -- Host: localhost Database: world
4 --
5 -- Server version 8.0.44
6
7 /*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
8 /*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
9 /*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
10 /*!40103 SET NAMES utf8mb4 */;
11 /*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
12 /*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
13 /*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
14 /*!40101 SET NAMES utf8 */;
15 /*!40101 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
16 /*!40101 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
17 /*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
18 /*!40101 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;
19
20 --
21 -- Dumping data for table 'city'
22 --
23 /*!40000 ALTER TABLE `city` DISABLE KEYS */;
24 INSERT INTO `city` VALUES
(1,'Rahat','Afghanistan',1780000),(2,'Qandahar','AFG','Qandahar',237500),(3,'Herat','AFG','Herat',184800),(4,'Mazar-e-Sharif','AFG','Malkh',127900),(5,'Amsterdam','NL','Noord-Holland',733200),(6,'Rotterdam','NL','Zuid-Holland',593321),(7,'Haag','NL','Zuid-Holland',440900),(8,'Utrecht','NL','Utrecht',23433),(9,'Groningen','NL','Groningen',172701),(10,'Breda','NL','Noord-Brabant',160398),(11,'Apel den','NL','Noord-Brabant',159398),(12,'Dordrecht','NL','Zuid-Brabant',159398),(13,'Eindhoven','NL','Noord-Brabant',159398),(14,'Den Haag','NL','Zuid-Holland',159398),(15,'Alkmaar','NL','Flevoland',142465),(16,'Arnhem','NL','Gelderland',139020),(17,'Saenstad','NL','Noord-Holland',139621),(20,'s-Hertogenbosch','NL','Noord-Brabant',129470),(21,'Aarsfoort','NL','Utrecht',126270),(22,'Maastricht','NL','Limburg',122087),(23,'Dordrecht','NL','Zuid-Holland',119811),(24,'Leiden','NL','Zuid-Holland',117196),(25,'Haarlem','NL','Noord-Holland',116222),(26,'Zoetermeer','NL','Zuid-Holland',110214),(27,'Emmen','NL','Drenthe',108111),(28,'Wolfe','NL','Overijssel',105260),(29,'Assen','NL','Drenthe',105260),(30,'Nijmegen','NL','Gelderland',103111),(31,'Tilburg','NL','Noord-Brabant',987213),(32,'Willemsstad','ANT','Curacao',2345),(33,'Ede','NL','Utrecht',279000),(34,'Almer','ALM','Tirana',279000),(35,'Alger','ALM','Algiers',216000),(36,'Oran','DZA','Oran',609823),(37,'Constantine','DZA','Constantine',443727),(38,'Annaba','DZA','Annaba',222518),(39,'Batna','DZA','Batna',183377),(40,'Sétif','DZA','Sétif',179055),(41,'Sidi Bel Abbès','DZA','Sidi Bel Abbès',177284),(42,'Tizi Ouzou','DZA','Tizi Ouzou',175218),(43,'Bejaia','DZA','Bejaia',173162),(44,'Mostaganem','DZA','Mostaganem',115213),(45,'Tébessa','DZA','Tébessa',112007),(46,'Tlemcen','DZA','Tlemcen',110242),(47,'Béchar','DZA','Béchar',107311),(50,'Tiaret','DZA','Tiaret',100118),(51,'Sétif-Chlef','DZA','Sétif-Chlef',967941),(52,'Ghardaïa','DZA','Ghardaïa',94145),(53,'Tafuna','ADM','Tutuila',3200),(54,'Fagatogo','ADM','Tutuila',2323),(55,'Andorra la Vella','AND','Andorra la Vella',21189),(56,'Lloida','ANG','Luanda',2022000),(57,'Huambo','AGO','Huambo',163100),(58,'Lobito','AGO','Benguela',130000),(59,'Benguela','AGO','Benguela',128300),(60,'Namibe','AGO','Namibe',118200),(61,'Santo António','AIA',-1,-598),(62,'Tchicapa Valley','AIA',-1,-598),(63,'Saint John','ATO'),+st John',24000),(64,'Dubai','ARE','Dubai',69181),(65,'Abu Dhabi','ARE','Abu Dhabi',396895),(66,'Sharjah','ARE','Sharjah',320998),(67,'al-Ayn','ARE','al-Ayn',14000),

```

7.5. mysqlimport:

`mysqlimport` client can help to import data from a file into a table. It works as an alternative to `LOAD DATA SQL` statement.

`mysqlimport` supports various options which can be specified in the command line. Use the following command to get the list of options supported by `mysqlimport` and commands to be executed:

```
mysqlimport --help
```

```

C:\Users\hp\mysqlimport --help
mysqlimport Ver 8.0.44 for Win64 on x86_64 (MySQL Community Server - GPL)

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

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

Loads tables from text files in various formats. The base name of the
text file must be the name of the table that should be used.
If one uses sockets to connect to the MySQL server, the server will open and
read the text file directly. In other cases the client will open the text
file. The SQL command "LOAD DATA INFILE" is used to import the rows.

Usage: mysqlimport [OPTIONS] database csvfile...
Default options are read from the following files in the given order:
C:\Windows\my.ini C:\Windows\my.cnf C:\my.ini C:\my.cnf D:\ProgramFiles\MySQL\MySQL Server 8.0\my.cnf
The following groups are read: mysqlimport client
The following options may be given as the first argument:
--print-defaults      Print the program argument list and exit.
--no-defaults        Don't read default options from any option file,
                   except for login file.
--defaults-file=#   Only read default options from the given file #.
--defaults-extra-file=# Read this file after the global files are read.
--defaults-group-suffix=suffix
                   Also read groups with concat(group, suffix)
--login-path#        Read this path from the login file.
--bind-address=name  IP address to bind to.
--character-sets-dir=name
                   Directory for character set files.
--default-character-set=name
                   Set the default character set.
--c, --columns=name  Use only these columns to import the data to. Give the
                   column names in a comma separated list. This is same as
                   giving columns to LOAD DATA INFILE.
-C, --compress       Use compression in server/client protocol.
-#, --debug[=#]       This is a non-debug version. Catch this and exit.
--debug-check        This is a non-debug version. Catch this and exit.
--debug-info         This is a non-debug version. Catch this and exit.
--default-auth=name  Default authentication client-side plugin to use.
-d, --delete         First delete all rows from table.
--enable-cleartext-plugin
                   Enable/disable the clear text authentication plugin.
--fields-terminated-by=name
                   Fields in the input file are terminated by the given
                   string.
--fields-enclosed-by=name
                   Fields in the import file are enclosed by the given

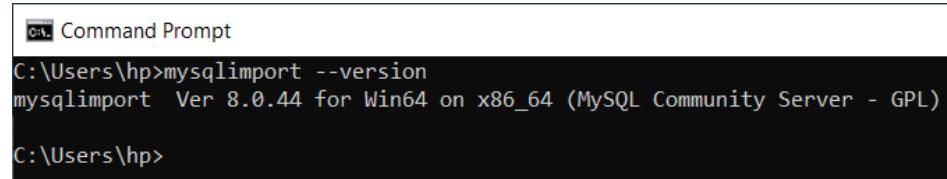
```

For example, use the following commands to get the current mysqlimport client version:

```
mysqlimport -V
```

or

```
mysqlimport --version
```



```
C:\Users\hp>mysqlimport --version
mysqlimport Ver 8.0.44 for Win64 on x86_64 (MySQL Community Server - GPL)
C:\Users\hp>
```

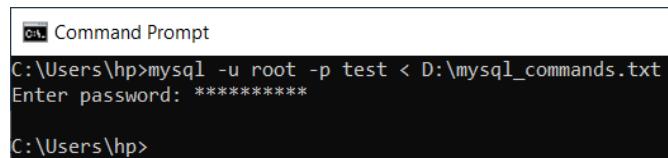
Before importing data from a file, first create the respective table in the database. For this, open D:\mysql_commands.txt file and enter the following commands in the file:

```
create table employee (employee_id int not null, first_name varchar(100),
last_name varchar(100), email varchar(100), phone_number varchar(20),
hire_date date, salary int, manager_id int, department_id int);
```



Then run the below mysql command to create table in test database. Enter root user password when prompted:

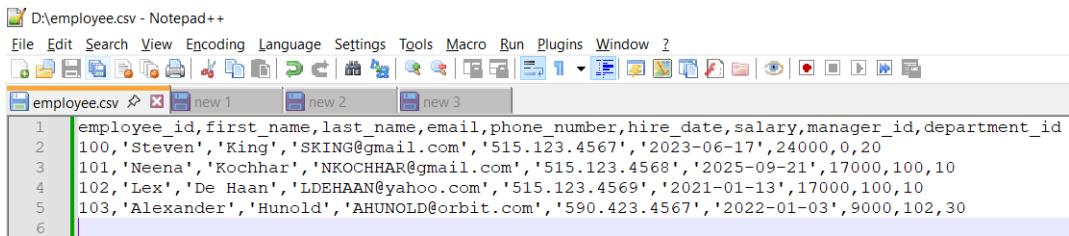
```
mysql -u root -p test < D:\mysql_commands.txt
```



```
C:\Users\hp>mysql -u root -p test < D:\mysql_commands.txt
Enter password: *****
C:\Users\hp>
```

Now, create a sample file named employee.csv file in D:\ and enter the following contents:

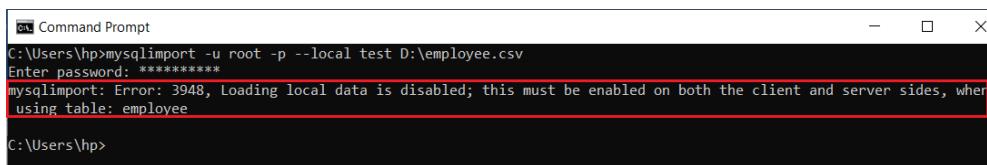
```
employee_id,first_name,last_name,email,phone_number,hire_date,salary,manager
_id,department_id
100,'Steven','King','SKING@gmail.com','515.123.4567','2023-06-17',24000,0,20
101,'Neena','Kochhar','NKOCHHAR@gmail.com','515.123.4568','2025-09-
21',17000,100,10
102,'Lex','De Haan','LDEHAAN@yahoo.com','515.123.4569','2021-01-
13',17000,100,10
103,'Alexander','Hunold','AHUNOLD@orbit.com','590.423.4567','2022-01-
03',9000,102,30
```



D:\employee.csv - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
employee.csv new 1 new 2 new 3
1 employee_id,first_name,last_name,email,phone_number,hire_date,salary,manager_id,department_id
2 100,'Steven','King','SKING@gmail.com','515.123.4567','2023-06-17',24000,0,20
3 101,'Neena','Kochhar','NKOCHHAR@gmail.com','515.123.4568','2025-09-21',17000,100,10
4 102,'Lex','De Haan','LDEHAAN@yahoo.com','515.123.4569','2021-01-13',17000,100,10
5 103,'Alexander','Hunold','AHUNOLD@orbit.com','590.423.4567','2022-01-03',9000,102,30
6

Use the following command to import data from D:\employee.csv file into test database.
Enter root user password when prompted:

```
mysqlimport -u root -p --local test D:\employee.csv
```

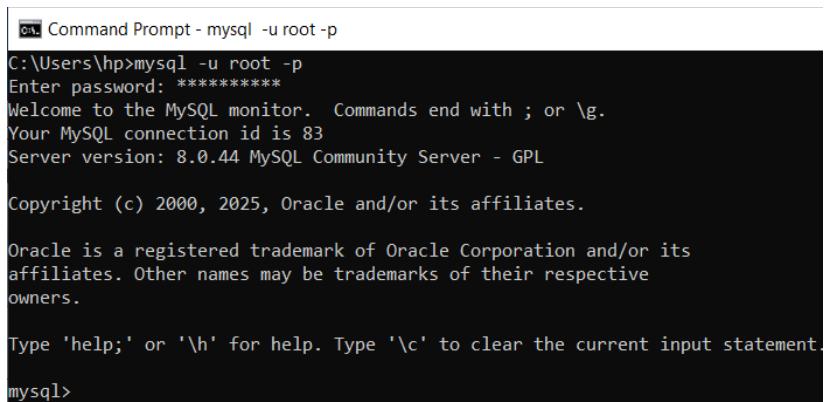


```
C:\Users\hp>mysqlimport -u root -p --local test D:\employee.csv
Enter password: *****
mysqlimport: Error: 3948, Loading local data is disabled; this must be enabled on both the client and server sides, when using table: employee
```

When the above command is executed, it might throw **mysqlimport: Error: 3948, Loading local data is disabled; this must be enabled on both the client and server sides, when using table: employee**. This is because local_infile variable is disabled by default.

First run the following command to connect to mysql client. Enter root user password when prompted:

```
mysql -u root -p
```



```
C:\Users\hp>mysql -u root -p
C:\Users\hp>mysql -u root -p
Enter password: *****
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 83
Server version: 8.0.44 MySQL Community Server - GPL

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

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>
```

Then, run the following SQL command to verify the status of local_infile variable:

```
show global variables like 'local_infile';
```

```
mysql> show global variables like 'local_infile';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| local_infile | OFF |
+-----+-----+
1 row in set (0.01 sec)

mysql>
```

Here, it displays the value OFF which means `local_infile` is disabled.

Use the following command to enable it:

```
set global local_infile=true;
```

```
mysql> set global local_infile=true;
Query OK, 0 rows affected (0.00 sec)

mysql> show global variables like 'local_infile';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| local_infile | ON |
+-----+-----+
1 row in set (0.00 sec)

mysql>
```

Then exit `mysql` client and run the following `mysqlimport` command. Enter `root` user password when prompted:

```
mysqlimport -u root -p --local test D:\employee.csv
```

```
Command Prompt
C:\Users\hp>mysqlimport -u root -p --local test D:\employee.csv
Enter password: *****
test.employee: Records: 5 Deleted: 0 Skipped: 0 Warnings: 45
C:\Users\hp>
```

Use the following `mysqlimport` command to import data from file using specific options. Enter `root` user password when prompted:

```
mysqlimport -u root -p --local --fields-terminated-by="," --lines-terminated-by="\n" test D:\employee.csv
```

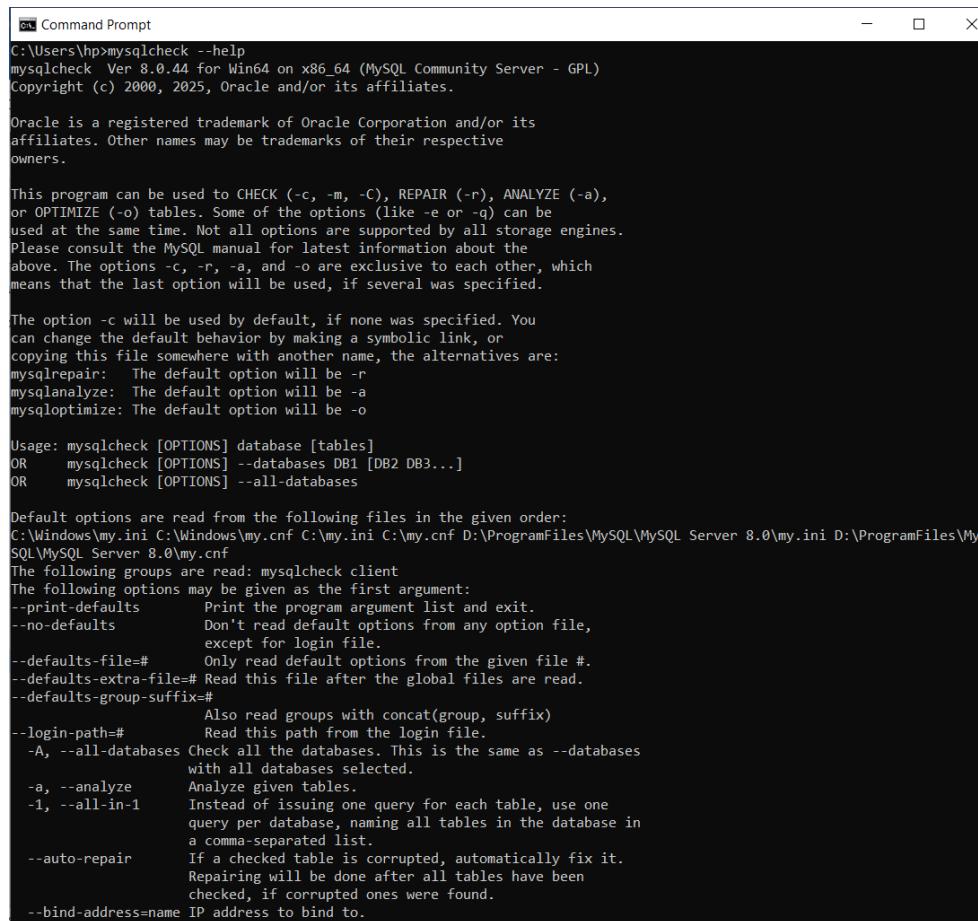
```
Command Prompt
C:\Users\hp>mysqlimport -u root -p --local --fields-terminated-by="," --lines-terminated-by="\n" test D:\employee.csv
Enter password: *****
test.employee: Records: 5 Deleted: 0 Skipped: 0 Warnings: 13
C:\Users\hp>
```

7.6. mysqlcheck:

`mysqlcheck` client is useful to for table maintenance operations such as checking table, repairing, analyzing and optimizing tables.

`mysqlcheck` supports various options which can be specified in the command line. Use the following command to get the list of options supported by `mysqlcheck` and commands to be executed:

```
mysqlcheck --help
```



The screenshot shows a Windows Command Prompt window with the title 'Command Prompt'. The window contains the help output for the mysqlcheck command. It includes the MySQL copyright notice, information about trademarks, a detailed description of the program's purpose and usage, and a comprehensive list of command-line options with their descriptions. The text is white on a black background.

```
C:\Users\hp>mysqlcheck --help
mysqlcheck Ver 8.0.44 for Win64 on x86_64 (MySQL Community Server - GPL)
Copyright (c) 2000, 2025, Oracle and/or its affiliates.

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

This program can be used to CHECK (-c, -m, -C), REPAIR (-r), ANALYZE (-a),
or OPTIMIZE (-o) tables. Some of the options (like -e or -q) can be
used at the same time. Not all options are supported by all storage engines.
Please consult the MySQL manual for latest information about the
above. The options -c, -r, -a, and -o are exclusive to each other, which
means that the last option will be used, if several was specified.

The option -c will be used by default, if none was specified. You
can change the default behavior by making a symbolic link, or
copying this file somewhere with another name, the alternatives are:
mysqlrepair: The default option will be -r
mysqlanalyze: The default option will be -a
mysqloptimize: The default option will be -o

Usage: mysqlcheck [OPTIONS] database [tables]
OR      mysqlcheck [OPTIONS] --databases DB1 [DB2 DB3...]
OR      mysqlcheck [OPTIONS] --all-databases

Default options are read from the following files in the given order:
C:\Windows\my.ini C:\Windows\my.cnf C:\my.ini C:\my.cnf D:\ProgramFiles\MySQL\MySQL Server 8.0\my.ini D:\ProgramFiles\My
SQL\MySQL Server 8.0\my.cnf
The following groups are read: mysqlcheck client
The following options may be given as the first argument:
--print-defaults      Print the program argument list and exit.
--no-defaults        Don't read default options from any option file,
                    except for login file.
--defaults-file=#   Only read default options from the given file #.
--defaults-extra-file=# Read this file after the global files are read.
--defaults-group-suffix=#    Also read groups with concat(group, suffix)
--login-path=#       Read this path from the login file.
-A, --all-databases Check all the databases. This is the same as --databases
with all databases selected.
-a, --analyze        Analyze given tables.
-1, --all-in-1       Instead of issuing one query for each table, use one
query per database, naming all tables in the database in
a comma-separated list.
--auto-repair       If a checked table is corrupted, automatically fix it.
Repairing will be done after all tables have been
checked, if corrupted ones were found.
--bind-address=name IP address to bind to.
```

For example, use the following commands to get the current `mysqlcheck` client version:

```
mysqlcheck -V
```

or

```
mysqlcheck --version
```

```
C:\ Command Prompt
C:\Users\hp>mysqlcheck -V
mysqlcheck Ver 8.0.44 for Win64 on x86_64 (MySQL Community Server - GPL)

C:\Users\hp>
```

Use the following command to check the `employee` table in `test` database for any errors.
Enter `root` user password when prompted:

```
mysqlcheck -u root -p test employee
```

```
C:\ Command Prompt
C:\Users\hp>mysqlcheck -u root -p test employee
Enter password: *****
test.employee                                     OK

C:\Users\hp>
```

Use the following command to analyze tables in all databases. Enter `root` user password when prompted:

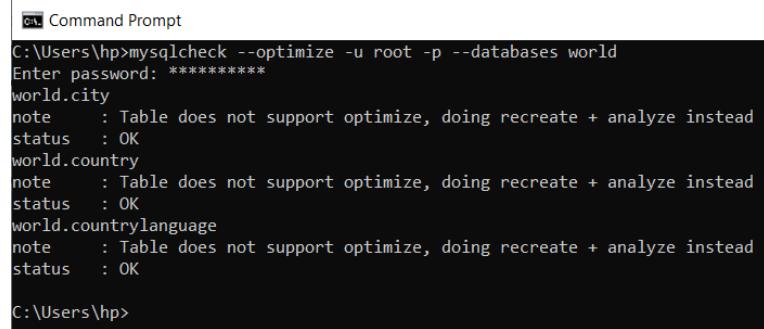
```
mysqlcheck --analyze -u root -p --all-databases
```

```
C:\ Command Prompt
C:\Users\hp>mysqlcheck --analyze -u root -p --all-databases
Enter password: *****
mysql.columns_priv          OK
mysql.component              OK
mysql.db                     OK
mysql.default_roles          OK
mysql.engine_cost            OK
mysql.func                   OK
mysql.general_log            note : The storage engine for the table doesn't support analyze
mysql.global_grants          OK
mysql.gtid_executed          OK
mysql.help_category          OK
mysql.help_keyword            OK
mysql.help_relation           OK
mysql.help_topic              OK
mysql.innodb_index_stats     OK
mysql.innodb_table_stats      OK
mysql.ndb_binlog_index        OK
mysql.password_history        OK
mysql.plugin                  OK
mysql.proc_priv               OK
mysql.proxies_priv            OK
mysql.replication_asynchronous_connection_fallover OK
mysql.replication_asynchronous_connection_fallover_managed OK
mysql.replication_group_configuration_version OK
mysql.replication_group_member_actions OK
mysql.role_edges               OK
mysql.server_cost              OK
mysql.servers                 OK
mysql.slave_master_info        OK
mysql.slave_relay_log_info      OK
mysql.slave_worker_info         OK
mysql.slow_log                note : The storage engine for the table doesn't support analyze
mysql.tables_priv              OK
mysql.time_zone               OK
mysql.time_zone_leap_second    OK
mysql.time_zone_name           OK
mysql.time_zone_transition     OK
mysql.time_zone_transition_type OK
mysql.user                     OK
sakila.actor                  OK
sakila.address                OK
sakila.category                OK
sakila.city                    OK
sakila.country                 OK
sakila.customer                OK
sakila.film                    OK
sakila.film_actor              OK
sakila.film_category           OK
sakila.film_text                OK
sakila.inventory                OK
sakila.language                 OK
sakila.payment                  OK
sakila.rental                   OK
sakila.staff                   OK
sakila.store                   OK
sys.sys_config                 OK
test.employee                  OK
world.city                     OK
world.country                  OK
world.countrylanguage           OK

C:\Users\hp>
```

Use the following command to optimize tables in `world` database. Enter `root` user password when prompted:

```
mysqlcheck --optimize -u root -p --databases world
```

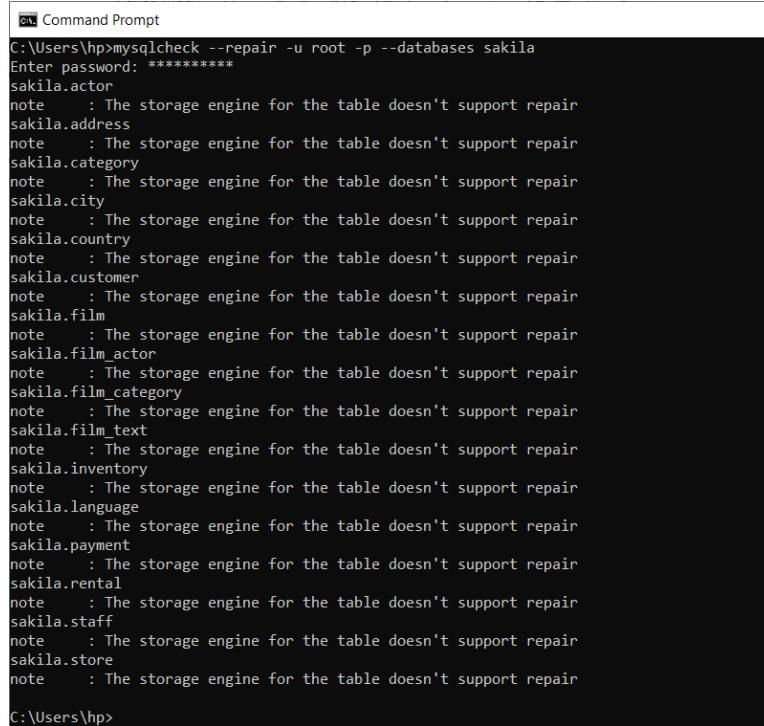


```
cmd Command Prompt
C:\Users\hp>mysqlcheck --optimize -u root -p --databases world
Enter password: *****
world.city
note      : Table does not support optimize, doing recreate + analyze instead
status    : OK
world.country
note      : Table does not support optimize, doing recreate + analyze instead
status    : OK
world.countrylanguage
note      : Table does not support optimize, doing recreate + analyze instead
status    : OK

C:\Users\hp>
```

Use the following command to repair tables in `sakila` database. Enter `root` user password when prompted:

```
mysqlcheck --repair -u root -p --databases sakila
```



```
cmd Command Prompt
C:\Users\hp>mysqlcheck --repair -u root -p --databases sakila
Enter password: *****
sakila.actor
note      : The storage engine for the table doesn't support repair
sakila.address
note      : The storage engine for the table doesn't support repair
sakila.category
note      : The storage engine for the table doesn't support repair
sakila.city
note      : The storage engine for the table doesn't support repair
sakila.country
note      : The storage engine for the table doesn't support repair
sakila.customer
note      : The storage engine for the table doesn't support repair
sakila.film
note      : The storage engine for the table doesn't support repair
sakila.film_actor
note      : The storage engine for the table doesn't support repair
sakila.film_category
note      : The storage engine for the table doesn't support repair
sakila.film_text
note      : The storage engine for the table doesn't support repair
sakila.inventory
note      : The storage engine for the table doesn't support repair
sakila.language
note      : The storage engine for the table doesn't support repair
sakila.payment
note      : The storage engine for the table doesn't support repair
sakila.rental
note      : The storage engine for the table doesn't support repair
sakila.staff
note      : The storage engine for the table doesn't support repair
sakila.store
note      : The storage engine for the table doesn't support repair

C:\Users\hp>
```

8. MySQL Shell:

MySQL Shell (`mysqlsh`) is an advanced command line interface to work with MySQL instances. It works similar to `mysql` to run **SQL** commands but it also offers scripting capabilities for **JavaScript** and **Python** languages. In addition to traditional database management, MySQL Shell provides APIs such as the **X DevAPI** for developing applications in JavaScript or Python that treat MySQL as both a relational and document store, and the **AdminAPI** for administering MySQL instances and configuring high-availability solutions like InnoDB Cluster, InnoDB ClusterSet, and InnoDB ReplicaSet by facilitating automated setup, provisioning, monitoring, etc.

MySQL Shell can execute code in three modes – JavaScript or Python or SQL – and process code either interactively or non-interactively (batch processing). By default, it processes the code in JavaScript mode which can be changed in two ways:

- Using `--sql` (for SQL mode) or `--py` (for Python mode) or `--js` (for JavaScript mode) options while starting `mysqlsh`.
- Using `\sql` or `\py` or `\js` commands after entering into MySQL Shell.

While MySQL Shell is installed along with MySQL Server, it can also be installed as a standalone tool on client machines from the [MySQL Community Downloads](#) page.

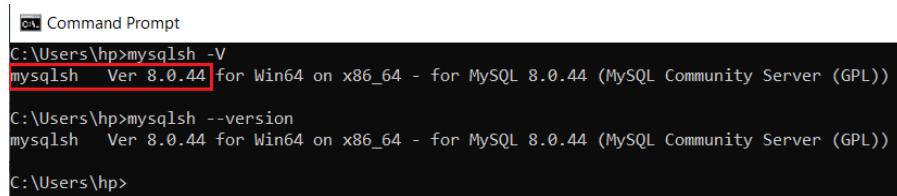
8.1. Verify MySQL Shell Installation:

Verify if `mysqlsh` has been installed successfully using the following command:

```
mysqlsh -V
```

or

```
mysqlsh --version
```

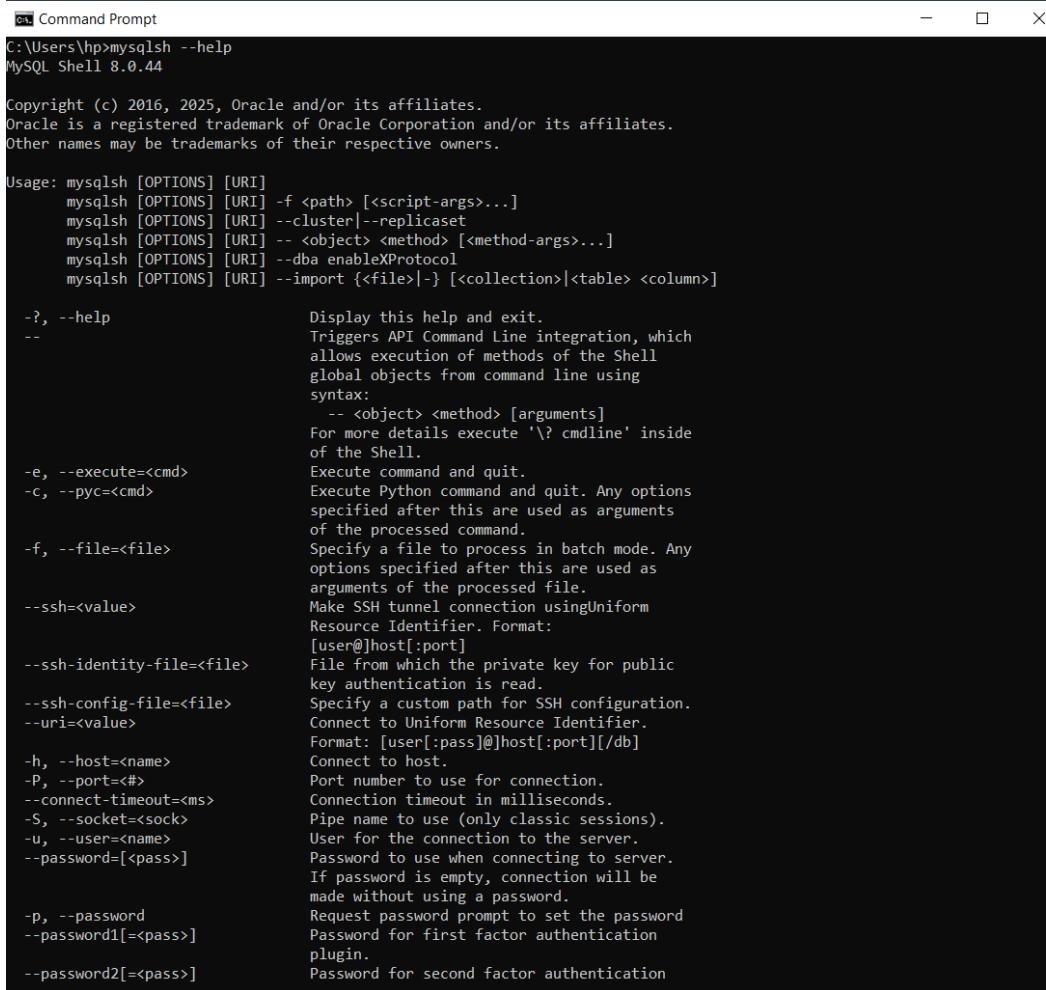


The screenshot shows a Windows Command Prompt window titled "Command Prompt". It contains two lines of text:
C:\Users\hp>mysqlsh -V
mysqlsh Ver 8.0.44 for Win64 on x86_64 - for MySQL 8.0.44 (MySQL Community Server (GPL))
C:\Users\hp>mysqlsh --version
mysqlsh Ver 8.0.44 for Win64 on x86_64 - for MySQL 8.0.44 (MySQL Community Server (GPL))
C:\Users\hp>

Here, it displays the `mysqlsh` version 8.0.44 version that was installed on the Windows system.

`mysqlsh` supports various options which can be specified in the command line. Use the following command to get the list of options supported by `mysqlsh`.

```
mysql --help
```



```
C:\ Command Prompt
C:\Users\hp>mysqlsh --help
MySQL Shell 8.0.44

Copyright (c) 2016, 2025, Oracle and/or its affiliates.
Oracle is a registered trademark of Oracle Corporation and/or its affiliates.
Other names may be trademarks of their respective owners.

Usage: mysqlsh [OPTIONS] [URI]
  mysqlsh [OPTIONS] [URI] -f <path> [<script-args>...]
  mysqlsh [OPTIONS] [URI] --cluster|--replicaset
  mysqlsh [OPTIONS] [URI] -- <object> <method> [<method-args>...]
  mysqlsh [OPTIONS] [URI] --dba enableXProtocol
  mysqlsh [OPTIONS] [URI] --import <{file|-}> [<collection>|<table> <column>]

-?, --help           Display this help and exit.
--                Triggers API Command Line integration, which
                allows execution of methods of the Shell
                global objects from command line using
                syntax:
                  -- <object> <method> [arguments]
                For more details execute '? cmdline' inside
                of the Shell.
-e, --execute=<cmd> Execute command and quit.
-c, --pyc=<cmd>     Execute Python command and quit. Any options
                    specified after this are used as arguments
                    of the processed command.
-f, --file=<file>   Specify a file to process in batch mode. Any
                    options specified after this are used as
                    arguments of the processed file.
--ssh=<value>      Make SSH tunnel connection using Uniform
                    Resource Identifier. Format:
                      [user@host[:port]]
                    File from which the private key for public
                    key authentication is read.
--ssh-identity-file=<file> Specify a custom path for SSH configuration.
--ssh-config-file=<file>
--uri=<value>        Connect to Uniform Resource Identifier.
                    Format: [user[:pass]@]host[:port][/:db]
--host=<name>       Connect to host.
--port=<#>           Port number to use for connection.
--connect-timeout=<ms> Connection timeout in milliseconds.
--socket=<sock>     Pipe name to use (only classic sessions).
--user=<name>        User for the connection to the server.
--password=[<pass>]  Password to use when connecting to server.
                    If password is empty, connection will be
                    made without using a password.
-p, --password      Request password prompt to set the password
--password1[=<pass>] Password for first factor authentication
plugin.
--password2[=<pass>] Password for second factor authentication
```

8.2. Start MySQL Shell:

Run the following command to start MySQL Shell:

```
mysqlsh
```



```
C:\ Command Prompt - mysqlsh
C:\Users\hp>mysqlsh
MySQL Shell 8.0.44

Copyright (c) 2016, 2025, Oracle and/or its affiliates.
Oracle is a registered trademark of Oracle Corporation and/or its affiliates.
Other names may be trademarks of their respective owners.

Type '\help' or '\?' for help; '\quit' to exit.
MySQL JS >
```

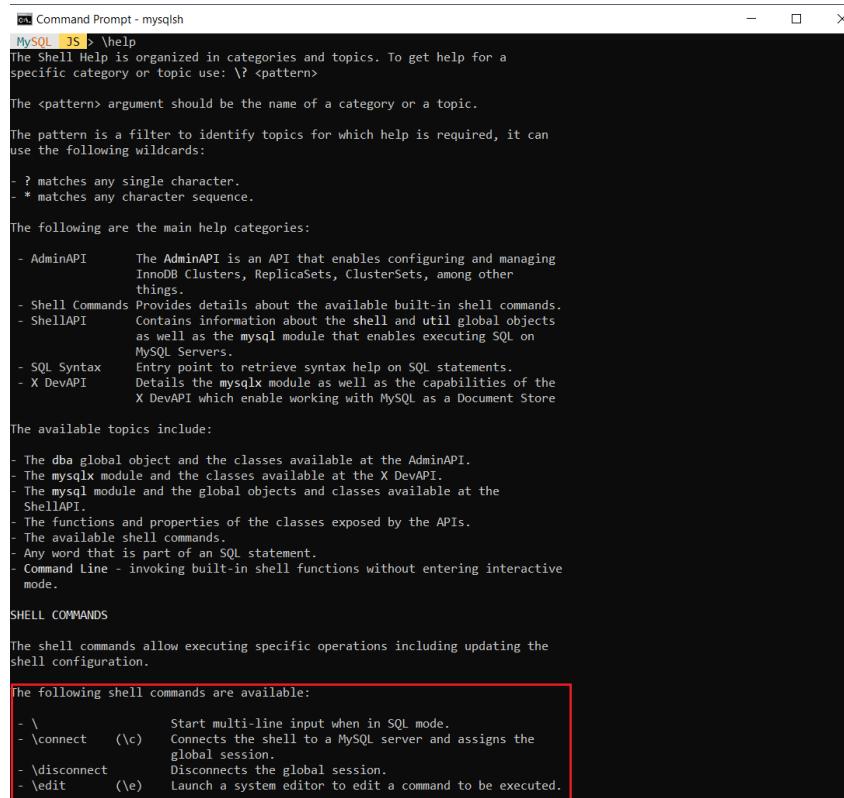
Once MySQL Shell is successfully connected, it displays **MySQL JS >** prompt which executes JavaScript code in interactive mode.

8.3. MySQL Shell Commands:

MySQL Shell provides many built-in commands to perform tasks like connecting to a MySQL instance, switching current scripting language, running reports, utilizing various utilities, etc. To ensure these commands operate independently of current scripting execution mode, they must always be prefixed with the backslash (\) escape character.

Use the following basic command to get more help on mysqlsh usage and available shell commands:

```
\?  
or  
\h  
or  
\help
```



The screenshot shows the MySQL Shell Command Prompt window. The command entered is '\help'. The output provides general help about the shell, including categories like AdminAPI, Shell Commands, SQL Syntax, X DevAPI, and available topics. It also lists shell commands such as \, \connect, \disconnect, and \edit. A red box highlights the section on shell commands.

```
MySQL [JS] > \help
The Shell Help is organized in categories and topics. To get help for a specific category or topic use: \? <pattern>
The <pattern> argument should be the name of a category or a topic.
The pattern is a filter to identify topics for which help is required, it can use the following wildcards:
- ? matches any single character.
- * matches any character sequence.

The following are the main help categories:
- AdminAPI      The AdminAPI is an API that enables configuring and managing InnoDB Clusters, ReplicaSets, ClusterSets, among other things.
- Shell Commands Provides details about the available built-in shell commands.
- ShellAPI       Contains information about the shell and util global objects as well as the mysql module that enables executing SQL on MySQL Servers.
- SQL Syntax    Entry point to retrieve syntax help on SQL statements.
- X DevAPI      Details the mysqlx module as well as the capabilities of the X DevAPI which enable working with MySQL as a Document Store

The available topics include:
- The dba global object and the classes available at the AdminAPI.
- The mysqlx module and the classes available at the X DevAPI.
- The mysql module and the global objects and classes available at the ShellAPI.
- The functions and properties of the classes exposed by the APIs.
- The available shell commands.
- Any word that is part of an SQL statement.
- Command Line - invoking built-in shell functions without entering interactive mode.

SHELL COMMANDS

The shell commands allow executing specific operations including updating the shell configuration.

The following shell commands are available:
- \           Start multi-line input when in SQL mode.
- \connect    (\c)  Connects the shell to a MySQL server and assigns the global session.
- \disconnect Disconnects the global session.
- \edit      (\e)  Launch a system editor to edit a command to be executed.
```

Following are some common commands being used:

- **\? or \h or \help:** Displays general help about MySQL Shell, or allows to search the online documentation.
- **\q or \quit or \exit:** Exits MySQL Shell.
- **\c or \connect:** Connects to a MySQL instance.
- **\reconnect:** Reconnects to the same MySQL instance.

- `\disconnect`: Disconnects the MySQL instance.
- `\s` or `\status`: Displays the current MySQL Shell status.
- `\js`: Switches execution mode to JavaScript.
- `\py`: Switches execution mode to Python.
- `\sql`: Switches execution mode to SQL.
- `\:`: Allows to write multi-line code in SQL mode.
- `\u` or `\use`: Allows to specify schema to use.
- `\.` or `\source`: Executes script file using the active language.
- `\history`: Allows to view and edit command line history.
- `\option`: Modifies MySQL Shell configuration options.
- `\show`: Runs the specified report using the provided options and arguments.
- `!` or `\system`: Executes the specified shell command.

To get more help of each of these commands, prefix the command name with `\?` or `\h` or `\help`.

For example, use the following command to get more help on `\connect` command:

```
\? \connect
```

```
MySQL JS > \? \connect
NAME
  \connect - Connects the shell to a MySQL server and assigns the global
  session.

SYNTAX
  \connect [<TYPE>] <URI>
  \c [<TYPE>] <URI>

DESCRIPTION
  TYPE is an optional parameter to specify the session type. Accepts the
  following values:
  - --mc, --mysql: create a classic MySQL protocol session (default port
    3306)
  - --mx, --mysqlx: create an X protocol session (default port 33060)
  - --ssh <SSHURI>: create an SSH tunnel to use as a gateway for db
    connection. This requires that db port is specified in advance.

  If TYPE is omitted, automatic protocol detection is done, unless the
  protocol is given in the URI.

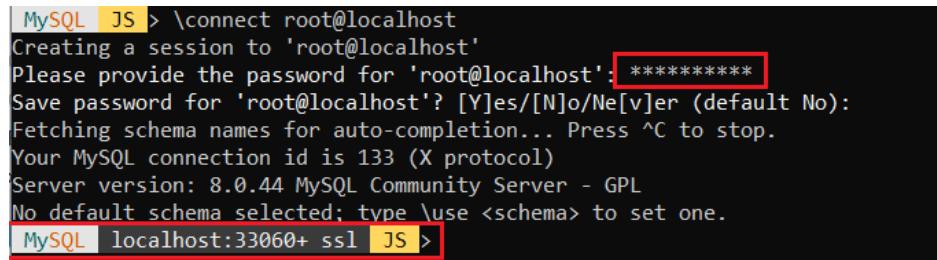
  URI format is: [user[:password]@]hostname[:port] and SSHURI format is:
  [user@]hostname[:port]

EXAMPLE
  \connect --mx root@localhost
  Creates a global session using the X protocol to the indicated URI.
MySQL JS >
```

8.4. Connect to MySQL Instance:

In MySQL Shell, run the following command to connect to a MySQL instance. Enter `root` user password when prompted after which it asks to save password for the user (*Enter Y or N if you wish to save or not*):

```
\connect root@localhost
```



The screenshot shows the MySQL Shell interface. The prompt is `MySQL JS >`. The user runs the command `\connect root@localhost`. The shell creates a session for 'root@localhost'. It prompts for the password, which is shown as a redacted string. It then asks if the user wants to save the password ('Y' or 'N'). The shell then fetches schema names for auto-completion. It displays the MySQL connection ID (133), the server version (8.0.44), and the fact that no default schema is selected. Finally, the prompt changes to `MySQL localhost:33060+ ssl JS >`, indicating a successful connection.

After successfully connected, the prompt will change to `MySQL localhost:33060+ ssl JS >` to run commands.

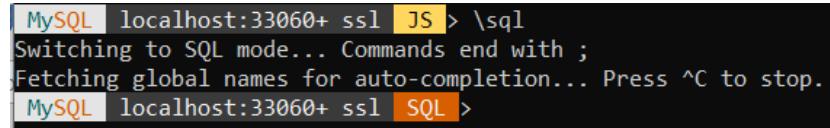
Note that `mysqlsh` also allows to connect to a MySQL instance before starting the shell. For example, use the below command to connect to MySQL and switch to SQL mode:

```
mysqlsh -h localhost -u root --sql
```

8.5. Run SQL Commands:

In MySQL Shell, first run the following command to switch to SQL execution mode:

```
\sql
```



The screenshot shows the MySQL Shell interface. The prompt is `MySQL localhost:33060+ ssl JS >`. The user runs the command `\sql`. The shell switches to SQL mode, indicating that commands end with a semicolon. It then fetches global names for auto-completion. Finally, the prompt changes to `MySQL localhost:33060+ ssl SQL >`.

Now the prompt changes to `MySQL localhost:33060+ ssl SQL >` to run SQL commands terminating with semi-colon.

Run the following command to list the available databases:

```
show databases;
```

```
MySQL [localhost:33060+ ssl] [SQL] > show databases;
+-----+
| Database |
+-----+
| database |
| information_schema |
| mysql |
| performance_schema |
| sakila |
| sys |
| test |
| world |
+-----+
8 rows in set (0.0018 sec)
MySQL [localhost:33060+ ssl] [SQL] >
```

Run the following command to see the list of tables in the `test` database:

```
show tables from test;
```

```
MySQL [localhost:33060+ ssl] [SQL] > show tables from test;
+-----+
| Tables_in_test |
+-----+
| employee |
+-----+
1 row in set (0.0025 sec)
MySQL [localhost:33060+ ssl] [SQL] >
```

Run the following command to see the list of columns in the `employee` table in `test` database:

```
show columns from test.employee;
```

```
MySQL [localhost:33060+ ssl] [SQL] > show columns from test.employee;
+-----+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| employee_id | int    | NO   |   | NULL    |       |
| first_name   | varchar(100) | YES  |   | NULL    |       |
| last_name    | varchar(100) | YES  |   | NULL    |       |
| email         | varchar(100) | YES  |   | NULL    |       |
| phone_number  | varchar(20)  | YES  |   | NULL    |       |
| hire_date     | date    | YES  |   | NULL    |       |
| salary        | int    | YES  |   | NULL    |       |
| manager_id    | int    | YES  |   | NULL    |       |
| department_id | int    | YES  |   | NULL    |       |
+-----+-----+-----+-----+-----+-----+
9 rows in set (0.0026 sec)
MySQL [localhost:33060+ ssl] [SQL] >
```

8.6. Run Python Code:

In MySQL Shell, first run the following command to switch to Python execution mode:

```
\python
```

```
MySQL localhost:33060+ ssl SQL > \py
Switching to Python mode...
MySQL localhost:33060+ ssl Py >
```

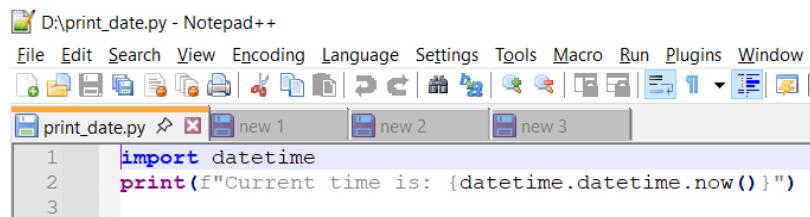
Now the prompt changes to MySQL localhost:33060+ ssl Py > to run Python code.

Run the following code to display current date and time:

```
import datetime
print(f"Current time is: {datetime.datetime.now()}")
```

```
MySQL localhost:33060+ ssl Py > import datetime
MySQL localhost:33060+ ssl Py > print(f"Current time is: {datetime.datetime.now()}")
Current time is: 2026-01-06 11:49:23.905700
MySQL localhost:33060+ ssl Py >
```

Instead of executing Python code line by line, MySQL Shell can execute the Python file directly. First save the above code in print_date.py file in D drive.



Run the following command in MySQL Shell to execute the D:\print_date.py file:

```
\. D:\print_date.py
```

```
MySQL localhost:33060+ ssl Py > \. D:\print_date.py
Current time is: 2026-01-06 12:11:49.167570
MySQL localhost:33060+ ssl Py >
```

Note that mysqlsh also allows to execute Python script without starting shell.

For example, use any of the below commands to run D:\print_date.py file using mysqlsh:

```
mysqlsh --py < D:\print_date.py
```

or

```
mysqlsh --py --file D:\print_date.py
```

While remaining in Python mode in MySQL Shell, single SQL query can be executed by prefixing with \sql before the SQL command as below:

```
\sql show tables from world;
```

```
MySQL localhost:33060+ ssl Py > \sql show tables from world;
Fetching global names for auto-completion... Press ^C to stop.
+-----+
| Tables_in_world |
+-----+
| city           |
| country        |
| countrylanguage |
+-----+
3 rows in set (0.0172 sec)
MySQL localhost:33060+ ssl Py >
```

8.7. Run JavaScript Code:

In MySQL Shell, first run the following command to switch to JavaScript execution mode:

```
\js
```

```
MySQL localhost:33060+ ssl Py > \js
Switching to JavaScript mode
MySQL localhost:33060+ ssl JS >
```

Now the prompt changes to MySQL localhost:33060+ ssl JS > to run JavaScript code.

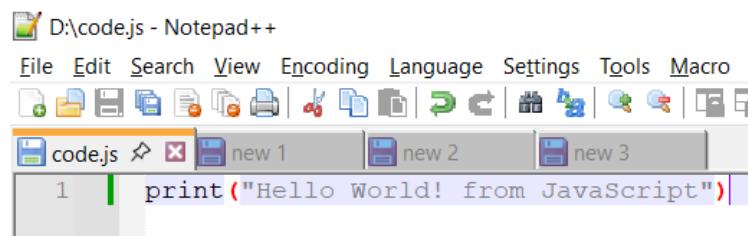
Run the following code to display a message:

```
print("Hello World! from JavaScript")
```

```
MySQL localhost:33060+ ssl JS > print("Hello World! from JavaScript")
Hello World! from JavaScript
MySQL localhost:33060+ ssl JS >
```

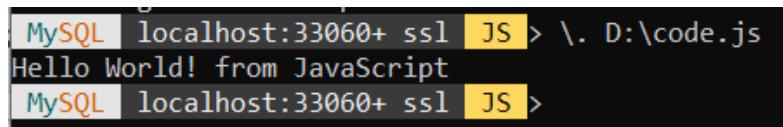
Instead of executing JavaScript code line by line, MySQL Shell can execute a JavaScript file directly.

First save the above code in code.js file in D drive.



Run the following command in MySQL Shell to execute the D:\code.js file:

```
\. D:\code.js
```



```
MySQL [localhost:33060+ ssl] JS > \. D:\code.js
Hello World! from JavaScript
MySQL [localhost:33060+ ssl] JS >
```

Note that mysqlsh also allows to execute JavaScript file without starting shell.

For example, use any of the below commands to run D:\code.js file using mysqlsh:

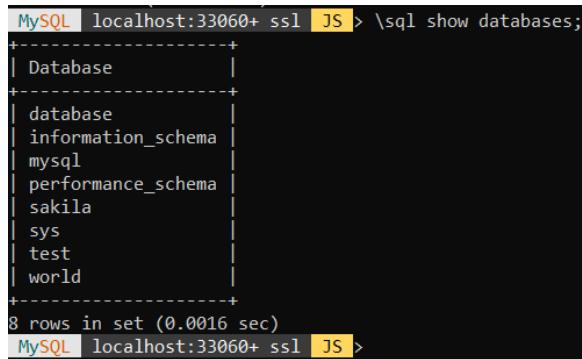
```
mysqlsh --js < D:\code.js
```

or

```
mysqlsh --js --file D:\code.js
```

While remaining in JavaScript mode in MySQL Shell, single SQL query can be executed by prefixing with \sql before the SQL command as below:

```
\sql show databases;
```



```
MySQL [localhost:33060+ ssl] JS > \sql show databases;
+-----+
| Database      |
+-----+
| database     |
| information_schema |
| mysql        |
| performance_schema |
| sakila       |
| sys          |
| test         |
| world        |
+-----+
8 rows in set (0.0016 sec)
MySQL [localhost:33060+ ssl] JS >
```

8.8. MySQL Shell Utilities:

MySQL Shell provides built-in utilities that are useful to perform various administrative and maintenance tasks such as data import/export, diagnostics collection and server upgrades. These utilities are accessible through global object called `util` which is available in JavaScript and Python modes only.

8.8.1. Dump Utilities:

Dump utilities include `util.dumpInstance()` which exports data from a MySQL instance, `util.dumpSchemas()` which exports data from specific schemas, and `util.dumpTables()` which exports data from selected tables or views into a set of local

files or an Oracle Cloud Infrastructure (OCI) Object Storage bucket. For complete details, refer to the [official MySQL documentation](#).

Follow the below syntax to use these dump utilities:

```
util.dumpInstance(outputUrl[, options])
util.dumpSchemas(schemas, outputUrl[, options])
util.dumpTables(schema, tables, outputUrl[, options])
```

For example, run the following command to perform a dry run to check for any compatibility issues before performing a dump to D:\\mysql_instance_dump directory using dumpInstance utility:

```
util.dumpInstance("D:/mysql_instance_dump", {dryRun: true})
```

```
MySQL [localhost:33060+ ssl] JS > util.dumpInstance("D:/mysql_instance_dump", {dryRun: true})
dryRun enabled, no locks will be acquired and no files will be created.
Acquiring global read lock
Global read lock acquired
Initializing - done
3 out of 7 schemas will be dumped and within them 19 tables, 7 views, 6 routines, 6 triggers.
1 out of 4 users will be dumped.
Gathering information - done
All transactions have been started
Locking instance for backup
Global read lock has been released
Writing global DDL files
Writing users DDL
Writing DDL - done
Starting data dump
0% (0 rows / ~52.46K rows), 0.00 rows/s, 0.00 B/s uncompressed, 0.00 B/s compressed
MySQL [localhost:33060+ ssl] JS >
```

Since there are no issues reported, it is ready for actual data dump. Run the following command to data dump to a local D:\\mysql_instance_dump directory:

```
util.dumpInstance("D:/mysql_instance_dump")
```

```

MySQL [localhost:33060+ ssl] JS > util.dumpInstance("D:/mysql_instance_dump")
Acquiring global read lock
Global read lock acquired
Initializing - done
3 out of 7 schemas will be dumped and within them 19 tables, 7 views, 6 routines, 6 triggers.
1 out of 4 users will be dumped.
Gathering information - done
All transactions have been started
Locking instance for backup
Global read lock has been released
Writing global DDL files
Writing users DDL
Running data dump using 4 threads.
NOTE: Progress information uses estimated values and may not be accurate.
Writing schema metadata - done
Writing DDL - done
Writing table metadata - done
Starting data dump
100% (52.57K rows / ~52.46K rows), 0.00 rows/s, 0.00 B/s uncompressed, 0.00 B/s compressed
Dump duration: 00:00:00
Total duration: 00:00:00s
Schemas dumped: 3
Tables dumped: 19
Uncompressed data size: 3.23 MB
Compressed data size: 802.72 KB
Compression ratio: 4.0
Rows written: 52570
Bytes written: 802.72 KB
Average uncompressed throughput: 3.23 MB/s
Average compressed throughput: 802.72 KB/s
MySQL [localhost:33060+ ssl] JS >

```

Once data dump is completed, it creates `mysql_instance_dump` folder in D drive with all necessary data that can imported to another MySQL instance.

Name	Date modified	Type	Size
@.done.json	1/6/2026 1:23 PM	JSON Source File	2 KB
@.json	1/6/2026 1:23 PM	JSON Source File	1 KB
@.post.sql	1/6/2026 1:23 PM	SQL Text File	1 KB
@.sql	1/6/2026 1:23 PM	SQL Text File	1 KB
@.users.sql	1/6/2026 1:23 PM	SQL Text File	2 KB
database.json	1/6/2026 1:23 PM	JSON Source File	1 KB
database.sql	1/6/2026 1:23 PM	SQL Text File	1 KB
sakila.json	1/6/2026 1:23 PM	JSON Source File	2 KB
sakila.sql	1/6/2026 1:23 PM	SQL Text File	13 KB
sakila@actor.json	1/6/2026 1:23 PM	JSON Source File	1 KB
sakila@actor.sql	1/6/2026 1:23 PM	SQL Text File	1 KB
sakila@actor@0.tsv.zst	1/6/2026 1:23 PM	WinRAR archive	2 KB
sakila@actor@0.tsv.zst.idx	1/6/2026 1:23 PM	IDX File	1 KB
sakila@actor_info.pre.sql	1/6/2026 1:23 PM	SQL Text File	1 KB
sakila@actor_info.sql	1/6/2026 1:23 PM	SQL Text File	2 KB
sakila@address.json	1/6/2026 1:23 PM	JSON Source File	1 KB
sakila@address.sql	1/6/2026 1:23 PM	SQL Text File	2 KB
sakila@address@0.tsv.zst	1/6/2026 1:23 PM	WinRAR archive	30 KB

Run the following command to import `sakila` and `world` databases to `D:\mysql_sakila_world_dump` directory using `dumpSchemas` utility:

```
util.dumpSchemas(["sakila","world"],"D:/mysql_sakila_world_dump")
```

```

MySQL [localhost:33060+ ssl] JS > util.dumpSchemas(["sakila","world"],"D:/mysql_sakila_world_dump")
Acquiring global read lock
Global read lock acquired
Initializing - done
2 schemas will be dumped and within them 19 tables, 7 views, 6 routines, 6 triggers.
Gathering information - done
All transactions have been started
Locking instance for backup
Global read lock has been released
Writing global DDL files
Running data dump using 4 threads.
NOTE: Progress information uses estimated values and may not be accurate.
Writing schema metadata - done
Writing DDL - done
Writing table metadata - done
Starting data dump
100% (52.57K rows / ~52.46K rows), 0.00 rows/s, 0.00 B/s uncompressed, 0.00 B/s compressed
Dump duration: 00:00:00s
Total duration: 00:00:00s
Schemas dumped: 2
Tables dumped: 19
Uncompressed data size: 3.23 MB
Compressed data size: 802.72 KB
Compression ratio: 4.0
Rows written: 52570
Bytes written: 802.72 KB
Average uncompressed throughput: 3.23 MB/s
Average compressed throughput: 802.72 KB/s
MySQL [localhost:33060+ ssl] JS >

```

Once data dump is completed, it creates `mysql_sakila_world_dump` folder in D drive with all necessary data that can imported to another MySQL instance.

This PC > Local Disk (D:) > mysql_sakila_world_dump				
	Name	Date modified	Type	Size
	@.done.json	1/6/2026 1:30 PM	JSON Source File	2 KB
	@.json	1/6/2026 1:30 PM	JSON Source File	1 KB
	@.post.sql	1/6/2026 1:30 PM	SQL Text File	1 KB
	@.sql	1/6/2026 1:30 PM	SQL Text File	1 KB
	sakila.json	1/6/2026 1:30 PM	JSON Source File	2 KB
	sakila.sql	1/6/2026 1:30 PM	SQL Text File	13 KB
	sakila@actor.json	1/6/2026 1:30 PM	JSON Source File	1 KB
	sakila@actor.sql	1/6/2026 1:30 PM	SQL Text File	1 KB
	sakila@actor@@0.tsv.zst	1/6/2026 1:30 PM	WinRAR archive	2 KB
	sakila@actor@@0.tsv.zst.idx	1/6/2026 1:30 PM	IDX File	1 KB
	sakila@actor_info.pre.sql	1/6/2026 1:30 PM	SQL Text File	1 KB
	sakila@actor_info.sql	1/6/2026 1:30 PM	SQL Text File	2 KB
	sakila@address.json	1/6/2026 1:30 PM	JSON Source File	1 KB
	sakila@address.sql	1/6/2026 1:30 PM	SQL Text File	2 KB
	sakila@address@@0.tsv.zst	1/6/2026 1:30 PM	WinRAR archive	30 KB
	sakila@address@@0.tsv.zst.idx	1/6/2026 1:30 PM	IDX File	1 KB
	sakila@category.json	1/6/2026 1:30 PM	JSON Source File	1 KB
	sakila@category.sql	1/6/2026 1:30 PM	SQL Text File	1 KB

Run the following command to import `city` and `country` tables in the `world` database to `D:\mysql_city_country_dump` directory using `dumpTables` utility:

```

util.dumpTables("world", ["city","country"],
"D:/mysql_city_country_dump")

```

```

[MySQL] localhost:33060+ ssl [JS] > util.dumpTables("world", ["city", "country"], "D:/mysql_city_country_dump")
Acquiring global read lock
Global read lock acquired
Initializing - done
2 tables and 0 views will be dumped.
Gathering information - done
All transactions have been started
Locking instance for backup
Global read lock has been released
Writing global DDL files
Running data dump using 4 threads.
NOTE: Progress information uses estimated values and may not be accurate.
Writing schema metadata - done
Writing DDL - done
Writing table metadata - done
Starting data dump
101% (4.32K rows / ~4.27K rows), 0.00 rows/s, 0.00 B/s uncompressed, 0.00 B/s compressed
Dump duration: 00:00:00s
Total duration: 00:00:00s
Schemas dumped: 1
Tables dumped: 2
Uncompressed data size: 176.37 KB
Compressed data size: 82.92 KB
Compression ratio: 2.1
Rows written: 4318
Bytes written: 82.92 KB
Average uncompressed throughput: 176.37 KB/s
Average compressed throughput: 82.92 KB/s
[MySQL] localhost:33060+ ssl [JS] >

```

Once data dump is completed, it creates `mysql_city_country_dump` folder in D drive with all necessary data that can imported to another MySQL instance.

This PC > Local Disk (D:) > mysql_city_country_dump				
	Name	Date modified	Type	Size
	@.done.json	1/6/2026 1:47 PM	JSON Source File	1 KB
	@.json	1/6/2026 1:47 PM	JSON Source File	1 KB
	@.post.sql	1/6/2026 1:47 PM	SQL Text File	1 KB
	@.sql	1/6/2026 1:47 PM	SQL Text File	1 KB
	world.json	1/6/2026 1:47 PM	JSON Source File	1 KB
	world.sql	1/6/2026 1:47 PM	SQL Text File	1 KB
	world@city.json	1/6/2026 1:47 PM	JSON Source File	1 KB
	world@city.sql	1/6/2026 1:47 PM	SQL Text File	1 KB
	world@city@@0.tsv.zst	1/6/2026 1:47 PM	WinRAR archive	68 KB
	world@city@@0.tsv.zst.idx	1/6/2026 1:47 PM	IDX File	1 KB
	world@country.json	1/6/2026 1:47 PM	JSON Source File	1 KB
	world@country.sql	1/6/2026 1:47 PM	SQL Text File	2 KB
	world@country@@0.tsv.zst	1/6/2026 1:47 PM	WinRAR archive	14 KB
	world@country@@0.tsv.zst.idx	1/6/2026 1:47 PM	IDX File	1 KB

8.8.2. Load Utility:

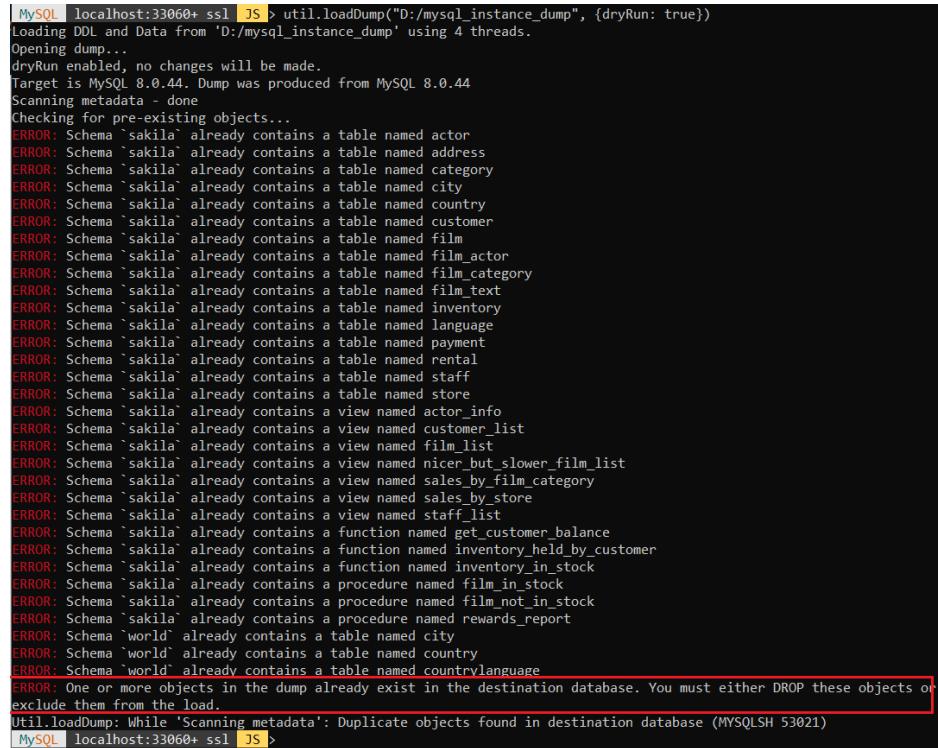
Loading utility include `util.loadDump()` which imports data that was exported using the dump utilities. It supports parallel loading of table chunks, progress tracking, and the ability to resume partially completed imports. For complete details, refer to the [official MySQL documentation](#).

Follow the below syntax to use this load utility:

```
util.loadDump(url[, options])
```

For example, run the following command to perform a dry run to check for any compatibility issues before performing load:

```
util.loadDump("D:/mysql_instance_dump", {dryRun: true})
```



The screenshot shows a terminal window with the MySQL command-line client. The user runs the command `util.loadDump("D:/mysql_instance_dump", {dryRun: true})`. The output shows numerous errors indicating that various tables and views already exist in the target database (MySQL 8.0.44) and cannot be created again. A red box highlights the final error message:

```
MySQL [localhost:33060+ ssl] JS > util.loadDump("D:/mysql_instance_dump", {dryRun: true})
Loading DDL and Data from 'D:/mysql_instance_dump' using 4 threads.
Opening dump...
dryRun enabled, no changes will be made.
Target is MySQL 8.0.44. Dump was produced from MySQL 8.0.44
Scanning metadata - done
Checking for pre-existing objects...
ERROR: Schema `sakila` already contains a table named actor
ERROR: Schema `sakila` already contains a table named address
ERROR: Schema `sakila` already contains a table named category
ERROR: Schema `sakila` already contains a table named city
ERROR: Schema `sakila` already contains a table named country
ERROR: Schema `sakila` already contains a table named customer
ERROR: Schema `sakila` already contains a table named film
ERROR: Schema `sakila` already contains a table named film_actor
ERROR: Schema `sakila` already contains a table named film_category
ERROR: Schema `sakila` already contains a table named film_text
ERROR: Schema `sakila` already contains a table named inventory
ERROR: Schema `sakila` already contains a table named language
ERROR: Schema `sakila` already contains a table named payment
ERROR: Schema `sakila` already contains a table named rental
ERROR: Schema `sakila` already contains a table named staff
ERROR: Schema `sakila` already contains a table named store
ERROR: Schema `sakila` already contains a view named actor_info
ERROR: Schema `sakila` already contains a view named customer_list
ERROR: Schema `sakila` already contains a view named film_list
ERROR: Schema `sakila` already contains a view named nicer_but_slower_film_list
ERROR: Schema `sakila` already contains a view named sales_by_film_category
ERROR: Schema `sakila` already contains a view named sales_by_store
ERROR: Schema `sakila` already contains a view named staff_list
ERROR: Schema `sakila` already contains a function named get_customer_balance
ERROR: Schema `sakila` already contains a function named inventory_held_by_customer
ERROR: Schema `sakila` already contains a function named inventory_in_stock
ERROR: Schema `sakila` already contains a procedure named film_in_stock
ERROR: Schema `sakila` already contains a procedure named film_not_in_stock
ERROR: Schema `sakila` already contains a procedure named rewards_report
ERROR: Schema `world` already contains a table named city
ERROR: Schema `world` already contains a table named country
ERROR: Schema `world` already contains a table named countrylanguage
ERROR: One or more objects in the dump already exist in the destination database. You must either DROP these objects or exclude them from the load.
Util.loadDump: While 'Scanning metadata': Duplicate objects found in destination database (MYSQLSH 53021)
MySQL [localhost:33060+ ssl] JS >
```

As you can see here, it reported many errors that the schemas already exist in the database and they should be dropped before performing the load.

8.8.3. Table Export Utility:

Table export utility include `util.exportTable()` which exports data from a MySQL relational table into a single data file stored either locally or an Oracle Cloud Infrastructure (OCI) Object Storage bucket in CSV, TSV and text formats. For complete details, refer to the [official MySQL documentation](#).

Follow the below syntax to use this utility:

```
util.exportTable(table, outputUrl[, options])
```

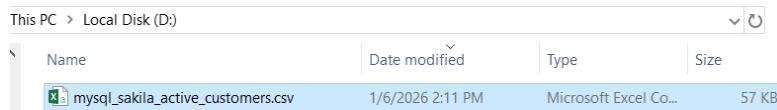
For example, run the following command to export active customers from `sakila.customer` table to a local `D:\mysql_sakila_active_customers.csv` file:

```
util.exportTable("sakila.customer",
"file:///D:/mysql_sakila_active_customers.csv", {"where":
"active=1"})
```

```
MySQL [localhost:33060+ ssl] > util.exportTable("sakila.customer", "file:///D:/mysql_sakila_active_customers.csv", {
  "where": "active=1")
Initializing - done
Gathering information - done
Running data dump using 1 thread.
NOTE: Progress information uses estimated values and may not be accurate.
Starting data dump
97% (584 rows / ~599 rows), 0.00 rows/s, 0.00 B/s
Dump duration: 00:00:00s
Total duration: 00:00:00s
Data size: 57.47 KB
Rows written: 584
Bytes written: 57.47 KB
Average throughput: 57.47 KB/s

The dump can be loaded using:
util.importTable("file:///D:/mysql_sakila_active_customers.csv", {
  "characterSet": "utf8mb4",
  "schema": "sakila",
  "table": "customer"
})
```

Once the table export is completed, it creates `mysql_sakila_active_customers.csv` file in D drive.



8.8.4. Parallel Table Import Utility:

Parallel Table Import Utility include `util.importTable()` which is useful for rapid import of large data files like CSV or TSV into a single relational table using parallel connections. It is much faster than a standard single-threaded `LOAD DATA` statement for large imports. It uses `LOAD DATA LOCAL INFILE` statements to import data which requires the `local_infile` system variable to be enabled before using this utility. For complete details, refer to the [official MySQL documentation](#).

Follow the below syntax to use this utility:

```
util.importTable ({file_name | file_list}, options)
```

Let us try to import `D:\mysql_sakila_active_customers.csv` file into `test.customer` table using `importTable` utility.

First, create `test.customer` table using the following command:

```
\sql CREATE TABLE test.customer (customer_id int NOT NULL
AUTO_INCREMENT,store_id int NOT NULL,first_name varchar(45) NOT
NULL,last_name varchar(45) NOT NULL,email varchar(50) DEFAULT
```

```
NULL,address_id int NOT NULL,active int NOT NULL DEFAULT  
'1',create_date datetime NOT NULL,last_update timestamp NULL DEFAULT  
CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,PRIMARY KEY  
(`customer_id`));
```

Then run the following command to import

D:\mysql_sakila_active_customers.csv file into test.customer table :

```
util.importTable("D:/mysql_sakila_active_customers.csv", {schema:  
"test", table: "customer", skipRows: 0, showProgress: true})
```

```
[MySQL] localhost:33060+ ssl [JS] > util.importTable("D:/mysql_sakila_active_customers.csv", {schema: "test", table: "cus  
tomer", skipRows: 0, showProgress: true})  
Util.importTable: A classic protocol session is required to perform this operation. (RuntimeError)  
[MySQL] localhost:33060+ ssl [JS] >
```

The parallel table import utility throws runtime error *A classic protocol session is required to perform this operation* since it requires a classic connection to the target server and does not support X Protocol connections.

8.8.5. JSON Import Utility:

JSON Import Utility such as `util.importJSON()` imports JSON documents from a file to a MySQL relational table or a collection in JSON format. If the table or collection does not exist in the database, this utility creates the default table or collection and imports data. For complete details, refer to the [official MySQL documentation](#).

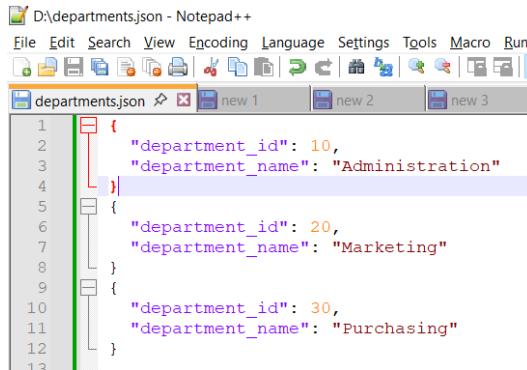
Follow the below syntax to use this utility:

```
util.importJSON (path, options)
```

First, create a simple JSON file D:\departments.json with the below contents:

```
{  
    "department_id": 10,  
    "department_name": "Administration"  
}  
{  
    "department_id": 20,  
    "department_name": "Marketing"  
}  
{  
    "department_id": 30,  
    "department_name": "Purchasing"  
}
```

Note: MySQL Shell does not accept a JSON file containing square brackets and commas in between array contents.



```
D:\departments.json - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run
departments.json new 1 new 2 new 3
1 {
2   "department_id": 10,
3   "department_name": "Administration"
4 }
5 {
6   "department_id": 20,
7   "department_name": "Marketing"
8 }
9 {
10  "department_id": 30,
11  "department_name": "Purchasing"
12 }
```

Then run the below command to import the above JSON file to a test.departments table in JSON format:

```
util.importJson("D:/departments.json", {schema: "test", table: "departments"})
```

```
MySQL localhost:33060+ ssl JS > util.importJson("D:/departments.json", {schema: "test", table: "departments"})
Importing from file "D:/departments.json" to table `test`.`departments` in MySQL Server at localhost:33060
...
Processed 198 bytes in 3 documents in 0.0769 sec (3.00 documents/s)
Total successfully imported documents 3 (3.00 documents/s)
MySQL localhost:33060+ ssl JS >
```

Run the below command to verify data in test.departments table:

```
\sql select * from test.departments;
```

```
MySQL localhost:33060+ ssl JS > \sql select * from test.departments;
+-----+-----+
| doc | id |
+-----+-----+
| {"department_id": 10, "department_name": "Administration"} | 1 |
| {"department_id": 20, "department_name": "Marketing"} | 2 |
| {"department_id": 30, "department_name": "Purchasing"} | 3 |
+-----+-----+
3 rows in set (0.0012 sec)
MySQL localhost:33060+ ssl JS >
```

8.8.6. Copy Utilities:

Copy utilities include `util.copyInstance()`, `util.copySchemas()`, and `util.copyTables()` which copy DDL and stream data directly between two MySQL instances without the need for intermediate local storage. It combines both the dump and load operations into one step. These utilities uses `LOAD DATA LOCAL INFILE` statements to

import data which requires the `local_infile` system variable to be enabled. For complete details, refer to the [official MySQL documentation](#).

Follow the below syntax to use these copy utilities:

```
util.copyInstance(connectionData[, options])
util.copySchemas(schemaList, connectionData[, options])
util.copyTables(schemaName, tablesList, connectionData[, options])
```

8.8.7. Upgrade Checker Utility:

Upgrade checker utility include `util.checkForServerUpgrade()` performs checks on a MySQL server instance to determine its readiness for an upgrade to a newer version. For complete details, refer to the [official MySQL documentation](#).

Follow the below syntax to use this utility:

```
util.checkForServerUpgrade (ConnectionData connectionData, Dictionary
options)
```

For example, run the following command to check for server upgrade:

```
util.checkForServerUpgrade()
```

```
[MySQL] localhost:33060+ ssl [JS] > util.checkForServerUpgrade()
The MySQL server at localhost:33060, version 8.0.44 - MySQL Community Server -
GPL. will now be checked for compatibility issues for upgrade to MySQL 8.0.44...
Util.checkForServerUpgrade: Detected MySQL Server version is 8.0.44. MySQL Shell cannot check MySQL server instances for
upgrade if they are at a version the same as or higher than the MySQL Shell version. (ArgumentError)
[MySQL] localhost:33060+ ssl [JS] >
```

Here, it displays an error *MySQL Shell cannot check MySQL server instances for upgrade if they are at a version the same as or higher than the MySQL Shell version* since MySQL Shell and MySQL server are both at same version.

8.8.8. Diagnostic Utilities:

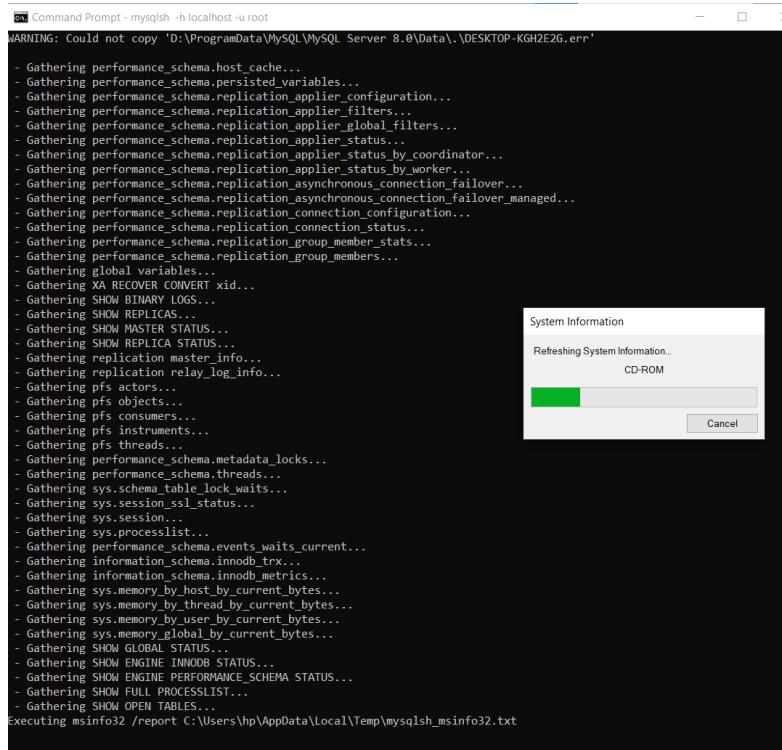
Diagnostic utilities include `util.debug.collectDiagnostics()`, `util.debug.collectHighLoadDiagnostics()`, `util.debug.collectSlowQueryDiagnostics()` which gather diagnostic information and performance metrics from the connected MySQL server and generate diagnostic reports on overall health, performance and SQL queries in YAML and TSV formats that help in troubleshooting system issues. For complete details, refer to the [official MySQL documentation](#).

Follow the below syntax to use these utilities:

```
util.debug.collectDiagnostics("path/", {options})
util.debug.collectHighLoadDiagnostics(path, {options})
util.debug.collectSlowQueryDiagnostics("path", "query", {options})
```

For example, run the following command to collect diagnostics of the current system:

```
util.debug.collectDiagnostics("D:/mysql_diagnostics")
```



It collects system information and writes to D:\mysql_diagnostics.zip file.

```
Gathering SHOW ENGINE INNODB STATUS...
- Gathering SHOW ENGINE PERFORMANCE_SCHEMA STATUS...
- Gathering SHOW FULL PROCESSLIST...
- Gathering SHOW OPEN TABLES...
Executing msinfo32 /report C:\Users\hp\AppData\Local\Temp\mysqlsh_msinfo32.txt
Collecting system information for DESKTOP-KGH2E2G (win32)
-> Executing date /T
-> Executing ver
-> Executing systeminfo
-> Executing tasklist /V

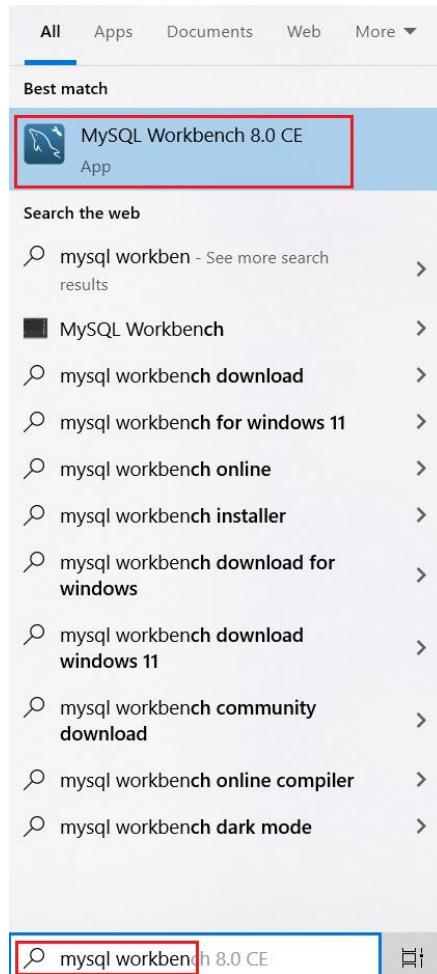
Diagnostics information was written to D:/mysql_diagnostics.zip
MySQL | localhost:33060+ ssl [S] >
```

Name	Date modified	Type	Size
mysql_diagnostics.zip	1/6/2026 3:27 PM	WinRAR ZIP archive	3,018 KB

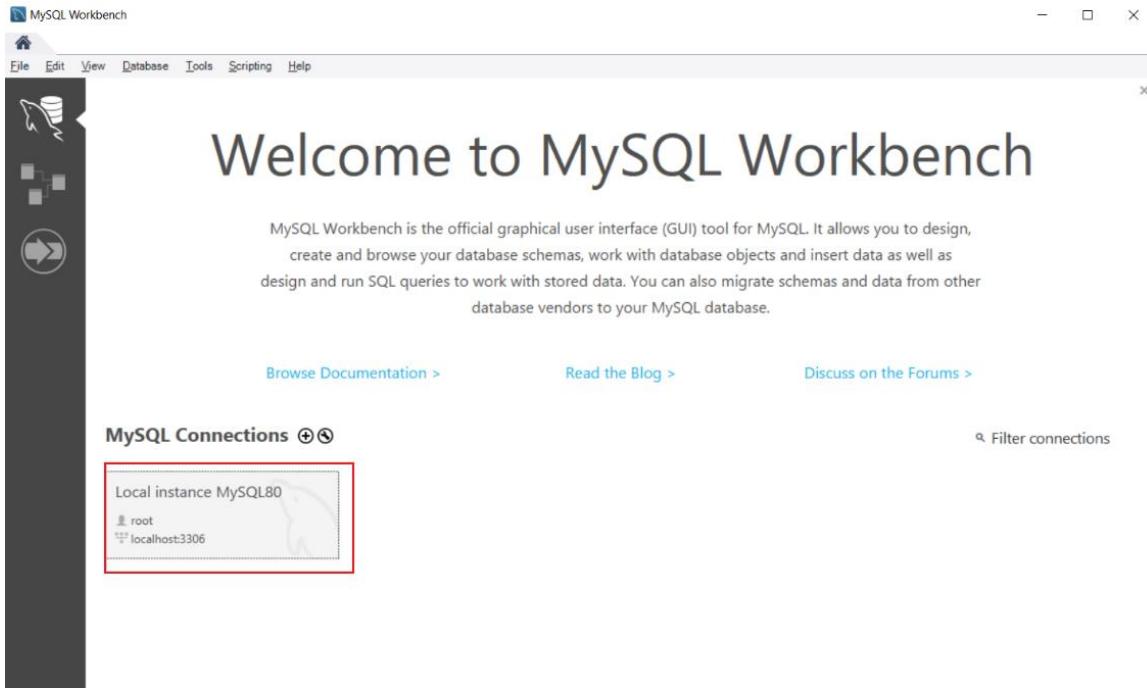
9. MySQL Workbench:

MySQL Workbench is a visual tool that provides a graphical user interface for working with MySQL databases that simplifies creating, executing and optimizing SQL queries.

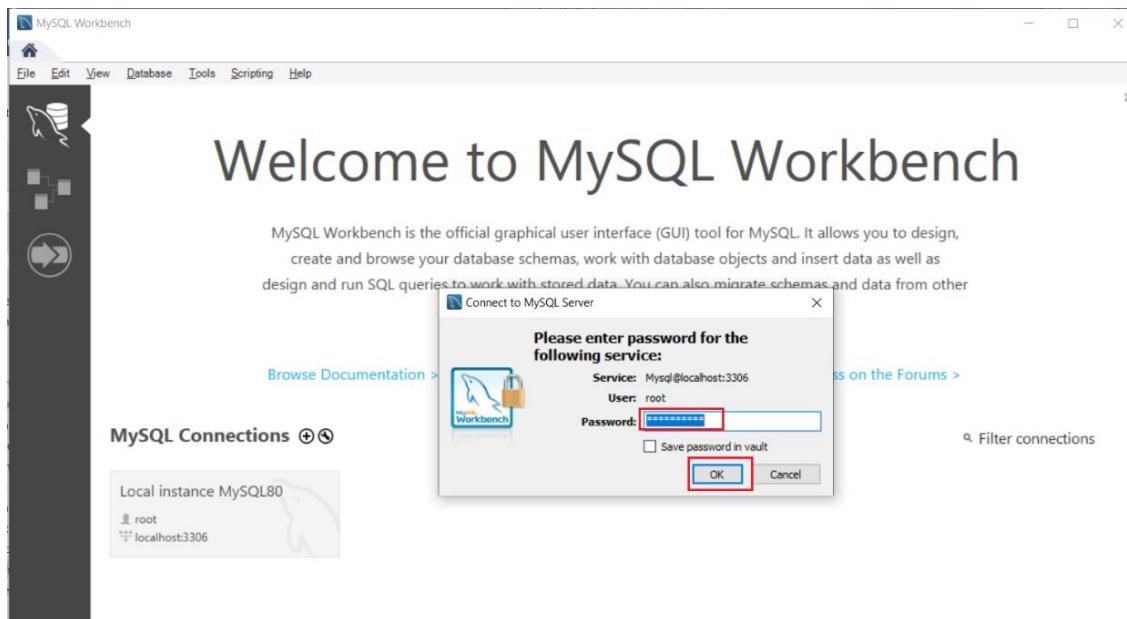
To open MySQL Workbench, search for **mysql workbench** in the task bar and choose the first option:



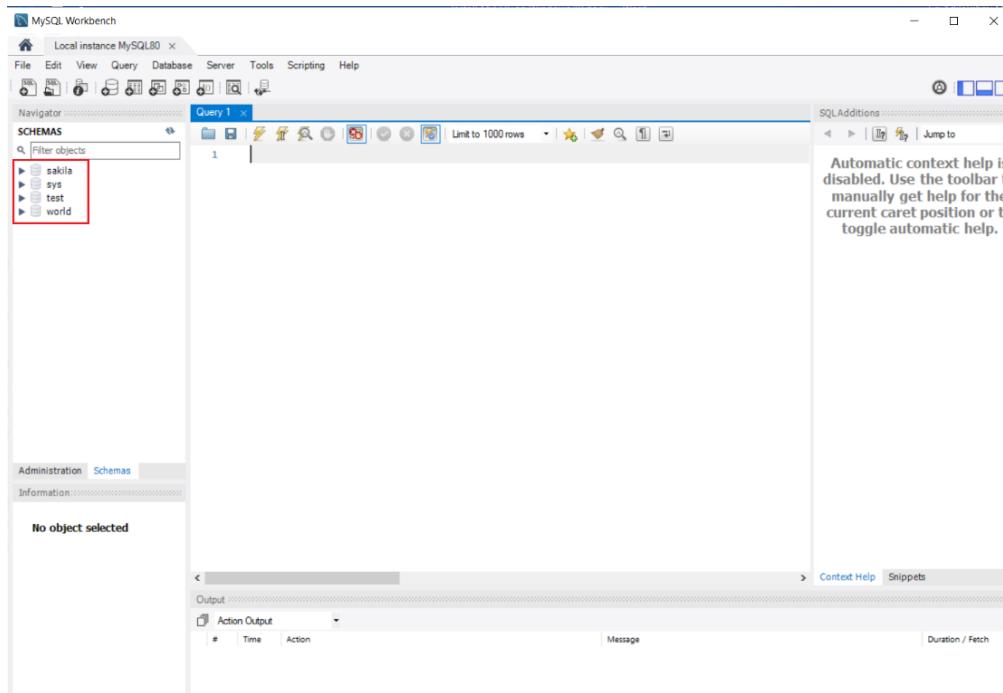
On the home screen, it displays the list of MySQL connections. Select **Local instance MySQL** to connect to the MySQL server installed locally:



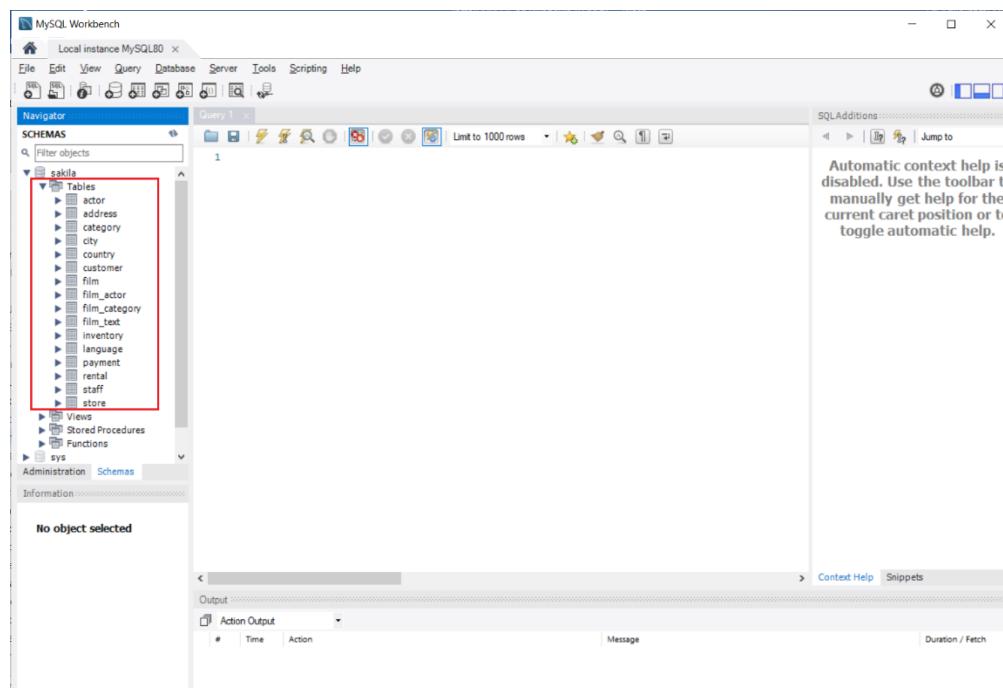
It prompts for root password and enter the password given at the time of installation and press **OK**.



Once it is successfully connected to the local MySQL server, it displays the list of databases available in the server:

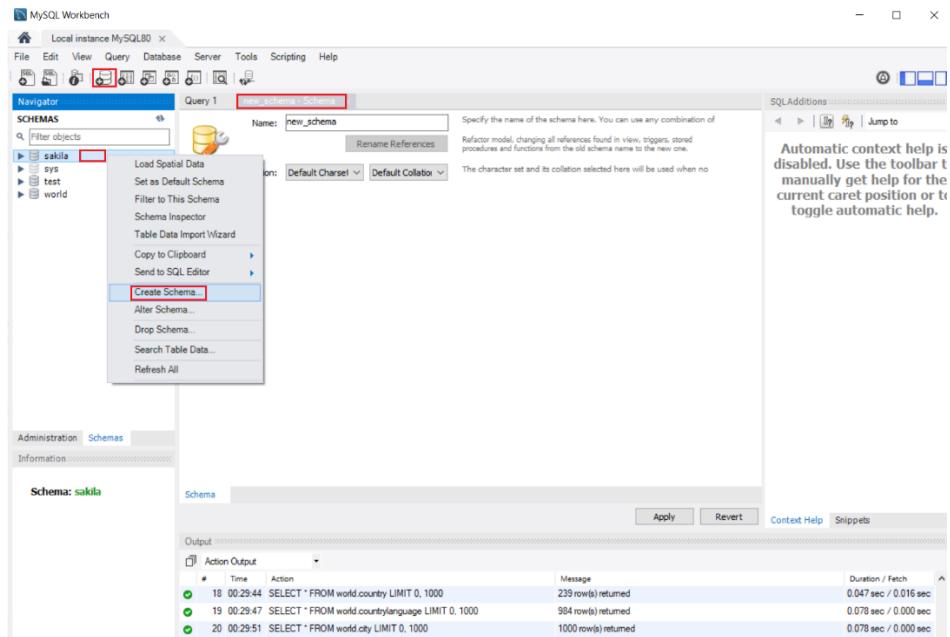


To see the list of tables in a database, expand the database name (for instance, **sakila**) and expand **Tables**.

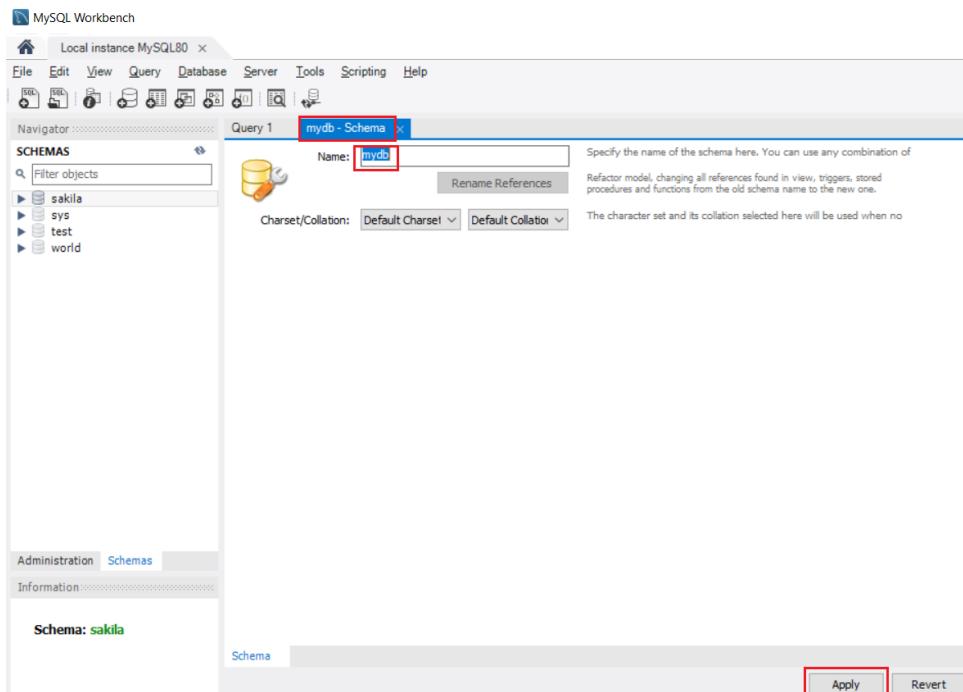


9.1. Create Database:

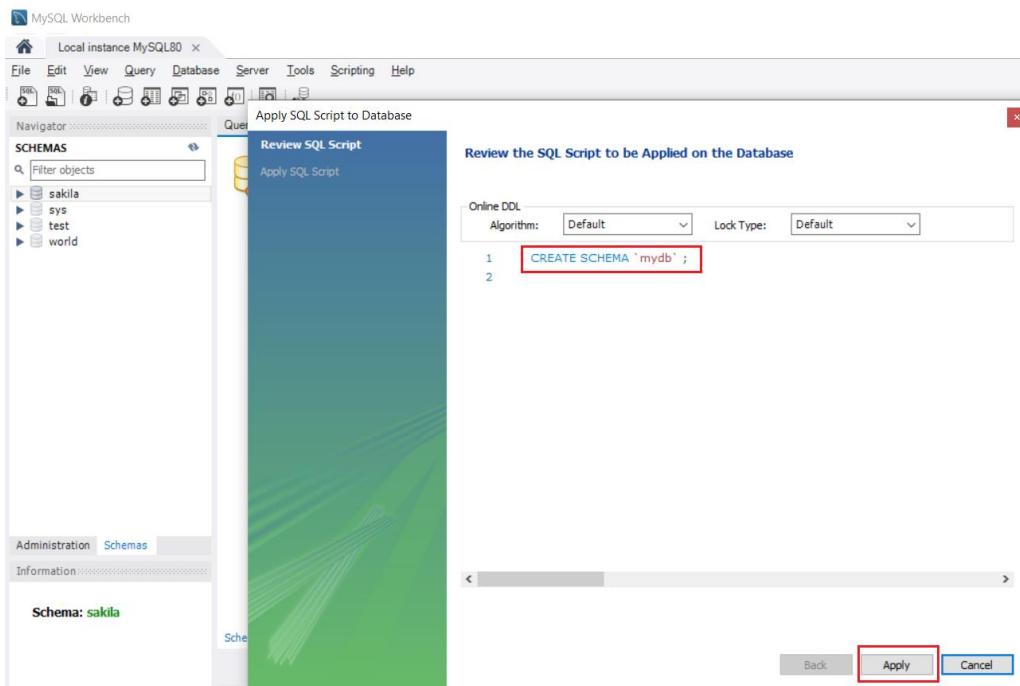
To create a new database, click on **Create new schema** icon on the top menu or right click on existing schema and select **Create Schema** option which opens **new_schema - Schema** tab.



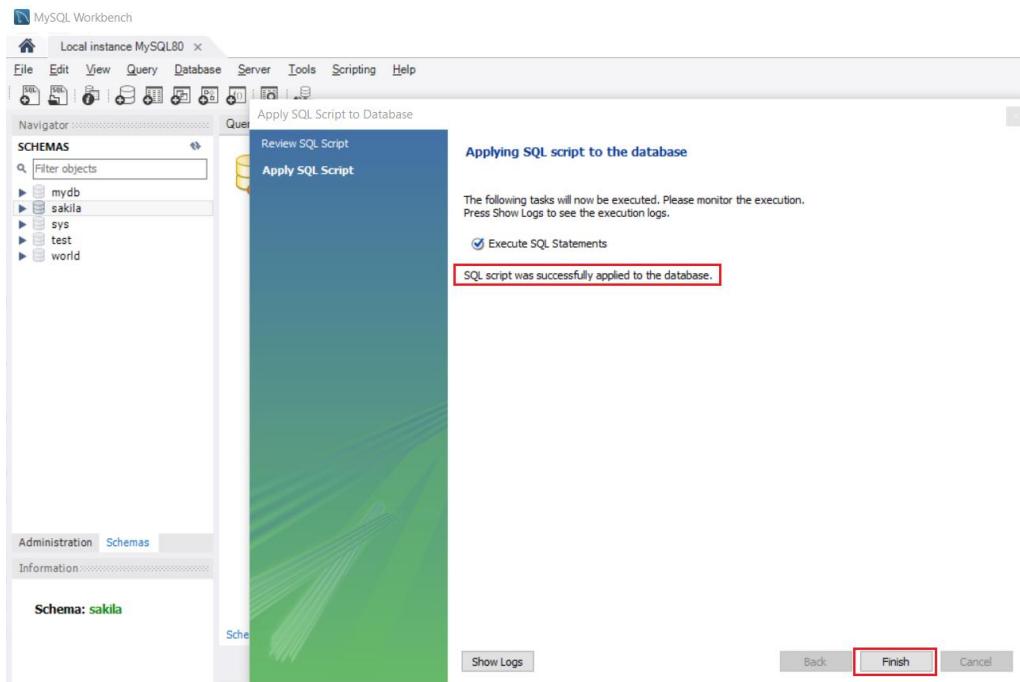
Provide the new database name such as `mydb` (after which you can observe the tab name changes to **mydb - Schema**) and click on **Apply** button.



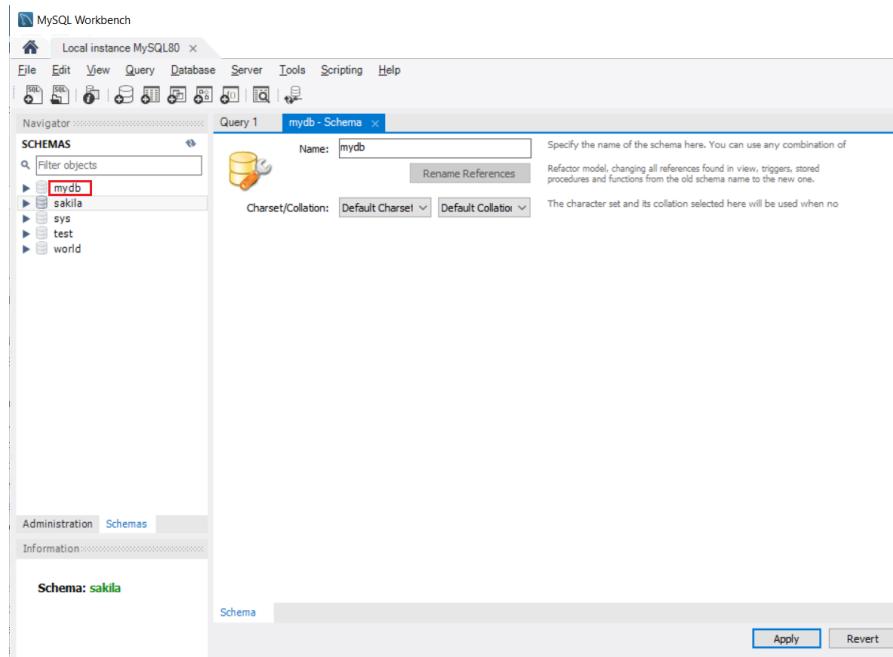
Review the SQL script provided and click on **Apply** button.



It executes the task and displays message that SQL script was successfully applied to the database. Click on **Finish** button to close.

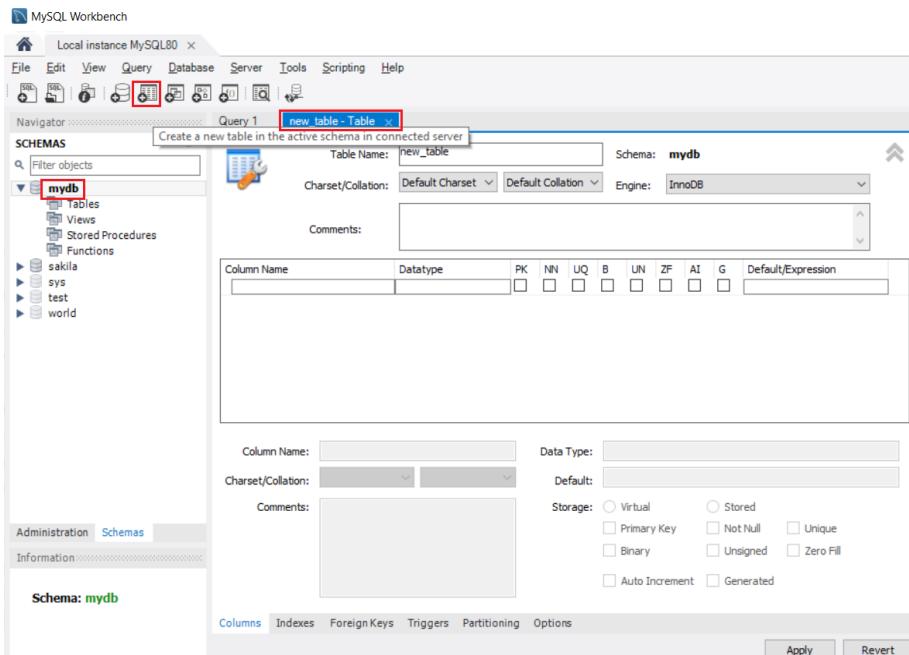


Under **Schemas** list, the new database **mydb** is visible. Close **mydb - Schema** tab.

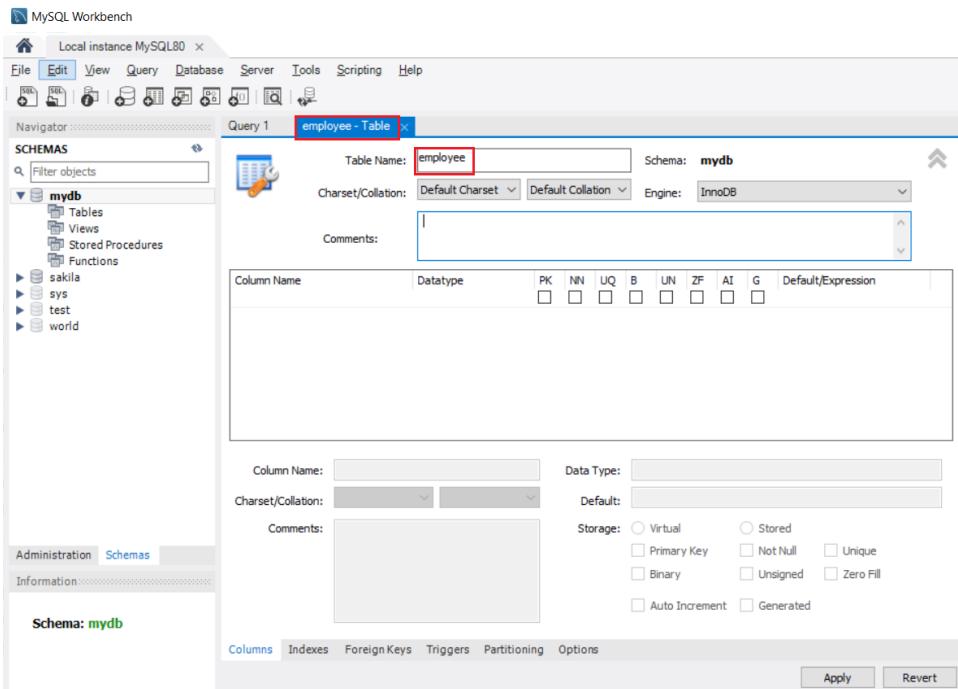


9.2. Create Table:

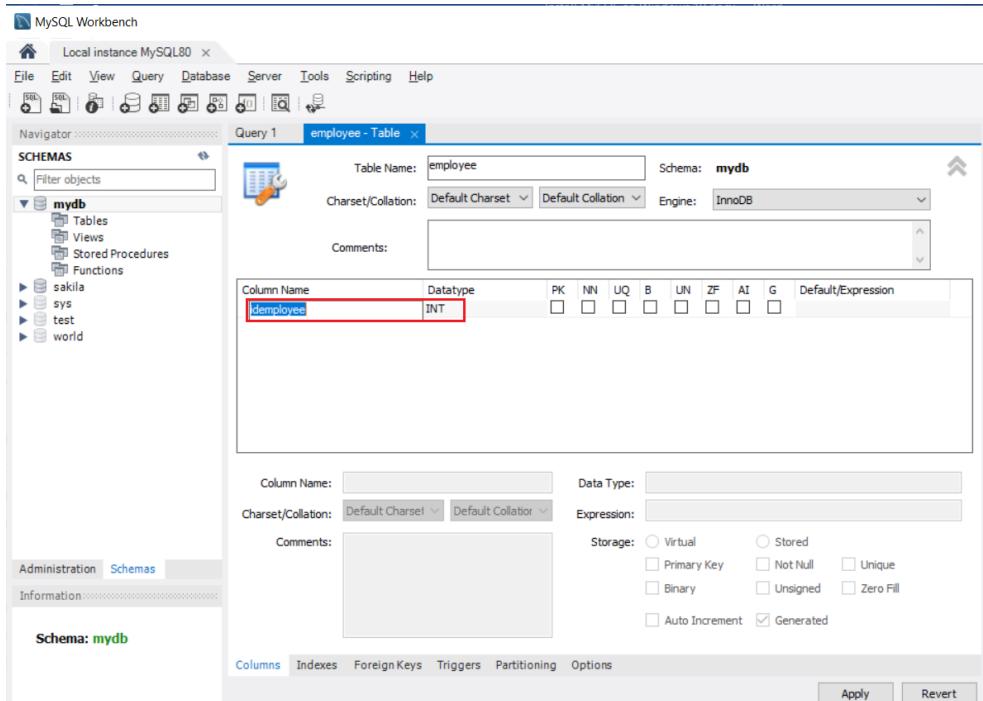
To create a new table, double click on **mydb** which selects the table and click on **Create new table** icon on the top menu which opens **new_table - Table** tab.



Provide the new table name such as `employee` (after which you can observe the tab name changes to **employee - Table**).



Click under **Column Name** which by default displays as `idemployee` and **Datatype** as `INT`.



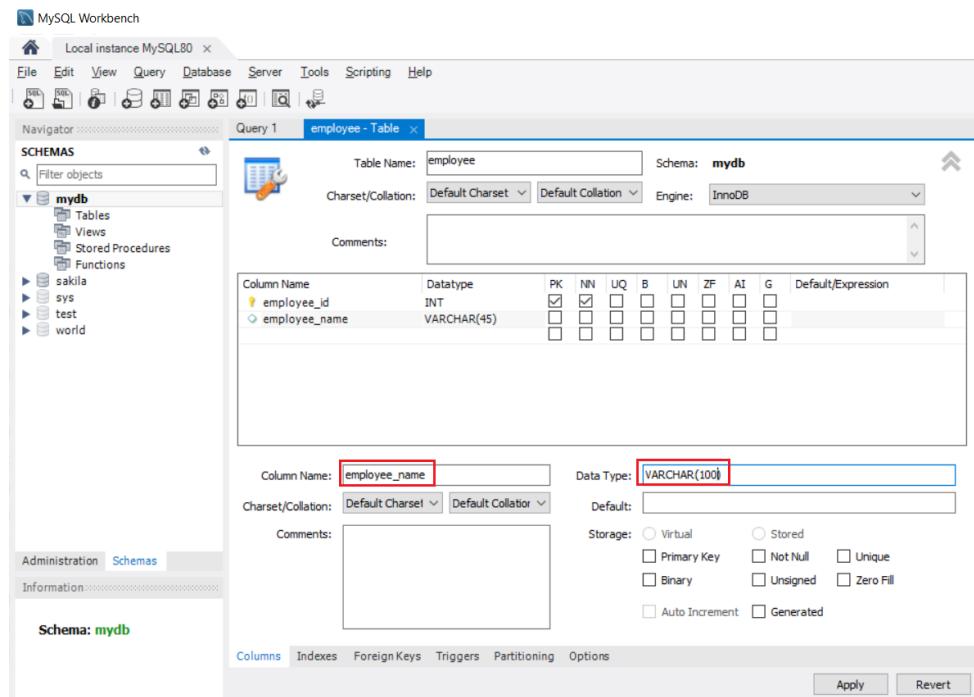
Update the **Column Name** to **employee_id** and keep **Datatype** as **INT** and it automatically selects **PK** (Primary Key) and **NN** (Not Null).

MySQL Workbench Screenshot showing the 'employee' table configuration. The 'employee_id' column is selected, highlighted with a red border. The 'Datatype' is set to 'INT'. The 'PK' and 'NN' checkboxes are checked. The 'Storage' section shows 'Primary Key' and 'Not Null' checked.

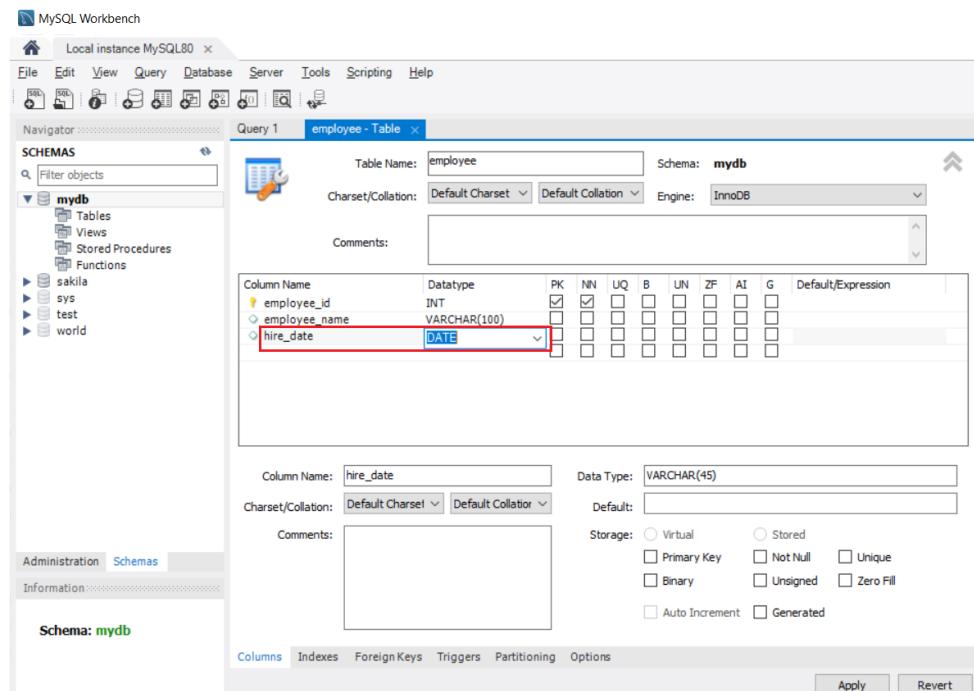
Then select other column where by default it displays **Column Name** as **employeecol** and **Datatype** as **VARCHAR(45)**.

MySQL Workbench Screenshot showing the 'employee' table configuration. The 'employeecol' column is selected, highlighted with a red border. The 'Datatype' is set to 'VARCHAR(45)'.

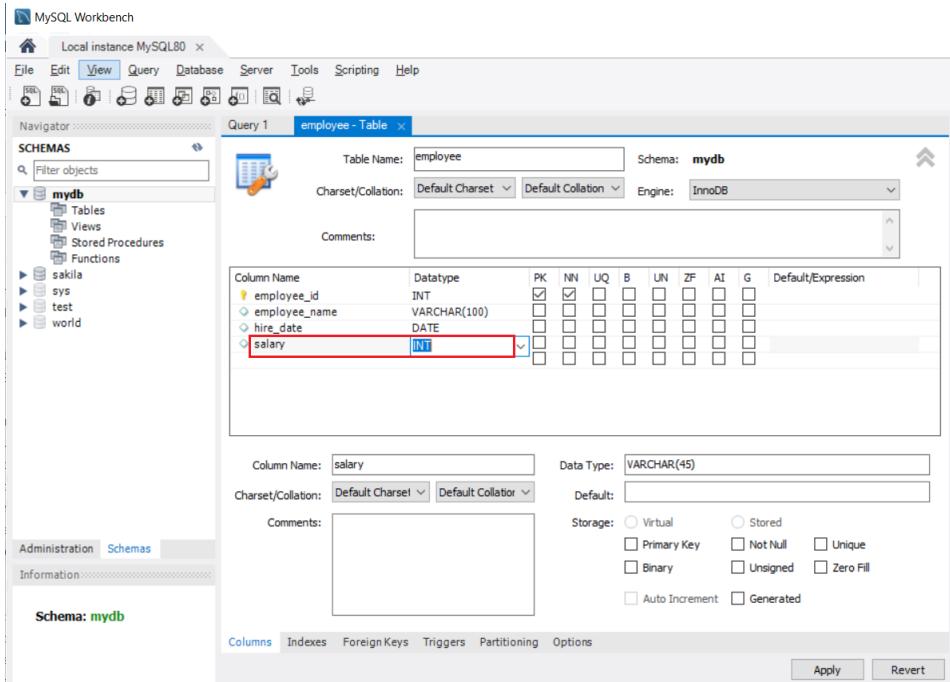
Select `employeecol` and update the **Column Name** to `employee_name` and **Datatype** to `VARCHAR(100)`.



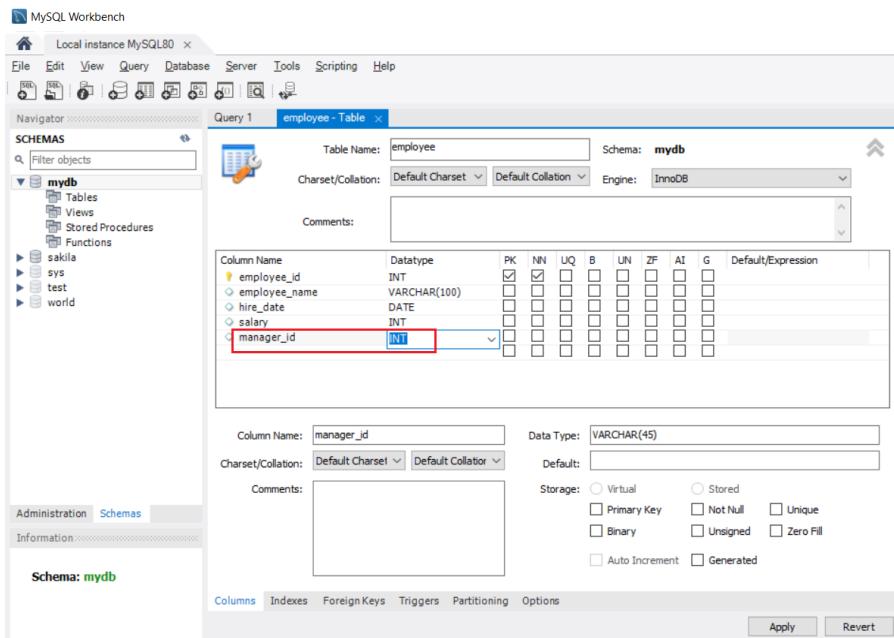
Then select another column and update the **Column Name** to `hire_date` and choose **Datatype** as `DATE` from the dropdown.



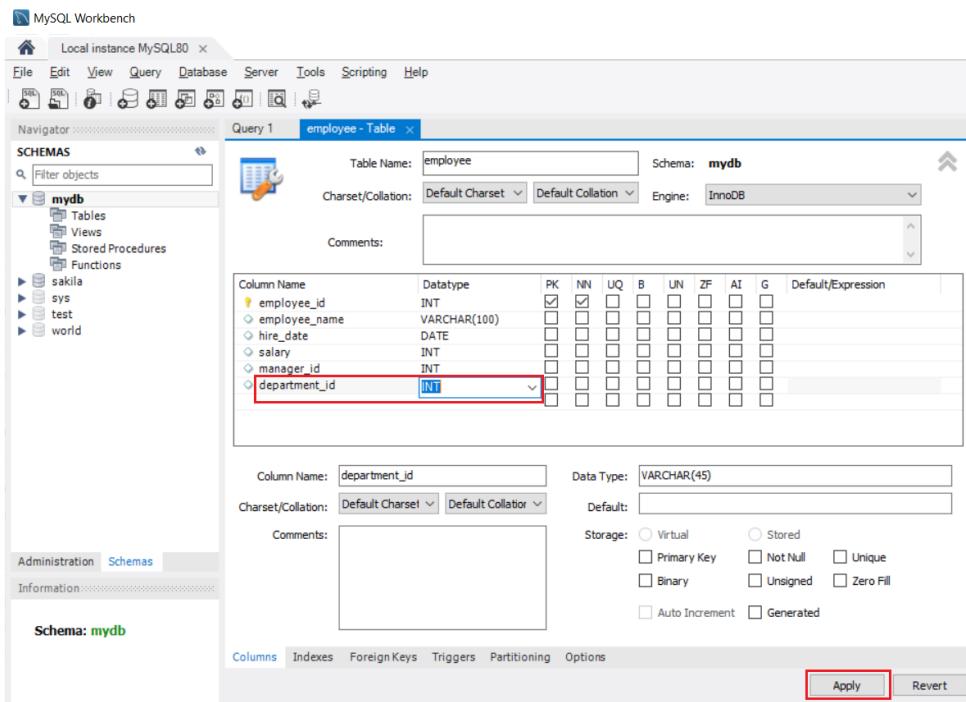
Then select another column and update the **Column Name** to `salary` and choose **Datatype** as `INT` from the dropdown.



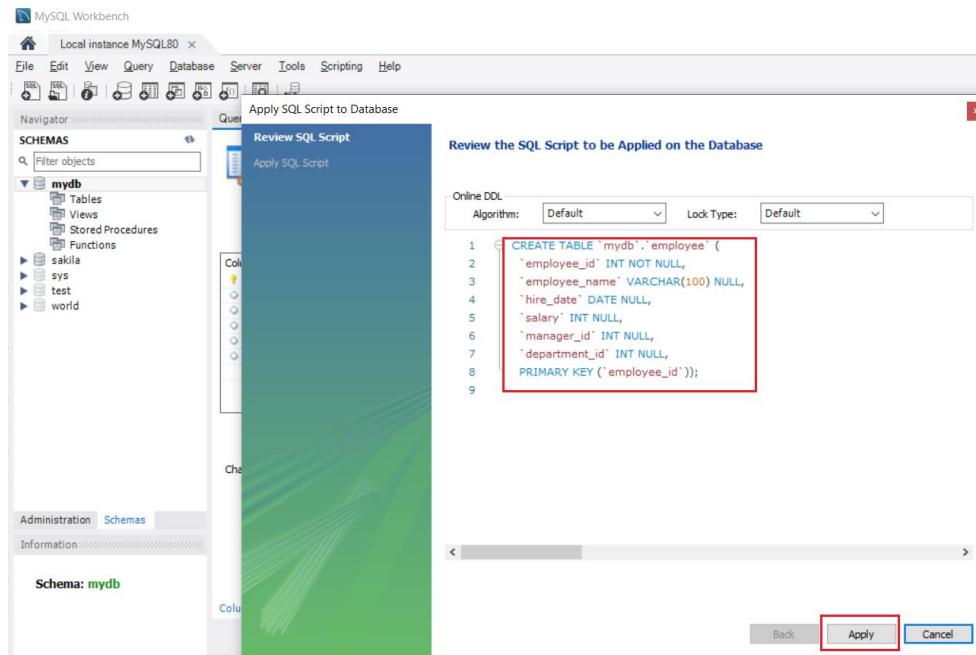
Then select another column and update the **Column Name** to `manager_id` and choose **Datatype** as `INT` from the dropdown.



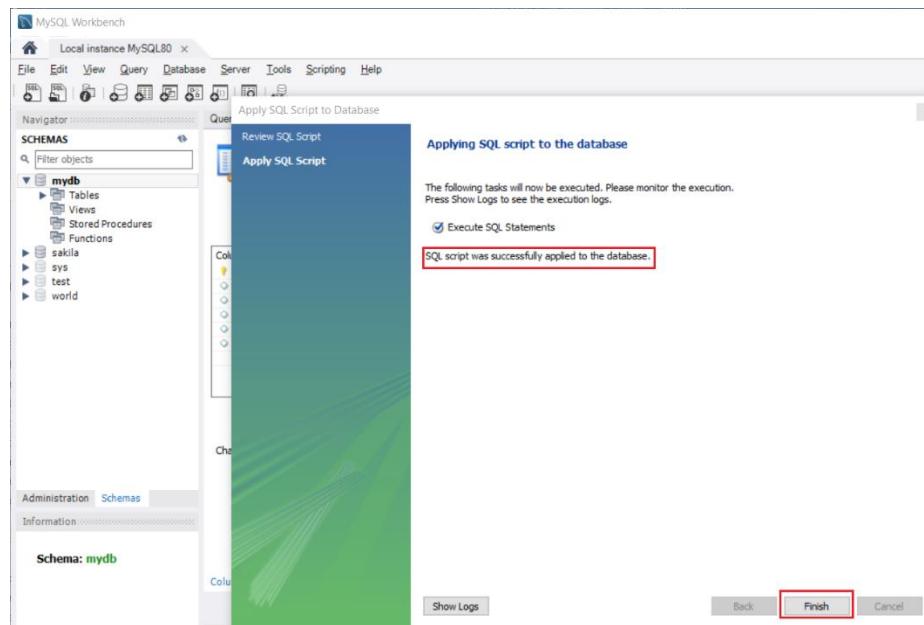
Then select another column and update the **Column Name** to `department_id` and choose **Datatype** as `INT` from the dropdown and click on **Apply** button.



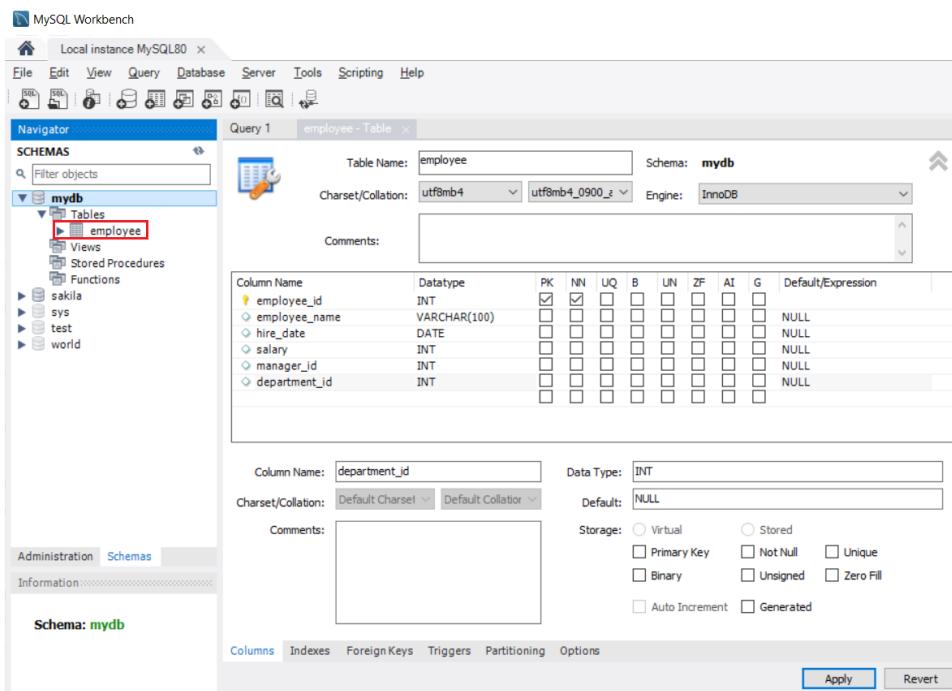
Review the SQL script provided and click on **Apply** button.



It executes the task and displays message that SQL script was successfully applied to the database. Click on **Finish** button to close.



Now, expand **Tables** under mydb on the right side where you see employee table. Close **employee - Table tab**.



9.3. Insert Data:

Now, select employee table and click on table icon which executes the select query and displays null data as there is no data available in the table.

The screenshot shows the MySQL Workbench interface. In the Navigator pane, the 'mydb' schema is selected, and the 'Tables' section shows the 'employee' table. A query window titled 'Query 1' contains the SQL command: 'SELECT * FROM mydb.employee;'. The result grid shows one row with all columns set to NULL ('NULL').

employee_id	employee_name	hire_date	salary	manager_id	department_id
NULL	NULL	NULL	NULL	NULL	NULL

Click on NULL under employee_id column which displays blank data and update the value to 100.

The screenshot shows the MySQL Workbench interface. The 'employee' table has been modified. The first row now has an 'employee_id' value of 100, while other columns remain NULL. The rest of the table structure and the 'Result Grid' interface are identical to the previous screenshot.

employee_id	employee_name	hire_date	salary	manager_id	department_id
100	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL

Similarly, select NULL under employee_name and update value to Steven King. Then select NULL under hire_date and update value to 2023-06-17. Then select NULL under salary and update value to 24000. Then select NULL under department_id and update value to 20.

The screenshot shows the MySQL Workbench interface with the 'employee' table selected in the 'Query 1' tab. The table has six columns: employee_id, employee_name, hire_date, salary, manager_id, and department_id. The first row contains the values: 100, Steven King, 2023-06-17, 24000, NULL, and 20 respectively. The entire row is highlighted with a red border.

employee_id	employee_name	hire_date	salary	manager_id	department_id
100	Steven King	2023-06-17	24000	NULL	20
NULL	NULL	NULL	NULL	NULL	NULL

Similarly, select NULL value in the second row and enter the following data:

employee_id: 101, employee_name: Neena Kochhar, hire_date: 2025-09-21, salary: 17000, manager_id: 100, department_id: 10

The screenshot shows the MySQL Workbench interface with the 'employee' table selected in the 'Query 1' tab. The table now has two rows. The first row is the same as before: 100, Steven King, 2023-06-17, 24000, NULL, and 20. The second row has been updated with the new data: 101, Neena Kochhar, 2025-09-21, 17000, 100, and 10. The second row is highlighted with a red border.

employee_id	employee_name	hire_date	salary	manager_id	department_id
100	Steven King	2023-06-17	24000	NULL	20
101	Neena Kochhar	2025-09-21	17000	100	10
NULL	NULL	NULL	NULL	NULL	NULL

Similarly, select NULL value in the third row and enter the following data:

employee_id: 102, employee_name: Lex De Haan, hire_date: 2021-01-13, salary: 17000, manager_id: 100, department_id: 10

The screenshot shows the MySQL Workbench interface. The left sidebar displays the Navigator with the 'mydb' schema selected, showing tables like 'employee', 'Views', 'Stored Procedures', and 'Functions'. The main area is a 'Query 1' window with the query 'SELECT * FROM mydb.employee;'. The results grid shows four rows of data. The fourth row, corresponding to employee_id 102, has all columns set to NULL (indicated by red boxes). The 'Result Grid' tab is active, and the 'Form Editor', 'Field Types', and 'Query Stats' tabs are visible on the right.

employee_id	employee_name	hire_date	salary	manager_id	department_id
100	Steven King	2023-06-26	24000	NULL	20
101	Neena Kochhar	2025-09-17	17000	100	10
102	Lex De Haan	2021-01-13	17000	100	10
NULL	NULL	NULL	NULL	NULL	NULL

Similarly, select NULL value in the fourth row and enter the following data. Then click on **Apply** button.

employee_id: 103, employee_name: Alexander Hunold, hire_date: 2022-01-03, salary: 9000, manager_id: 102, department_id: 30

The screenshot shows the MySQL Workbench interface. In the Navigator pane, the 'mydb' schema is selected, and the 'Tables' section shows the 'employee' table. The 'Query 1' tab contains the SQL command: 'SELECT * FROM mydb.employee;'. The Result Grid displays the following data:

employee_id	employee_name	hire_date	salary	manager_id	department_id
100	Steven King	2023-06-01	24000	NULL	20
101	Nneka Kochhar	2025-09-15	17000	100	10
102	Lex De Haan	2021-01-15	17000	100	10
103	Alexander Hunold	2022-01-15	9000	102	30

The last row (employee_id 103) is highlighted with a red box. The 'Apply' button at the bottom right of the result grid is also highlighted with a red box.

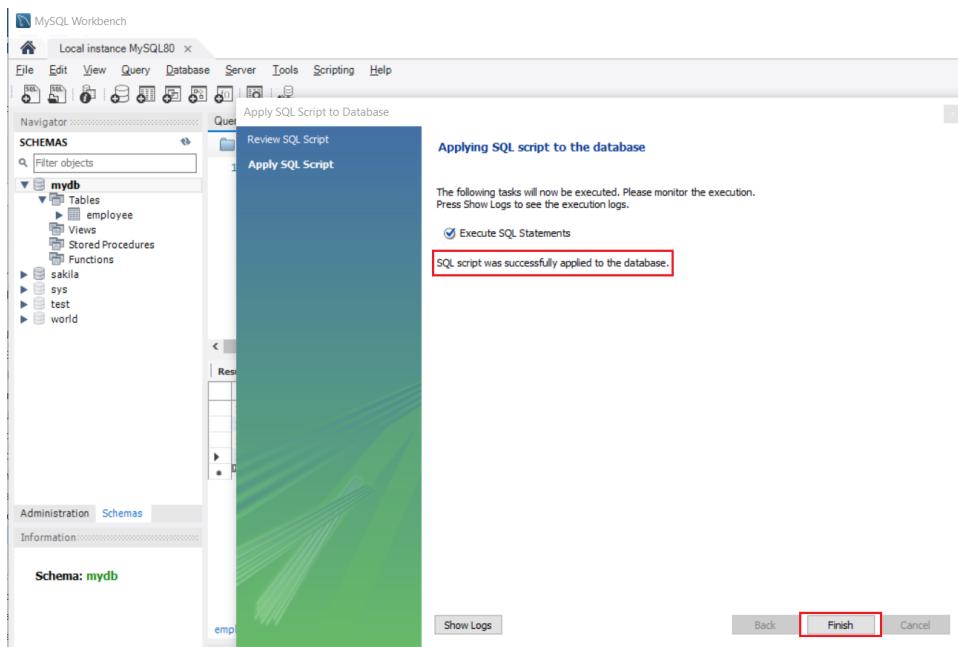
Review the SQL script provided and click on **Apply** button.

The screenshot shows the MySQL Workbench interface with the 'Review SQL Script' dialog box open. The dialog title is 'Review the SQL Script to be Applied on the Database'. The SQL script content area shows the following code:

```
1 INSERT INTO `mydb`.`employee` ('employee_id', 'employee_name', 'hire_date'
2 INSERT INTO `mydb`.`employee` ('employee_id', 'employee_name', 'hire_date'
3 INSERT INTO `mydb`.`employee` ('employee_id', 'employee_name', 'hire_date'
4 INSERT INTO `mydb`.`employee` ('employee_id', 'employee_name', 'hire_date'
5
```

The 'Apply' button at the bottom right of the dialog is highlighted with a red box.

It executes the task and displays message that SQL script was successfully applied to the database. Click on **Finish** button to close.



9.4. Select Data:

To select data from a table, click on table icon next to employee table which executes the select query and displays all records available in the table.

employee_id	employee_name	hire_date	salary	manager_id	department_id
100	Steven King	2023-06-17	24000	NULL	20
101	Neena Kochhar	2025-09-21	17000	100	10
102	Lex De Haan	2021-01-13	17000	100	10

We can modify the select query to filter records and displays data.

Write the following query to display department 10 employees, select the query and click on **Execute** button to execute query and display results:

```
SELECT * FROM mydb.employee WHERE department_id=10;
```

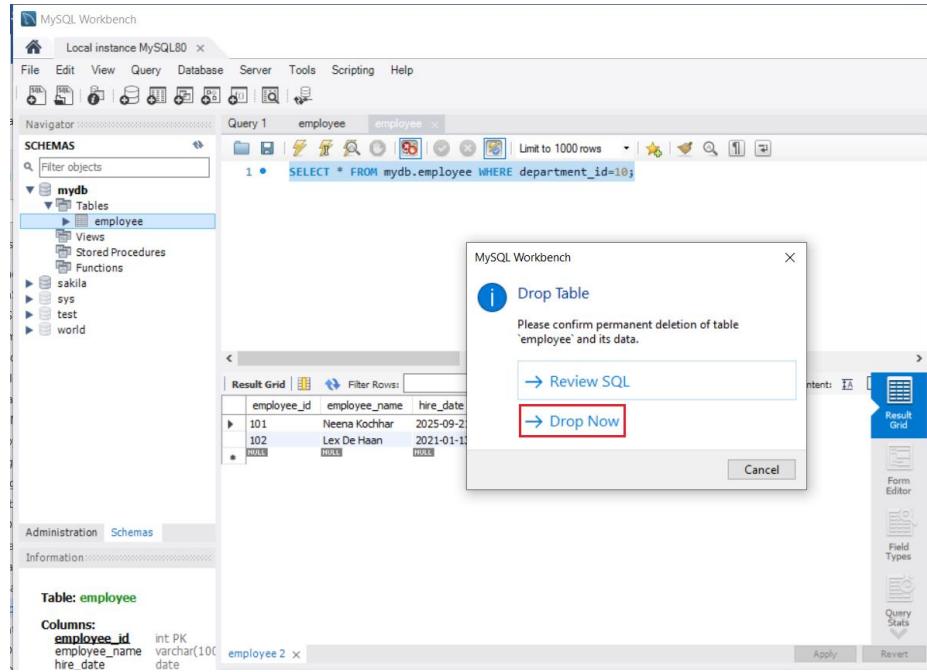
The screenshot shows the MySQL Workbench interface. In the Navigator pane, the 'mydb' schema is selected, and the 'Tables' section shows the 'employee' table. In the Query Editor (Query 1), the SQL query `SELECT * FROM mydb.employee WHERE department_id=10;` is entered and highlighted with a red box. Below the query, the results are displayed in a Result Grid. The grid has columns: employee_id, employee_name, hire_date, salary, manager_id, and department_id. Two rows are visible: one for Neena Kochhar (employee_id 101) and one for Lex De Haan (employee_id 102). Both rows are also highlighted with a red box. The 'Result Grid' tab is active on the right side of the interface.

9.5. Drop Table:

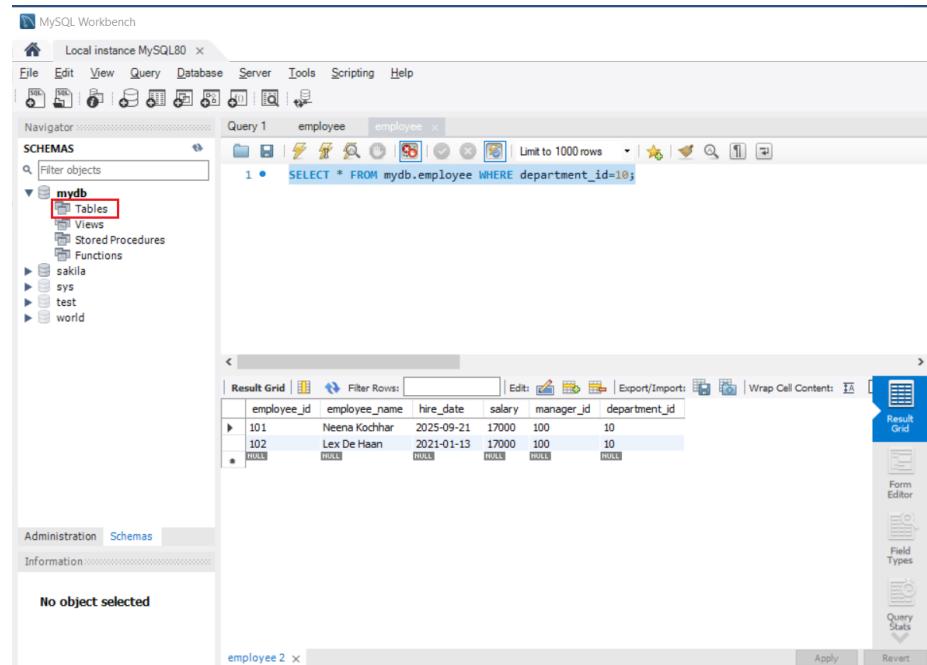
To drop a specific table, right click on table name and select **Drop Table** option.

The screenshot shows the MySQL Workbench interface. The 'mydb' schema is selected in the Navigator. The 'Tables' section shows the 'employee' table, which is currently selected and highlighted with a red box. A context menu is open over the 'employee' table, with the 'Drop Table...' option highlighted with a red box. Other options in the menu include 'Select Rows - Limit 1000', 'Table Inspector', 'Copy to Clipboard', 'Table Data Export Wizard', 'Table Data Import Wizard', 'Send to SQL Editor', 'Create Table...', 'Create Table Like...', 'Alter Table...', 'Table Maintenance...', 'Truncate Table...', 'Search Table Data...', and 'Refresh All'. The 'Result Grid' tab is active on the right side of the interface.

It opens a popup message where you can choose **Review SQL** if you want to review before dropping. Otherwise, select **Drop Now** to drop the table and its data permanently.

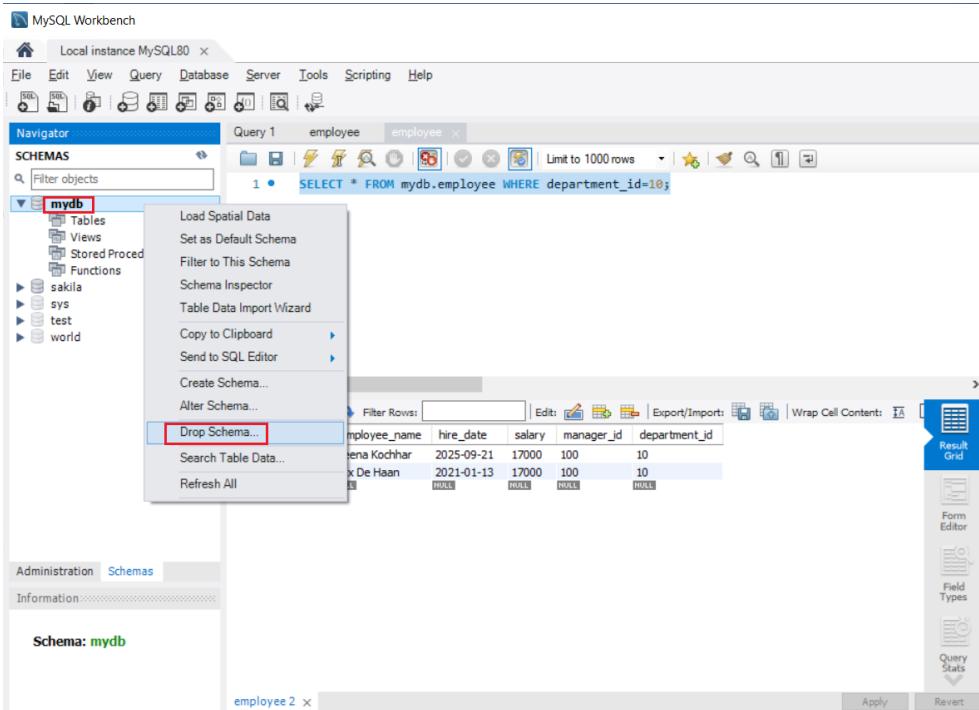


You can observe that `employee` table has disappeared under **Tables** section in the `mydb` database.

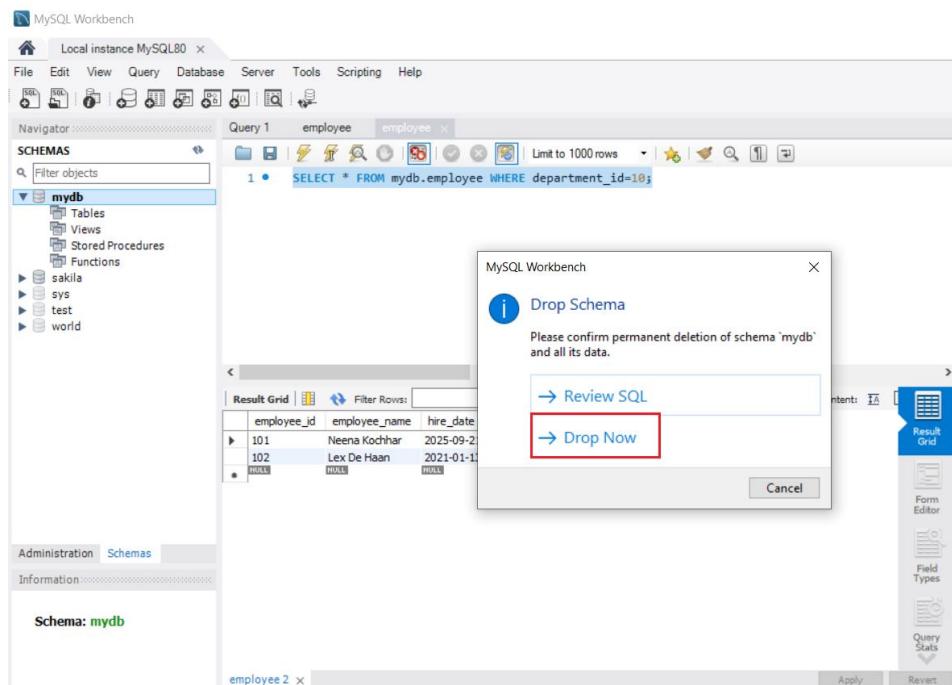


9.6. Drop Database:

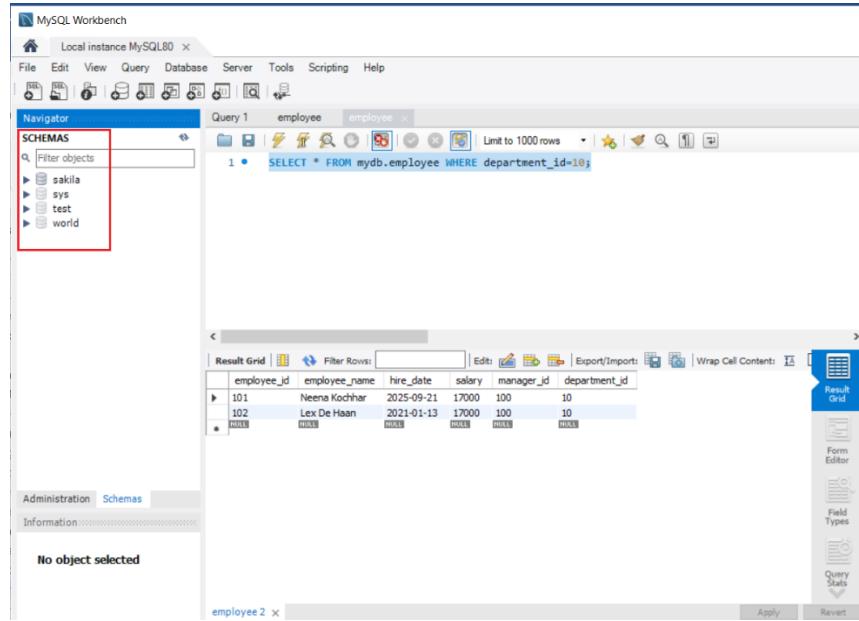
To drop a specific database, right click on database name and select **Drop Schema** option.



It opens a popup message where you can choose **Review SQL** if you want to review before dropping. Otherwise, select **Drop Now** to drop the schema and its entire contents permanently.



You can observe that mydb table has disappeared under **Schemas** section.

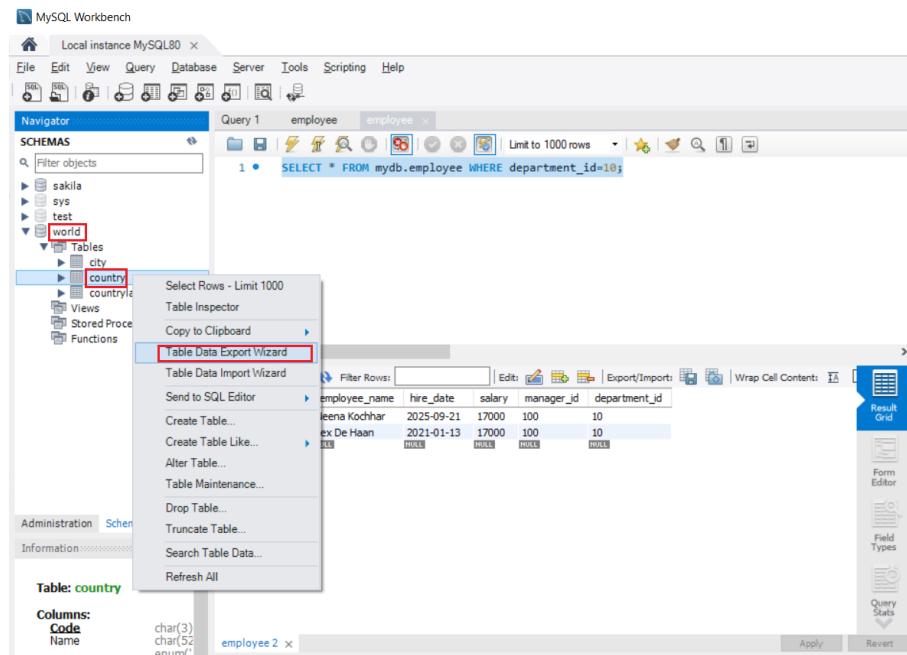


The screenshot shows the MySQL Workbench interface. In the top-left corner, there's a red box highlighting the 'Navigator' tab. Below it, the 'SCHEMAS' section is visible, showing the 'world' schema. The main query editor window contains a SQL query: 'SELECT * FROM mydb.employee WHERE department_id=10;'. The results grid shows two rows of employee data:

employee_id	employee_name	hire_date	salary	manager_id	department_id
101	Neena Kochhar	2025-09-21	17000	100	10
102	Lex De Haan	2021-01-13	17000	100	10
NULL	NULL	NULL	NULL	NULL	NULL

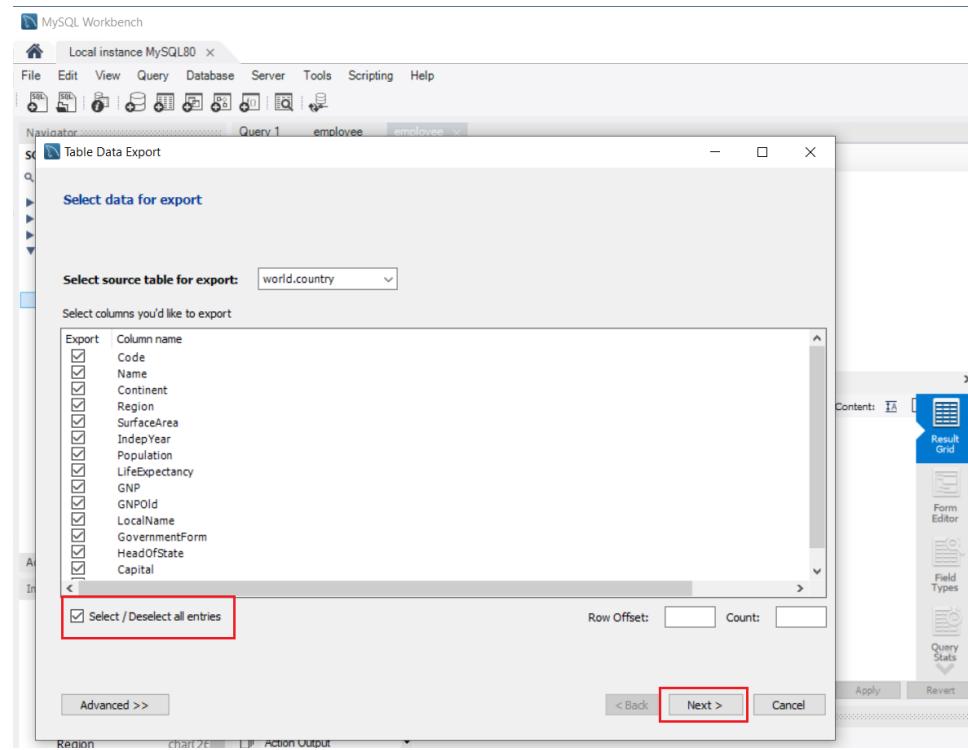
9.7. Export Table:

To export a specific table, such as country table under world schema, right click on the table name and select **Table Data Export Wizard** option.

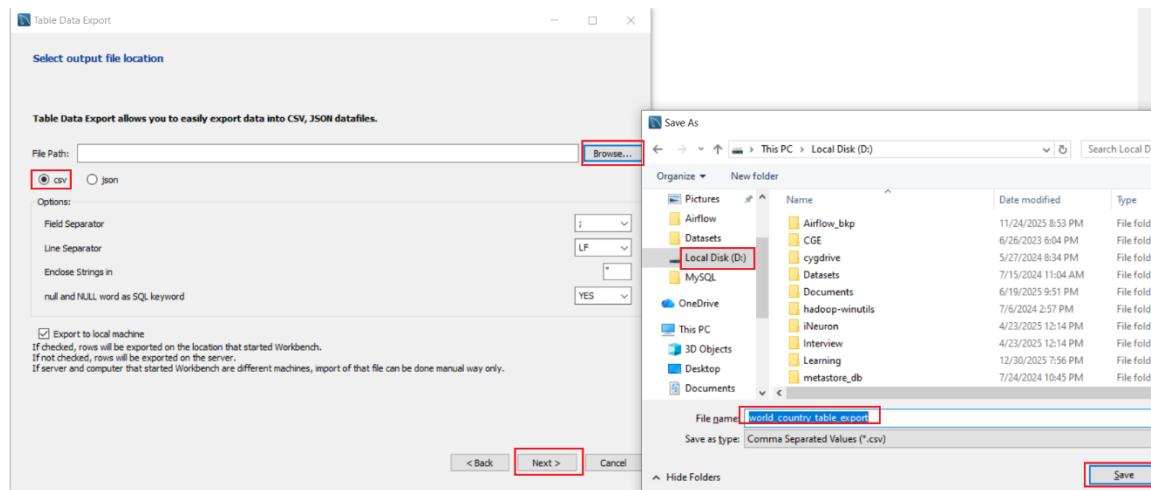


This screenshot shows the MySQL Workbench interface with the 'Navigator' tab selected. The 'world' schema is expanded, and the 'country' table is selected. A context menu is open over the 'country' table, with the 'Table Data Export Wizard' option highlighted by a red box. The menu also includes other options like 'Select Rows - Limit 1000', 'Table Inspector', 'Copy to Clipboard', and 'Table Data Import Wizard'. The main query editor window shows the same SELECT query and result set as the previous screenshot.

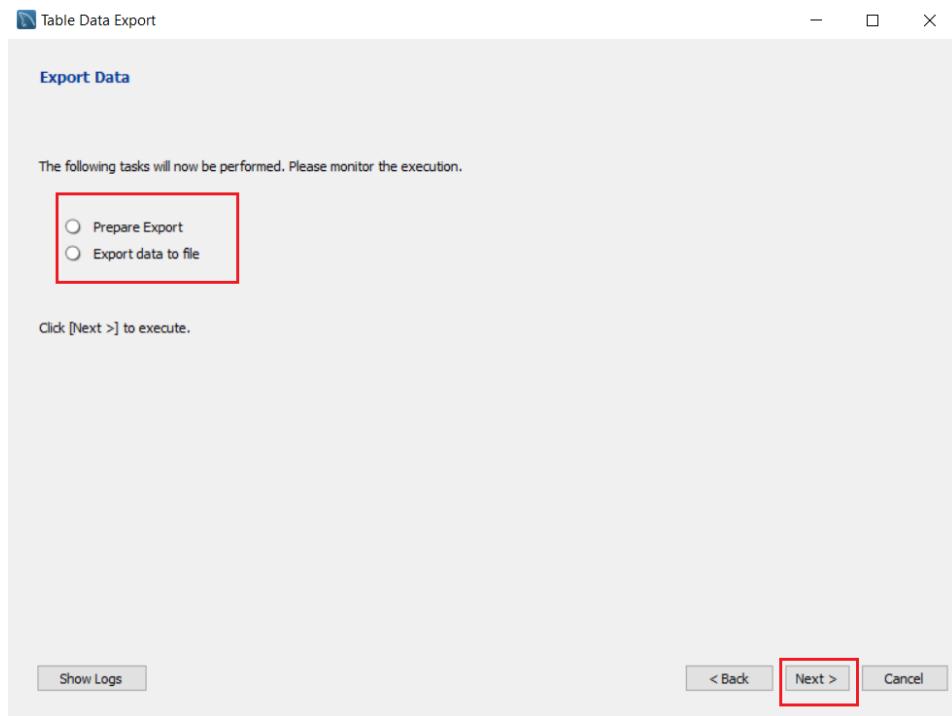
Keep all columns selected (or select required columns for export) and click on **Next** button.



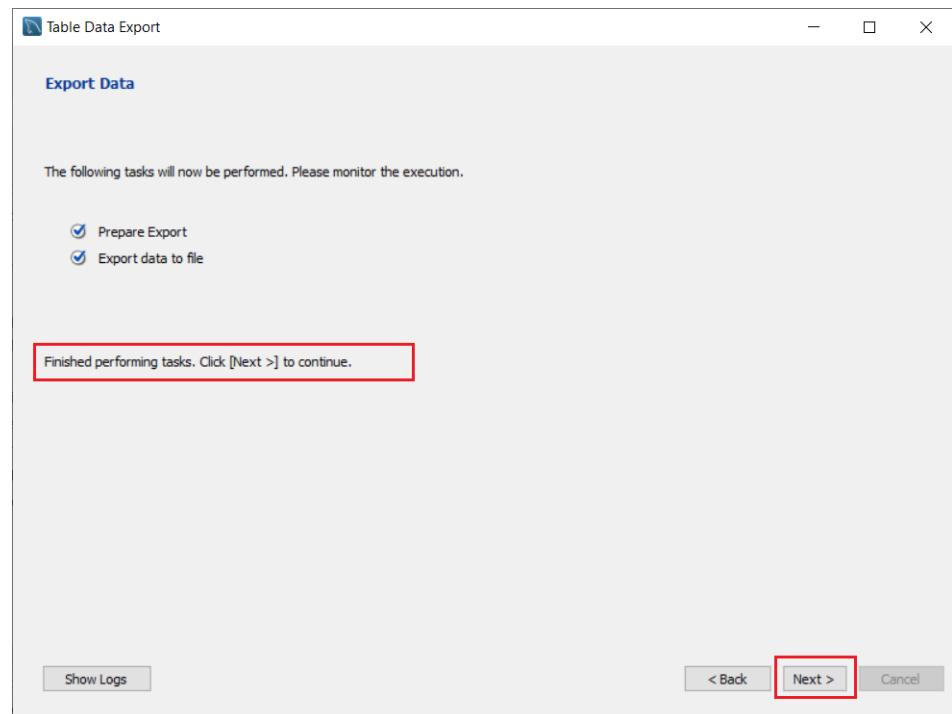
Select the file format (CSV or JSON) and click on **Browse** button to select the desired directory and provide the file name and click on **Save**. Then click on **Next** to continue.



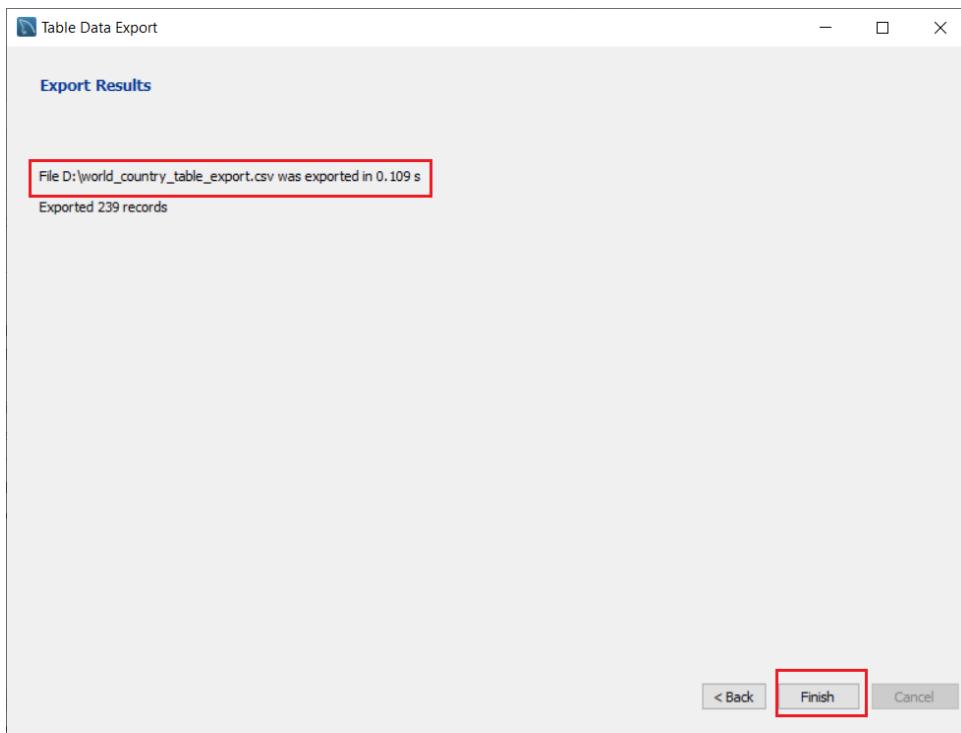
It displays the list of tasks to be performed. Click on **Next** to continue.



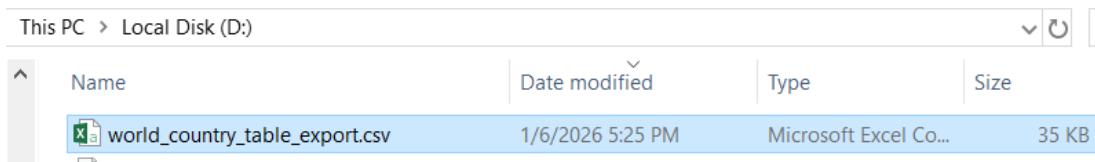
In few seconds, data export would be finished. Click **Next** to continue.



It displays the file location where the data got exported successfully. Click on **Finish** to exit the wizard.

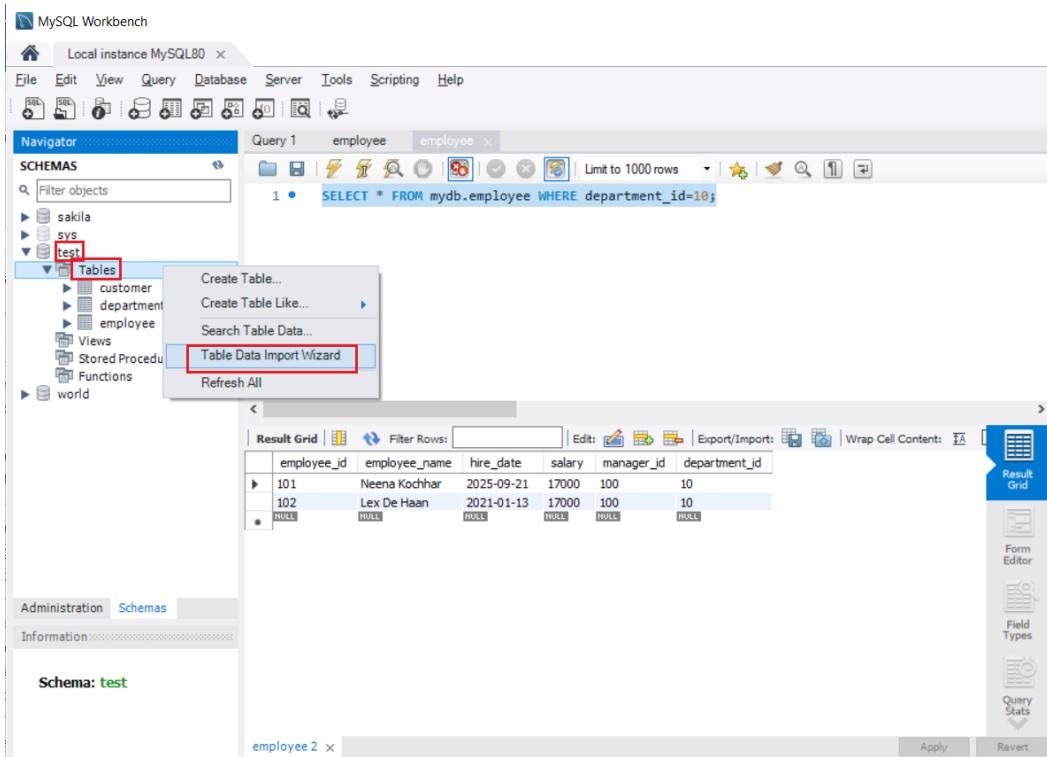


Once finished, navigate to the directory and review the CSV file created.

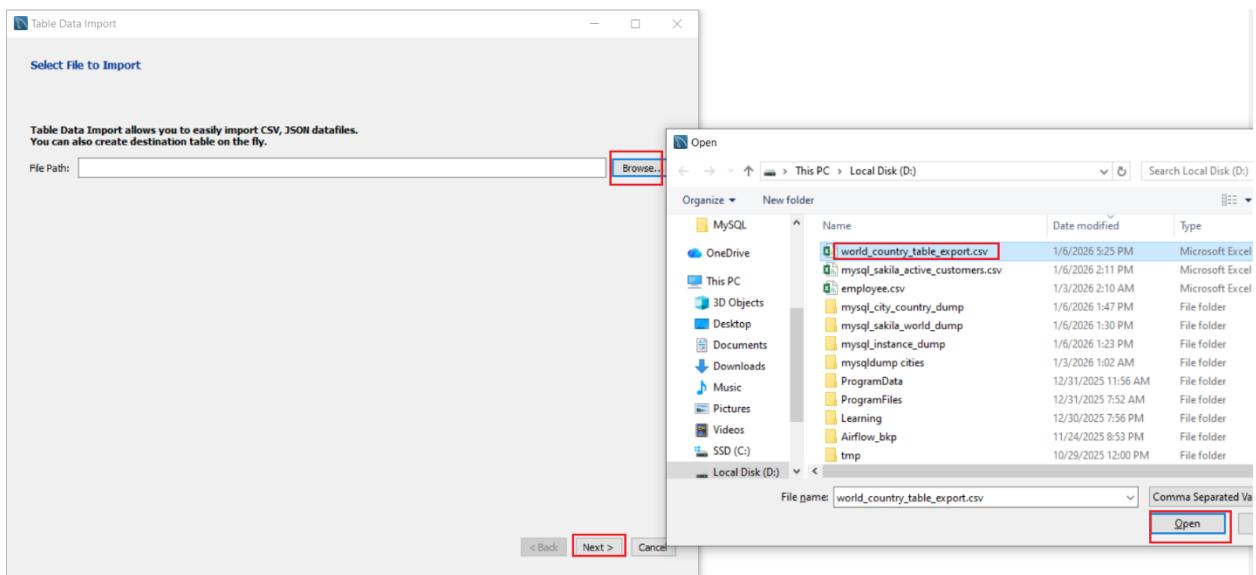


9.8. Import Table:

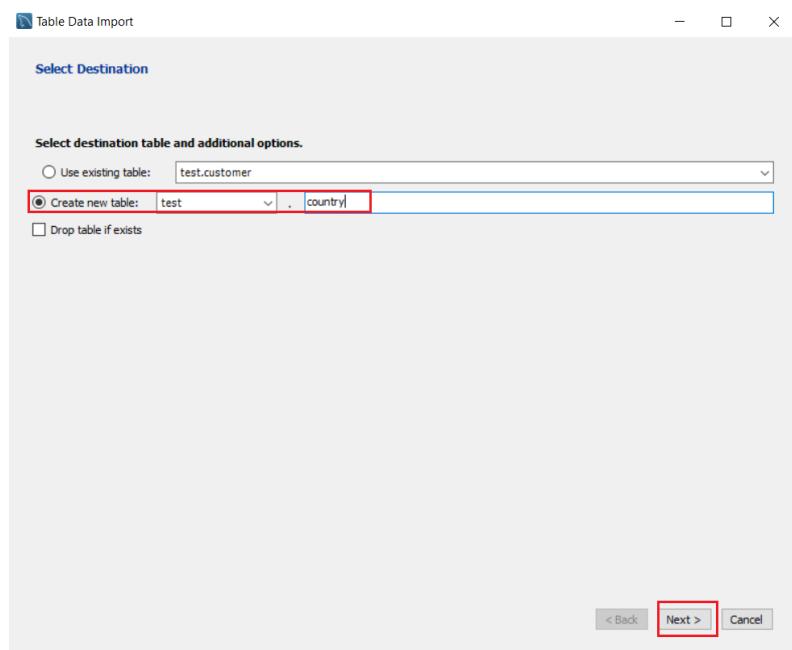
To import data right click on the **Tables** folder under specific schema such as `test` and select **Table Data Export Wizard** option.



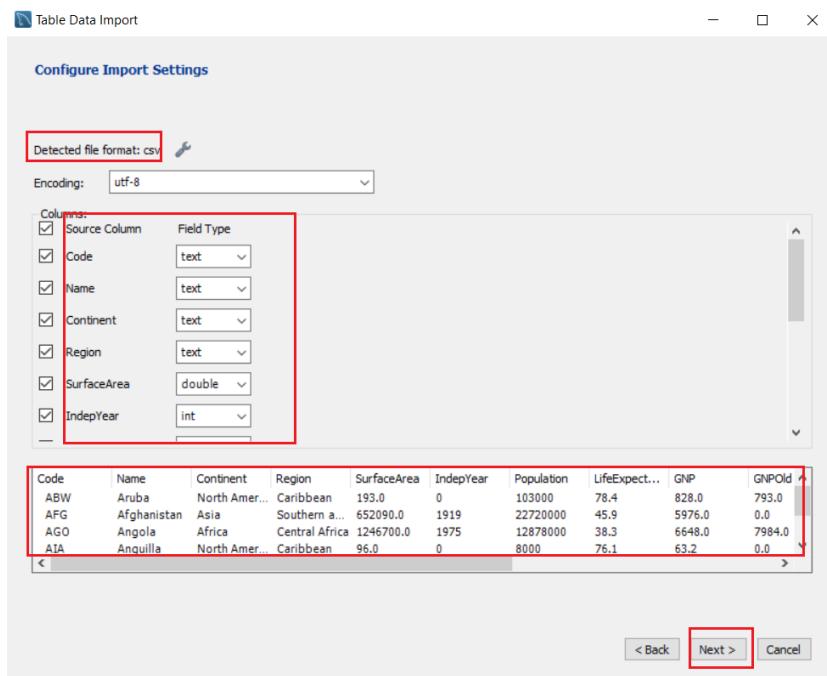
Click on **Browse** button and navigate to the directory and select the file to be imported and click **Open**. Then click on **Next** to continue.



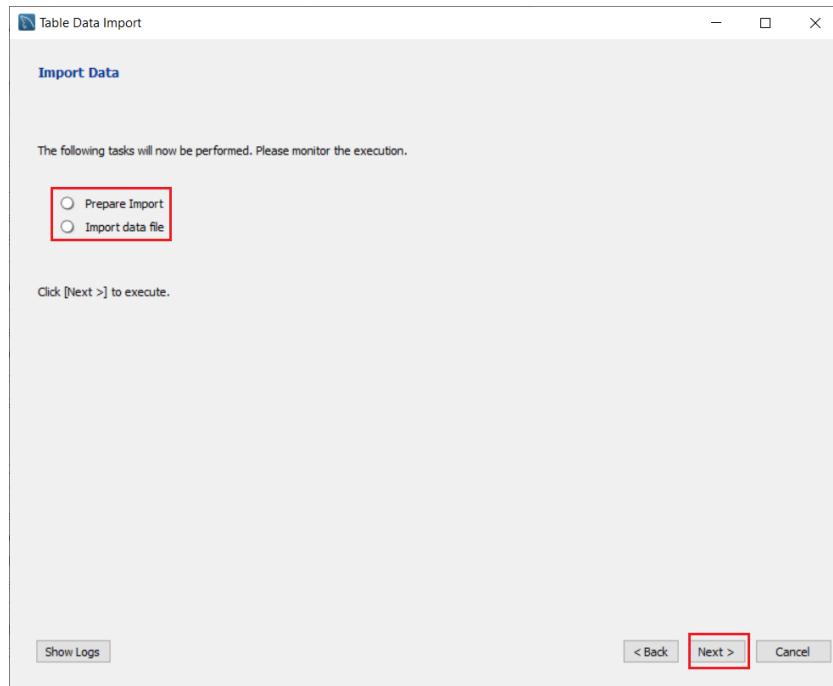
By default, it tries to create new table with the file name. Provide the respective table name and click on **Next** to continue.



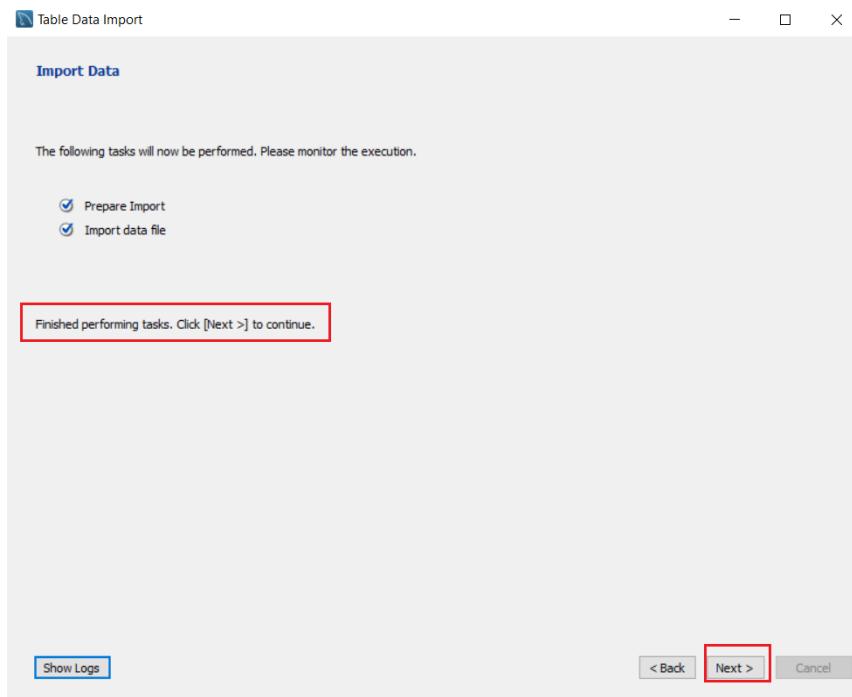
It automatically detects the file format and displays the list of columns with auto-detected datatypes and sample data. Review data types based on sample data and click on **Next** to continue.



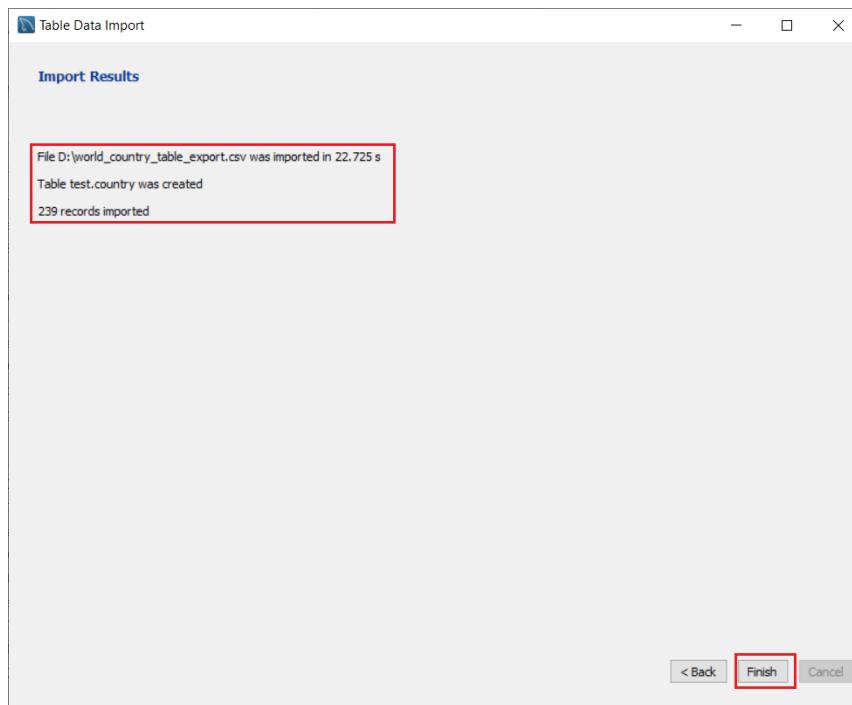
It displays the list of tasks to be performed. Click on **Next** to continue.



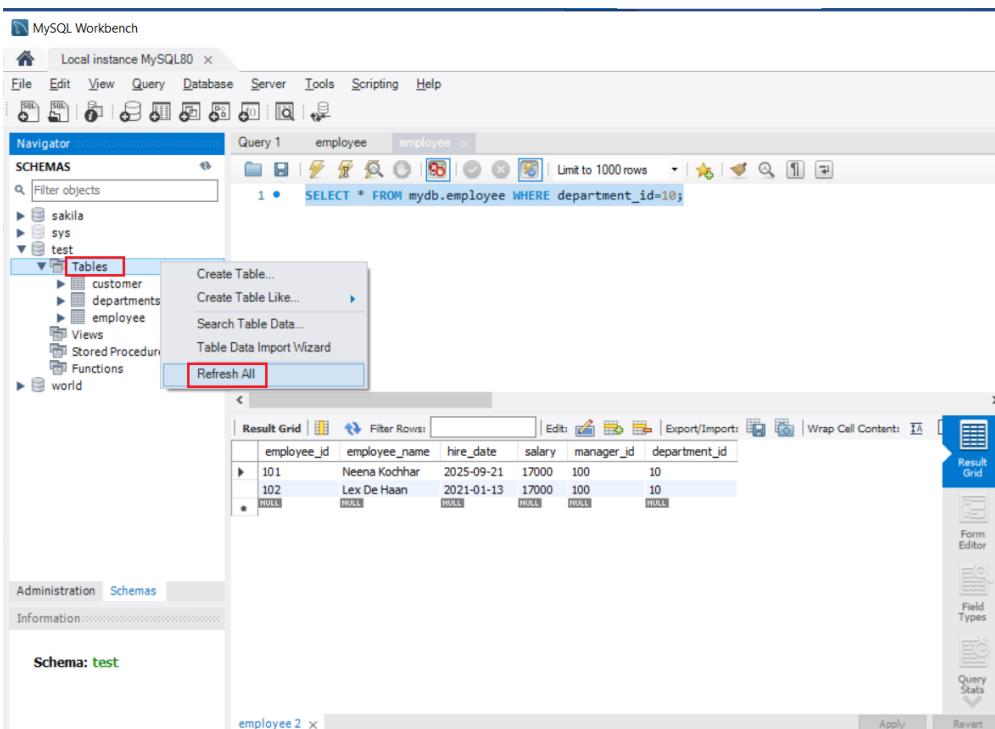
In few seconds, data import would be finished. Click **Next** to continue.



It displays the number of records imported to a table from the file. Click on **Finish** to exit the wizard.



Once finished, right click on **Tables** folder under the schema where table got imported and select **Refresh All** option.



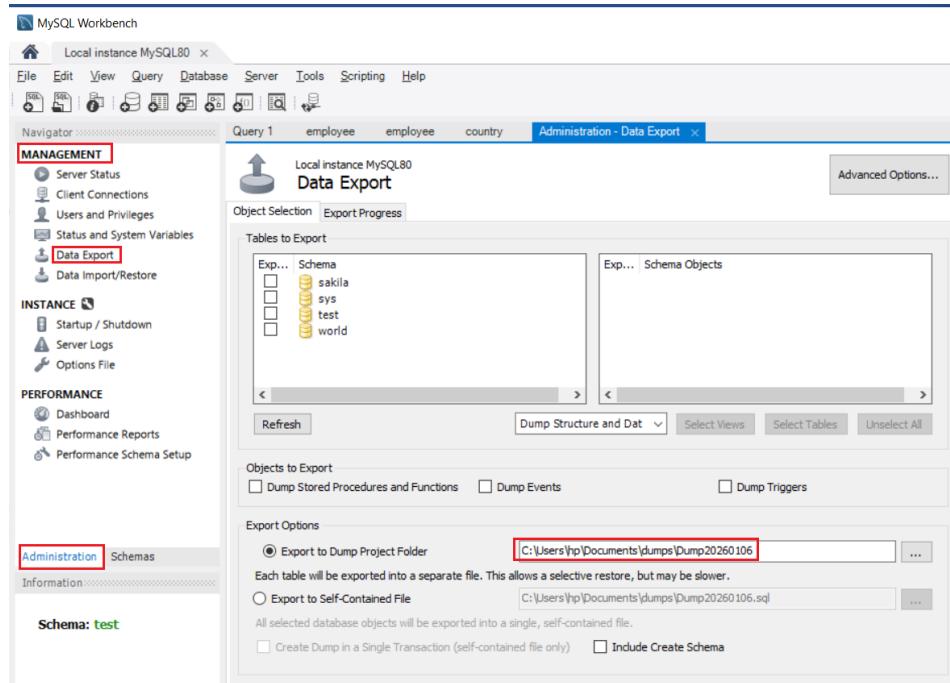
You can see the `country` table created. Click on table icon next to `country` table which executes the select query and displays all records available in the table.

The screenshot shows the MySQL Workbench interface. In the Navigator pane, under the 'test' schema, the 'Tables' section contains a table named 'country'. A red box highlights this table. In the central Query Editor pane, a SQL query is displayed: 'SELECT * FROM test.country;'. The Result Grid pane below shows the data from the 'country' table, also highlighted with a red box. The columns are: Code, Name, Continent, Region, SurfaceArea, IndepYear, Population, and LifeExp. The data includes entries for countries like Aruba, Afghanistan, Angola, Anguilla, Albania, Andorra, Netherlands Antilles, United Arab Emirates, Argentina, Armenia, American Samoa, Antarctica, and French Southern territories.

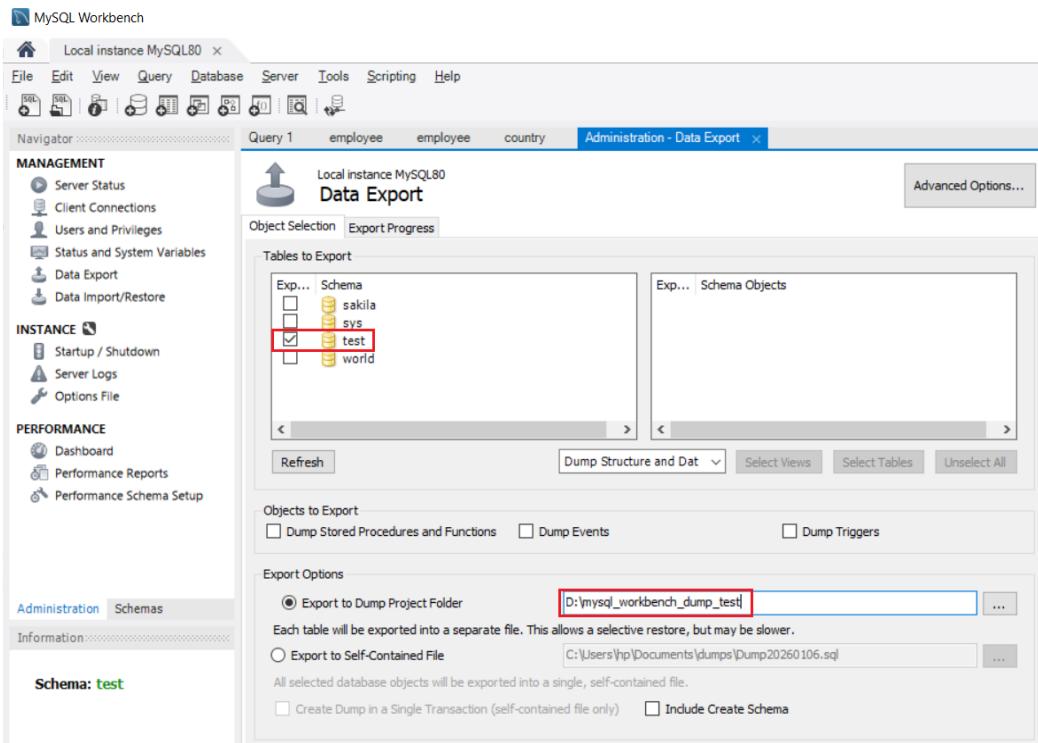
Code	Name	Continent	Region	SurfaceArea	IndepYear	Population	LifeExp
ABW	Aruba	North America	Caribbean	193	0	103000	78.4
AFG	Afghanistan	Asia	Southern and Central Asia	652090	1919	23720000	45.9
AGO	Angola	Africa	Central Africa	1246700	1975	12878000	38.3
ATA	Anguilla	North America	Caribbean	96	0	8000	76.1
ALB	Albania	Europe	Southern Europe	28748	1912	3401200	71.6
AND	Andorra	Europe	Southern Europe	468	1278	78000	83.5
ANT	Netherlands Antilles	North America	Caribbean	800	0	217000	74.7
ARE	United Arab Emirates	Asia	Middle East	83600	1971	2441000	74.1
ARG	Argentina	South America	South America	2780400	1816	37032000	75.1
ARM	Armenia	Asia	Middle East	29800	1991	3520000	66.4
ASM	American Samoa	Oceania	Polynesia	199	0	68000	75.1
ATA	Antarctica	Antarctica	Antarctica	13120000	0	0	0
ATF	French Southern ter...	Antarctica	Antarctica	7780	0	0	0

9.9. Export Database:

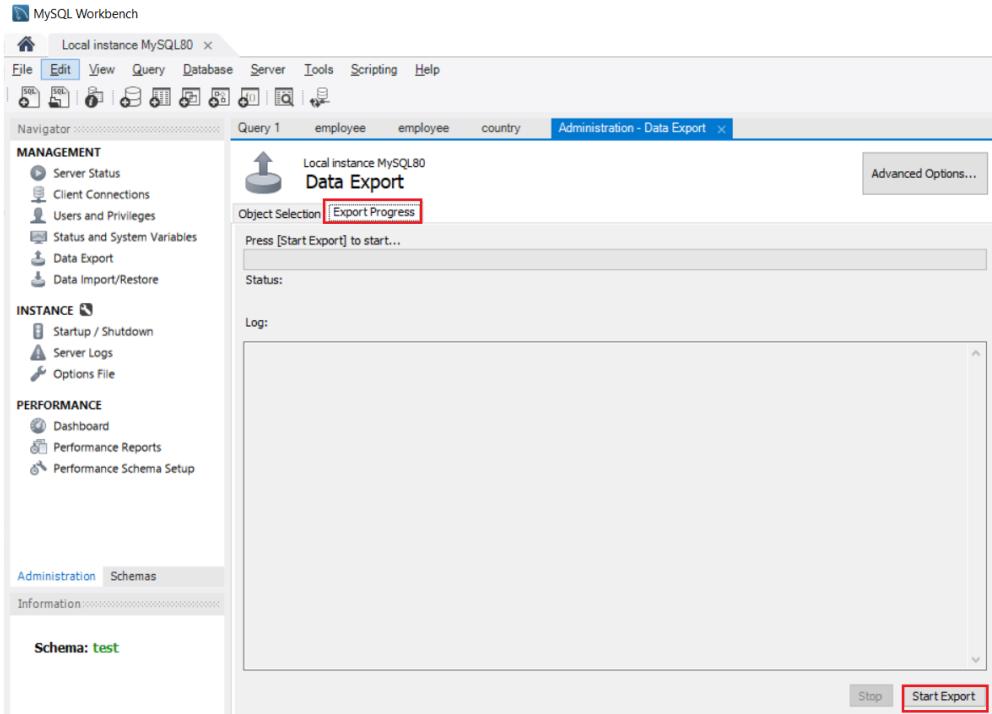
To export a specific database, click on **Administration** tab on the right side and select **Data Export** under **Management** which opens up **Administration-Data Export** tab with list of schemas and the default dump directory.



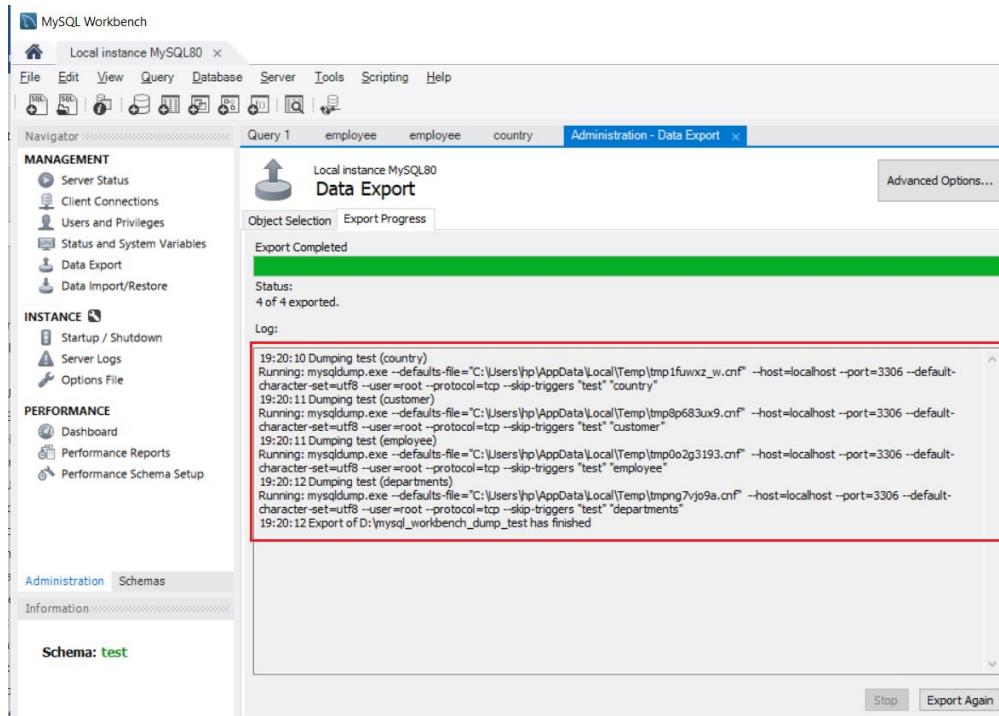
Select the database/schema name such as `test` and provide the appropriate location for data dump.



Select **Export Progress** tab and click on **Start Export** to begin with the data export.



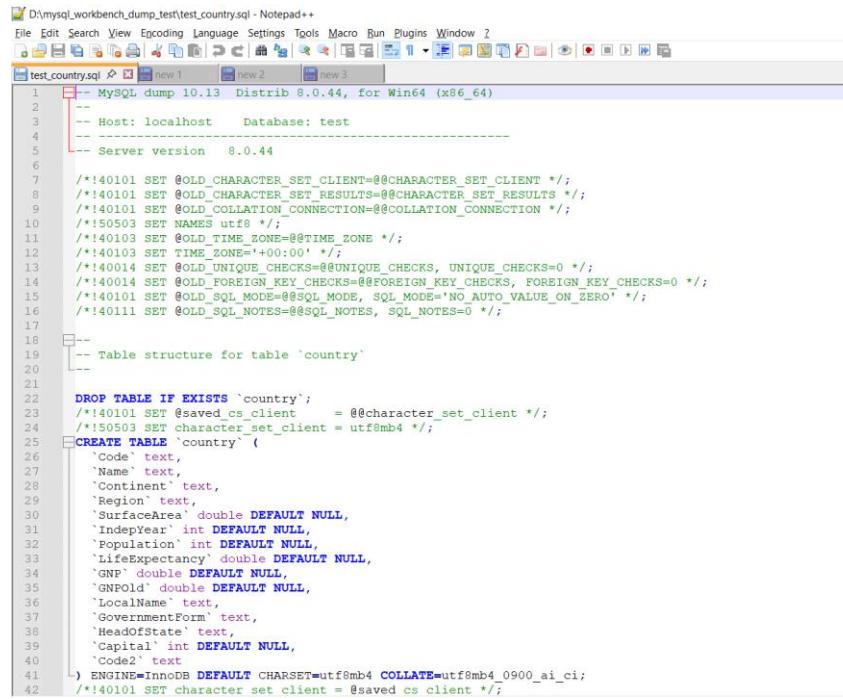
In few seconds, the data export would be finished. Close **Administration-Data Export** tab.



Once finished, go to the location where database dump created and you can see SQL files created

This PC > Local Disk (D:) > mysql_workbench_dump_test			
Name	Date modified	Type	Size
test_country.sql	1/6/2026 7:20 PM	SQL Text File	36 KB
test_customer.sql	1/6/2026 7:20 PM	SQL Text File	3 KB
test_departments.sql	1/6/2026 7:20 PM	SQL Text File	3 KB
test_employee.sql	1/6/2026 7:20 PM	SQL Text File	3 KB

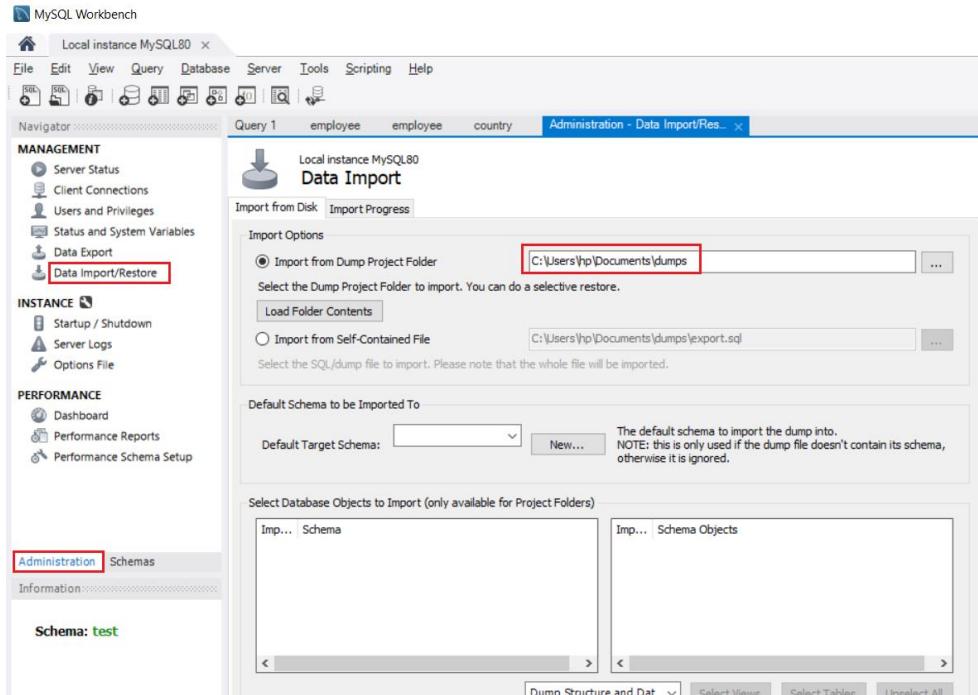
Review the SQL file which contains create and insert statements for the respective table to be created in the target database.



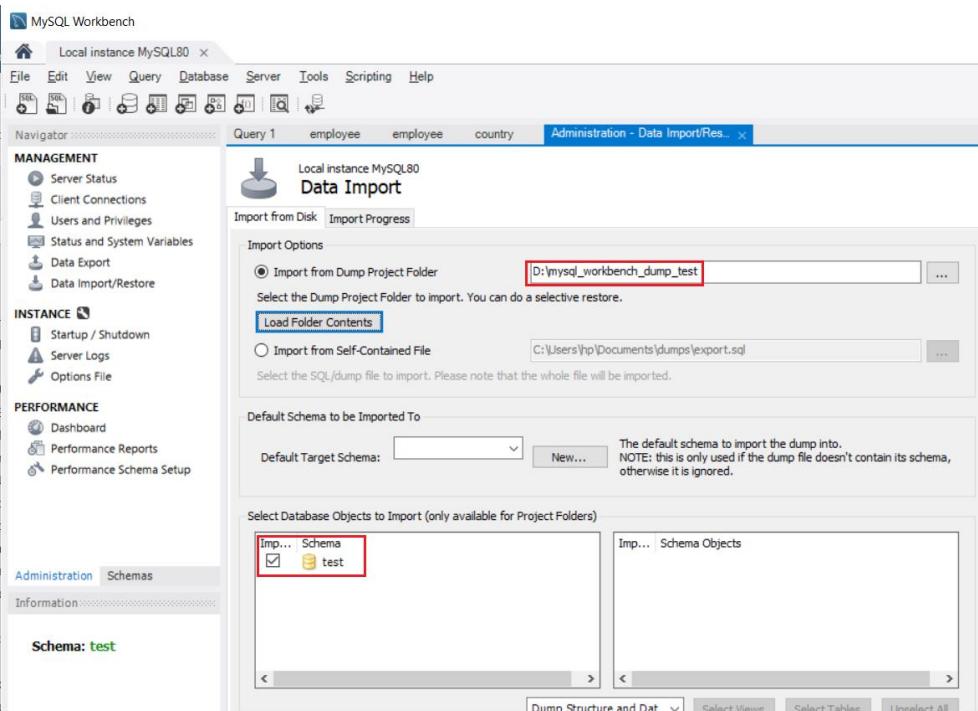
```
D:\mysql_workbench_dump_test\test_country.sql - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
test_country.sql new 1 new 2 new 3
1 -- MySQL dump 10.13 Distrib 8.0.44, for Win64 (x86_64)
2 --
3 -- Host: localhost      Database: test
4 --
5 -- Server version     8.0.44
6
7 /*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
8 /*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
9 /*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
10 /*!50503 SET NAMES utf8 */;
11 /*!40103 SET @OLD_TIME_ZONE=@TIMEZONE */;
12 /*!40103 SET TIME_ZONE='+00:00' */;
13 /*!40014 SET @OLD_UNIQUE_CHECKS=@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
14 /*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
15 /*!40101 SET @OLD_SQL_MODE=@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
16 /*!40111 SET @OLD_SQL_NOTES=@SQL_NOTES, SQL_NOTES=0 */;
17
18 /**
19 -- Table structure for table `country`
20 */
21
22 DROP TABLE IF EXISTS `country`;
23 /*!40101 SET @saved_cs_client      = @@character_set_client */;
24 /*!50503 SET character_set_client = utf8mb4 */;
25 CREATE TABLE `country` (
26   `Code` text,
27   `Name` text,
28   `Continent` text,
29   `Region` text,
30   `SurfaceArea` double DEFAULT NULL,
31   `IndepYear` int DEFAULT NULL,
32   `Population` int DEFAULT NULL,
33   `LifeExpectancy` double DEFAULT NULL,
34   `GNP` double DEFAULT NULL,
35   `GNPold` double DEFAULT NULL,
36   `LocalName` text,
37   `GovernmentForm` text,
38   `HeadofState` text,
39   `Capital` int DEFAULT NULL,
40   `Code2` text,
41   ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
42 /*!40101 SET character_set_client = @saved_cs_client */;
```

9.10. Import Database:

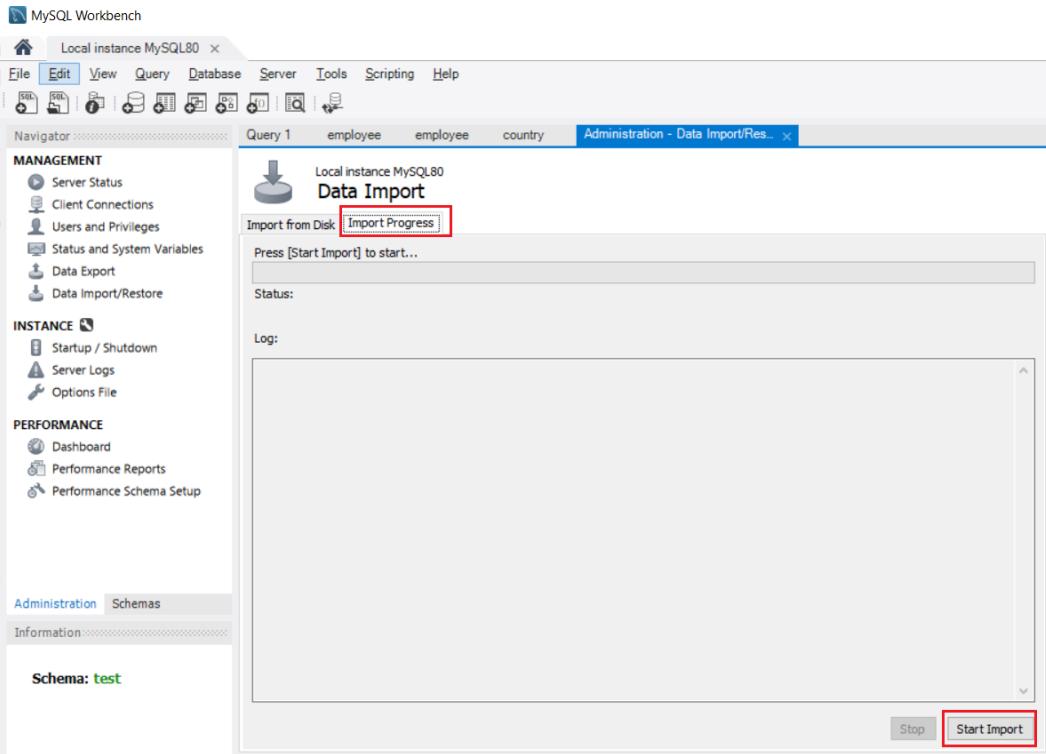
To import a specific database, click on **Administration** tab on the right side and select **Data Import/Restore** under **Management** which opens up **Administration-Data Import** tab with the default dump directory.



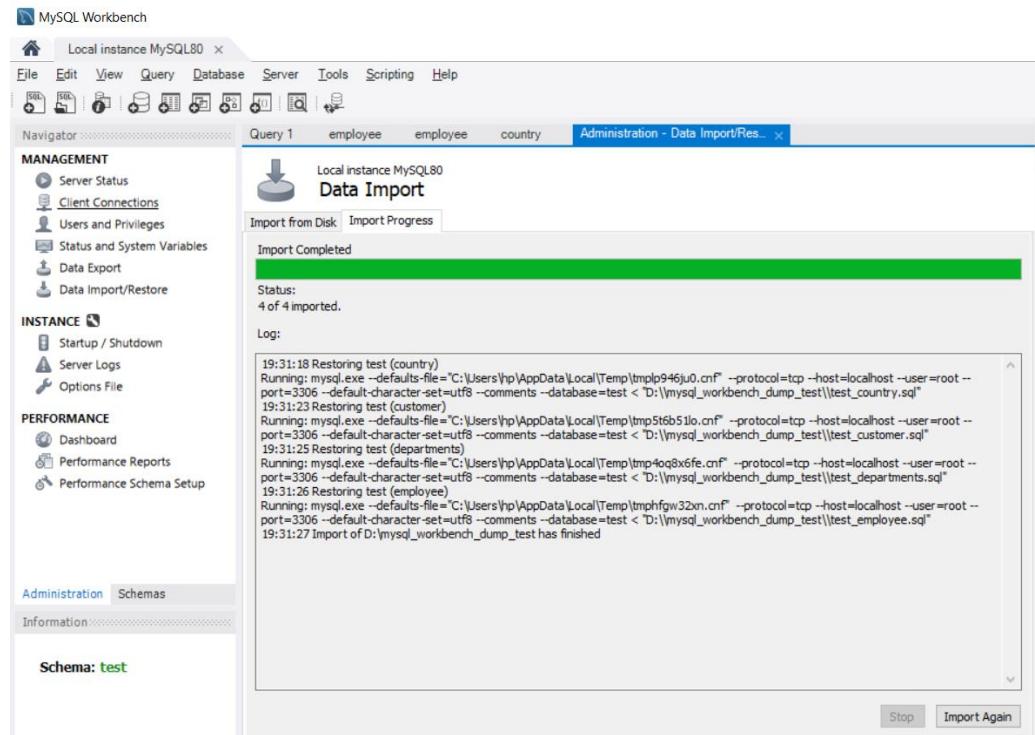
Provide the appropriate location where the database dump exists. Then it displays the list of databases to be imported



Select **Import Progress** tab and click on **Start Import** to begin with the database import.



In few seconds, the data import would be finished. Close **Administration-Data Import** tab.



Once finished, right click on **Tables** folder under the schema and select **Refresh All** option to see the list of tables created.

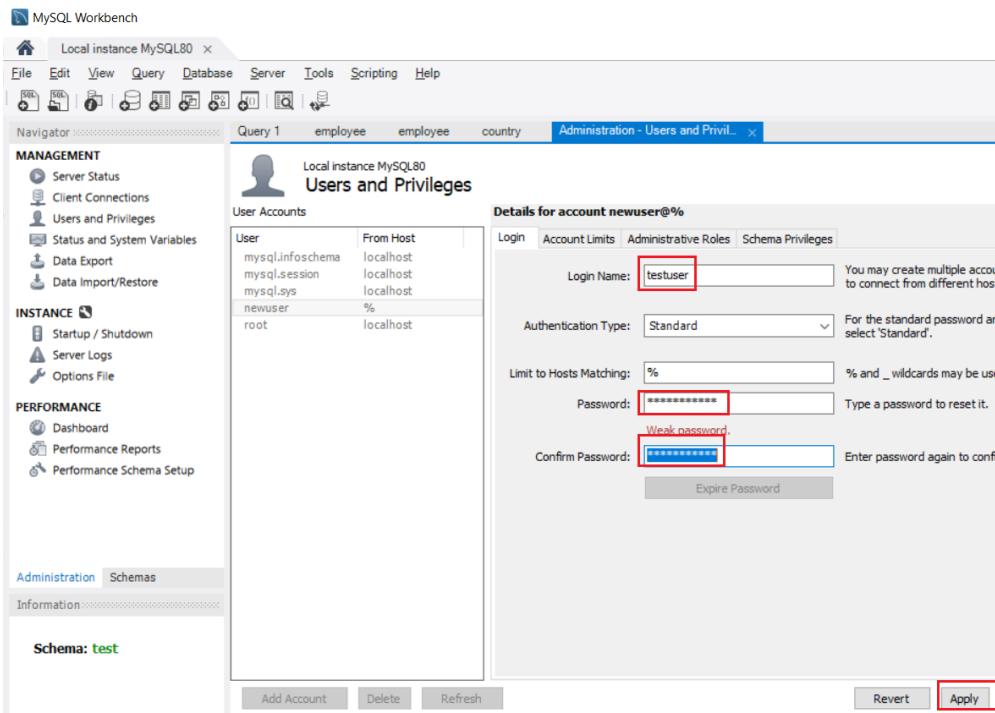
The screenshot shows the MySQL Workbench interface. In the Navigator pane, the 'test' schema is selected, and the 'Tables' folder is highlighted with a red box. A context menu is open over the 'Tables' folder, with the 'Refresh All' option highlighted with a red box. The main query editor window displays a SELECT query and its results for the 'country' table.

9.11. Create User:

To create a user, click on **Administration** tab on the right side and select **Users and Privileges** under **Management** which opens up **Administration- Users and Privileges** tab with the list of users available. Click on **Add Account** to create user.

The screenshot shows the MySQL Workbench interface with the 'Administration' tab selected in the sidebar. The 'Users and Privileges' section is selected under the 'Management' category. The 'Administration - Users and Privileges' tab is active, showing a list of existing users ('mysql.infoschema', 'mysql.session', 'mysql.sys', 'root') with their respective host entries ('localhost'). A red box highlights the 'Add Account' button at the bottom left of the tab area. The 'Information' and 'Schema: test' tabs are also visible in the bottom left.

Provide the new user Id and password and click on **Apply** button. Close **Administration- Users and Privileges** tab.



Congratulations!! You have successfully installed MySQL and executed SQL commands from various MySQL command line utilities and Workbench GUI.