**MySQL-SQL**

## WHAT IS SQL?

* SQL stands for Structured Query Language
* Used for managing and manipulating relational databases.
* SQL lets you access and manipulate databases
* SQL became a standard of the American National Standards Institute (ANSI) in 1986, and of the International Organization for Standardization (ISO) in 1987.

## WHAT CAN SQL DO?

* SQL can execute queries against a database
* SQL can retrieve data from a database
* SQL can insert records in a database
* SQL can update records in a database
* SQL can delete records from a database
* SQL can create new databases
* SQL can create new tables in a database
* SQL can create stored procedures in a database
* SQL can create views in a database
* SQL can set permissions on tables, procedures, and views

## LIST OF RELATIONAL DATABASE MANAGEMENT SYSTEMS

* MySQL
* PostgreSQL
* Oracle Database
* Microsoft SQL Server
* SQLite
* IBM Db2
* MariaDB

## CASE SENSITIVE OR NOT?

* KEYWORDS AND IDENTIERS ARE CASE INSENSITIVE LITERALS ARE CASE SENSITIVE.

**WHAT DO YOU MEAN BY DBMS? WHAT ARE ITS DIFFERENT TYPES?**

Database is a structured collection of data.

A Database Management System (DBMS) is a software application that interacts with the user, applications and the database itself to capture and analyse data.

A DBMS allows a user to interact with the database using query language such as SQL. The data stored in the database can be modified, retrieved and deleted and can be of any type like strings, numbers, images etc.

**THERE ARE TWO TYPES OF DBMS:**

**Relational Database Management System:** The data is stored in relations (tables). Example – MySQL, Oracle SQL.

**Non-Relational Database Management System:** There is no concept of relations, tuples and attributes. Example – Mongo

## WHAT ARE THE DIFFERENT SUBSETS OF SQL?

The standard SQL commands to interact with relational databases are CREATE, SELECT, INSERT, UPDATE, DELETE and DROP. These commands can be classified into the following groups based on their nature –

### **DDL - Data Definition Language**

|  |  |
| --- | --- |
|  | **Command & Description** |
| 1 | **CREATE**  Creates a new table, a view of a table, or other object in the database. |
| 2 | **ALTER**  Modifies an existing database object, such as a table. |
| 3 | **DROP**  Deletes an entire table, a view of a table or other objects in the database. |

### **DML - Data Manipulation Language**

|  |  |
| --- | --- |
|  | **Command & Description** |
| 1 | **SELECT**  Retrieves certain records from one or more tables. |
| 2 | **INSERT**  Creates a record. |
| 3 | **UPDATE**  Modifies records. |
| 4 | **DELETE**  Deletes records. |

### **DCL - Data Control Language**

|  |  |
| --- | --- |
|  | **Command & Description** |
| 1 | **GRANT**  Gives a privilege to user. |
| 2 | **REVOKE**  Takes back privileges granted from user. |

### **DQL - Data Query Language**

|  |  |
| --- | --- |
|  | **Command & Description** |
| 1 | **SELECT**  The SELECT statement is used to retrieve data from one or more tables. |
| 2 | **DISTINCT**  The DISTINCT keyword is used with SELECT to retrieve unique values from a specified column or a combination of columns. |
| 3 | **FROM**  **The FROM clause specifies the table or tables from which you want to retrieve data.** |
| 4 | **WHERE**  **The WHERE clause is used to filter rows based on a specified condition. It allows you to retrieve only the rows that meet the criteria you specify.** |
| 5 | **ORDER BY**  **The ORDER BY clause is used to sort the result set in ascending (ASC) or descending (DESC) order based on one or more columns.** |
| 6 | **GROUP BY**  **The GROUP BY clause is used to group rows with the same values in one or more columns into summary rows.** |
| 7 | **HAVING**  **The HAVING clause is used to filter the results of a GROUP BY query based on a condition applied to the aggregated values.** |

**WHAT DO YOU MEAN BY TABLE AND FIELD IN SQL?**

A table refers to a collection of data in an organised manner in form of rows and columns. A field refers to the number of columns in a table. For example:

**Table:** StudentInformation

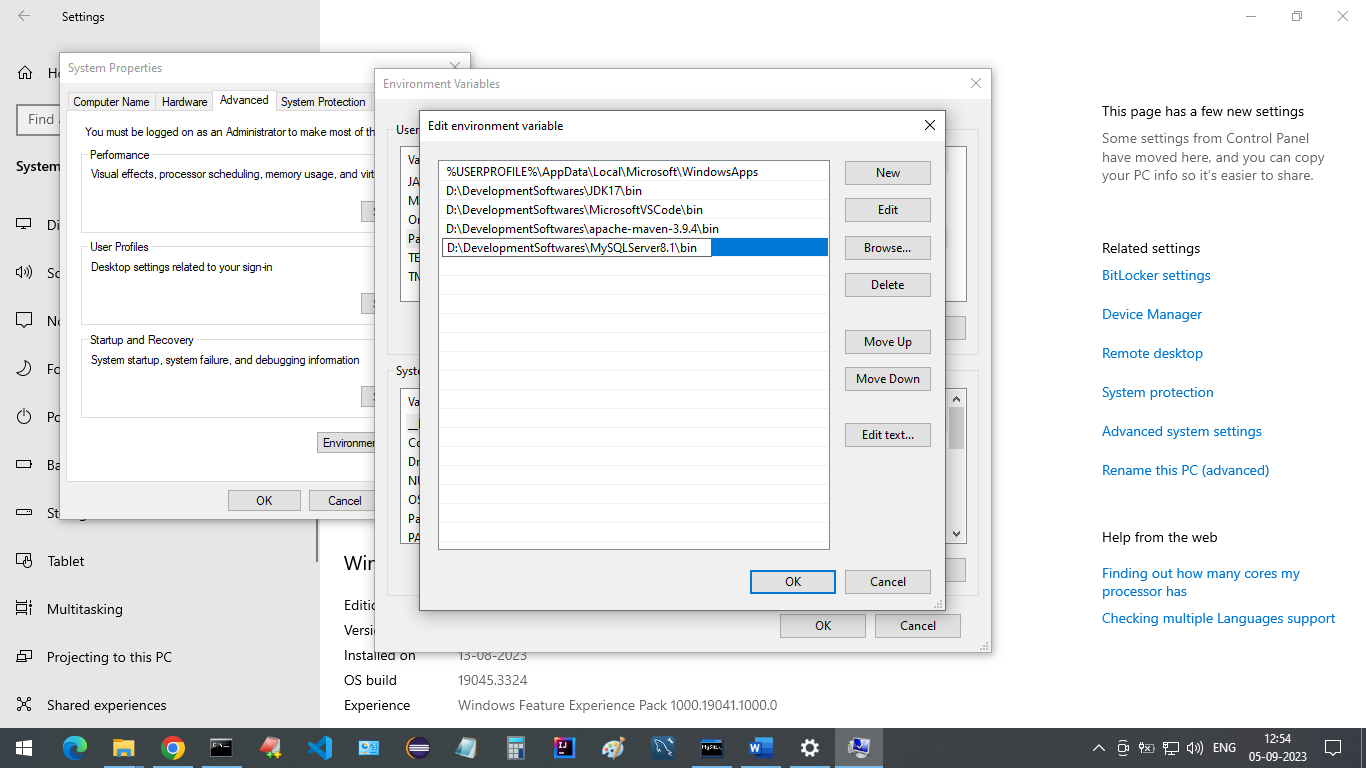
**Field:** StudentId, StudentName, StudentMarks

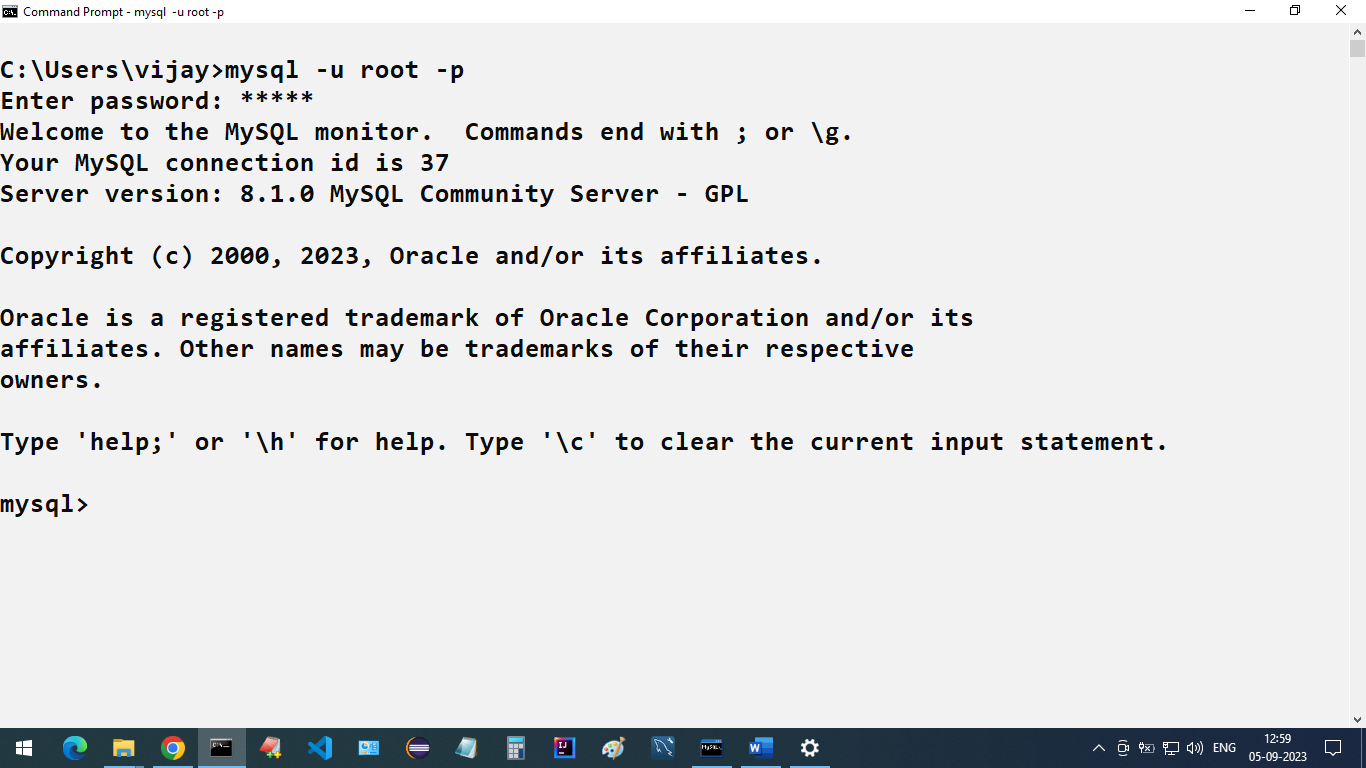
**Please follow the link to Learn how download and install MySQL Database and MySQL Workbench**

[**https://rb.gy/3hcwf**](https://rb.gy/3hcwf)

**Note:** If you are getting error while installing MySQL Server and MySQL Workbench like ‘MySql Workbench installer requires Visual C++ 2015’ then follow <https://aka.ms/vs/17/release/vc_redist.x64.exe> this link and download and install this piece of software.

**To Access the SQL Prompt from the windows command Line client set the path**





**TO CREATE A NEW USER**

CREATE USER 'new\_user'@'localhost' IDENTIFIED BY 'password';

new\_user is the name we’ve given to our new user account and the IDENTIFIED BY ‘password’ section sets a passcode for this user. You can replace these values with your own, inside the quotation marks.

In order to grant all privileges of the database for a newly created user, execute the following command:

GRANT ALL PRIVILEGES ON \* . \* TO 'new\_user'@'localhost';

For changes to take effect immediately flush these privileges by typing in the command:

FLUSH PRIVILEGES;

**TO DISPLAY ALL THE USERS**

**mysql> SELECT user, host FROM mysql.user;**

## SEMICOLON AFTER SQL STATEMENTS?

* Some database systems require a semicolon at the end of each SQL statement.
* Semicolon is the standard way to separate each SQL statement in database systems that allow more than one SQL statement to be executed in the same call to the server.

**CREATE NEW USER WITH PASSWORD**

**mysql> CREATE USER 'manager'@'localhost' IDENTIFIED BY 'admin';**

**mysql> CREATE USER 'vijay'@'localhost' IDENTIFIED BY 'admin';**

**DROP EXISTING USER**

**mysql> DROP USER 'manager'@'localhost';**

**mysql> DROP USER 'vijay'@'localhost';**

**GRANT ALL PRIVILIAGES TO THE NEW USER**

**GRANT ALL PRIVILEGES ON \*.\* TO 'vijay'@'localhost';**

**TO CHECK CURRENT USER PRIVILIAGES**

**mysql> SHOW GRANTS FOR 'root'@'localhost';**

**TO GIVE ALL THE PRIVILIAGES**

**mysql> GRANT ALL PRIVILEGES ON \* . \* TO 'root'@'localhost';**

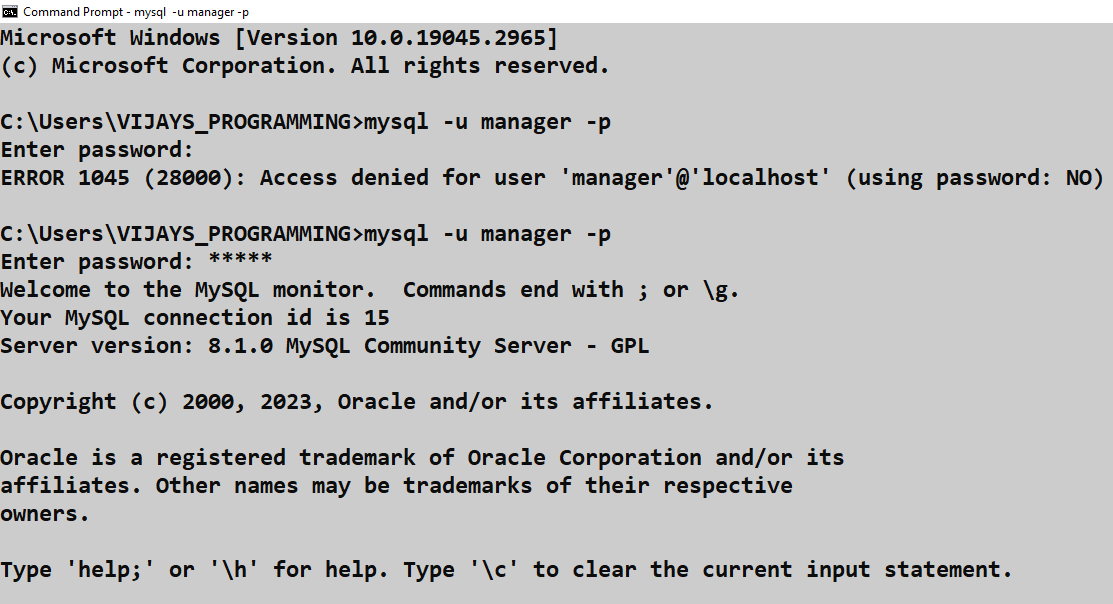
**mysql> GRANT ALL PRIVILEGES ON \* . \* TO 'vijay'@'localhost';**

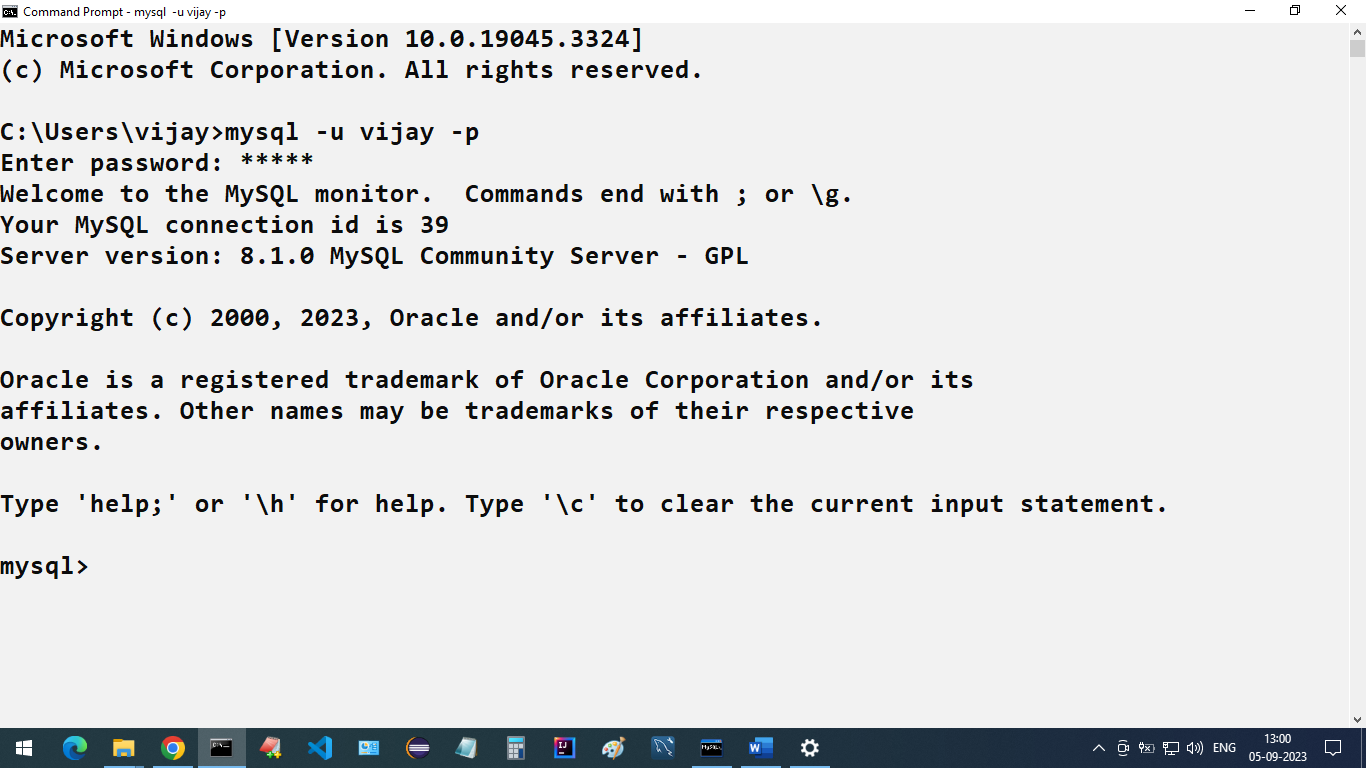
**FOR CHANGES TO TAKE EFFECT IMMEDIATELY**

mysql> FLUSH PRIVILEGES;

**Note:** Starting from MySQL 5.7.3, the FLUSH PRIVILEGES; statement is no longer strictly required after executing GRANT or REVOKE statements. The server automatically reloads the grant tables in these cases.

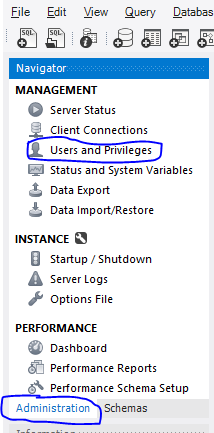
**TO OPEN SQL PROMPT USING NEW USER**



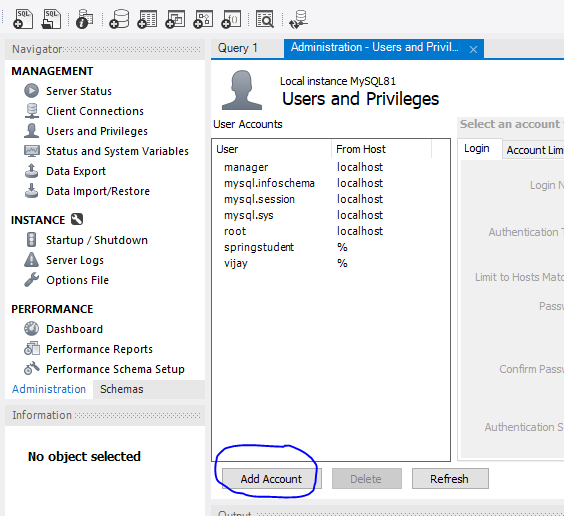


**TO CREATE A NEW USER IN THE MYSQL WORKBENCH**

1. Log in to any connection
2. Click on Administration on the left hand side

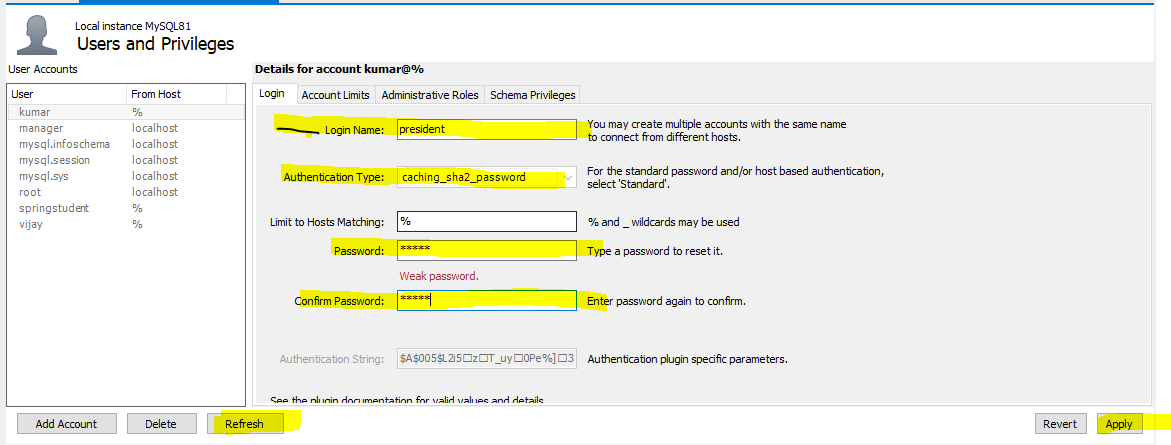


1. Click on Users and Privileges
2. Click on Add account to create a new account



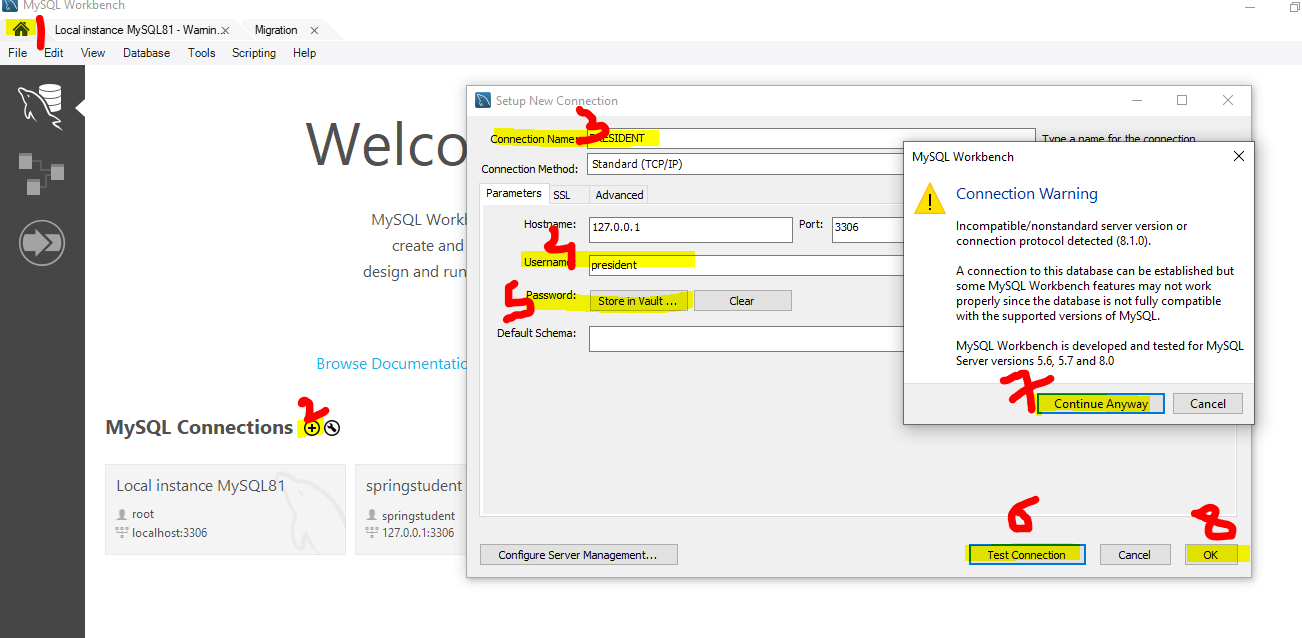
1. Fill in the deatails

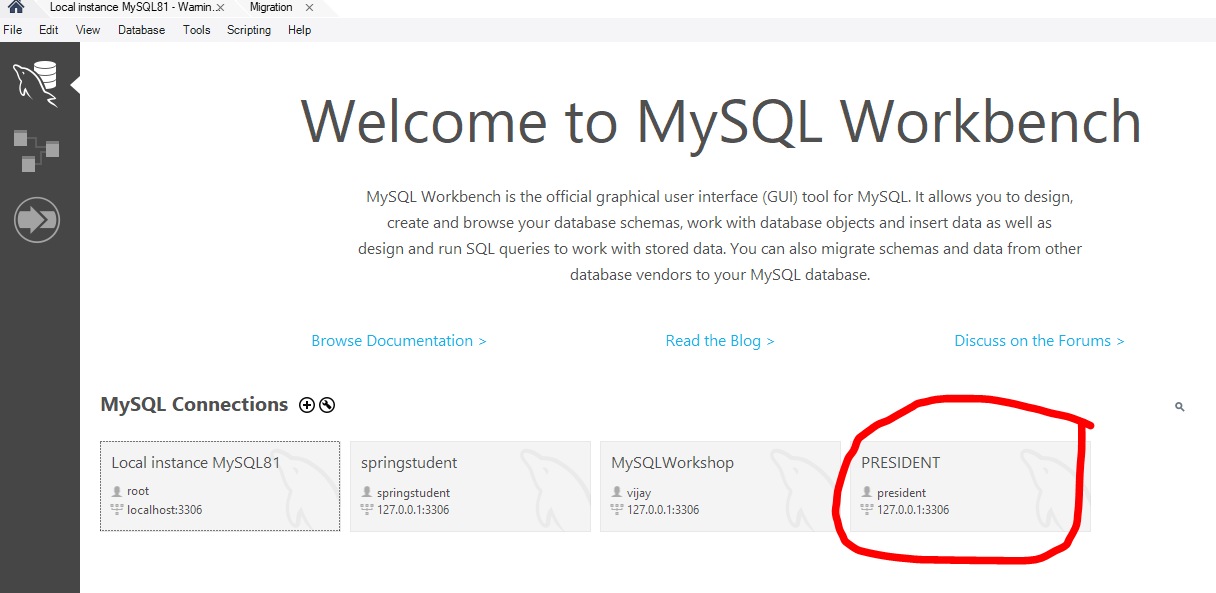
**Note:** Authentication type should be same as other accounts(check for root)



1. Click on Apply and Refresh

**TO ADD A NEW CONNECTION TO THE MYSQL WORKBENCH HOME**





**Note:** While creating the connection the user must be available(created already). Password is the user password that you have given at the time of creating a user.

**TO CLEAR THE SCREEN IN MYSQL**

mysql> \! Cls

**TO DISPLAY ALL THE DATABASES**

**SHOW DATABASES** command to get list of databases. Run the following query to show list of databases.

mysql> SHOW DATABASES;

+--------------------+

| Database |

+--------------------+

| information\_schema |

| mysql |

| mysql\_notes |

| performance\_schema |

| student\_tracker |

| sys |

+--------------------+

**TO CREATE A NEW DATABASE**

mysql> CREATE DATABASE MYSQL\_NOTES;

Query OK, 1 row affected (0.01 sec)

mysql> SHOW DATABASES;

+--------------------+

| Database |

+--------------------+

| information\_schema |

| mysql |

| mysql\_notes |

| performance\_schema |

| student\_tracker |

| sys |

+--------------------+

**TO DELETE A DATABASE**

mysql> DROP DATABASE MYSQL\_NOTES;

Query OK, 0 rows affected (0.01 sec)

mysql> SHOW DATABASES;

+--------------------+

| Database |

+--------------------+

| information\_schema |

| mysql |

| performance\_schema |

| student\_tracker |

| sys |

+--------------------+

**TO SET OR SELECT A DATABASE**

* Before doing anything first we need to connect to a database.

mysql> USE MYSQL\_NOTES;

Database changed

**TO CHECK CURRENTLY WHICH DATABASE YOU ARE IN**

mysql> SELECT DATABASE();

+-------------+

| DATABASE() |

+-------------+

| mysql\_notes |

+-------------+

1 row in set (0.00 sec)

**CREATING A SIMPLE TABLE**

**CREATE TABLE STUDENT (ID INTEGER, FIRST\_NAME VARCHAR(90), AGE INTEGER, COURSE VARCHAR(10));**

**Query OK, 0 rows affected (0.03 sec)**

* INTEGER is a data type synonym for INT.
* You can use both INT and INTEGER datatype to specify number types.

mysql> INSERT INTO STUDENT VALUES (101, 'ARUN', 20, 'CSE');

Query OK, 1 row affected (0.02 sec)

mysql> INSERT INTO STUDENT VALUES (102, 'BHAVESH', 21, 'ISE');

Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO STUDENT VALUES (103, 'CHAITANYA', 22, 'ECE');

Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO STUDENT VALUES (104, 'DEEPIKA', 23, 'MECH');

Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO STUDENT VALUES (105, 'DHANUSH', 24, 'DS');

Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO STUDENT VALUES (106, 'EKTA', 25, 'AI');

Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO STUDENT VALUES (107, 'GAURAV', 26, 'ARCH');

Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO STUDENT VALUES (108, 'HARSHITA', 27, 'CHEMICAL');

Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO STUDENT VALUES (109, 'ISHAAN', 28, 'CIVIL');

Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO STUDENT VALUES (110, 'JANU', 29, 'EEE');

Query OK, 1 row affected (0.00 sec)

**TO DISPLAY ALL THE RECORDS WITH ALL THE COLUMNS**

mysql> SELECT \* FROM STUDENT;

+------+------------+------+----------+