

ACCA5026 Database Application Development Tutorial Tasks

1: What is MYSQL?

1.2 List some features of MYSQL.

1.3: Install MYSQL on Windows

Steps:

1. Visit the Official MySQL Website

<https://dev.mysql.com/downloads/mysql/>

2. Select the Downloads option.

General Availability (GA) Releases Archives Help

MySQL Installer 8.0.35

Note: MySQL 8.0 is the final series with MySQL Installer. As of MySQL 8.1, use a MySQL product's MSI or Zip archive for installation. MySQL Server 8.1 and higher also bundle MySQL Configurator, a tool that helps configure MySQL Server.

Select Version:
8.0.35

Select Operating System:
Microsoft Windows

Windows (x86, 32-bit), MSI Installer (mysql-installer-web-community-8.0.35.0.msi)	8.0.35	2.1M	Download
Windows (x86, 32-bit), MSI Installer (mysql-installer-community-8.0.35.0.msi)	8.0.35	288.6M	Download

MD5: 214df2ccdf83eb5edc6ca7c115792406 | [Signature](#)

MD5: 2cfda448a2971b6b5323775ef9e8d012 | [Signature](#)

Go to the Downloads Section

On the MySQL homepage, Click on the” **No thanks, just start my download**” link to proceed MySql downloading.

MySQL Community Downloads

Login Now or Sign Up for a free account.

An Oracle Web Account provides you with the following advantages:

- Fast access to MySQL software downloads
- Download technical White Papers and Presentations
- Post messages in the MySQL Discussion Forums
- Report and track bugs in the MySQL bug system

[Login »](#)
using my Oracle Web account

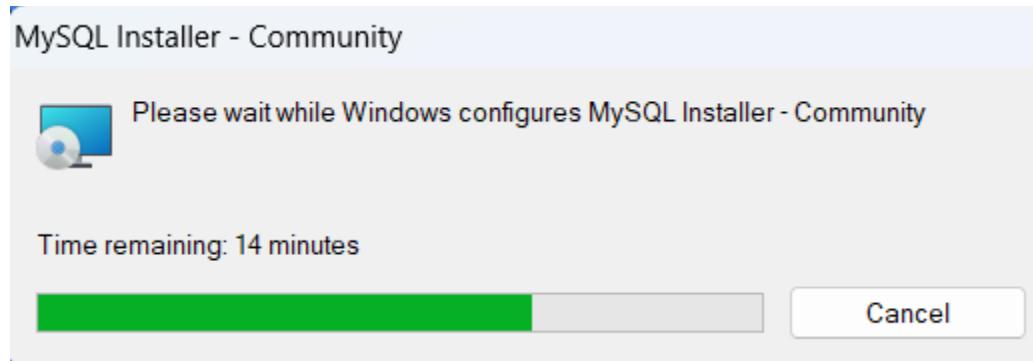
[Sign Up »](#)
for an Oracle Web account

MySQL.com is using Oracle SSO for authentication. If you already have an Oracle Web account, click the Login link. Otherwise, you can signup for a free account by clicking the Sign Up link and following the instructions.

[No thanks, just start my download.](#)

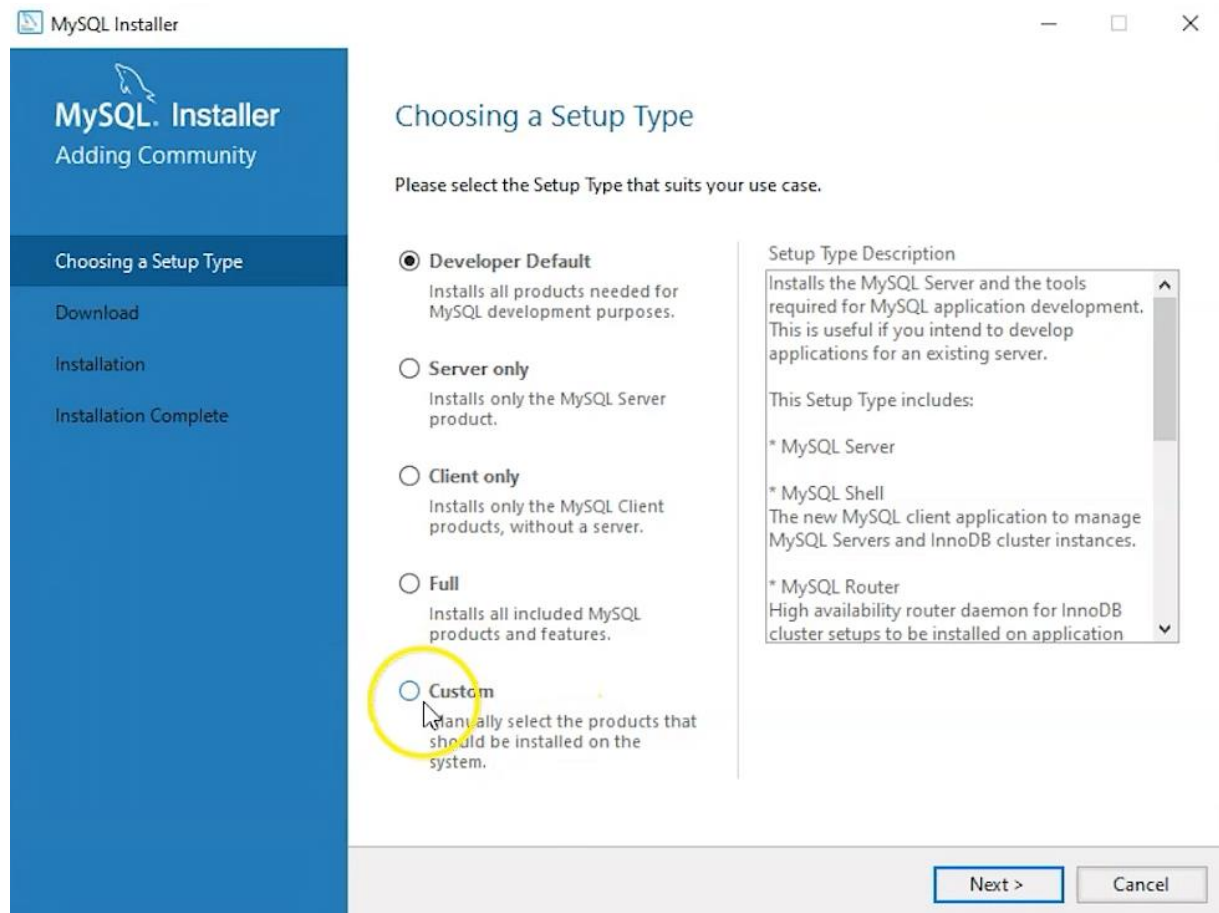
3: Run the Installer

After MySQL downloading **MySQL.exe** file , go to your Downloads folder, find the file, and double-click to run the installer.

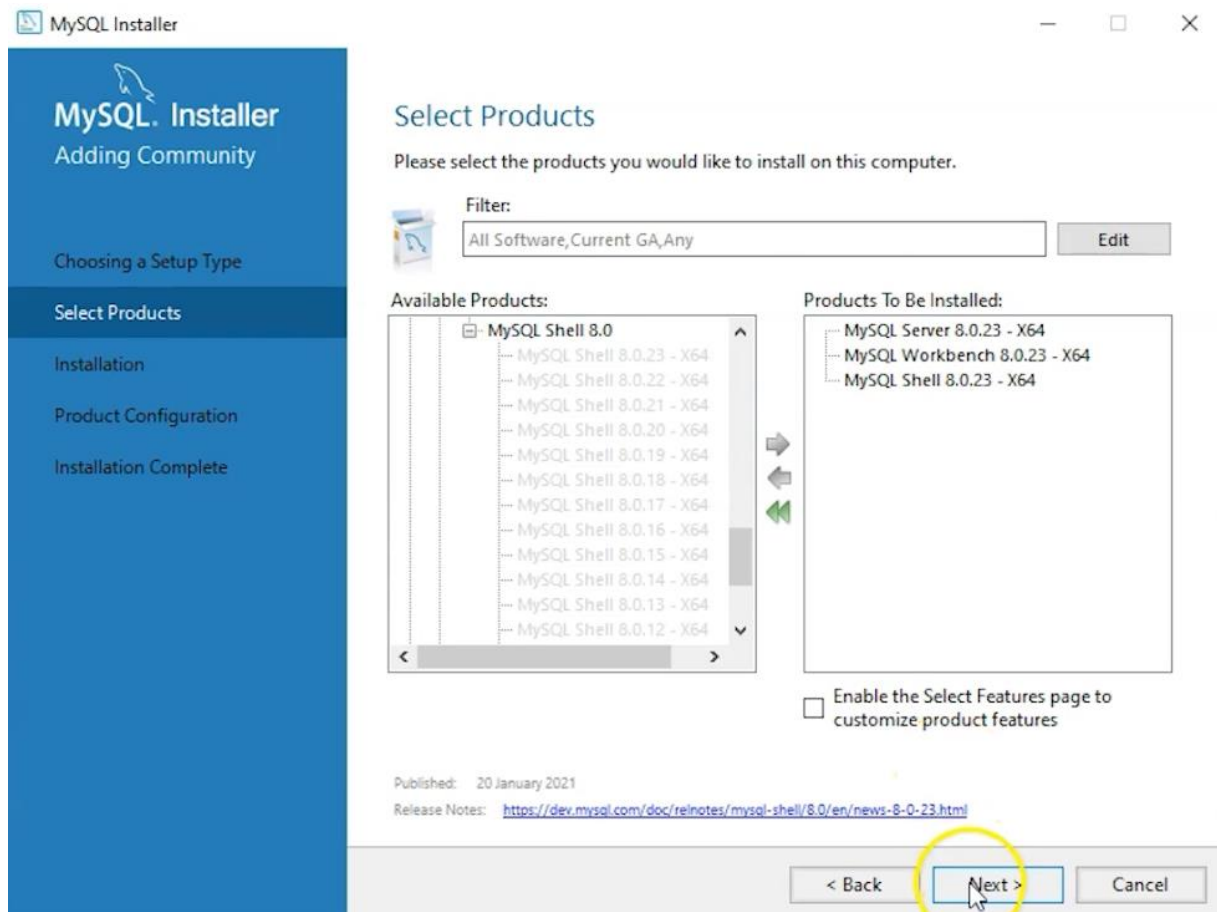


4: Choose Setup Type

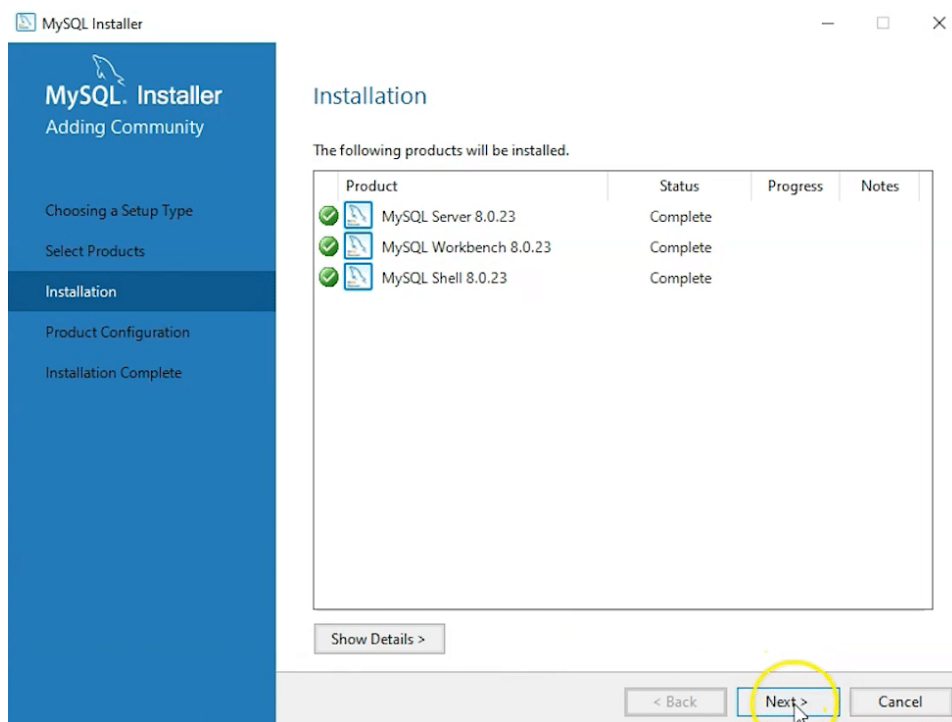
The installer will instruct you to choose the setup type. Here, select Custom. Click “Next” to proceed. With this, you will install MySQL server, MySQL Workbench, and MySQL shell.



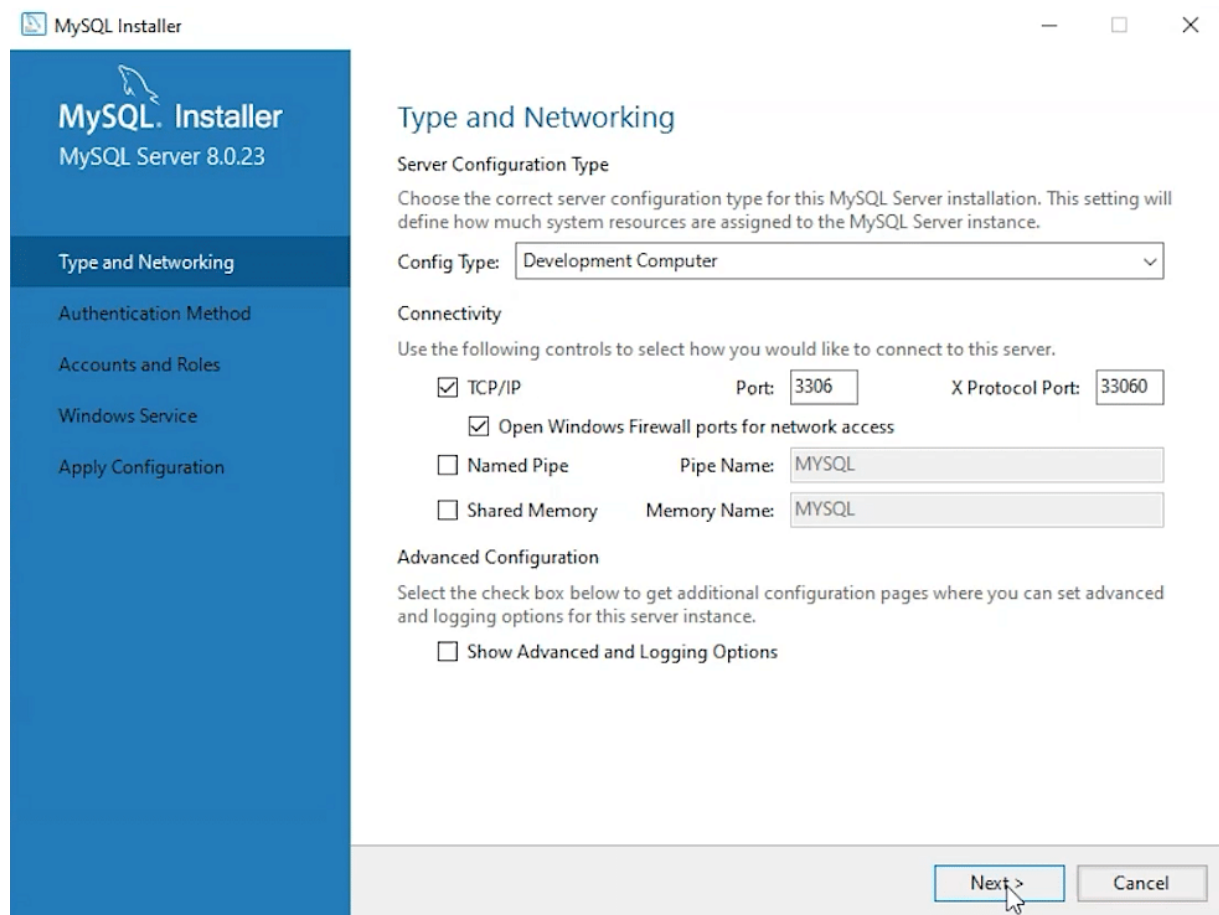
5: Open MySQL Servers, select the server you want to install, and move it to the Products/Features to be installed window section. Now, expand Applications, choose MySQL Workbench and MySQL shell. Move both to 'Products/Features to be installed'.



6. Click on the Next button. Now, click on the Execute button to download and install the MySQL server, MySQL Workbench, and the MySQL shell.



7: Once the product is ready to configure, click on Next. Under Type and Networking, go with the default settings and select Next.



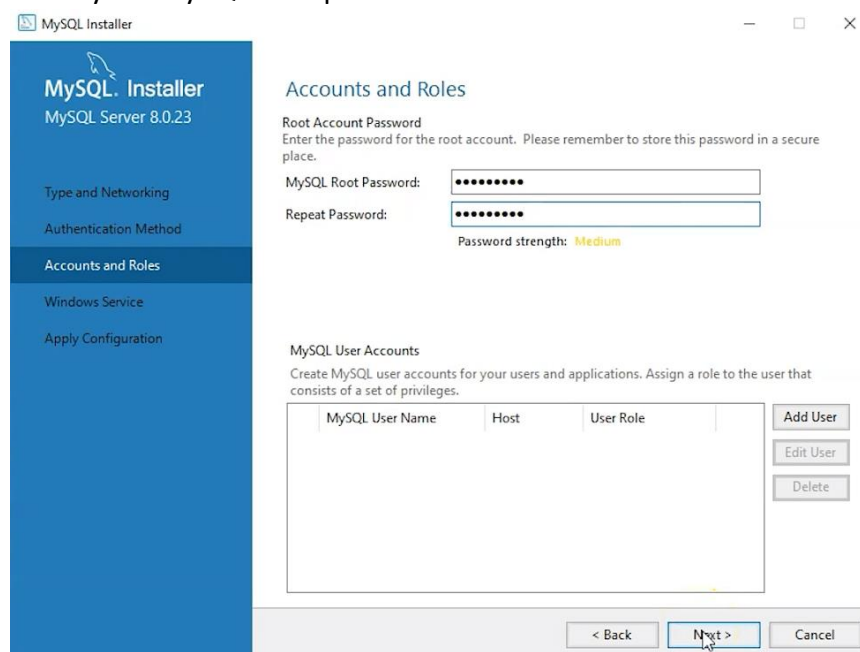
The screenshot shows the 'Type and Networking' configuration screen of the MySQL Installer for MySQL Server 8.0.23. The left sidebar contains a navigation menu with the following items: 'Type and Networking' (selected), 'Authentication Method', 'Accounts and Roles', 'Windows Service', and 'Apply Configuration'. The main content area is titled 'Type and Networking' and includes the following sections:

- Server Configuration Type:** A text box with the instruction 'Choose the correct server configuration type for this MySQL Server installation. This setting will define how much system resources are assigned to the MySQL Server instance.' Below it, a dropdown menu is set to 'Development Computer'.
- Connectivity:** A section with the instruction 'Use the following controls to select how you would like to connect to this server.' It contains three options:
 - ☒ TCP/IP: Port is 3306, X Protocol Port is 33060.
 - ☒ Open Windows Firewall ports for network access.
 - ☐ Named Pipe: Pipe Name is MYSQL.
 - ☐ Shared Memory: Memory Name is MYSQL.
- Advanced Configuration:** A section with the instruction 'Select the check box below to get additional configuration pages where you can set advanced and logging options for this server instance.' It contains one option:
 - ☐ Show Advanced and Logging Options.

At the bottom right, there are 'Next >' and 'Cancel' buttons. A mouse cursor is pointing at the 'Next >' button.

8. For authentication, use the recommended strong password encryption.

9. Set your MySQL Root password and click on next.

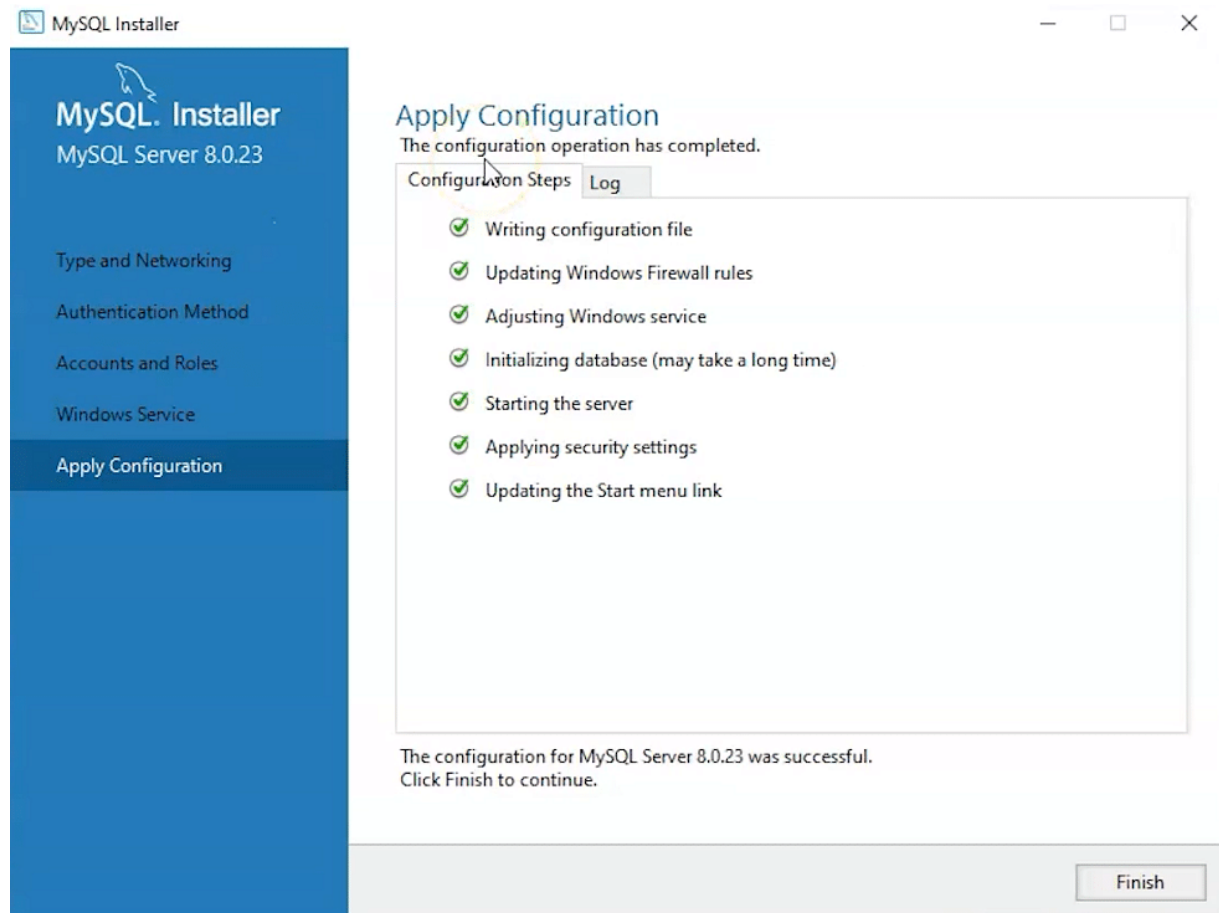


The screenshot shows the 'Accounts and Roles' configuration screen of the MySQL Installer for MySQL Server 8.0.23. The left sidebar contains a navigation menu with the following items: 'Type and Networking', 'Authentication Method', 'Accounts and Roles' (selected), 'Windows Service', and 'Apply Configuration'. The main content area is titled 'Accounts and Roles' and includes the following sections:

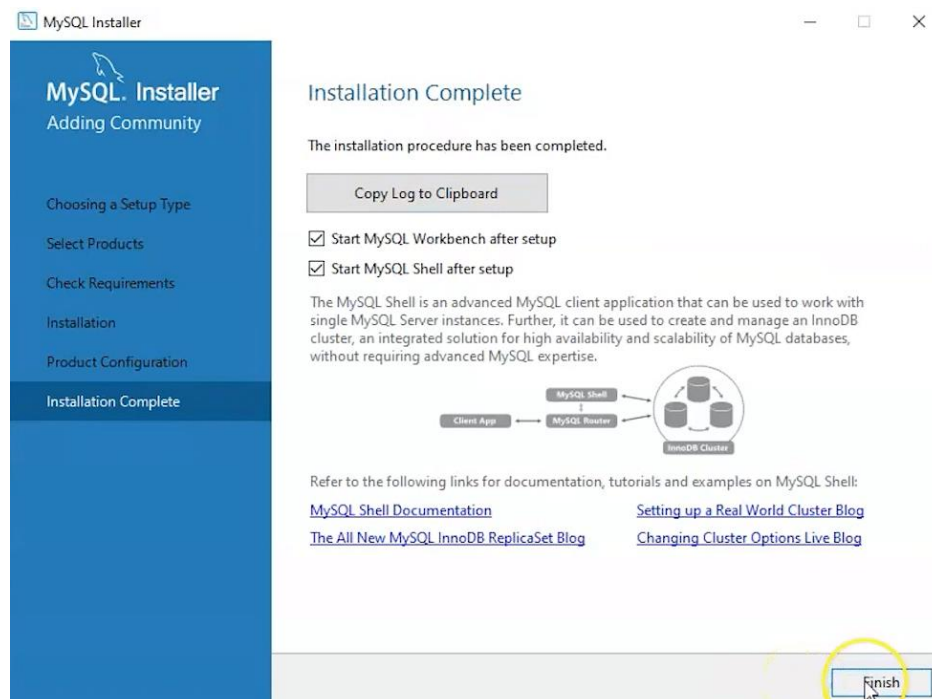
- Root Account Password:** A section with the instruction 'Enter the password for the root account. Please remember to store this password in a secure place.' It contains two text boxes: 'MySQL Root Password:' and 'Repeat Password:'. Both boxes contain a series of dots representing a masked password. Below the text boxes, it says 'Password strength: Medium'.
- MySQL User Accounts:** A section with the instruction 'Create MySQL user accounts for your users and applications. Assign a role to the user that consists of a set of privileges.' It contains a table with the following columns: 'MySQL User Name', 'Host', and 'User Role'. To the right of the table are three buttons: 'Add User', 'Edit User', and 'Delete'.

At the bottom, there are '< Back', 'Next >', and 'Cancel' buttons. A mouse cursor is pointing at the 'Next >' button.

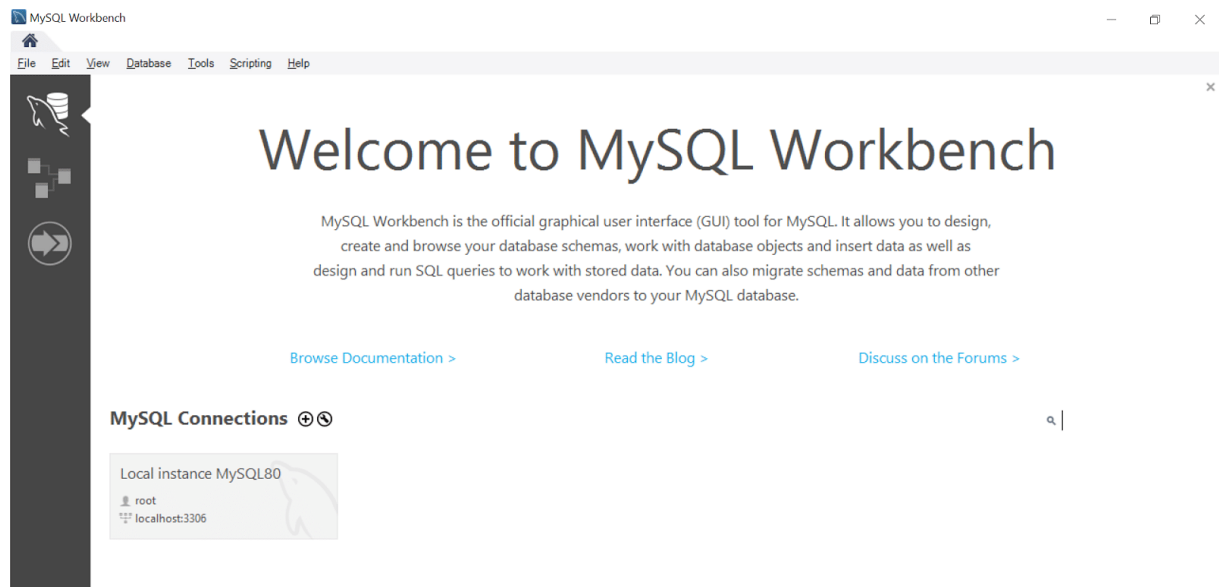
10. Go for the default windows service settings and under apply configuration, click on execute. Once the configuration is complete, click on finish.



11. Complete the installation. This will now launch the MySQL Workbench and the MySQL Shell.



12. Once MySQL Workbench is installed, select the Local instance and enter the password.



MySQL Workbench Is Ready To Use

Now, you can use the MySQL query tab to write your SQL queries.

Create Table in MySQL

A relational database consists of multiple related tables. A table is made up of rows and columns, which store structured data like customers, products, employees, etc.

To create a new table, you use the CREATE TABLE statement.

The following illustrates the basic syntax of the CREATE TABLE statement in MySQL:

```
CREATE TABLE [IF NOT EXISTS] table_name (  
    column1 datatype(length) column_constraint,  
    column2 datatype(length) column_constraint,  
    column3 datatype(length) column_constraint,  
    table_constraints  
);
```

MySQL includes several **column constraints** to ensure data integrity:

- **NOT NULL:** Ensures that a column cannot contain NULL values.
- **UNIQUE:** Ensures all the values in a column are unique across the table.
- **PRIMARY KEY:** Uniquely identifies each row in the table. A table can have only one primary key.
- **CHECK:** Ensures that values in a column satisfy a specific condition or rule (since MySQL 8.0).

- **FOREIGN KEY:** Links a column in one table to a column in another table, establishing a relationship between the two.

Note: Table constraints can be applied to more than one column and help enforce rules across multiple columns.

Try the below MYSQL statements by using MySQL Workbench or MySQL Online Editor

Launch MySQL Workbench on your computer. and click on the "Local instance MySQL"
Once connected, click the **SQL +** button to open a new SQL query window.

If you do not have access to MySQL Workbench, you can use an online MySQL editor (<https://onecompiler.com/mysql>)

Those who are using online editor can skip the process of creating database and goto step3

Step 1: Create a New Database

1. Once connected, click on the **SQL Editor** (or the plus icon next to it) to open a new query tab.
2. To create a new database, run the following SQL command:

```
CREATE DATABASE school;
```

3. After executing the command, you should see a message saying, "**Query OK**". The database school is now created.

Step 2: Select the Database Now that the database is created, you need to select it to make it active. Use the following command:

```
USE school;
```

Step 3: Creating Tables

Table 1: Creating a student Table

Step 1: Create a Table To create a table for storing student records, use the following SQL code:

```
CREATE TABLE student (
  student_id INT AUTO_INCREMENT PRIMARY KEY,
  first_name VARCHAR(50) NOT NULL,
  last_name VARCHAR(50) NOT NULL,
  date_of_birth DATE,
  is_enrolled BOOLEAN DEFAULT TRUE
);
```

This creates a table with:

- **student_id:** A unique identifier for each student, auto-incremented.
- **first_name:** The student's first name.

- last_name: The student's last name.
- date_of_birth: The student's date of birth.
- is_enrolled: Whether the student is currently enrolled.

Step 2: Verify Table Creation To check if the table has been created successfully, use the following command:

```
SHOW TABLES;
```

You should see the student table in the list of tables.

Table 2: Creating a course Table

Step 1: Create the course Table Now let's create a table for storing course records:

```
CREATE TABLE course (
  course_id INT AUTO_INCREMENT PRIMARY KEY,
  course_name VARCHAR(100) NOT NULL,
  course_code VARCHAR(10) UNIQUE NOT NULL,
  credits INT CHECK (credits >= 0),
  is_active BOOLEAN DEFAULT TRUE
);
```

This creates a table with:

- course_id: A unique identifier for each course.
- course_name: The name of the course.
- course_code: A unique code for each course.
- credits: The number of credits for the course, restricted to non-negative values.
- is_active: Whether the course is currently active.

Step 2: Verify Table Structure To check the structure of the course table, use the DESCRIBE command:

```
DESCRIBE course;
```

You will see details about the columns and data types used in the table.

Creating Tables (Extra Exercises)

Table 3: Creating an Instructor Table

Step 1: Create the Instructor Table

To create a table for storing instructor records, use the following SQL code:

```
CREATE TABLE instructor (
  instructor_id INT AUTO_INCREMENT PRIMARY KEY,
  first_name VARCHAR(50) NOT NULL,
```

```
last_name VARCHAR(50) NOT NULL,  
hire_date DATE,  
department VARCHAR(100),  
is_active BOOLEAN DEFAULT TRUE  
);
```

This creates a table with:

- **instructor_id**: A unique identifier for each instructor, auto-incremented.
- **first_name**: The instructor's first name.
- **last_name**: The instructor's last name.
- **hire_date**: The date the instructor was hired.
- **department**: The department in which the instructor teaches.
- **is_active**: Whether the instructor is currently active.

Step 2: Verify Table Creation

To check if the instructor table has been created successfully, use the following command:

```
SHOW TABLES;
```

You should see the instructor table in the list of tables.

Table 4: Creating a Enrolment Table

Step 1: Create the Enrolment Table

Now let's create a table for storing enrolment records:

```
CREATE TABLE enrolment (  
  enrolment_id INT AUTO_INCREMENT PRIMARY KEY,  
  student_id INT,  
  course_id INT,  
  enrolment_date DATE,  
  FOREIGN KEY (student_id) REFERENCES student(student_id),  
  FOREIGN KEY (course_id) REFERENCES course(course_id)  
);
```

This creates a table with:

- **enrolment_id**: A unique identifier for each enrolment record, auto-incremented.
- **student_id**: A reference to the student enrolled (linked to the student table).
- **course_id**: A reference to the course (linked to the course table).
- **enrolment_date**: The date the student enrolled in the course.

Step 2: Verify Table Structure

To check the structure of the enrolment table, use the DESCRIBE command:

```
DESCRIBE enrollment;
```

You will see details about the columns and data types used in the table.

Table 5: Creating a Grades Table

Step 1: Create the Grades Table

Next, let's create a table for storing grades:

```
CREATE TABLE grades (  
  grade_id INT AUTO_INCREMENT PRIMARY KEY,  
  enrolment_id INT,  
  grade DECIMAL(3, 2) CHECK (grade >= 0 AND grade <= 4.00),  
  FOREIGN KEY (enrolment_id) REFERENCES enrolment(enrolment_id)  
);
```

This creates a table with:

- **grade_id**: A unique identifier for each grade record, auto-incremented.
- **enrolment_id**: A reference to the enrolment record (linked to the enrolment table).
- **grade**: The grade awarded, restricted to a decimal value between 0 and 4.00.

Step 2: Verify Table Creation

To check if the grades table has been created successfully, use the following command:

```
SHOW TABLES;
```

You should see the grades table in the list of tables.

employee_id: An auto-incrementing integer that serves as the primary key.

first_name and last_name: Strings that store the employee's first and last names, with the constraint that they cannot be NULL.

hire_date: A date field to store the hire date.
