

Exercise: Databases

Problems for exercises and homework for the ["Software Technologies" course @ Software University](#).

1. Follow a SQL Tutorial

Practice and learn SQL skills, following a simple SQL tutorial:

1. Open the following [Tutorial](#) in W3Shools:

- [Create](#) your own profile in W3Shools and Login to track your progress.
- Take the first 10 topics from the Tutorial.
- Read the theory paragraph in every topic.
- Example – [Intro](#) Topic:

SQL Tutorial
SQL HOME
SQL Intro
SQL Syntax
SQL Select
SQL Select Distinct
SQL Where
SQL And, Or, Not
SQL Order By
SQL Insert Into
SQL Null Values
SQL Update
SQL Delete
SQL Select Top
SQL Min and Max
SQL Count, Avg, Sum
SQL Like
SQL Wildcards

Introduction to SQL

< Previous

Next >

SQL is a standard language for accessing and manipulating databases.

What is SQL?

- SQL stands for Structured Query Language
- 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

SQL Tutorial
SQL HOME
SQL Intro
SQL Syntax
SQL Select
SQL Select Distinct
SQL Where
SQL And, Or, Not
SQL Order By
SQL Insert Into
SQL Null Values
SQL Update
SQL Delete
SQL Select Top
SQL Min and Max
SQL Count, Avg, Sum
SQL Like
SQL Wildcards
SQL In

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

SQL is a Standard - BUT....

Although SQL is an ANSI/ISO standard, there are different versions of the SQL language.

However, to be compliant with the ANSI standard, they all support at least the major commands (such as `SELECT`, `UPDATE`, `DELETE`, `INSERT`, `WHERE`) in a similar manner.

SQL Tutorial
SQL HOME
SQL Intro
SQL Syntax
SQL Select
SQL Select Distinct
SQL Where
SQL And, Or, Not
SQL Order By
SQL Insert Into
SQL Null Values

Using SQL in Your Web Site

To build a web site that shows data from a database, you will need:

- An RDBMS database program (i.e. MS Access, SQL Server, MySQL)
- To use a server-side scripting language, like PHP or ASP
- To use SQL to get the data you want
- To use HTML / CSS to style the page

SQL Tutorial
SQL HOME
SQL Intro
SQL Syntax
SQL Select
SQL Select Distinct
SQL Where
SQL And, Or, Not
SQL Order By
SQL Insert Into
SQL Null Values
SQL Update
SQL Delete
SQL Select Top
SQL Min and Max
SQL Count, Avg, Sum
SQL Like
SQL Wildcards
SQL In

RDBMS

RDBMS stands for Relational Database Management System.

RDBMS is the basis for SQL, and for all modern database systems such as MS SQL Server, IBM DB2, Oracle, MySQL, and Microsoft Access.

The data in RDBMS is stored in database objects called tables. A table is a collection of related data entries and it consists of columns and rows.

Look at the "Customers" table:

Example

Get your own SQL Server

```
SELECT * FROM Customers;
```

Try It Yourself >

SQL Tutorial
SQL HOME
SQL Intro
SQL Syntax
SQL Select
SQL Select Distinct
SQL Where
SQL And, Or, Not
SQL Order By
SQL Insert Into
SQL Null Values
SQL Update
SQL Delete
SQL Select Top

Every table is broken up into smaller entities called fields. The fields in the Customers table consist of CustomerID, CustomerName, ContactName, Address, City, PostalCode and Country. A field is a column in a table that is designed to maintain specific information about every record in the table.

A record, also called a row, is each individual entry that exists in a table. For example, there are 91 records in the above Customers table. A record is a horizontal entity in a table.

A column is a vertical entity in a table that contains all information associated with a specific field in a table.

< Previous

Log in to track progress

Next >

2. In every topic you can try to write SQL by yourself:

SQL Statement:

[Get your own SQL server](#)

```
SELECT * FROM Customers;
```

Edit the SQL Statement, and click "Run SQL" to see the result.

[Run SQL »](#)

Result:

Number of Records: 91

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico

Your Database:

Tablename	Records
Customers	91
Categories	8
Employees	10
OrderDetails	518
Orders	196
Products	77
Shippers	3
Suppliers	29

[Restore Database](#)

- On the right side of the screen, you will be able to see all the tables that are created for you for the example:

SQL Statement:

[Get your own SQL server](#)

```
SELECT * FROM Categories;
```

Edit the SQL Statement, and click "Run SQL" to see the result.

[Run SQL »](#)

Your Database:

Tablename	Records
Customers	91
Categories	8
Employees	10
OrderDetails	518
Orders	196
Products	77
Shippers	3
Suppliers	29

[Restore Database](#)

- Click on "Run SQL" and see the selected table appeared in the "Result" box:

[Run SQL »](#)

Result:

Number of Records: 8

CategoryID	CategoryName	Description
1	Beverages	Soft drinks, coffees, teas, beers, and ales
2	Condiments	Sweet and savory sauces, relishes, spreads, and seasonings
3	Confections	Desserts, candies, and sweet breads
4	Dairy Products	Cheeses
5	Grains/Cereals	Breads, crackers, pasta, and cereal
6	Meat/Poultry	Prepared meats
7	Produce	Dried fruit and bean curd
8	Seafood	Seaweed and fish

- You can try to **SELECT** another table from the Database:

SQL Statement:

```
SELECT * FROM Employees;
```

Result:

Number of Records: 10

EmployeeID	LastName	FirstName	BirthDate	Photo	Notes
1	Davolio	Nancy	1968-12-08	EmpID1.pic	Education includes a BA in psychology from Colorado State University. She also completed (The Art of the Cold Call). Nancy is a member of 'Toastmasters International'.
2	Fuller	Andrew	1952-02-19	EmpID2.pic	Andrew received his BTS commercial and a Ph.D. in international marketing from the University of Dallas. He is fluent in French and Italian and reads German. He joined the company as a sales representative, was promoted to sales manager and was then named vice president of sales. Andrew is a member of the Sales Management Roundtable, the Seattle Chamber of Commerce, and the Pacific Rim Importers Association.
3	Leverling	Janet	1963-08-	EmpID3.pic	Janet has a BS degree in chemistry from Boston College). She

- Try to **SELECT** the **OrderDetails** table:

SQL Statement:

Get your own S

```
SELECT * FROM OrderDetails;
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

Result:

Number of Records: 518

OrderDetailID	OrderID	ProductID	Quantity
1	10248	11	12
2	10248	42	10
3	10248	72	5

- Now try to **SELECT** only those entries, that refer to a product with ID 11:

SQL Statement:

```
SELECT * FROM OrderDetails WHERE ProductID = 11;
```

Result:

Number of Records: 9

OrderDetailID	OrderID	ProductID	Quantity
1	10248	11	12
130	10296	11	12
211	10327	11	50
281	10353	11	12
314	10365	11	24
426	10407	11	30
492	10434	11	6
514	10442	11	30

- Now try to **SELECT** only those entries, that have **ProductID = 11** and **Quantity not less than 12**:

SQL Statement:

```
SELECT * FROM OrderDetails WHERE ProductID = 11 AND Quantity >= 12;
```

Result:

Number of Records: 7

OrderDetailID	OrderID	ProductID	Quantity
1	10248	11	12
130	10296	11	12
211	10327	11	50
281	10353	11	12
314	10365	11	24
426	10407	11	30
514	10442	11	30

3. In the end, test what you have learned by passing through the exercise, from the [HOME topic](#):

SQL Tutorial

SQL HOME

SQL Intro

SQL Syntax

SQL Select

SQL Select Distinct

SQL Where

SQL And, Or, Not

SQL Order By

SQL Insert Into

SQL Null Values

SQL Update

SQL Delete

SQL Select Top

SQL Min and Max

SQL Count, Avg, Sum

SQL Like

SQL Wildcards

SQL In

SQL Exercises

Test Yourself With Exercises

Exercise:

Insert the missing statement to get all the columns from the **Customers** table.

* FROM Customers;

Submit Answer »

[Start the Exercise](#)

4. You can try the SQL Quiz Test also:

SQL Tutorial

SQL HOME

SQL Intro

SQL Syntax

SQL Select

SQL Select Distinct

SQL Where

SQL And, Or, Not

SQL Quiz Test

Test your SQL skills at W3Schools!

Start SQL Quiz!

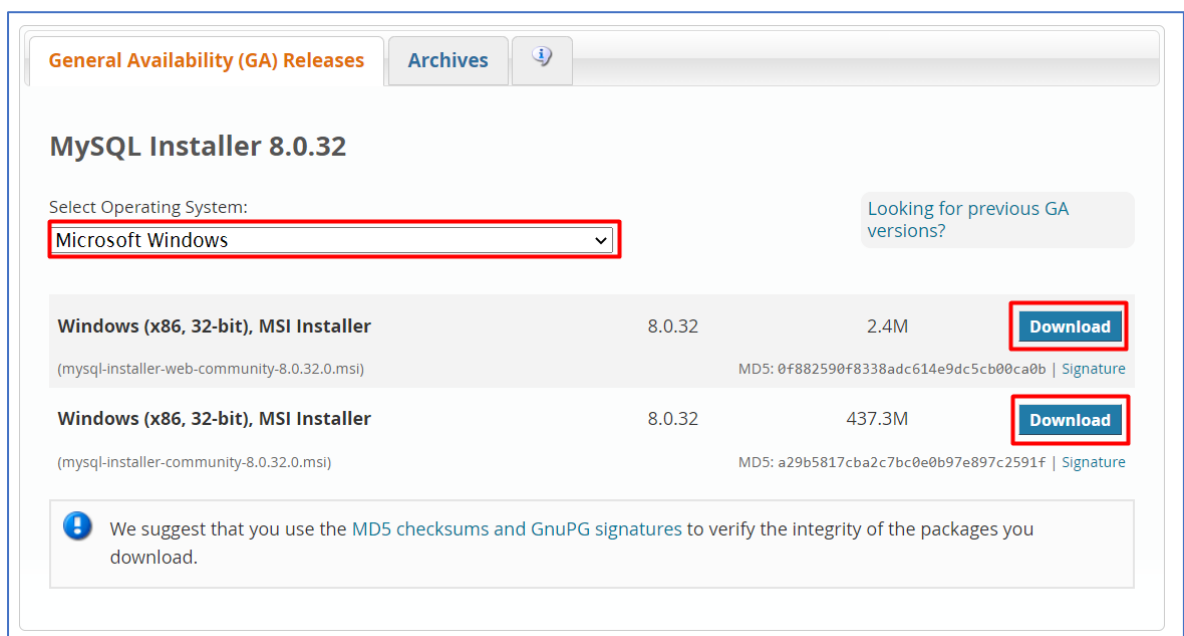
2. MySQL and Workbench Installation

1. MySQL Workbench:

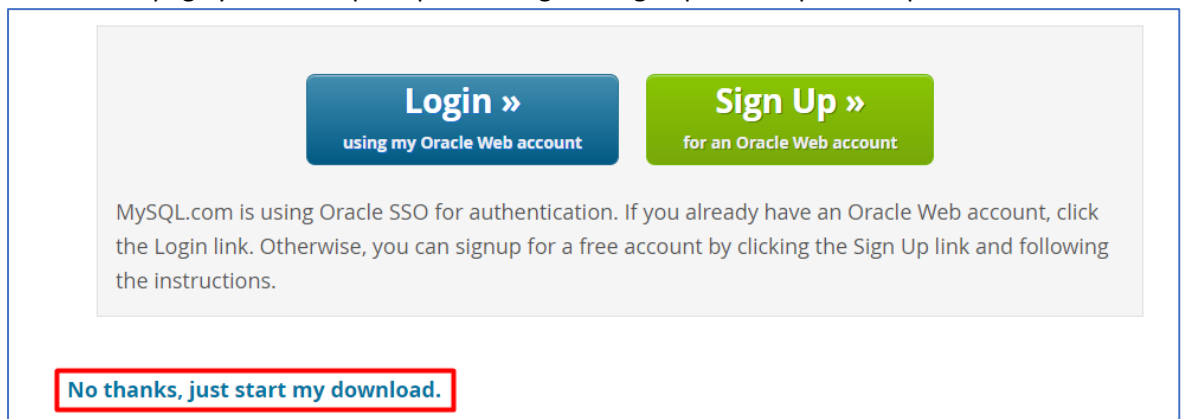
- A visual tool for database design, management, and administration, while **MySQL Server is the actual database management system** that stores, manages, and retrieves data. QA engineers need these tools to create, manipulate, and verify databases during testing, ensuring that the application's data-related functionalities work correctly and meet requirements.

2. Download MySQL Installer:

- Navigate to the official MySQL download site at <https://dev.mysql.com/downloads/installer/>
- Choose your operating system and click one of the two download buttons. It doesn't if you choose mysql-installer-web-community or mysql-installer-community. They are quite the same. The first one downloads the files needed while installing them, the other pre downloads the files needed and then installs them.

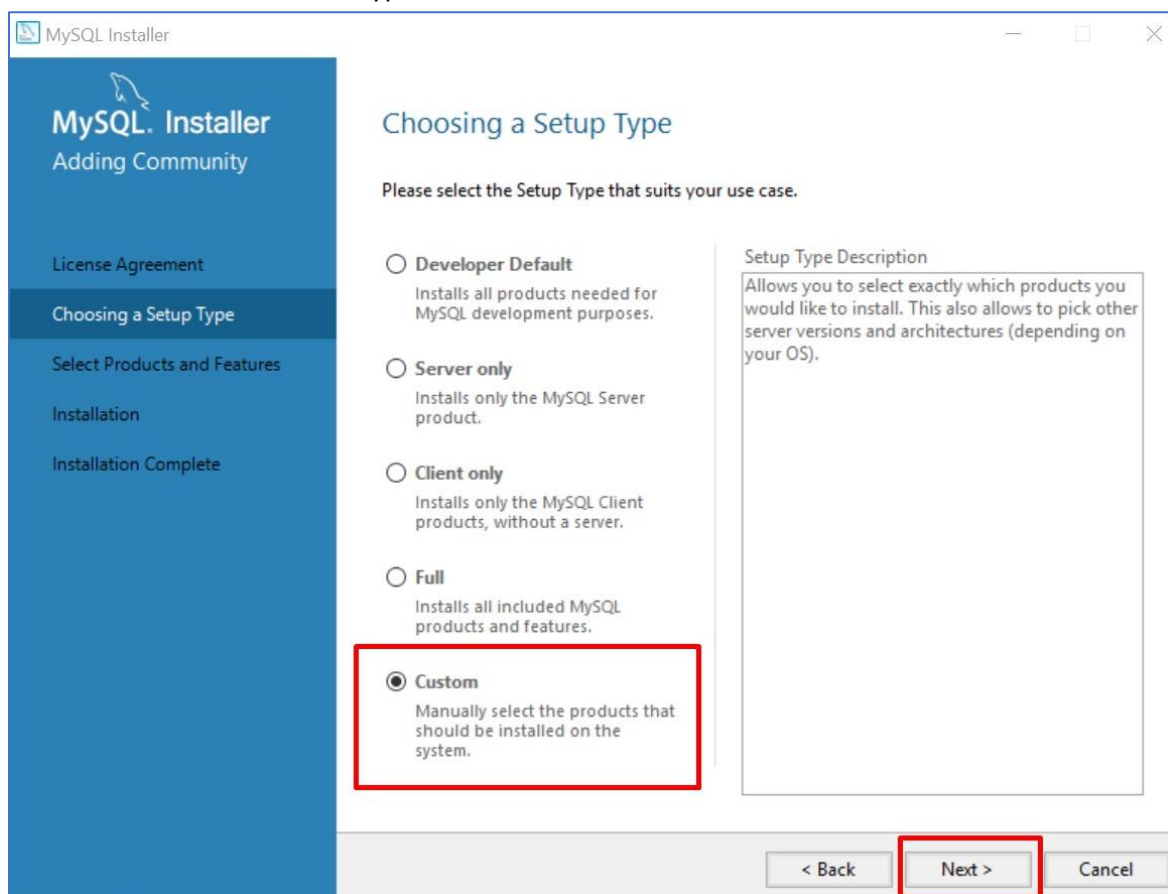


- On the next page you will be prompted to Login or Sign up. Just skip this step.

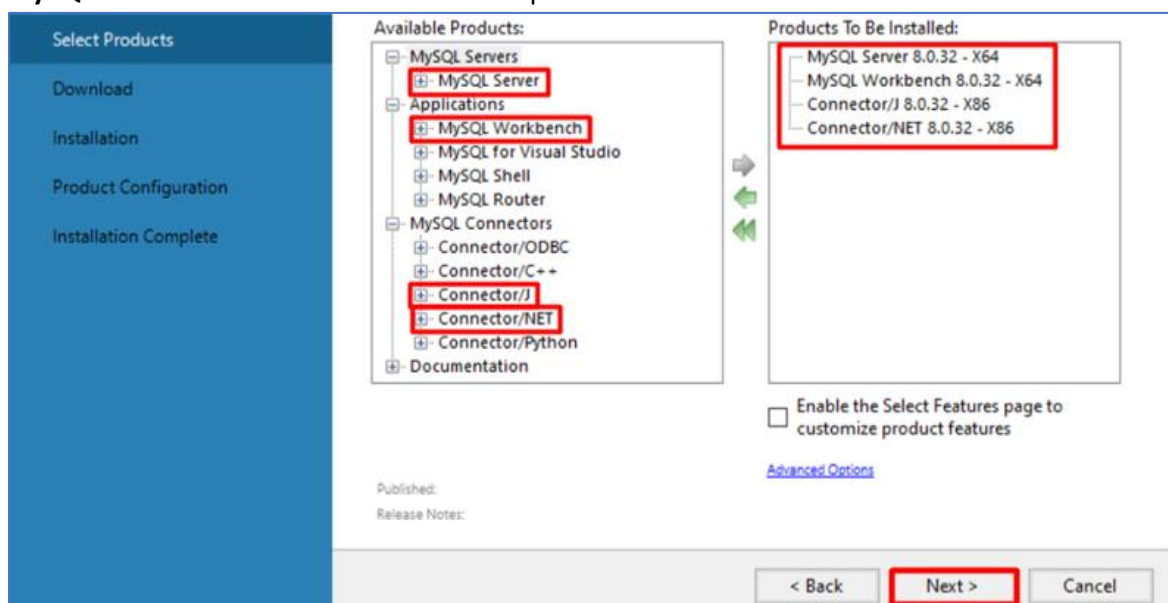


3. Run MySQL Installer:

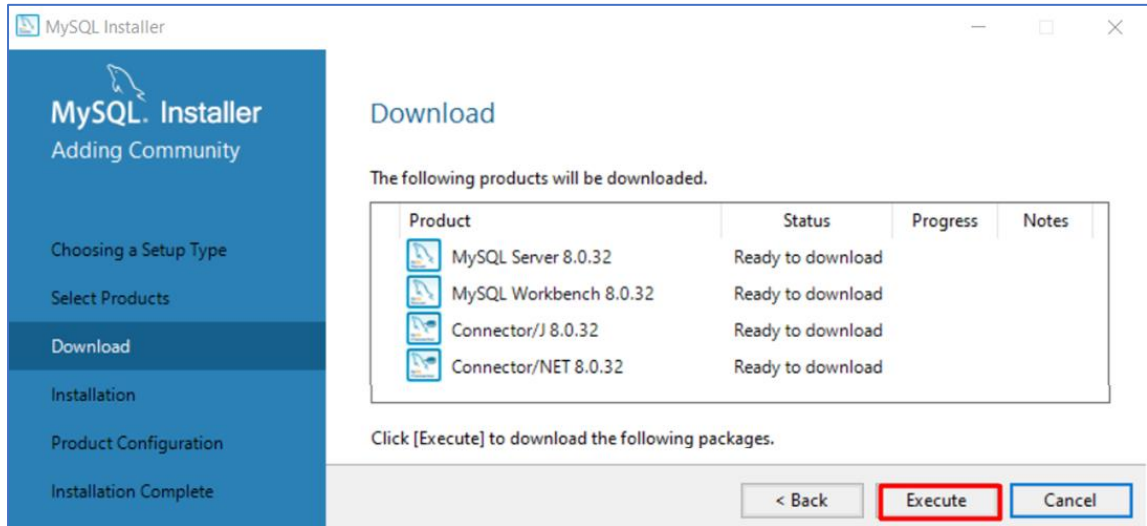
- Once the installer is downloaded, run the executable file.
- Choose "Custom" installation type. Then click next.



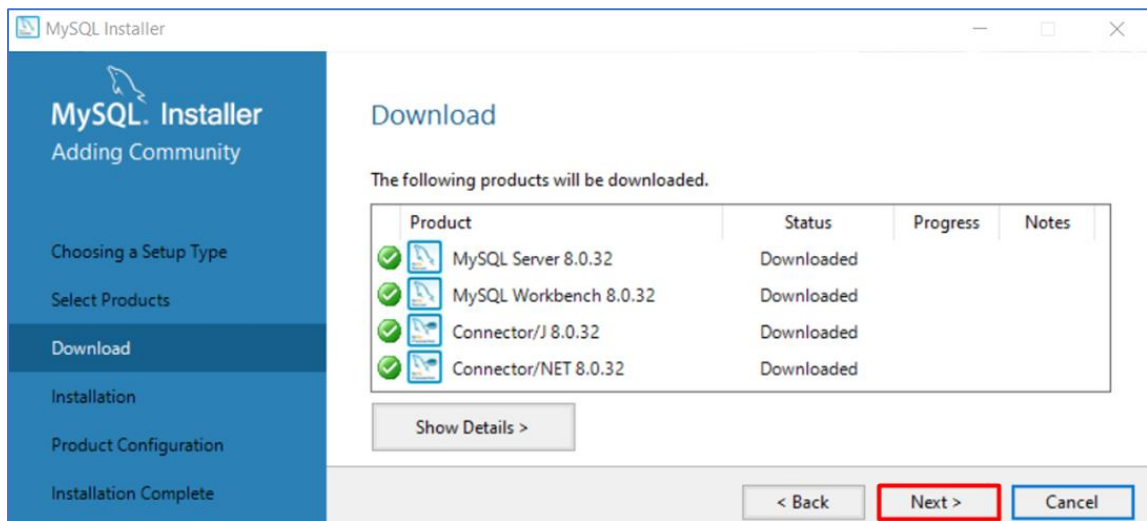
- All the features we need are **MySQL Server**, **MySQL Workbench**, **Connector/J**, **Connector/NET** and **MySQL Workbench**. All other features are optional and won't be needed for now.



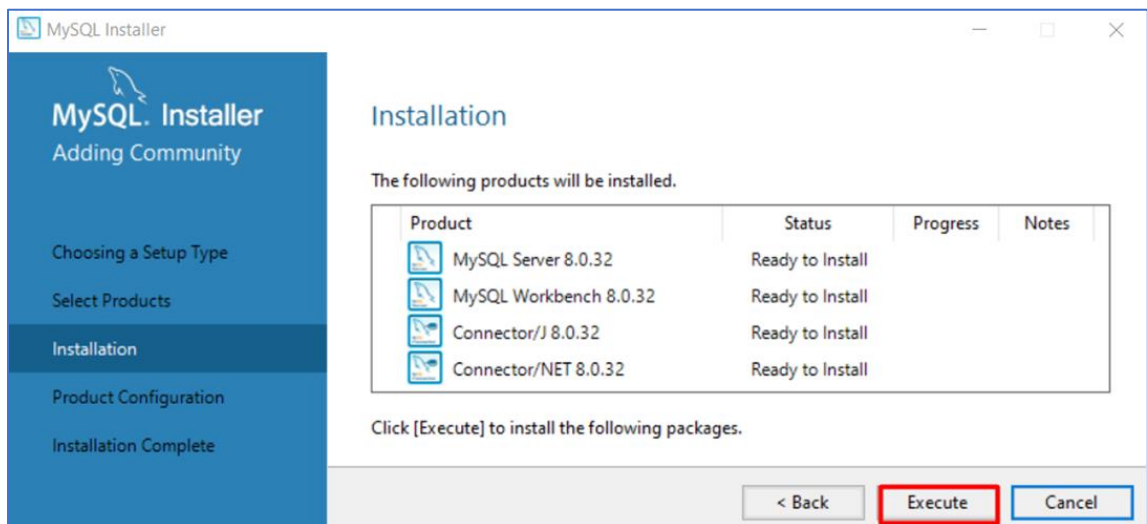
- Click **Execute** and the setup will download the selected features.



- Wait for the files to download and click **Next**:

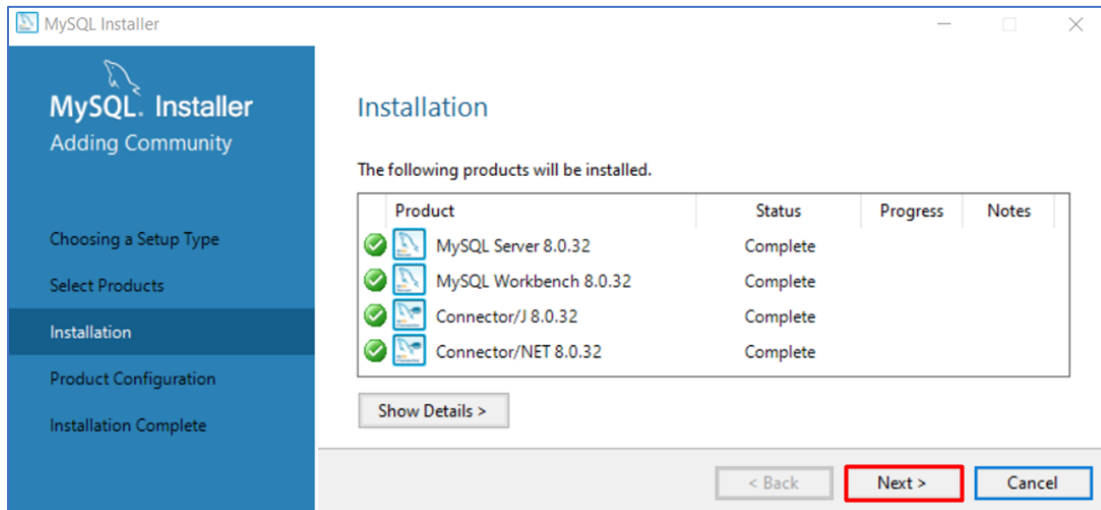


- Click **Execute** to install the files:



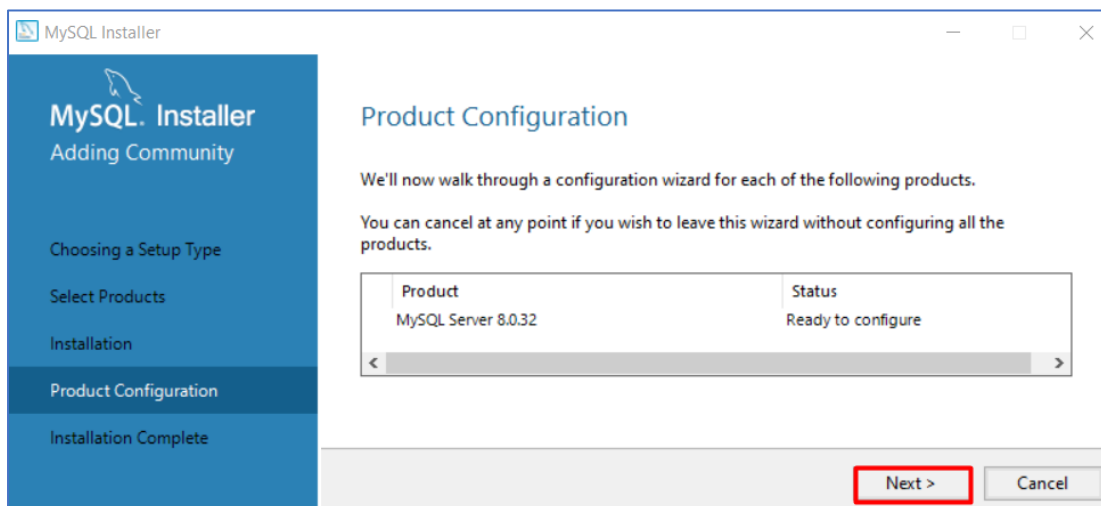
Connector/J and Connector/NET (client libraries for Java and .NET) are optional. We shall not need them.

- Wait for the installation to complete and click **Next** to start configuration wizard:

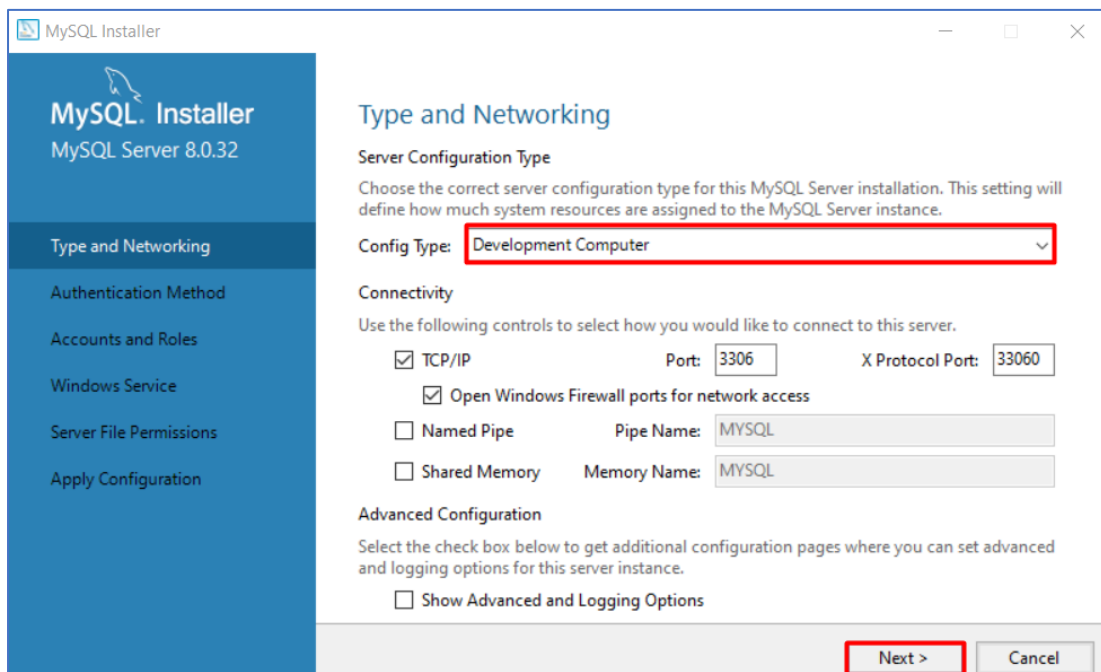


4. Now, you have to configure your product.

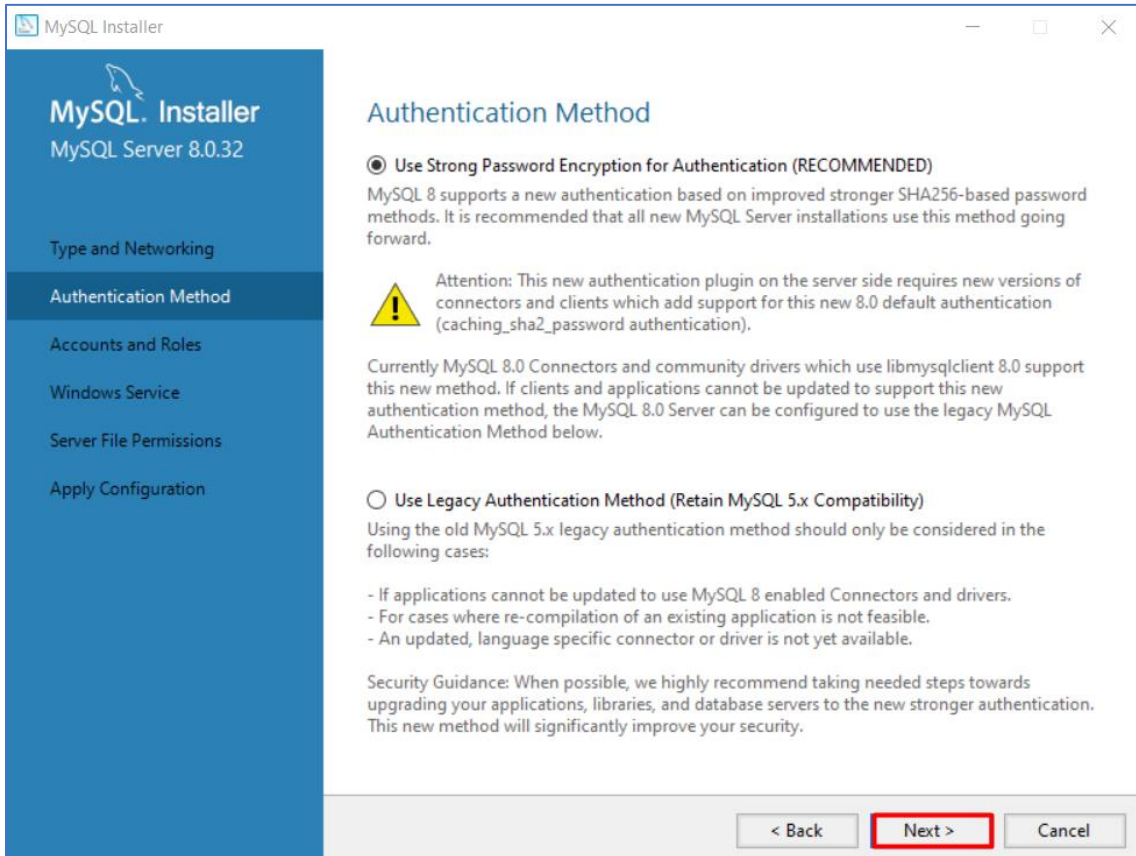
- Click Next:



- Choose the configuration type to be a **Development Computer** and click on Next:



- **Choose Authentication method** and click **Next**:




The screenshot shows the 'Authentication Method' screen of the MySQL Installer for MySQL Server 8.0.32. The left sidebar contains a list of steps: Type and Networking, Authentication Method (highlighted), Accounts and Roles, Windows Service, Server File Permissions, and Apply Configuration. The main area is titled 'Authentication Method' and features two radio button options. The first option, 'Use Strong Password Encryption for Authentication (RECOMMENDED)', is selected. It includes a warning icon and text explaining that MySQL 8 supports a new authentication method based on improved stronger SHA256-based password methods, and that it is recommended for all new installations. A second option, 'Use Legacy Authentication Method (Retain MySQL 5.x Compatibility)', is also present, with text explaining its use cases and a security guidance note. At the bottom right, there are three buttons: '< Back', 'Next >' (highlighted with a red box), and 'Cancel'.

MySQL Installer
MySQL Server 8.0.32

Type and Networking
Authentication Method
Accounts and Roles
Windows Service
Server File Permissions
Apply Configuration

Authentication Method

☒ **Use Strong Password Encryption for Authentication (RECOMMENDED)**
MySQL 8 supports a new authentication based on improved stronger SHA256-based password methods. It is recommended that all new MySQL Server installations use this method going forward.

 Attention: This new authentication plugin on the server side requires new versions of connectors and clients which add support for this new 8.0 default authentication (caching_sha2_password authentication).

Currently MySQL 8.0 Connectors and community drivers which use libmysqlclient 8.0 support this new method. If clients and applications cannot be updated to support this new authentication method, the MySQL 8.0 Server can be configured to use the legacy MySQL Authentication Method below.

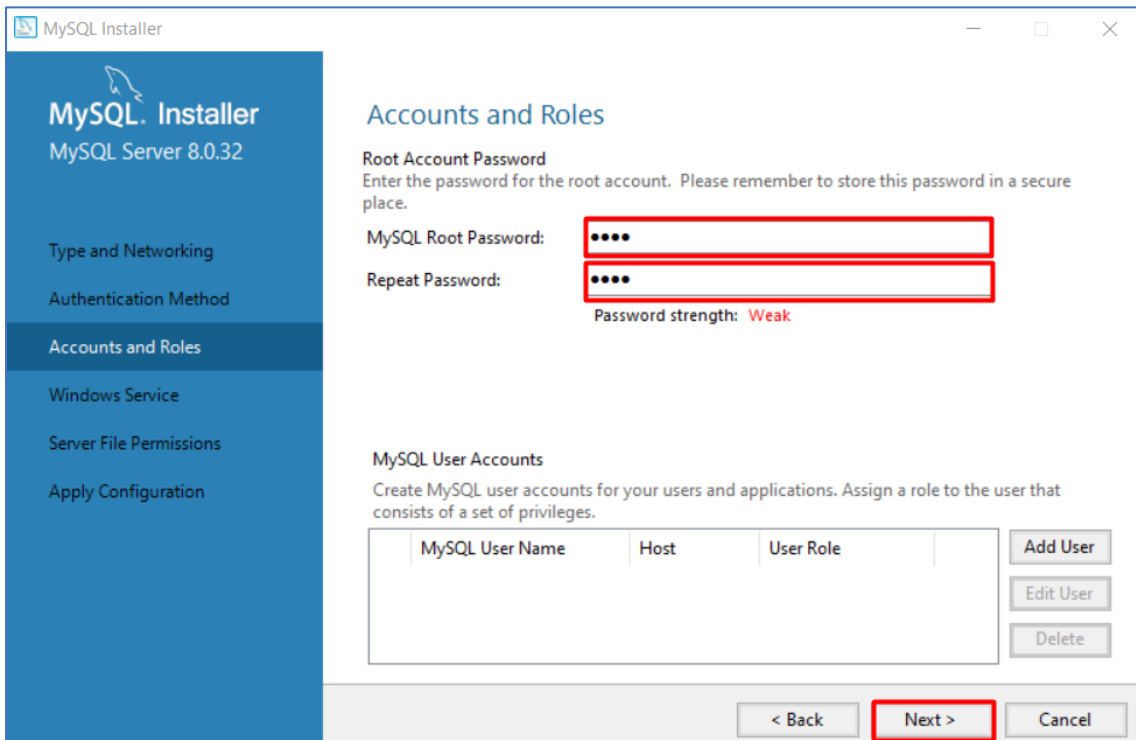
☐ **Use Legacy Authentication Method (Retain MySQL 5.x Compatibility)**
Using the old MySQL 5.x legacy authentication method should only be considered in the following cases:

- If applications cannot be updated to use MySQL 8 enabled Connectors and drivers.
- For cases where re-compilation of an existing application is not feasible.
- An updated, language specific connector or driver is not yet available.

Security Guidance: When possible, we highly recommend taking needed steps towards upgrading your applications, libraries, and database servers to the new stronger authentication. This new method will significantly improve your security.

< Back **Next >** Cancel

- **Set password to the Root account** and click **Next**:



The screenshot shows the 'Accounts and Roles' screen of the MySQL Installer for MySQL Server 8.0.32. The left sidebar is the same as the previous screen, with 'Accounts and Roles' highlighted. The main area is titled 'Accounts and Roles' and contains a section for 'Root Account Password'. It prompts the user to enter a password for the root account and provides two input fields: 'MySQL Root Password' and 'Repeat Password', both of which are highlighted with red boxes. Below these fields, the password strength is indicated as 'Weak'. There is also a section for 'MySQL User Accounts' with a table to manage users and buttons for 'Add User', 'Edit User', and 'Delete'. At the bottom right, there are three buttons: '< Back', 'Next >' (highlighted with a red box), and 'Cancel'.

MySQL Installer
MySQL Server 8.0.32

Type and Networking
Authentication Method
Accounts and Roles
Windows Service
Server File Permissions
Apply Configuration

Accounts and Roles

Root Account Password
Enter the password for the root account. Please remember to store this password in a secure place.

MySQL Root Password:

Repeat Password:

Password strength: **Weak**

MySQL User Accounts
Create MySQL user accounts for your users and applications. Assign a role to the user that consists of a set of privileges.

MySQL User Name	Host	User Role
-----------------	------	-----------

Add User
Edit User
Delete

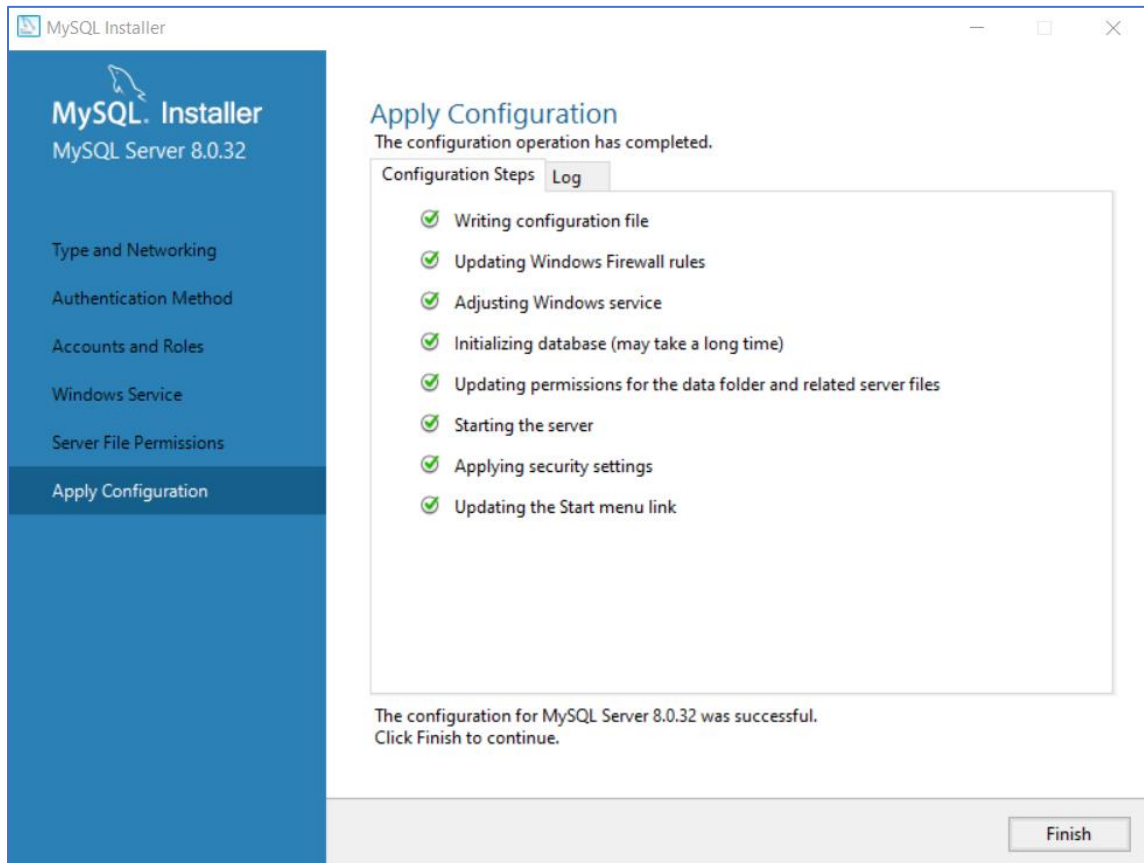
< Back **Next >** Cancel

- Here you can set the **MySQL Server** to run as **Windows Service** and to **start automatically at Windows start up**. Otherwise, **you must start up MySQL every time before working with a database**. Decide whichever suits you best.

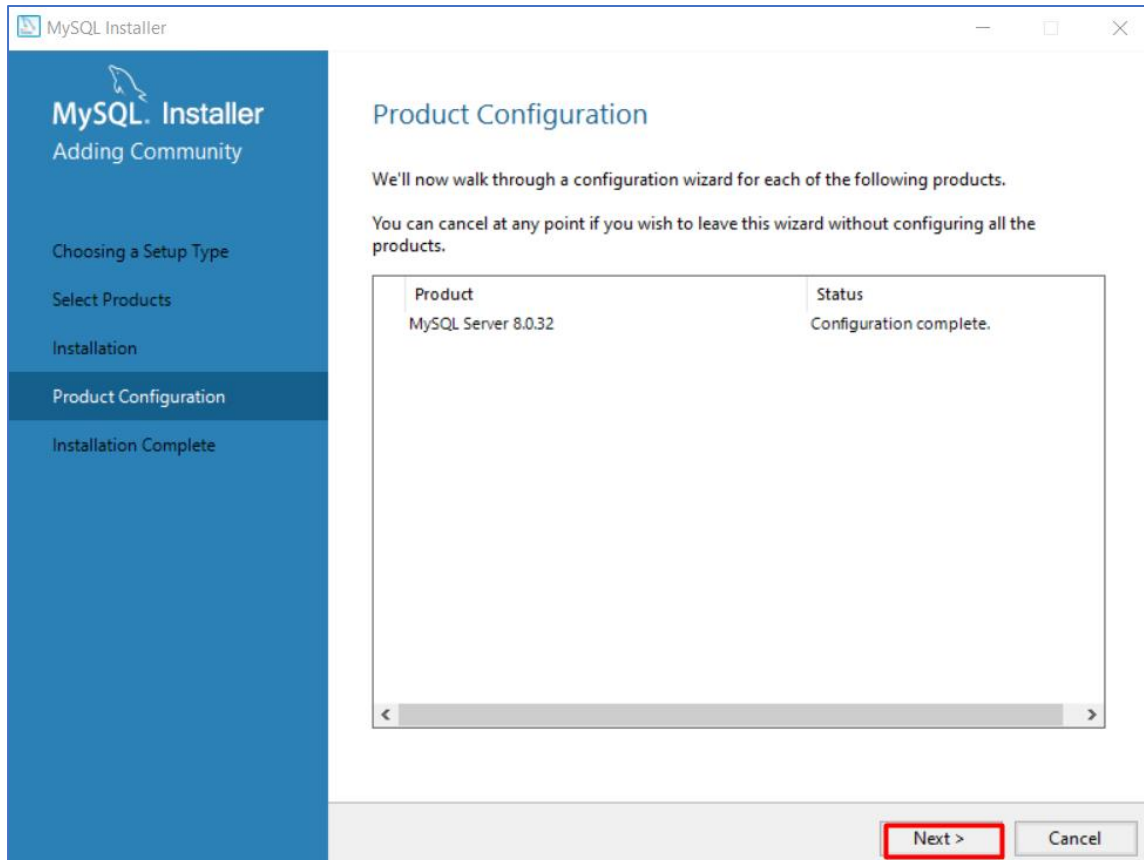
- Server File Permission** – Leave the chosen option as it as it is. Next:

- Hit Execute** and wait for the **Configuration to be applied**:

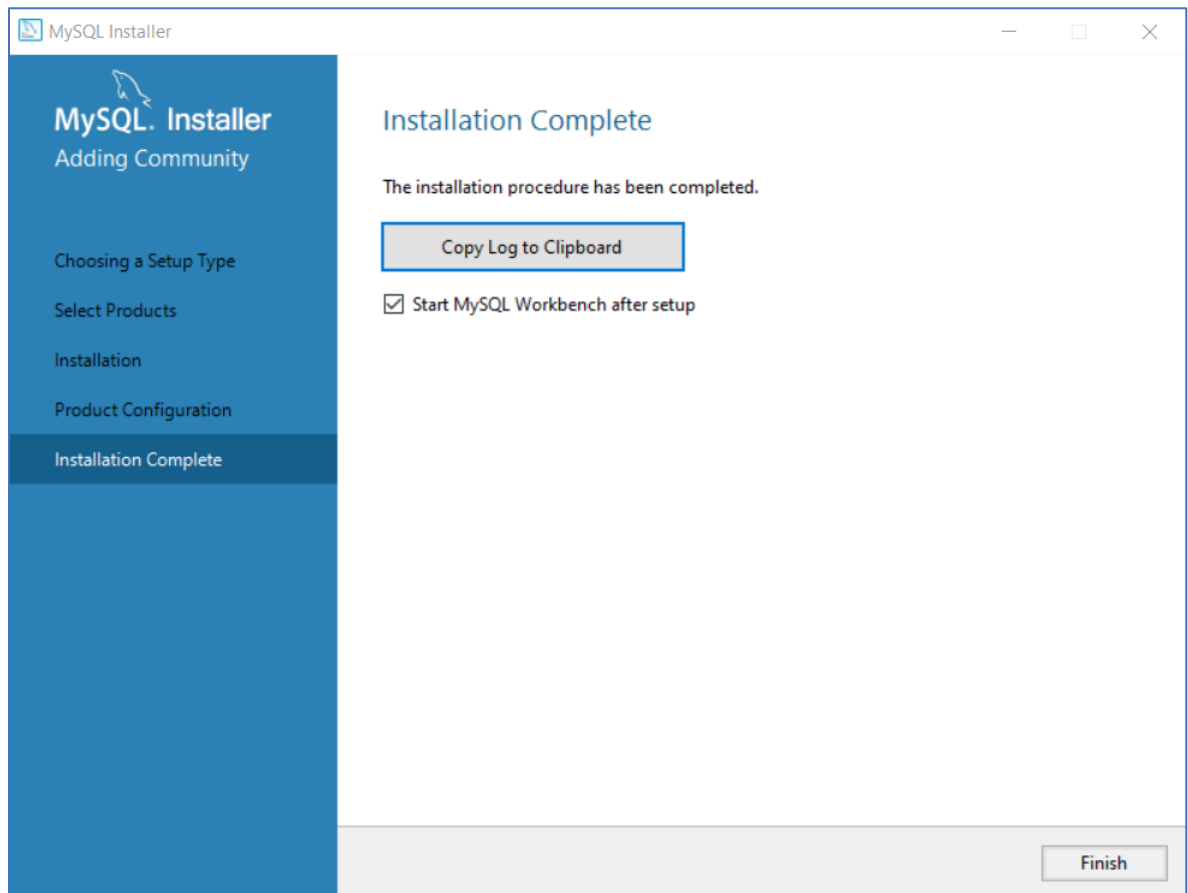
- Click **Finish**:



- Another **Next**: 😊

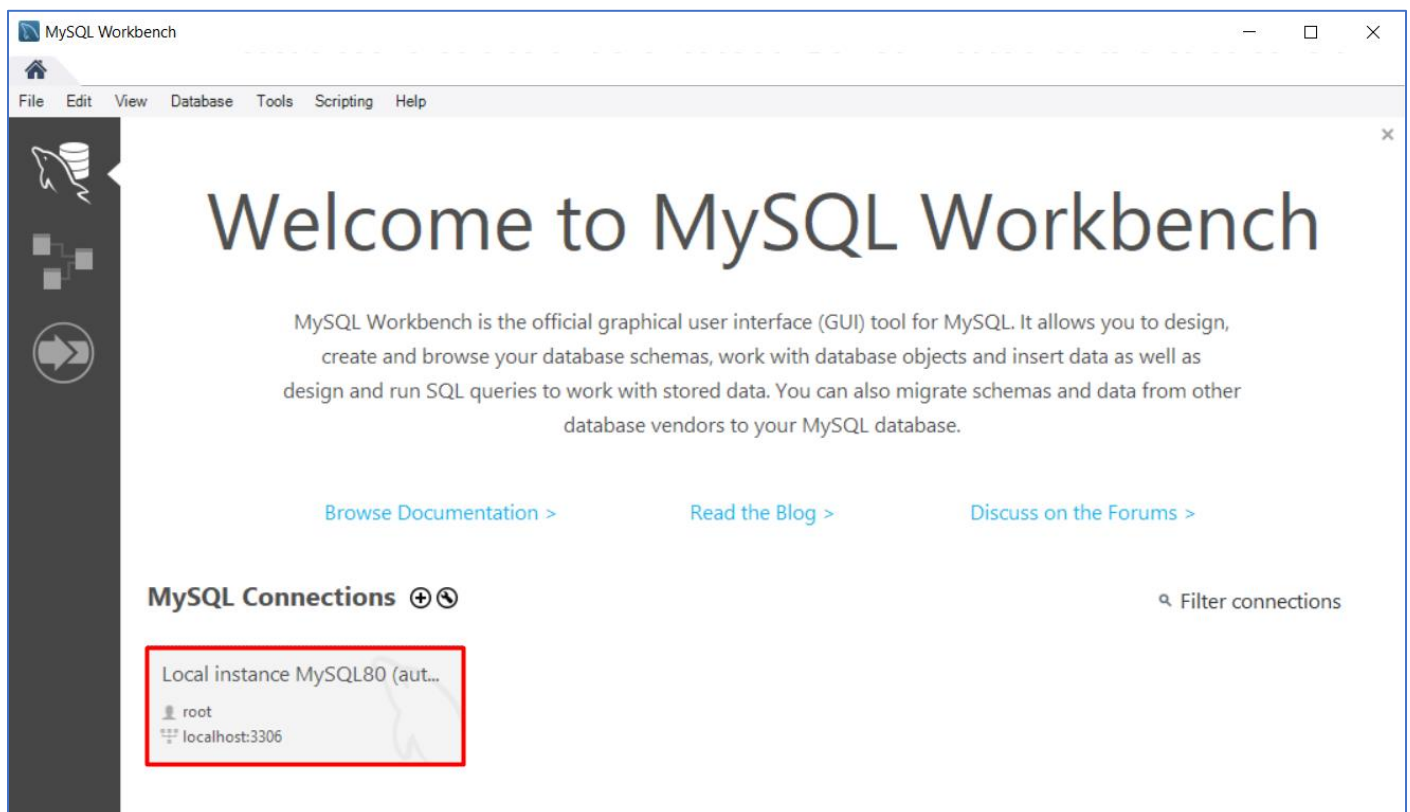


1. And one final **Finish**:

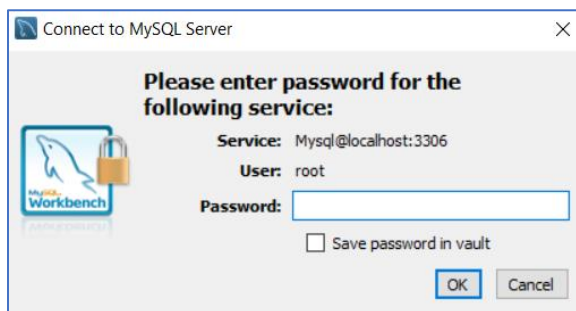


3. Simple Database and Queries

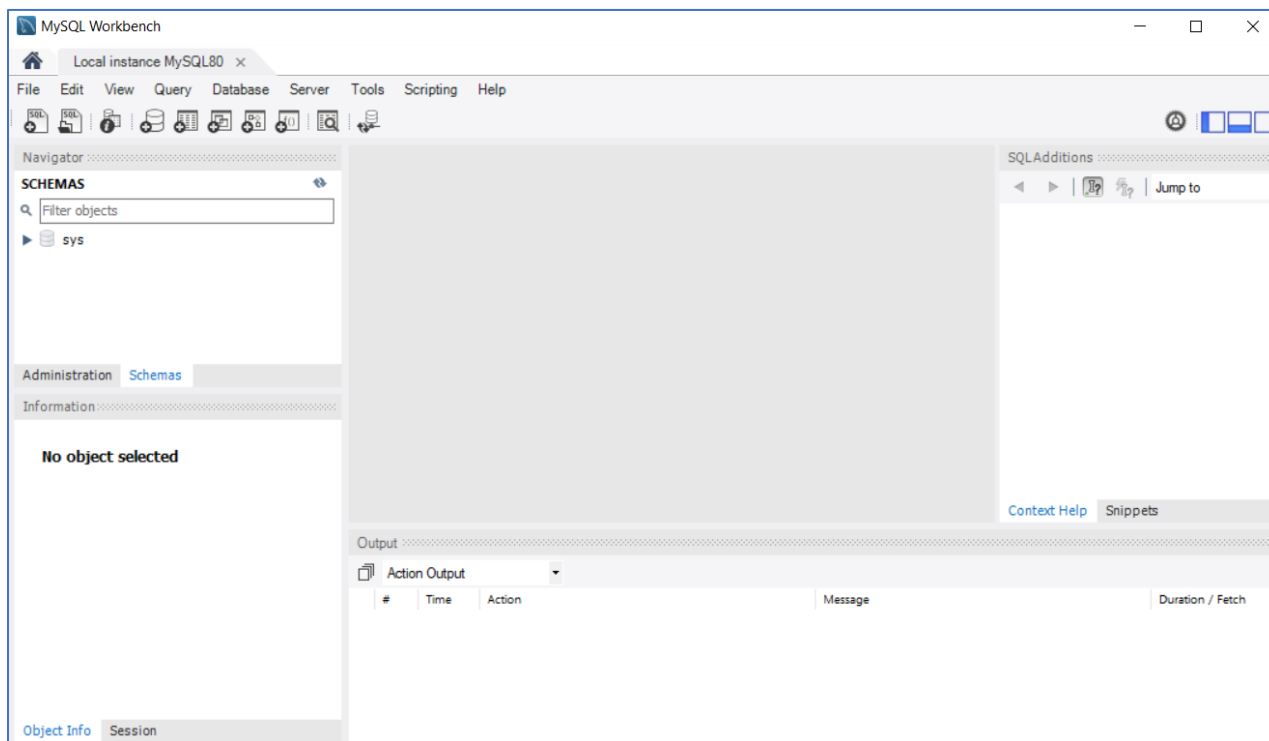
1. Open Workbench and connect to the local server.



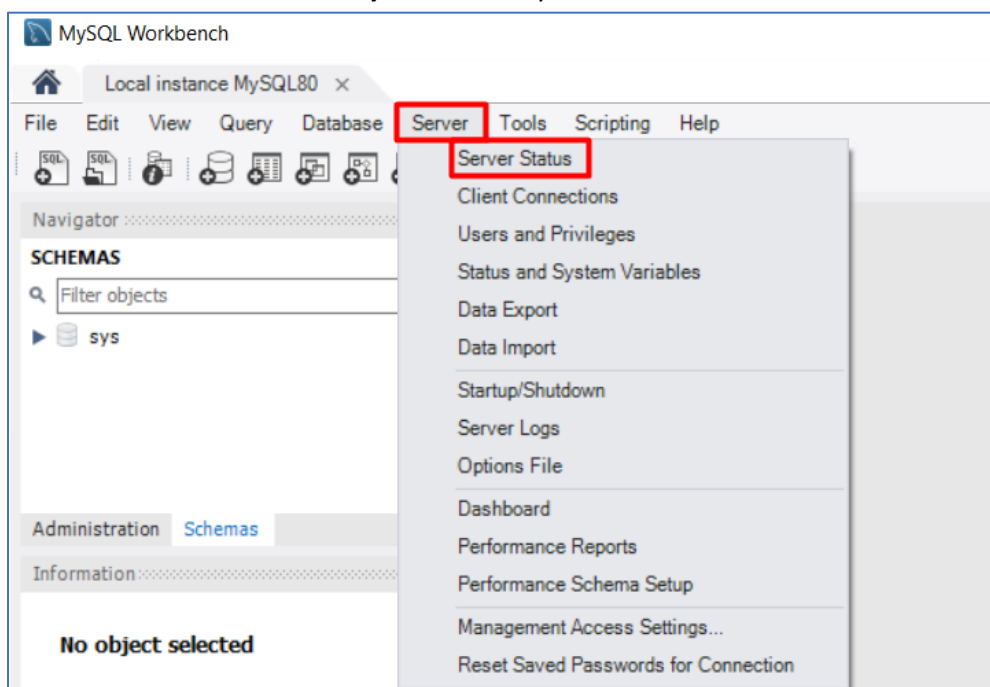
2. Enter the **password** you previously created.

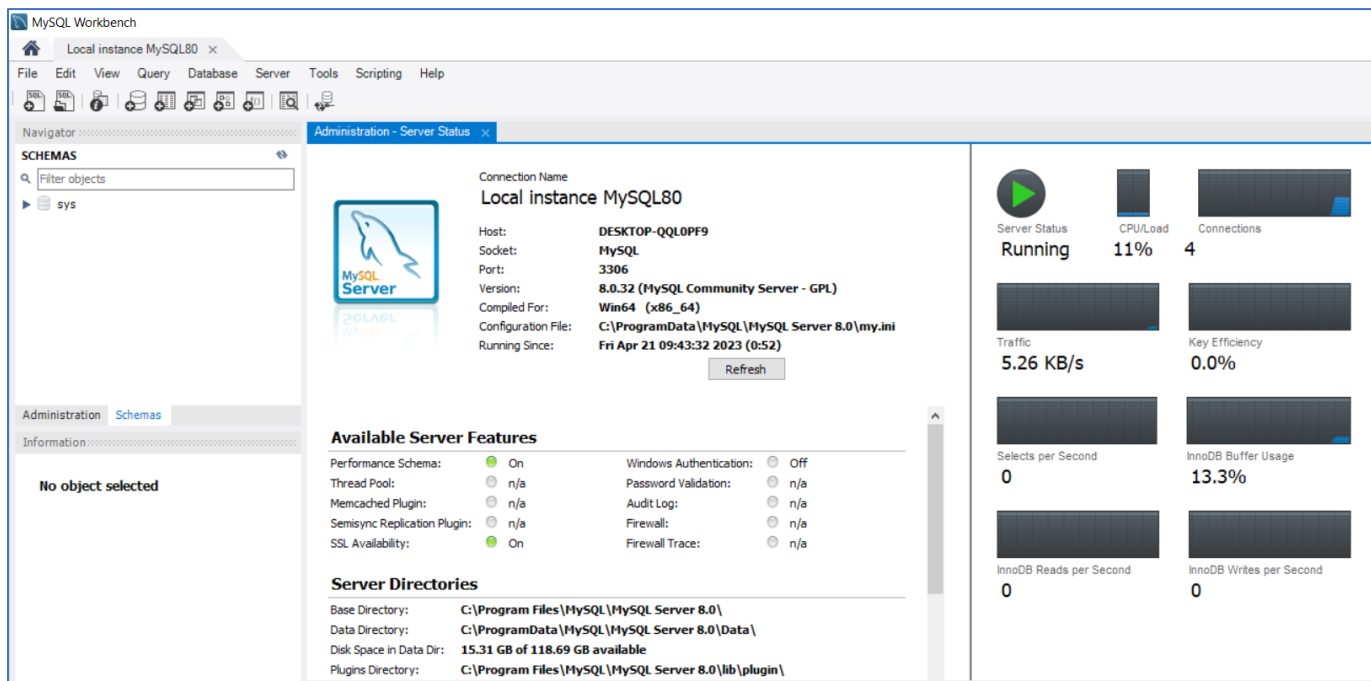


3. This is how **Workbench** looks like.

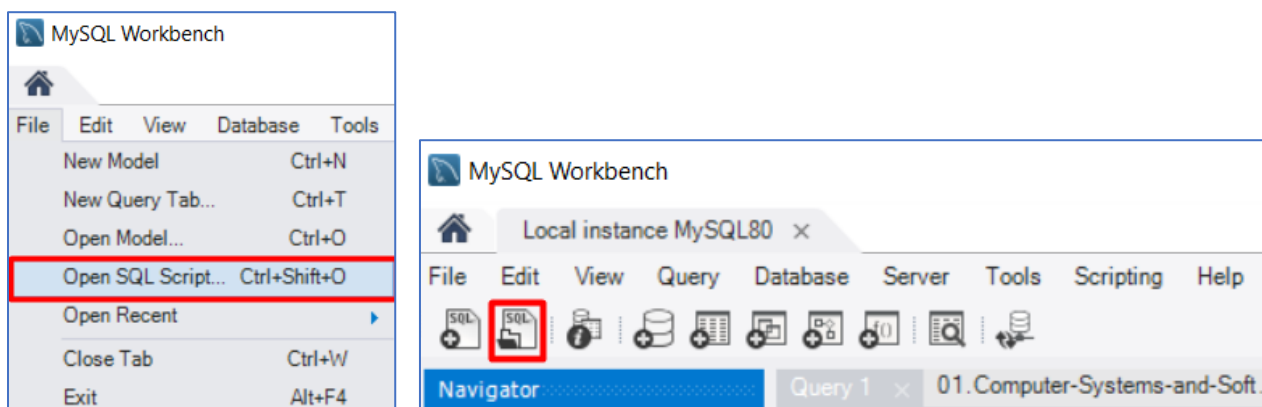


4. You can check the **status of your server** if you'd like.

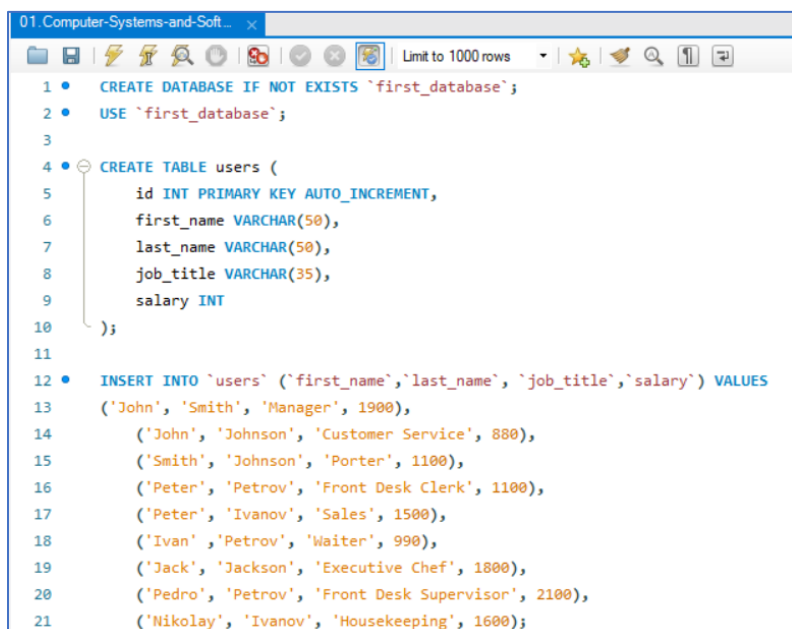




5. Now, we will open an existing SQL script, which will create a simple database, containing just one table and populate it with records. You are provided with the file "**01.Computer-Systems-and-Software-Exercise-MySQL-Database.sql**". You can open it in two ways:

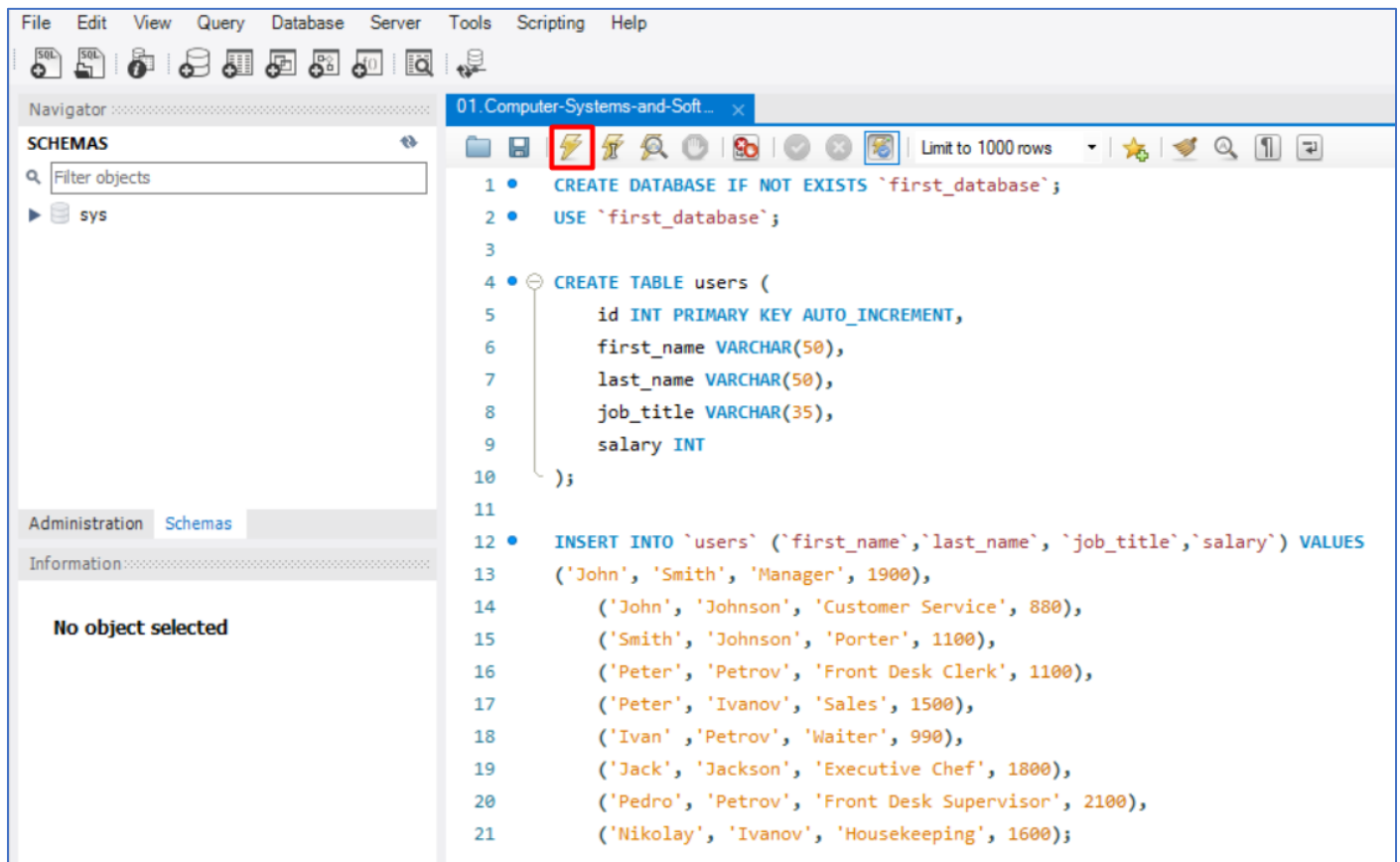


6. This is how the file **would look like** when you open it.



7. As you can see SQL is very logical and user friendly. The given script will create a database called "first_database" (if such database doesn't already exist), then it will use the database to create a table "users" in it. Each user will have an id, first_name, last_name, job_title and salary fields. After creating the table, it will populate it with values.

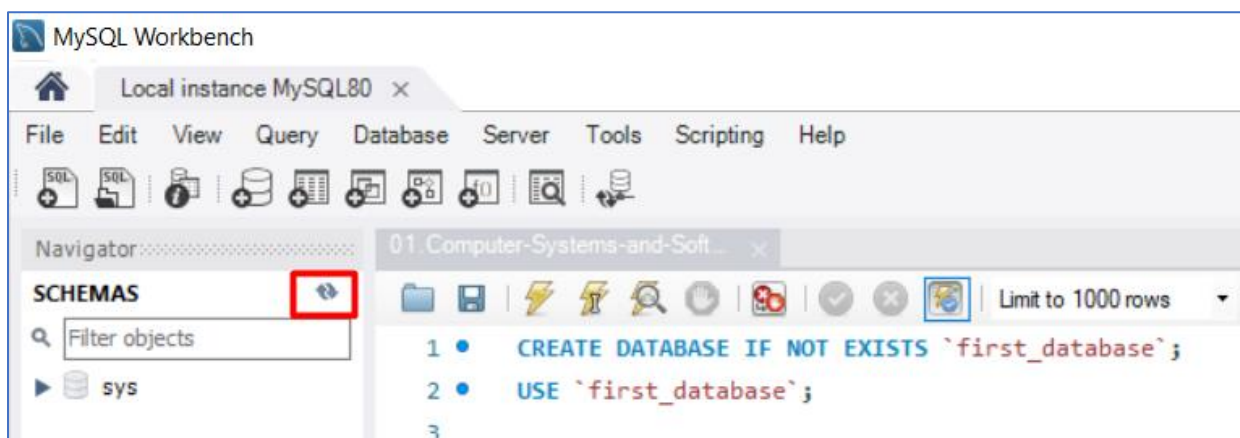
In order to **run the script**, hit the **yellow bolt button**.



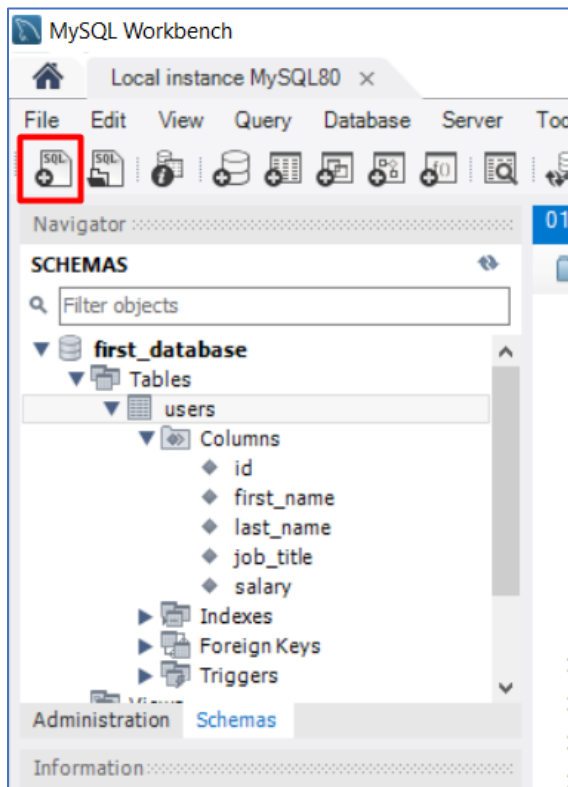
8. In the output section you should see the **commands executed**.

Output				
Action Output				
#	Time	Action	Message	
✓ 1	09:55:52	CREATE DATABASE IF NOT EXISTS `first_database`	1 row(s) affected	
✓ 2	09:55:52	USE `first_database`	0 row(s) affected	
✓ 3	09:55:52	CREATE TABLE users (id INT PRIMARY KEY AUTO_INCREMENT, first_name VARCHAR(50), last_name VARCHAR(50), job_title VARCHAR(35), salary INT)	0 row(s) affected	
✓ 4	09:55:52	INSERT INTO `users` (`first_name`,`last_name`,`job_title`,`salary`) VALUES ('John', 'Smith', 'Manager', 1900), ('John', 'Johnson', 'Customer Service', 880), ('Smith', 'Johnson', 'Porter', 1100), ('Peter', 'Petrov', 'Front Desk Clerk', 1100), ('Peter', 'Ivanov', 'Sales', 1500), ('Ivan', 'Petrov', 'Waiter', 990), ('Jack', 'Jackson', 'Executive Chef', 1800), ('Pedro', 'Petrov', 'Front Desk Supervisor', 2100), ('Nikolay', 'Ivanov', 'Housekeeping', 1600);	9 row(s) affected Records: 9 Duplicates: 0 Warnings: 0	

9. Hit the refresh button to **see the newly created database**.



10. As you can see, the database was created with all its attributes. Now, let's see how it looks like and write a few queries. **Click on the "Create a new SQL tab..." button.**



11. Now, we will **select all the records** from "users" table. Write the following query:

```
SELECT * FROM users;
```

Now hit the other bolt button, which has something like an "I" on it.

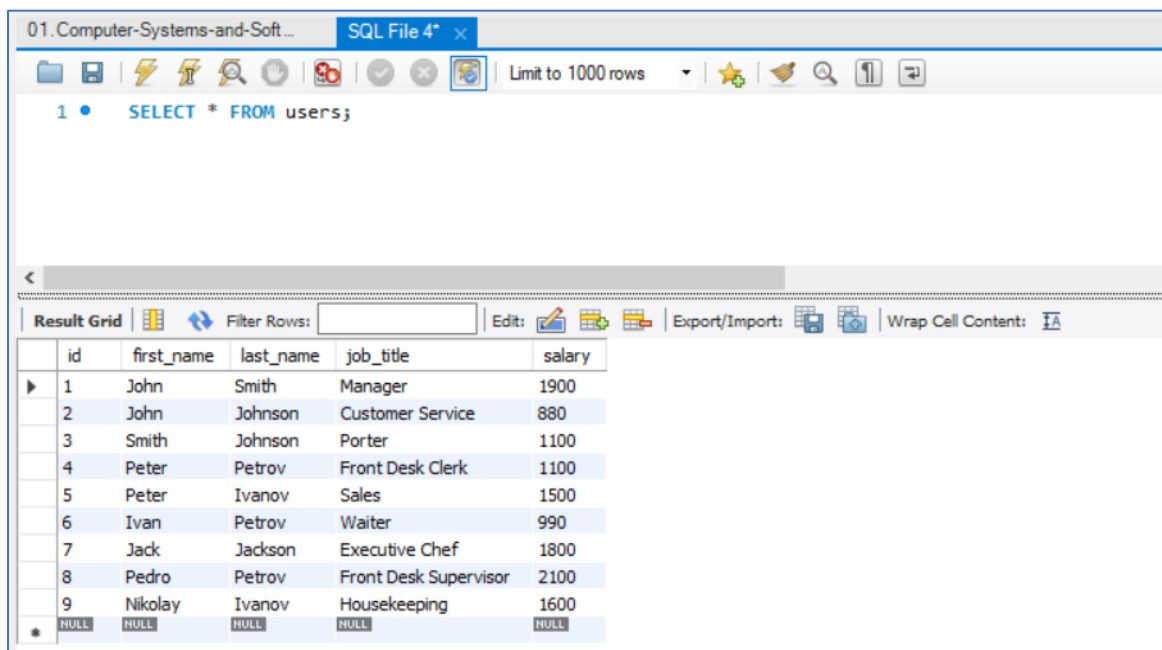
Note: The difference between those two buttons is, as follows:



- Executes the selected portion of the script or everything, if there's no selection



- Executes the statement under the keyboard cursor



12. Now, that we know how all records look like, let's execute a second **query**, which will select all records, where **salary is equal or more than 1500**.

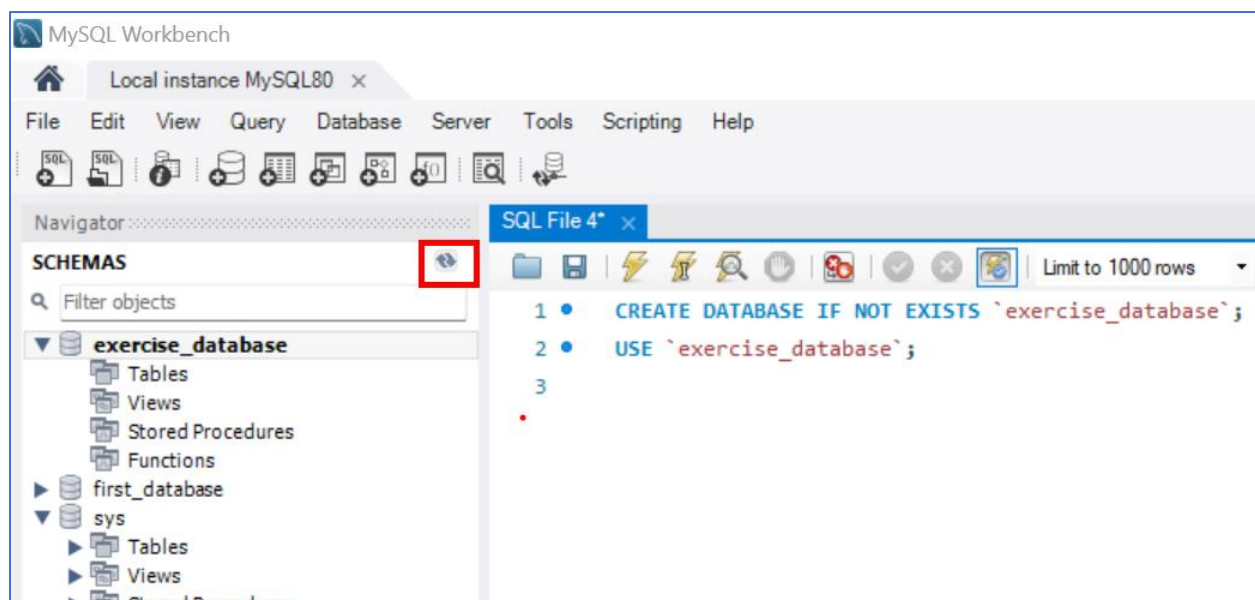
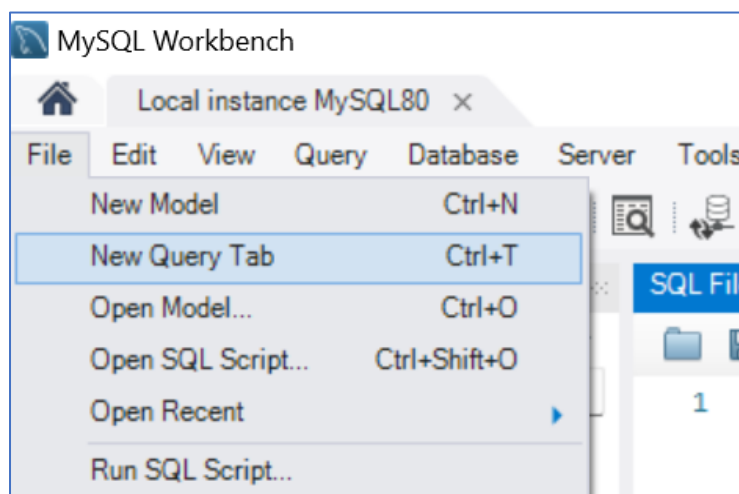
```
2
3 • SELECT * FROM users WHERE salary >=1500;
4
```

13. Try a few queries by yourself.

- Try to select all with last name "Petrov".
- Try to select all with first name "Peter".
- Try to select all with salary below 1800.

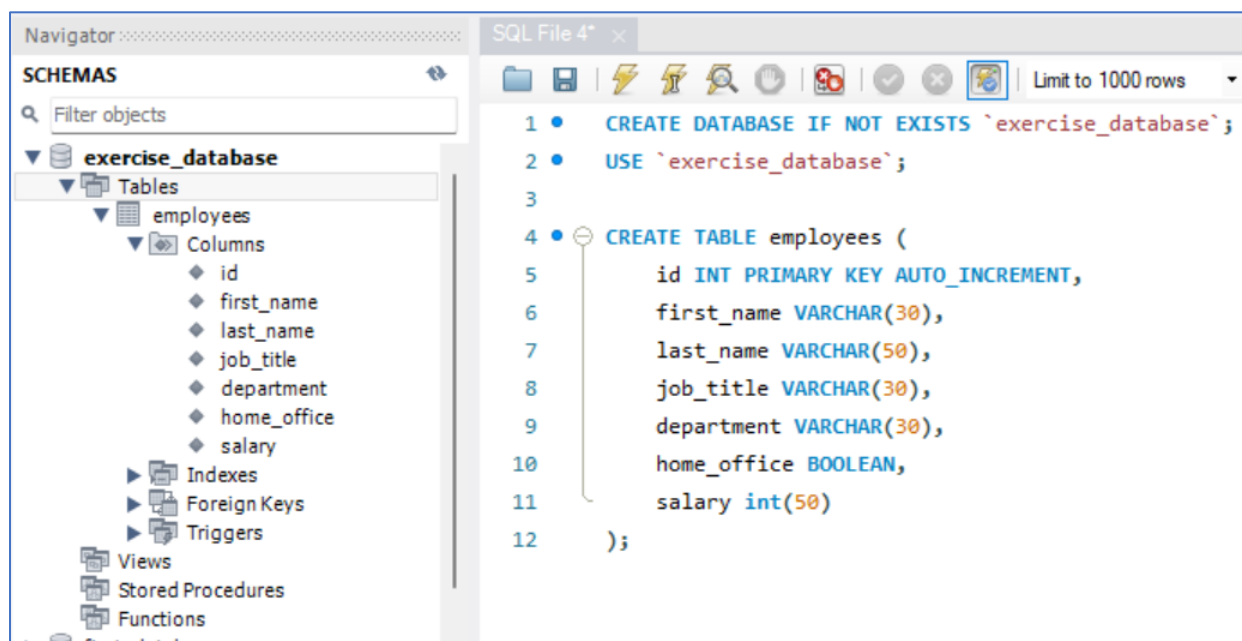
4. Create a Database and Execute Basic Operations

1. Open a **New Query Tab** and **Create a new Database** named "**exercise_database**":
 - Do not forget to **refresh** in order to see the newly created database:



2. Create a New Table named "employees" with the following columns:

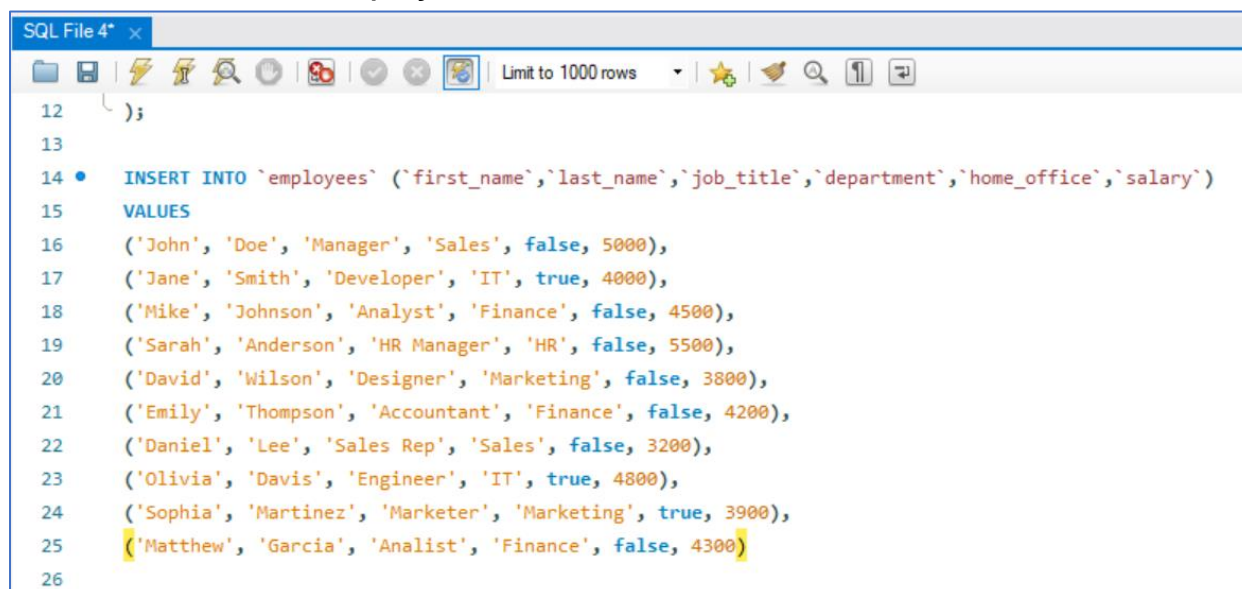
- id, first_name, last_name, job_title, department, home_office, salary



The screenshot shows a SQL Navigator window with a tree view on the left and a SQL editor on the right. The tree view shows the 'exercise_database' schema with a table 'employees' containing columns: id, first_name, last_name, job_title, department, home_office, and salary. The SQL editor shows the following code:

```
1 • CREATE DATABASE IF NOT EXISTS `exercise_database`;
2 • USE `exercise_database`;
3
4 • CREATE TABLE employees (
5     id INT PRIMARY KEY AUTO_INCREMENT,
6     first_name VARCHAR(30),
7     last_name VARCHAR(50),
8     job_title VARCHAR(30),
9     department VARCHAR(30),
10    home_office BOOLEAN,
11    salary int(50)
12 );
```

3. Insert some records in the "employees" table:



The screenshot shows a SQL editor window with the following code:

```
12 );
13
14 • INSERT INTO `employees` (`first_name`,`last_name`,`job_title`,`department`,`home_office`,`salary`)
15 VALUES
16 ('John', 'Doe', 'Manager', 'Sales', false, 5000),
17 ('Jane', 'Smith', 'Developer', 'IT', true, 4000),
18 ('Mike', 'Johnson', 'Analyst', 'Finance', false, 4500),
19 ('Sarah', 'Anderson', 'HR Manager', 'HR', false, 5500),
20 ('David', 'Wilson', 'Designer', 'Marketing', false, 3800),
21 ('Emily', 'Thompson', 'Accountant', 'Finance', false, 4200),
22 ('Daniel', 'Lee', 'Sales Rep', 'Sales', false, 3200),
23 ('Olivia', 'Davis', 'Engineer', 'IT', true, 4800),
24 ('Sophia', 'Martinez', 'Marketer', 'Marketing', true, 3900),
25 ('Matthew', 'Garcia', 'Analist', 'Finance', false, 4300)
26
```

4. Now try **several basic queries** in order to **select** the inserted **records from the table**:

26

27 • `SELECT * FROM `employees``

Result Grid | Filter Rows: | Edit: | Export/Import:

	id	first_name	last_name	job_title	department	home_office	salary
▶	1	John	Doe	Manager	Sales	0	5000
	2	Jane	Smith	Developer	IT	1	4000
	3	Mike	Johnson	Analyst	Finance	0	4500
	4	Sarah	Anderson	HR Manager	HR	0	5500
	5	David	Wilson	Designer	Marketing	0	3800
	6	Emily	Thompson	Accountant	Finance	0	4200
	7	Daniel	Lee	Sales Rep	Sales	0	3200
	8	Olivia	Davis	Engineer	IT	1	4800
	9	Sophia	Martinez	Marketer	Marketing	1	3900
	10	Matthew	Garcia	Analist	Finance	0	4300
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

28 • `SELECT * FROM `employees` WHERE `department` = 'IT';`

Result Grid | Filter Rows: | Edit: | Export/Import:

	id	first_name	last_name	job_title	department	home_office	salary
▶	2	Jane	Smith	Developer	IT	1	4000
	8	Olivia	Davis	Engineer	IT	1	4800
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

29 • `SELECT * FROM `employees` WHERE `department` = 'Marketing' AND `home_office` = TRUE;`

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

	id	first_name	last_name	job_title	department	home_office	salary
▶	9	Sophia	Martinez	Marketer	Marketing	1	3900
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL