

STRUCTURED QUERY LANGUAGE

SQL

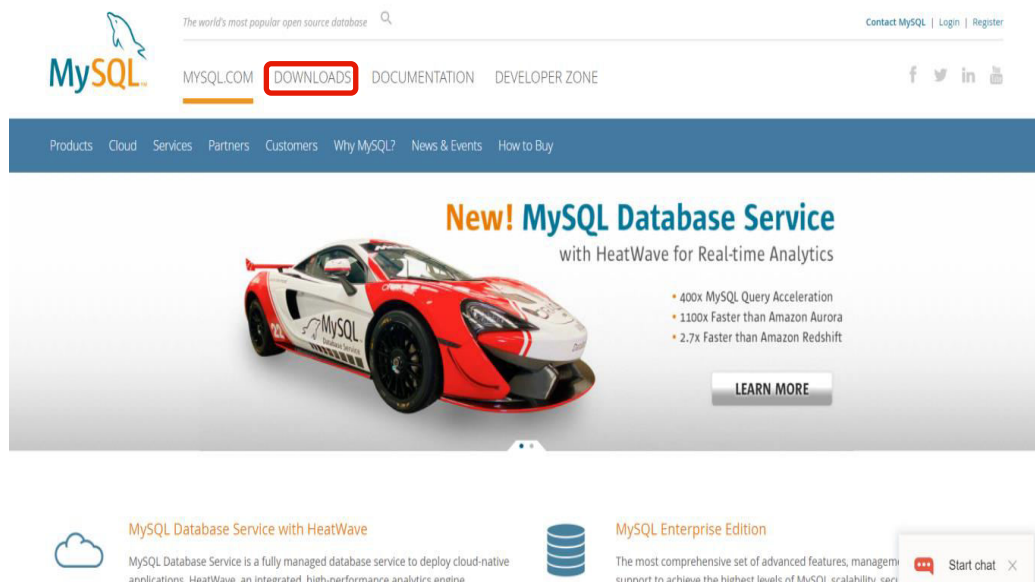
- SQL stands for Structured Query Language and it is a standard and domain specific computer language used for managing data in relational database management system which stores data in the form of tables and also to retrieve and update data in a database.
- SQL works with database like DB2, MySQL, PostgreSQL, Oracle, SQLite, SQL Server, Sybase, MS Access

MySQL

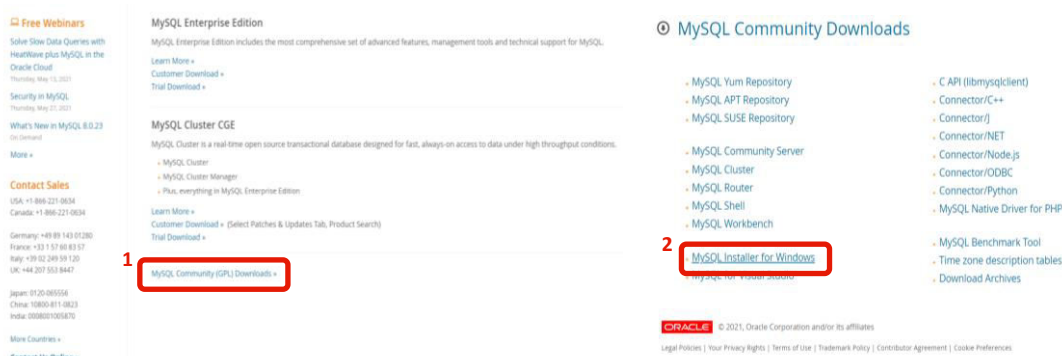
- MySQL is an Oracle-backed open source relational database management system (RDBMS) based on Structured Query Language (SQL).

Installation

- **Prerequisite for Windows**
 - ✓ Microsoft .NET Framework 4.5.2
 - ✓ Microsoft Visual C++ Redistributable for Visual Studio 2019
 - ✓ Microsoft Windows 10 or Windows Server 2019
- **Downloading MySQL 8.0.25**
 - ✓ **Step-1:** Go to the official website <https://www.mysql.com/> and click on the downloads as shown in the figure



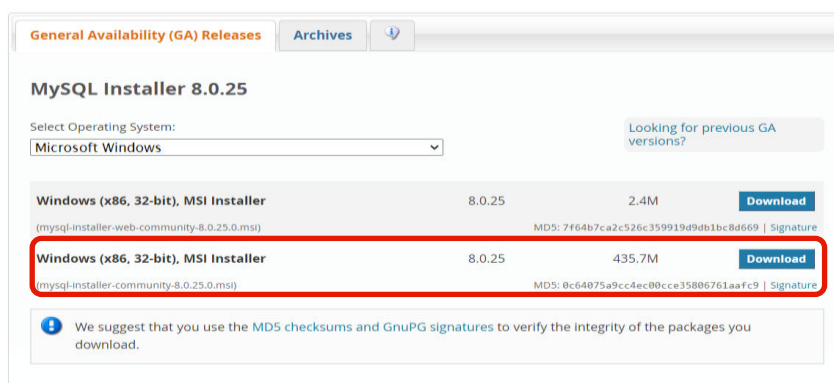
- ✓ **Step-2:** Click on the MySQL Community(GPL) Downloads and the click MySQL Installer for windows as follows



✓ **Step-3:** Click on Window(x86, 32-bit), MSI Installer with 435.7M

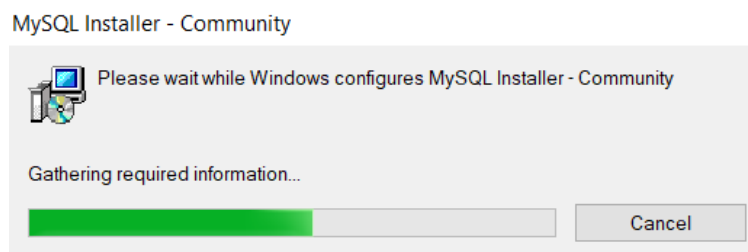
MySQL Community Downloads

MySQL Installer

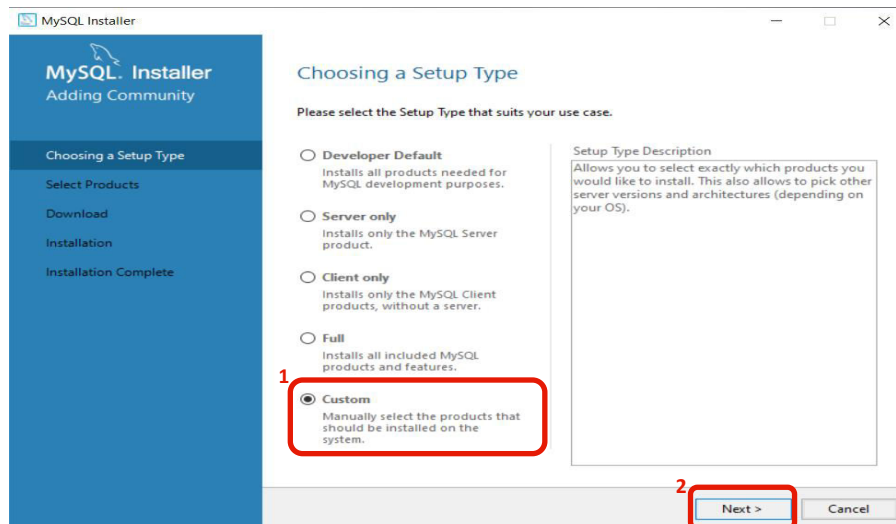


Installing MySQL

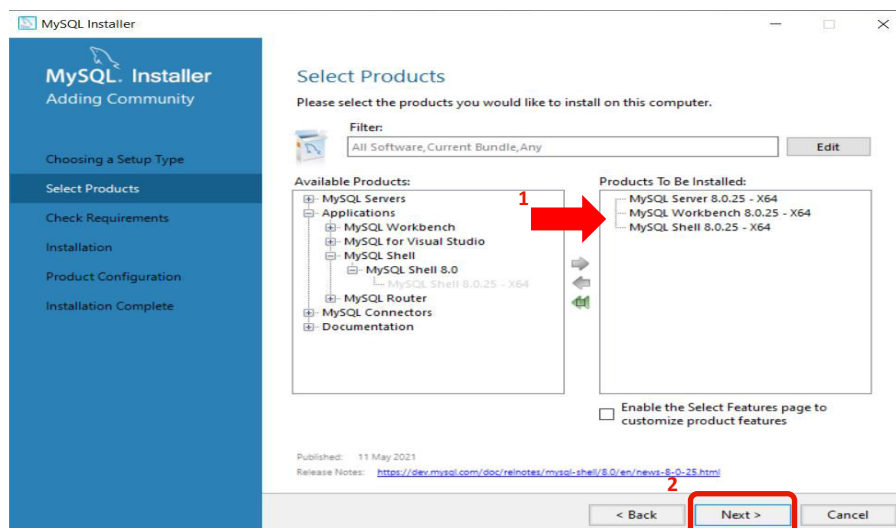
✓ **Step-1:** Double click the **MSI installer .exe file**. It will give the following screen.
(The exe file can be found in Downloads folder in the PC)



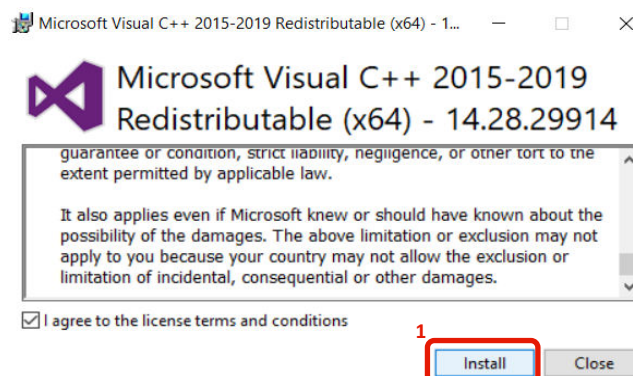
✓ **Step-2:** In the next wizard, **choosing the Setup Type**, There are several types available, and you need to choose the appropriate option to install MySQL product and features. Here, we are going to select the **Custom** option and click on the **Next button**.

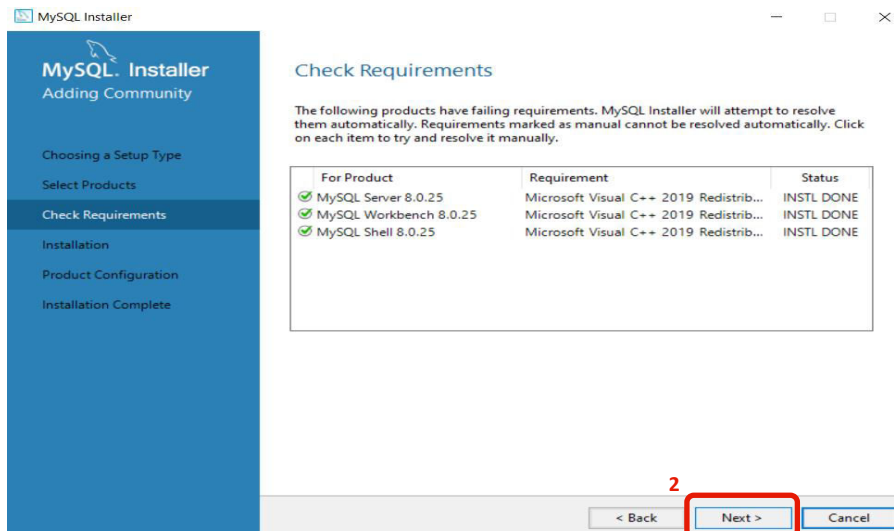


- ✓ **Step-3:** In **Select Product** we need MySQL Server, MySQL Workbench, MySQL Shell. So, we select **MySQL Server 8.0.25-x64**, **MySQL Workbench 8.0.25-x64**, **MySQL Shell 8.0.25-x64** from available products and drag them to products to be installed panel and click **next**

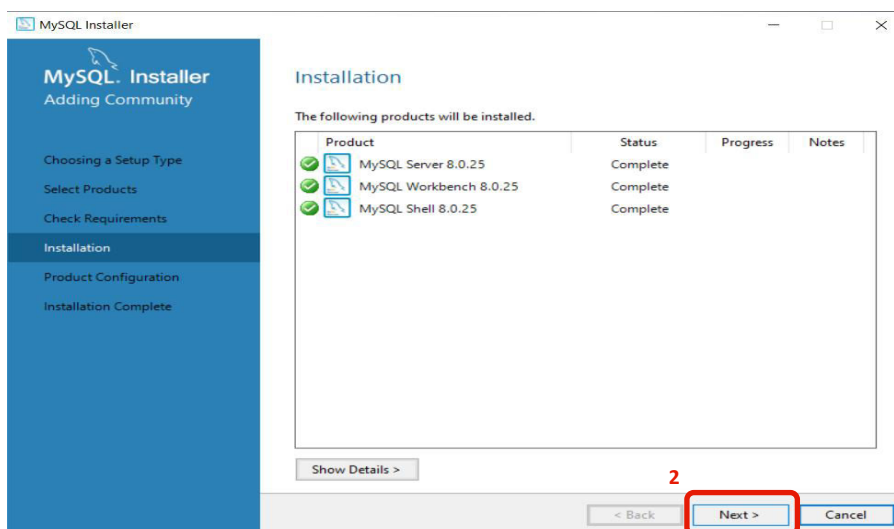
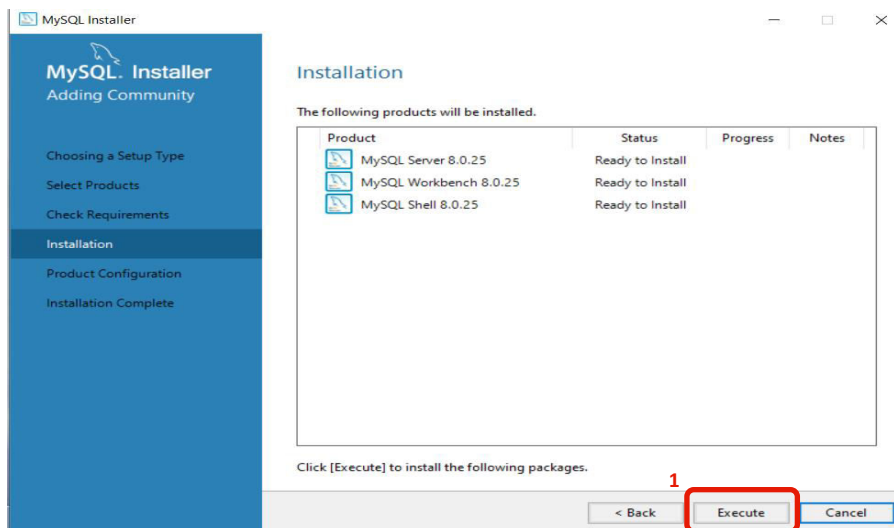


- ✓ **Step-4:** In **Check Requirements**, wizard checks the all the prerequisite are available, if not it prompts to **install the required software** after the agreement and installation click **next** as shown in the figure

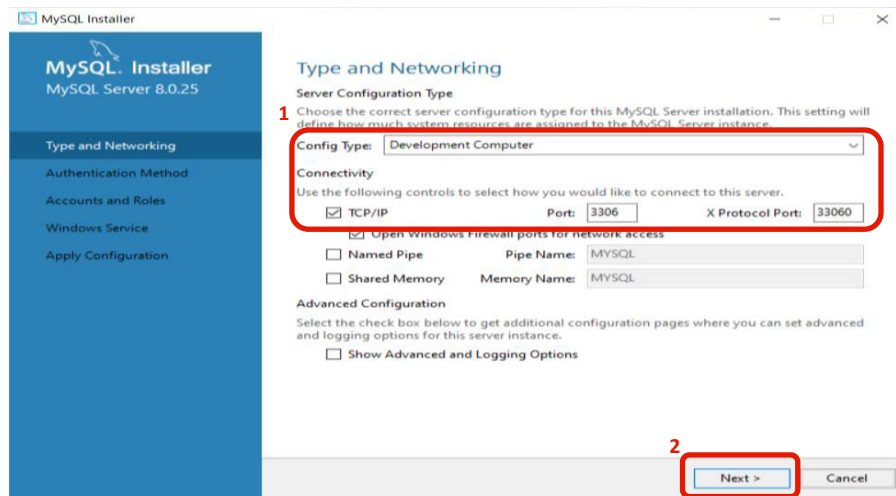




- ✓ **Step-5:** In Installation Click **Execute** to install the selected packages after the status is changed to complete click **Next** as shown

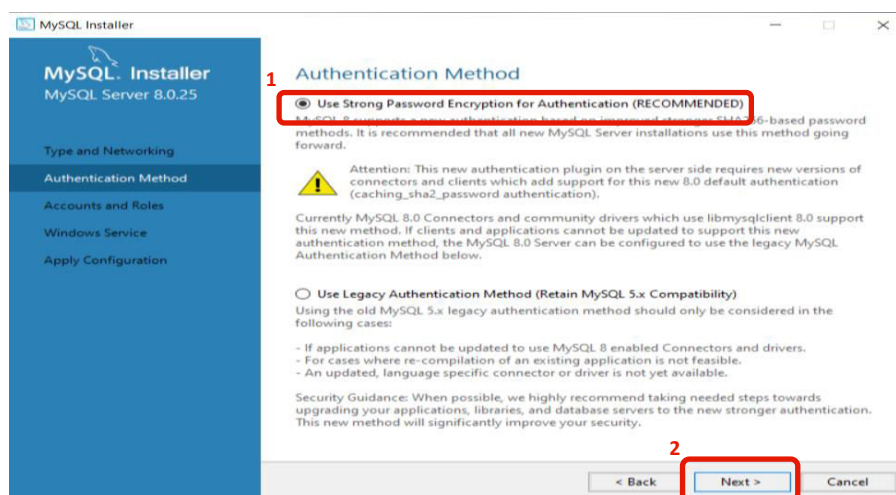


- ✓ **Step-6:** In **type and networking** we choose configuration type, port as shown and click **next**



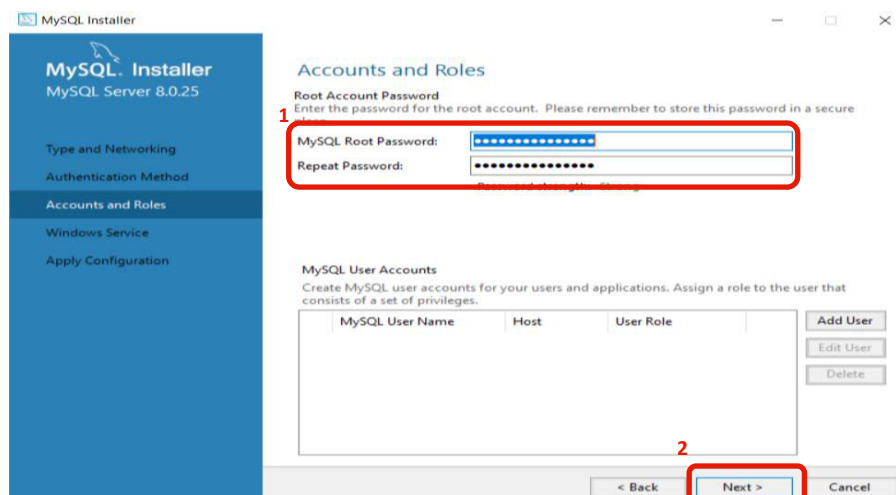
The screenshot shows the 'Type and Networking' screen of the MySQL Installer. On the left sidebar, 'Type and Networking' is selected. The main area is titled 'Type and Networking' and contains a 'Server Configuration Type' section with a dropdown menu set to 'Development Computer'. Below this is a 'Connectivity' section with a red box around the 'TCP/IP' checkbox, the 'Port' field set to '3306', and the 'X Protocol Port' field set to '33060'. There are also checkboxes for 'Open Windows Firewall ports for network access', 'Named Pipe', and 'Shared Memory'. At the bottom, there is an 'Advanced Configuration' section with a checkbox for 'Show Advanced and Logging Options'. A red box labeled '2' highlights the 'Next >' button at the bottom right.

- ✓ **Step-7:** In **Authentication method** choose, Use strong Password Encryption and click **next**



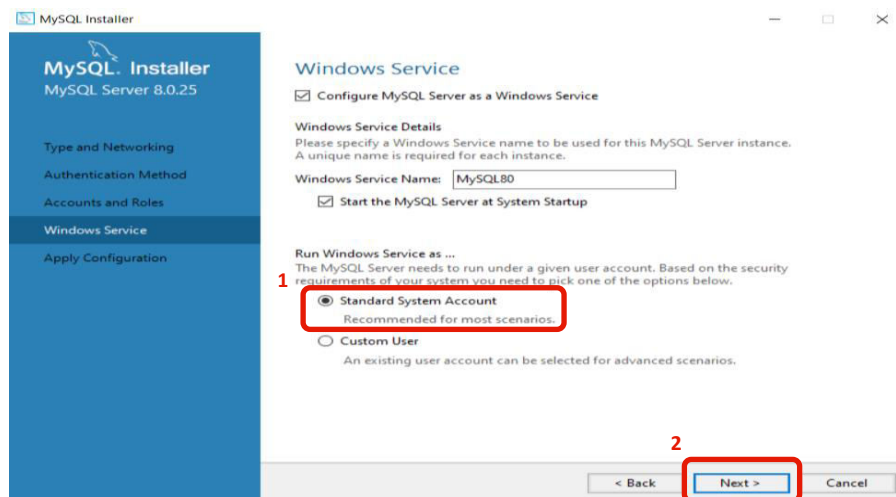
The screenshot shows the 'Authentication Method' screen of the MySQL Installer. On the left sidebar, 'Authentication Method' is selected. The main area is titled 'Authentication Method' and contains two radio button options. The first option, 'Use Strong Password Encryption for Authentication (RECOMMENDED)', is selected and highlighted with a red box labeled '1'. Below this option is a warning icon and text about the new authentication plugin. The second option is 'Use Legacy Authentication Method (Retain MySQL 5.x Compatibility)'. At the bottom, there is a red box labeled '2' highlighting the 'Next >' button.

- ✓ **Step-8:** In **Accounts and roles** set a **password for root account** and remember the password and click **next**

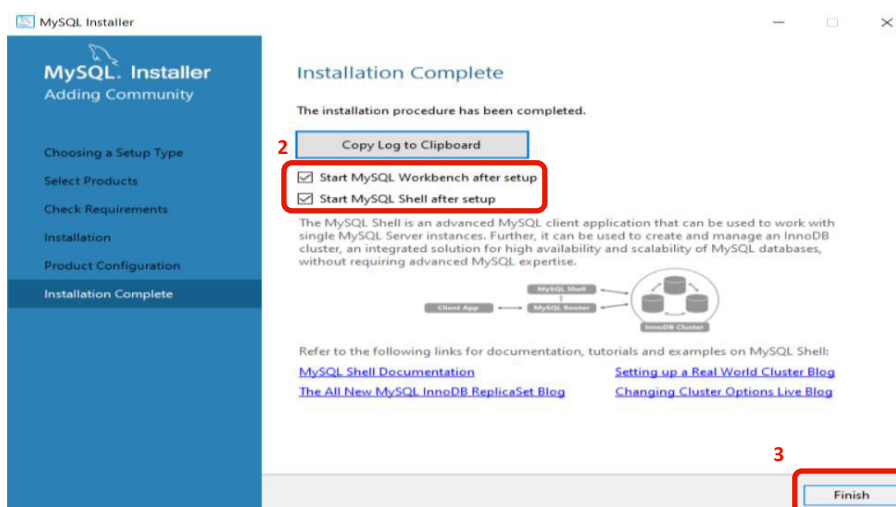
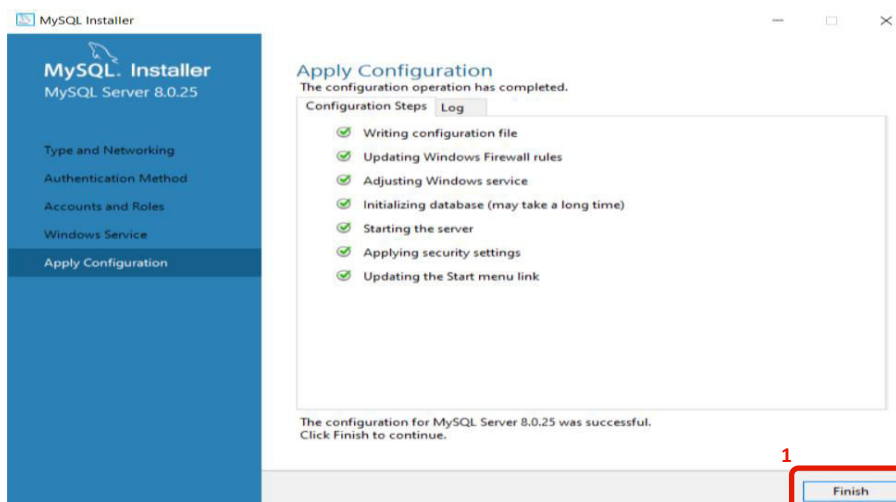


The screenshot shows the 'Accounts and Roles' screen of the MySQL Installer. On the left sidebar, 'Accounts and Roles' is selected. The main area is titled 'Accounts and Roles' and contains a 'Root Account Password' section with a red box labeled '1' around the 'MySQL Root Password' and 'Repeat Password' fields. Below this is a 'MySQL User Accounts' section with a table for creating user accounts. At the bottom, there is a red box labeled '2' highlighting the 'Next >' button.

✓ **Step-9: Select Standard System Account in Window Services and click next**

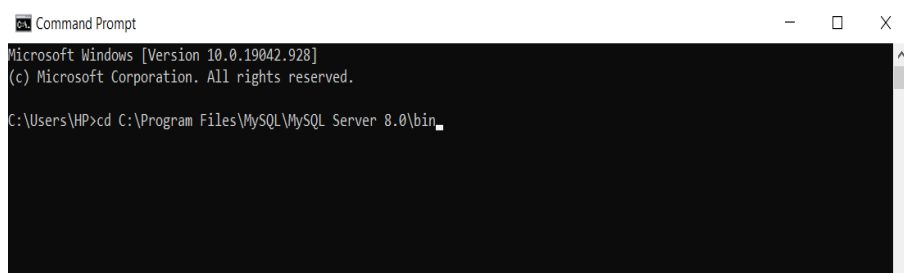


✓ **Step-10: After the configuration operations has completed click on finish and tick the boxes to start workbench and shell and click on finish. Installation is completed**



▪ Verify MySQL Installation


- ✓ **Step-1:** Open the Command Prompt and navigate to the bin folder of MySQL Server 8.0.25 i.e. `c:\program files\MySQL\MySQL Server 8.0\bin`



```
Command Prompt
Microsoft Windows [Version 10.0.19042.928]
(c) Microsoft Corporation. All rights reserved.

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

- ✓ **Step-2:** Run the command `mysql -m root -p`



```
Command Prompt
Microsoft Windows [Version 10.0.19042.928]
(c) Microsoft Corporation. All rights reserved.

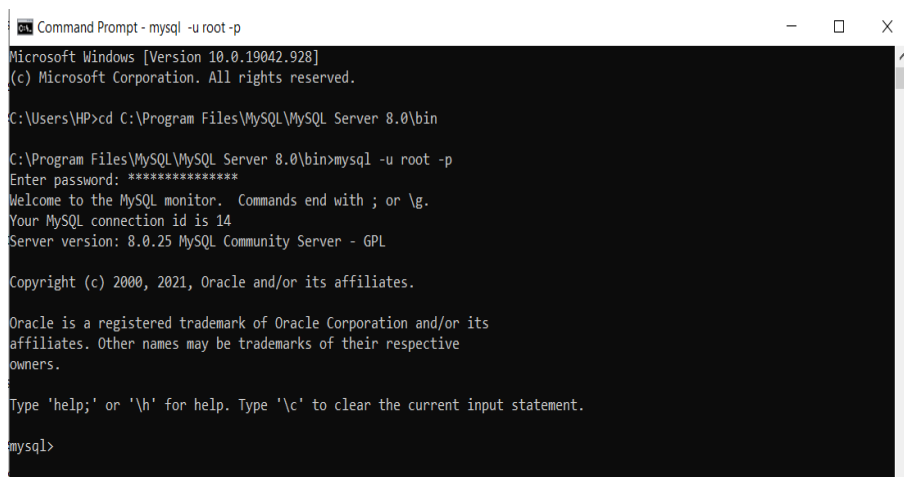
C:\Users\HP>cd C:\Program Files\MySQL\MySQL Server 8.0\bin
C:\Program Files\MySQL\MySQL Server 8.0\bin>mysql -m root -p
```

- ✓ **Step-3:** It will prompt you to enter the password and enter password created during the installation then it will show you the MySQL command line prompt as shown in the figure



```
Command Prompt - mysql -u root -p
Microsoft Windows [Version 10.0.19042.928]
(c) Microsoft Corporation. All rights reserved.

C:\Users\HP>cd C:\Program Files\MySQL\MySQL Server 8.0\bin
C:\Program Files\MySQL\MySQL Server 8.0\bin>mysql -u root -p
Enter password: *****
```



```
Command Prompt - mysql -u root -p
Microsoft Windows [Version 10.0.19042.928]
(c) Microsoft Corporation. All rights reserved.

C:\Users\HP>cd C:\Program Files\MySQL\MySQL Server 8.0\bin
C:\Program Files\MySQL\MySQL Server 8.0\bin>mysql -u root -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 14
Server version: 8.0.25 MySQL Community Server - GPL

Copyright (c) 2000, 2021, 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.

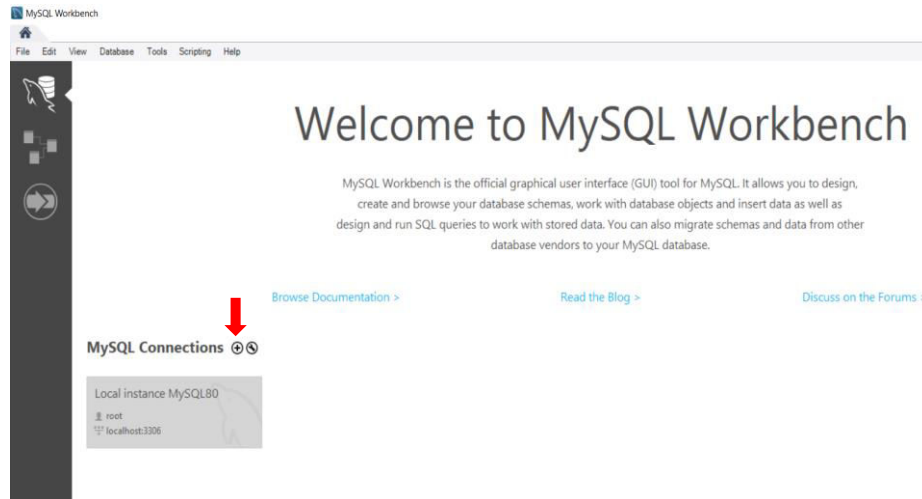
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

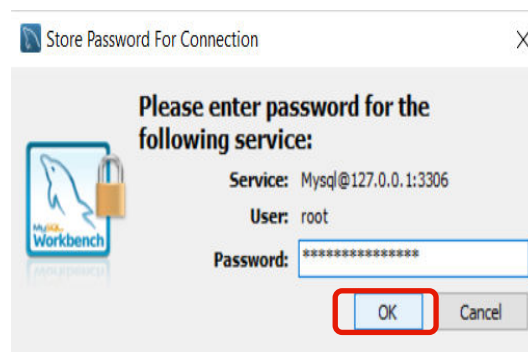
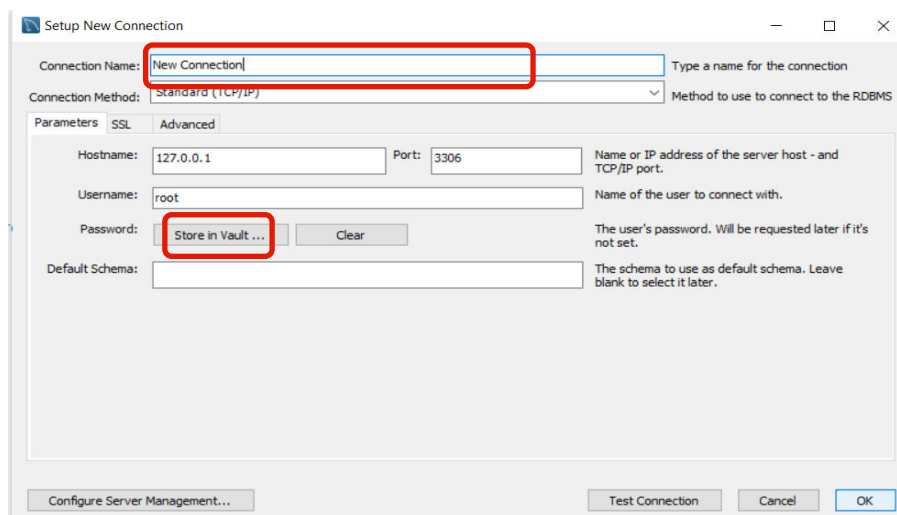

Working with MySQL Workbench

MySQL New Connections

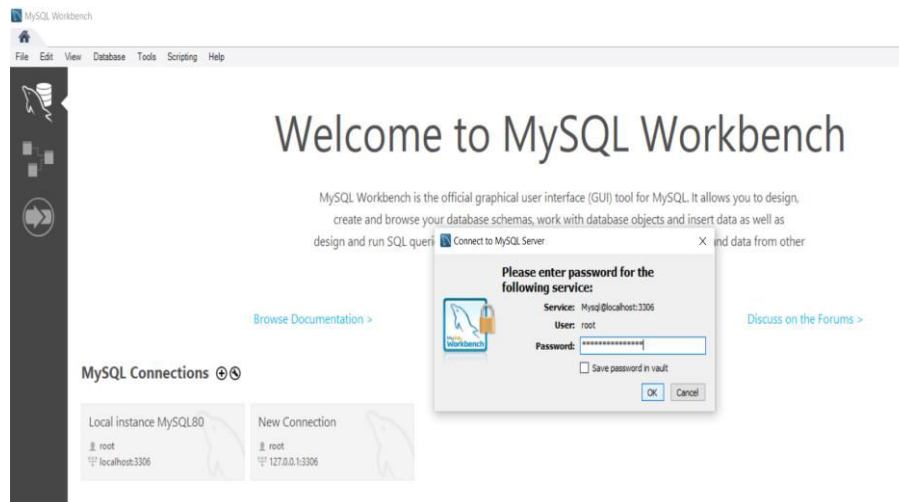
- ✓ **Step-1:** Launch the MySQL Workbench and the following screen appear and you will find an Local instance which is already created during installation. To create a new connection click on the plus icon as shown in the figure



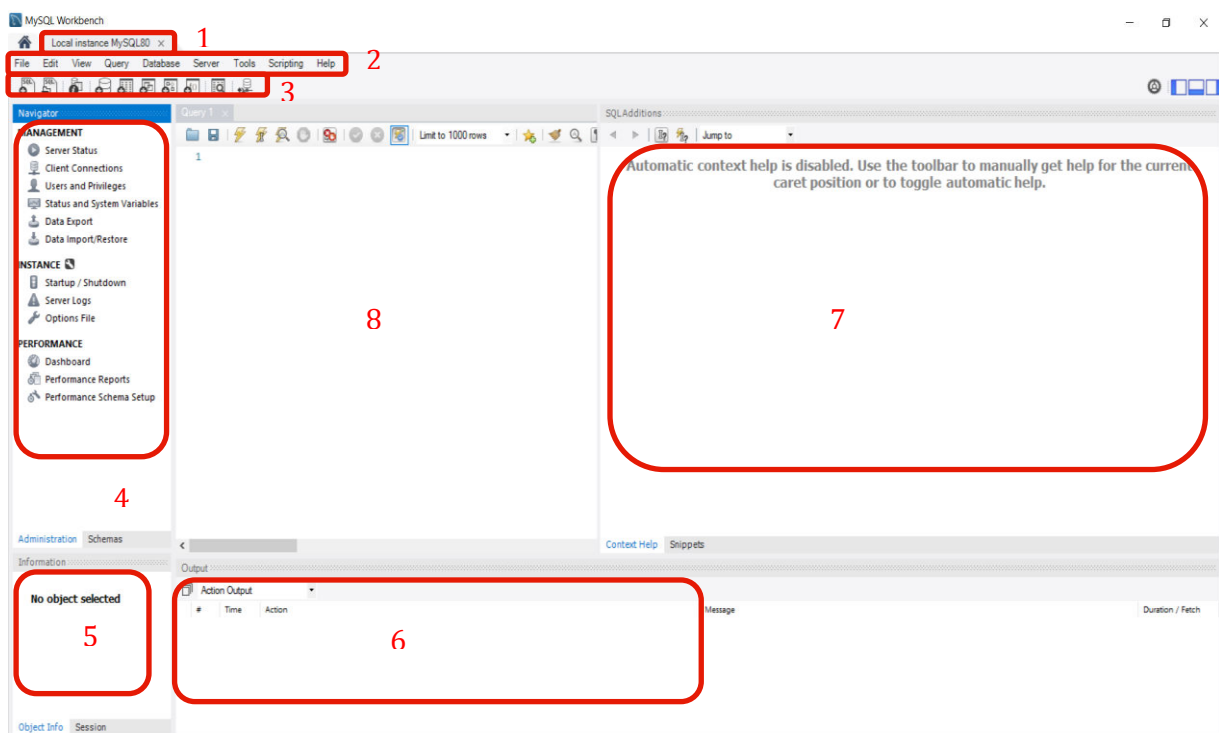
- ✓ **Step-2:** Fill the box to create a connection, such as connection name. Then click the Store in Vault button to write the password for the given user account and click Ok



- ✓ **Step-3:** Click on any of the connection created and give the related password to launch the MySQL Workbench screen



▪ Understanding Workbench



1. Connection tab-Information of current connection
2. Main menu bar- contains file,edit,view,query,server,tools and many more options
3. Main tool bar-contains icons to create schema,table,views,functions etc.
4. Administration tool-shows the information of the created tables, functions etc
5. Side bar panel-gives the information of selected object
6. Query result-it gives the output of the query written
7. Help panel
8. SQL Visual Editor- the panel where the query is written

SQL (Structured Query Language)

- Used to communicate with database.
- Used to perform task such as Updation, Retrieval, Insertion and deletion of data in the database.

SQL Commands

- SELECT - extracts data from a database
- UPDATE - updates data in a database
- DELETE - deletes data from a database
- INSERT INTO - inserts new data into a database
- CREATE DATABASE - creates a new database
- ALTER DATABASE - modifies a database
- CREATE TABLE - creates a new table
- ALTER TABLE - modifies a table
- DROP TABLE - deletes a table
- CREATE INDEX - creates an index (search key)
- DROP INDEX - deletes an index

Relational Database Management System (RDBMS)

- Basic for SQL.
- Stores data in the database in the collection of tables.
- RDBMS Systems: MySQL, Oracle

Tables

- Set of data that are organized in a model with rows and columns.
- Most common and simplest form of data storage in a relational database.

Fields

- Table with specific number of columns.
- For customer table it consist of fields ID, NAME, AGE, ADDRESS, and SALARY.

Schema

Schema diagram / ER diagram (Entity relationship)

Schema is a blueprint.

SQL data types:

Int, float, varchar(), Date, char, Blob

Create database dbName;

dbName;

usedbName;

create table Employee()

dbName1

create table Employee1 ()

Drop DB

The DROP DATABASE statement is used to drop an existing SQL database.

Syntax:

DROP DATABASE *databasename*;

Primary Key

- Specify a row uniquely.
- Contain UNIQUE values, and cannot contain NULL values.

Foreign Key

- One table related to primary key of other table.

Unique Key

- Provides uniqueness for column or set of columns.

Join

- Table combines with any field.

1. Inner Join: Returns dataset that have matching values in both tables.
2. Left Outer Join: Returns all records from the left table and matched records from the right table.
3. Right Outer Join: Returns all records from the right table and matched records from the left table.
4. Outer (Full) Join: Combined result of both left and right tables.
5. Cross Join: It is same as the Cartesian product.

Number of rows in the first table * Number of tables in the second row.

SQL View

View is a virtual table based on the result-set of an SQL statement.

Contains rows and columns.

Create view vName as Select * from Employee where age < 50;

Select * from vName;

SQL Index

Indexes are used to retrieve data from the database more quickly than otherwise.

The users cannot see the indexes, they are just used to speed up searches/queries.

Create Index Syntax

CREATE INDEX index_name ON table_name;

Subquery

A Subquery is used to return data that will be used in the main query as a condition to further restrict the data to be retrieved.

Correlated Subqueries: They are used to select data from a table referenced in the outer query. The subquery is known as a correlated because the subquery is related to the outer query. In this type of queries, a table alias also called a correlation name must be used.

Inline View: It is not a real view but a subquery in the FROM clause of a SELECT statement.

Order by

Used to sort the data in ascending or descending order, based on one or more columns. Some databases sort the query results in an ascending order by default.

Group by

Used in collaboration with the SELECT statement to arrange identical data into groups.

Stored Procedures

These are nothing but functions/Methods which will hold verified SQL statements...

Inline View

It is not a real view but a subquery in the FROM clause of a SELECT statement.

Syntax:

```
SELECT column_list] FROM (SELECT column_list FROM table_name WHERE  
condition) AS new_name;
```

PROGRAMES & OUTPUT

- Creating table, adding values to table
use employeeedatabase1;
show tables;
select*from employeee;

	Dep_No	Dep_Name	Dep_Location	Project_ID
▶	111	HR	Bangalore	11
	112	IT	Chennai	12
	113	Sales	Pune	15
	114	Research	Hyderabad	13
	115	Teacher	Mumbai	14
*	NULL	NULL	NULL	NULL

	Emp_ID	Emp_Name	Emp_Age	Emp_Dep	Dep_ID	Emp_Location
▶	101	Ravi	32	HR	111	Banglore
	102	Ishitha	25	Sales	113	Chennai
	103	Rupa	23	IT	112	Hyderabad
	104	Kiran	27	Sales	113	Rajpur
	105	Swetha	30	HR	111	Vizag
	106	keeru	24	Teacher	123	palakkad
*	NULL	NULL	NULL	NULL	NULL	NULL

- Adding column to table

	Emp_ID	Emp_Name	Emp_Age	Emp_Dep	Dep_ID	Emp_Location	Emp_Salary	Emp_Gender
▶	101	Ravi	32	HR	111	Banglore	15000	M
	102	Ishitha	25	Sales	113	Chennai	20000	F
	103	Rupa	23	IT	112	Hyderabad	25000	F
	104	Kiran	27	Sales	113	Rajpur	10000	M
	105	Swetha	30	HR	111	Vizag	15000	F
	106	keeru	24	Teacher	123	palakkad	40000	F
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

- Filter Results with Where Statement

select Emp_Name,Emp_Age from employeeedatabase1.employeee where Emp_Location='chennai';

	Emp_Name	Emp_Age
▶	Ishitha	25

- Delete values in Table

	Emp_ID	Emp_Name	Emp_Age	Emp_Dep	Dep_ID	Emp_Location	Emp_Salary	Emp_Gender
▶	101	Ravi	32	HR	111	Banglore	15000	M
	102	Ishitha	25	Sales	113	Chennai	20000	F
	103	Rupa	23	IT	112	Hyderabad	25000	F
	104	Kiran	27	Sales	113	Rajpur	10000	M
	105	Swetha	30	HR	111	Vizag	15000	F

- Order by

select avg(Emp_Salary) AS Avg_Salary,Emp_Location from
 employeeedatabase1.employeee group by Emp_Location having
 Avg_Salary>12000 order by avg(Emp_Salary) desc;

Result Grid		
	Avg_Salary	Emp_Location
▶	40000	palakkad
	25000	Hyderabad
	20000	Chennai
	15000	Banglore
	15000	Vizag

- Inner join

Select

employeee.Emp_Name,employeee.Dep_ID,department.Dep_Location,
 department.Dep_Name

from employeeedatabase1.employeee

inner join employeeedatabase1.department

on employeee.Dep_ID=department.Dep_No;

Result Grid				
	Emp_Name	Dep_ID	Dep_Location	Dep_Name
▶	Ravi	111	Bangalore	HR
	Ishitha	113	Pune	Sales
	Rupa	112	Chennai	IT
	Kiran	113	Pune	Sales
	Swetha	111	Bangalore	HR

- Outer join-Left join

select

employeee.Dep_ID,employeee.Emp_Location,department.Dep_Name,d
epartment.Dep_Location

from employeeedatabase1.employeee

left join employeeedatabase1.department

on employeee.Dep_ID=department.Dep_No;

	Dep_ID	Emp_Location	Dep_Name	Dep_Location
▶	111	Banglore	HR	Bangalore
	113	Chennai	Sales	Pune
	112	Hyderabad	IT	Chennai
	113	Rajpur	Sales	Pune
	111	Vizag	HR	Bangalore
	123	palakkad	NULL	NULL

- Outer join-Right join

select

employeee.Emp_Name,employeee.Dep_ID,department.Dep_Location,d
epartment.Dep_Name

from employeeedatabase1.employeee

right join employeeedatabase1.department

on employeee.Dep_ID=department.Dep_No;

	Emp_Name	Dep_ID	Dep_Location	Dep_Name
▶	Swetha	111	Bangalore	HR
	Ravi	111	Bangalore	HR
	Rupa	112	Chennai	IT
	Kiran	113	Pune	Sales
	Ishitha	113	Pune	Sales
	NULL	NULL	Hyderabad	Research
	NULL	NULL	Mumbai	Teacher

- Cartesian join/ Cross join

SELECT* FROM employeeedatabase1.department CROSS JOIN
employeeedatabase1.employeee where Emp_Salary>12000;

Dep_No	Dep_Name	Dep_Location	Project_ID	Emp_ID	Emp_Name	Emp_Age	Emp_Dep	Dep_ID	Emp_Location	Emp_Salary	Emp_Gender
111	HR	Bangalore	11	106	keeru	24	Teacher	123	palakkad	40000	F
111	HR	Bangalore	11	105	Swetha	30	HR	111	Vizag	15000	F
111	HR	Bangalore	11	103	Rupa	23	IT	112	Hyderabad	25000	F
111	HR	Bangalore	11	102	Ishitha	25	Sales	113	Chennai	20000	F
111	HR	Bangalore	11	101	Ravi	32	HR	111	Banglore	15000	M
112	IT	Chennai	12	106	keeru	24	Teacher	123	palakkad	40000	F
112	IT	Chennai	12	105	Swetha	30	HR	111	Vizag	15000	F
112	IT	Chennai	12	103	Rupa	23	IT	112	Hyderabad	25000	F
112	IT	Chennai	12	102	Ishitha	25	Sales	113	Chennai	20000	F
112	IT	Chennai	12	101	Ravi	32	HR	111	Banglore	15000	M
113	Sales	Pune	15	106	keeru	24	Teacher	123	palakkad	40000	F
113	Sales	Pune	15	105	Swetha	30	HR	111	Vizag	15000	F
113	Sales	Pune	15	103	Rupa	23	IT	112	Hyderabad	25000	F
113	Sales	Pune	15	102	Ishitha	25	Sales	113	Chennai	20000	F
113	Sales	Pune	15	101	Ravi	32	HR	111	Banglore	15000	M
114	Research	Hyderabad	13	106	keeru	24	Teacher	123	palakkad	40000	F
114	Research	Hyderabad	13	105	Swetha	30	HR	111	Vizag	15000	F
114	Research	Hyderabad	13	103	Rupa	23	IT	112	Hyderabad	25000	F
114	Research	Hyderabad	13	102	Ishitha	25	Sales	113	Chennai	20000	F
114	Research	Hyderabad	13	101	Ravi	32	HR	111	Banglore	15000	M
115	Teacher	Mumbai	14	106	keeru	24	Teacher	123	palakkad	40000	F

- Sub-Query using Correlated query and Inline view

SELECT* FROM employeeedatabase1.employeee e1 where 2=(select count(Emp_ID) from employeeedatabase1.employeee e2 where e2.Emp_Salary>e1.Emp_Salary);-----> **Correlation**

	Emp_ID	Emp_Name	Emp_Age	Emp_Dep	Dep_ID	Emp_Location	Emp_Salary	Emp_Gender
▶	102	Ishitha	25	Sales	113	Chennai	20000	F
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

- SELECT*FROM (SELECT SUM(Emp_Salary) AS "Total Salary",Emp_Gender AS "Gender" From employeeedatabase1.employeee GROUP BY Emp_Gender) AS GENDER;.....>Inline view

	Total Salary	Gender
▶	25000	M
	100000	F