MySql Installation

Contents

MySql Installation

1. What is Mysql?

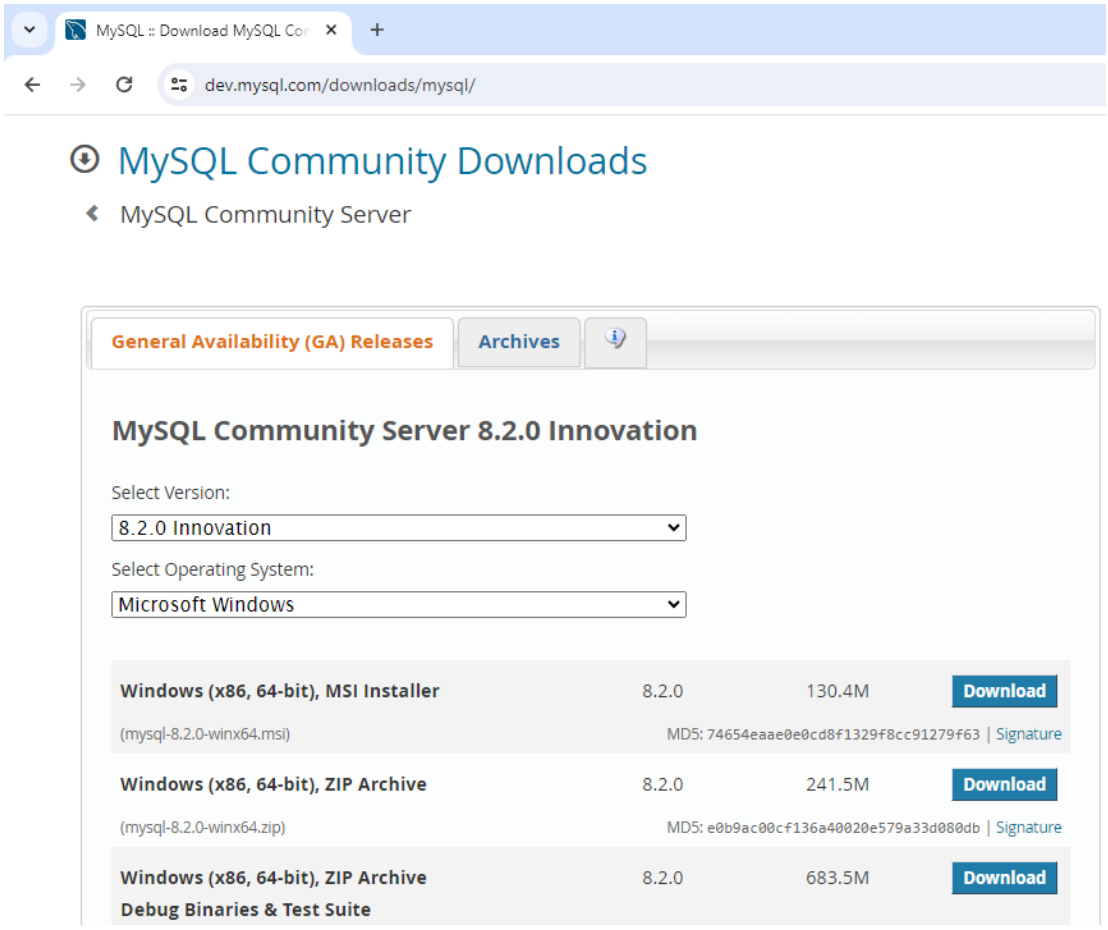
- ✓ MySQL is an open-source relational database management system.
- ✓ SQL is a query language to create, modify and extract data from the relational database, as well as control user access to the database.

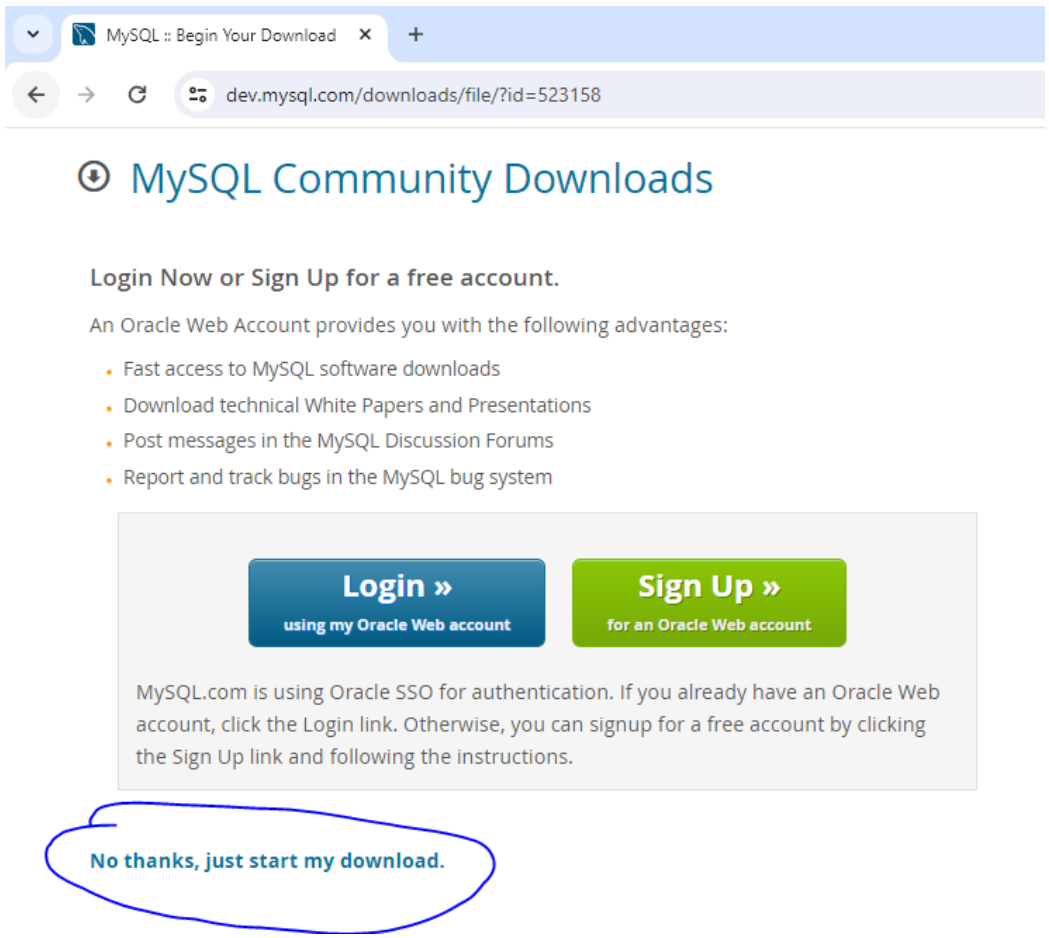
2. Logo

3. Steps to download and install

✓ Click on below url

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

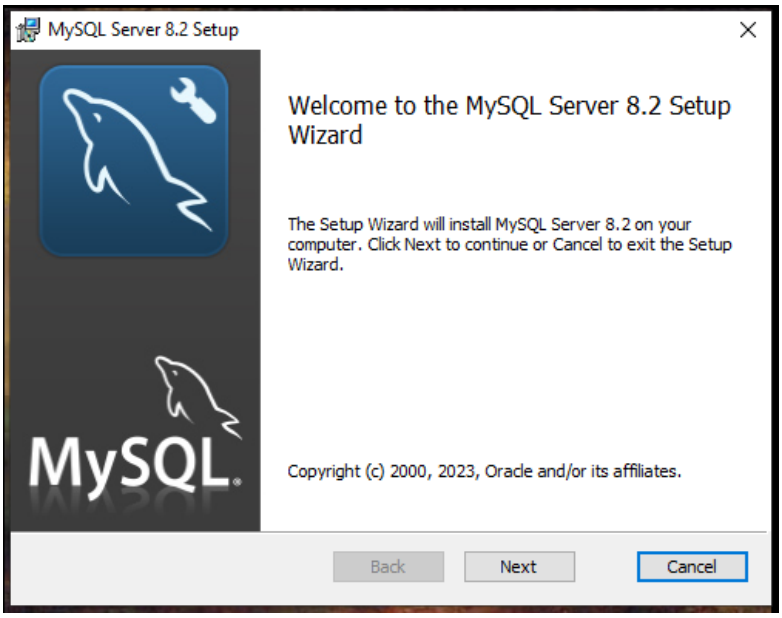


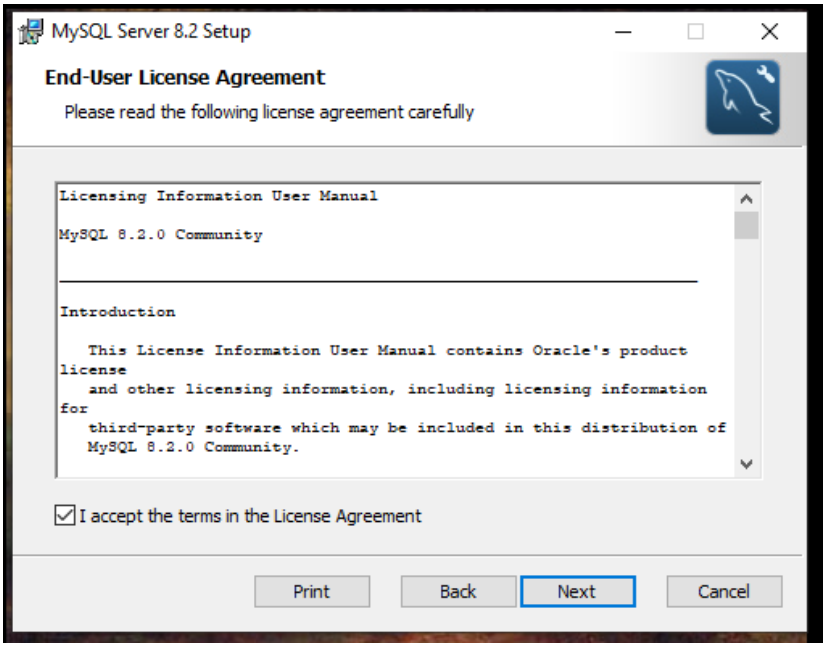


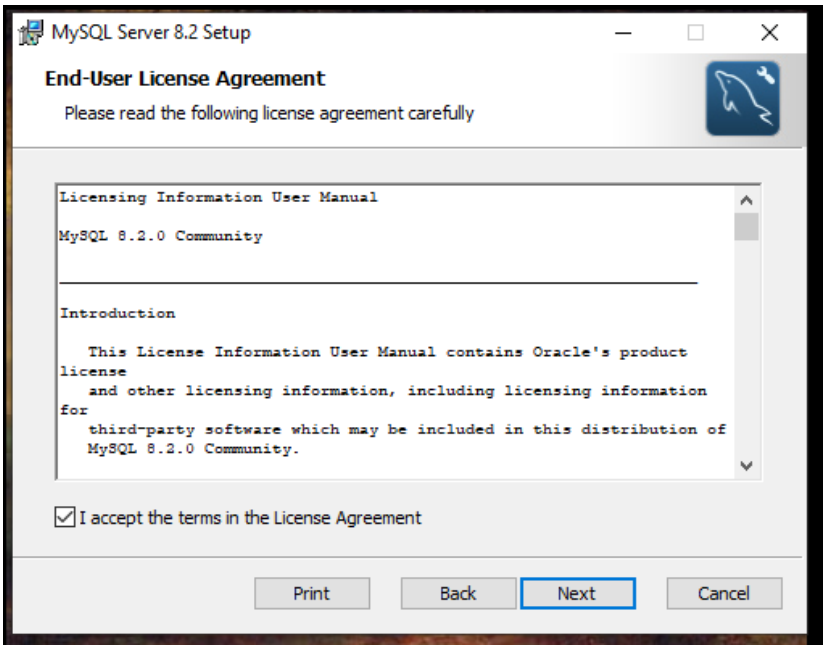
✓ Its .exe file

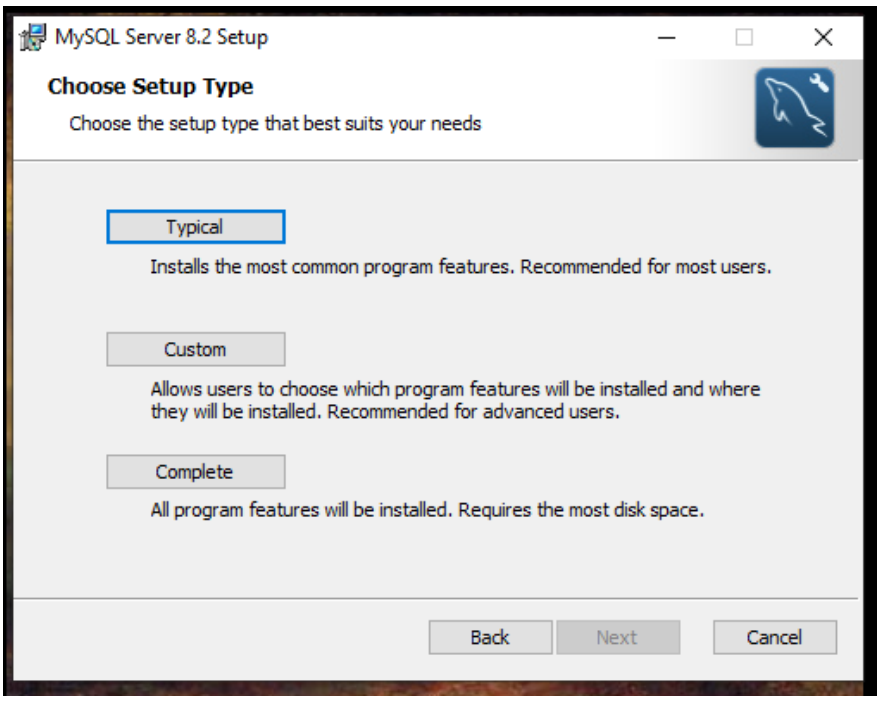
mysql-8.2.0-winx64

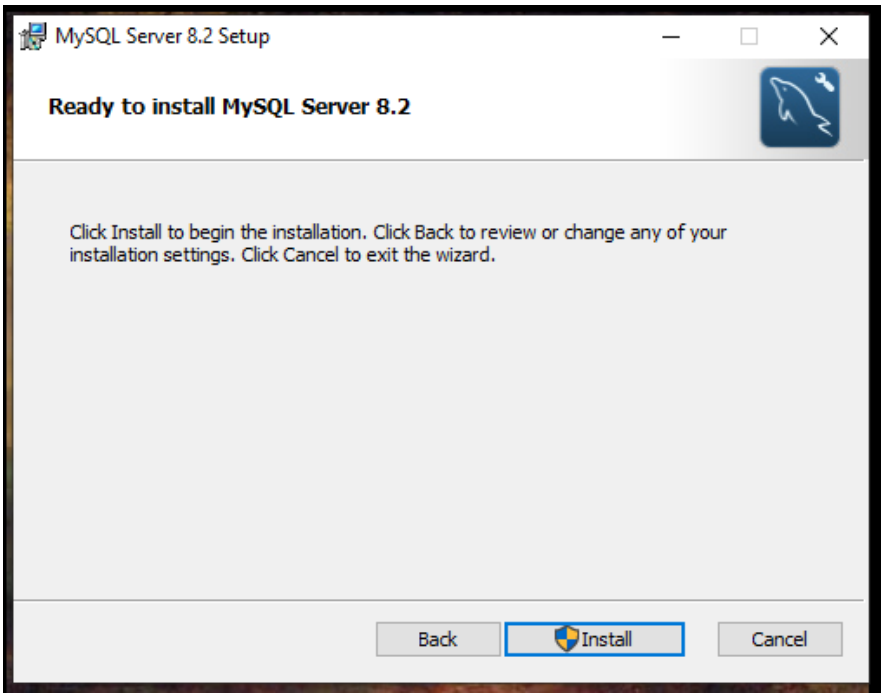
✓ Give double click on software.

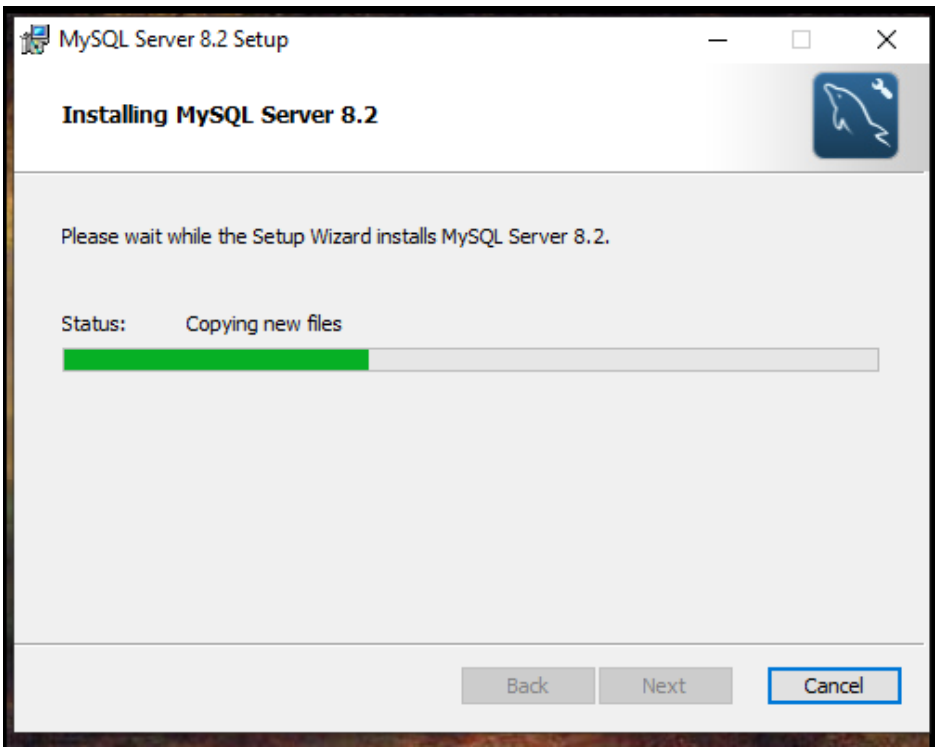




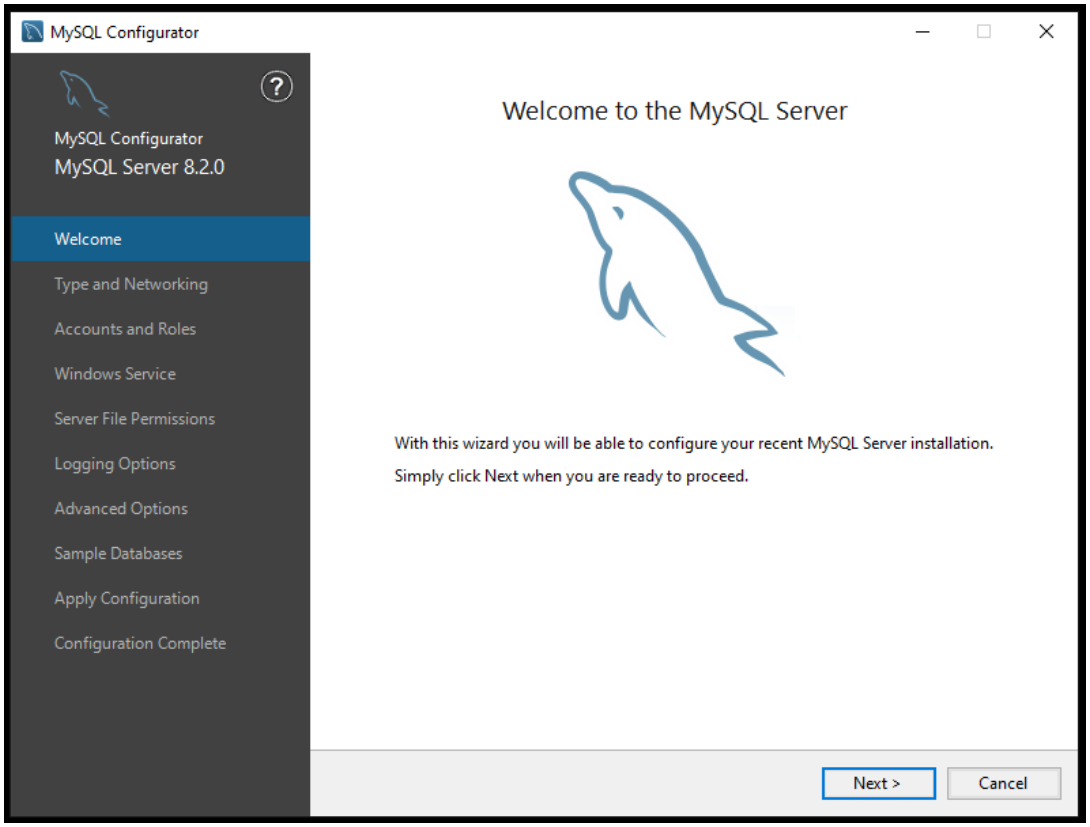

MySQL Server 8.2 Setup
End-User License Agreement
Please read the following license agreement carefully
Licensing Information User Manual
MySQL 8.2.0 Community
Introduction
This License Information User Manual contains Oracle's product license
and other licensing information, including licensing information for
third-party software which may be included in this distribution of
MySQL 8.2.0 Community.
I accept the terms in the License Agreement
Print
Back
Next
Cancel

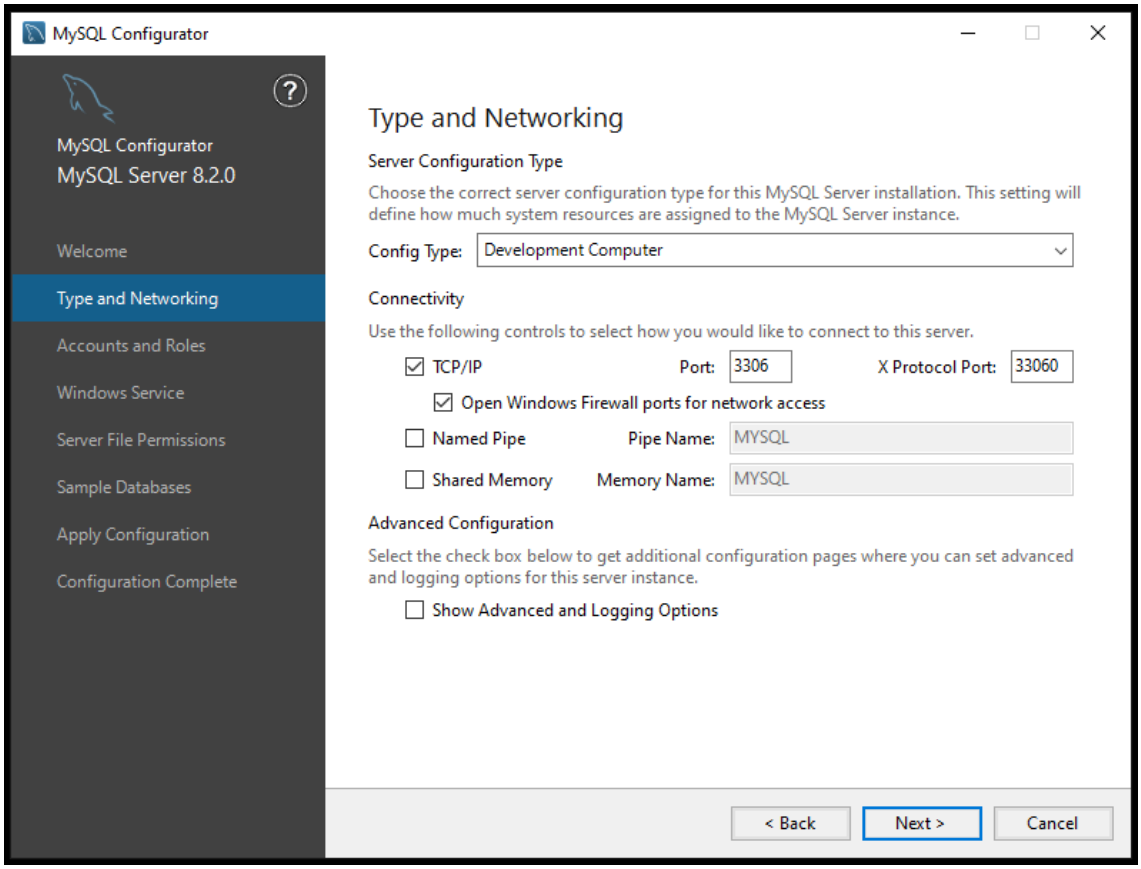

MySQL Server 8.2 Setup
Choose Setup Type
Choose the setup type that best suits your needs
Typical
Installs the most common program features. Recommended for most users.
Custom
Allows users to choose which program features will be installed and where they will be installed. Recommended for advanced users.
Complete
All program features will be installed. Requires the most disk space.
Back
Next
Cancel

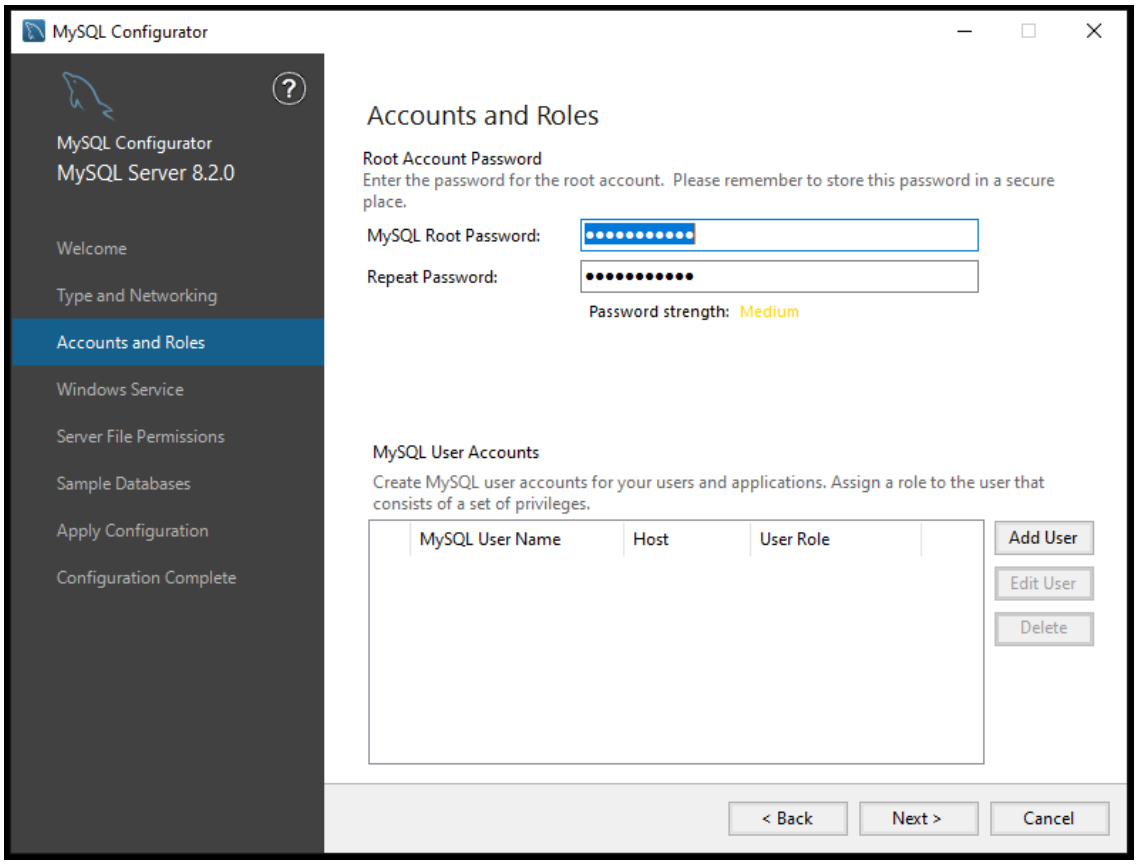

MySQL Server 8.2 Setup
Ready to install MySQL Server 8.2
Click Install to begin the installation. Click Back to review or change any of your installation settings. Click Cancel to exit the wizard.
Back
Install
Cancel

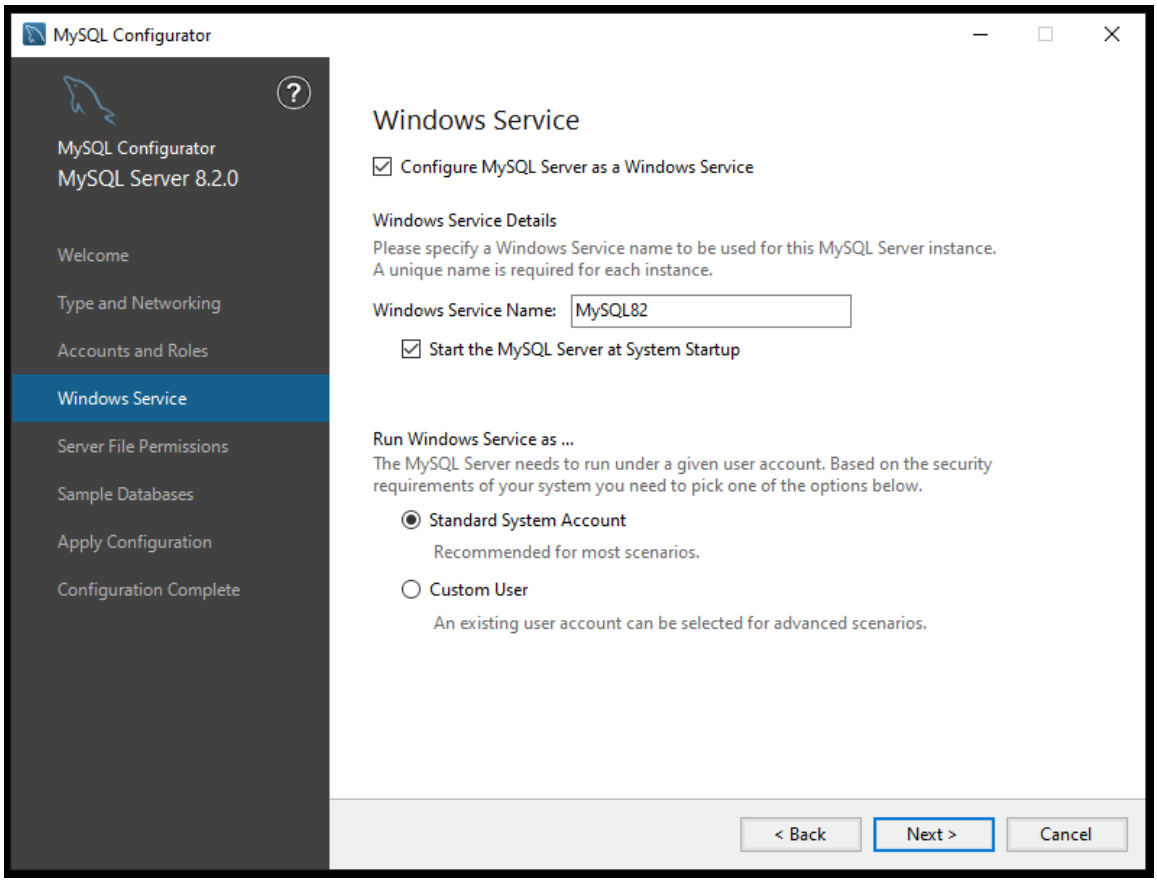

MySQL Server 8.2 Setup
Installing MySQL Server 8.2
Please wait while the Setup Wizard installs MySQL Server 8.2.
Status: Copying new files
Back
Next
Cancel

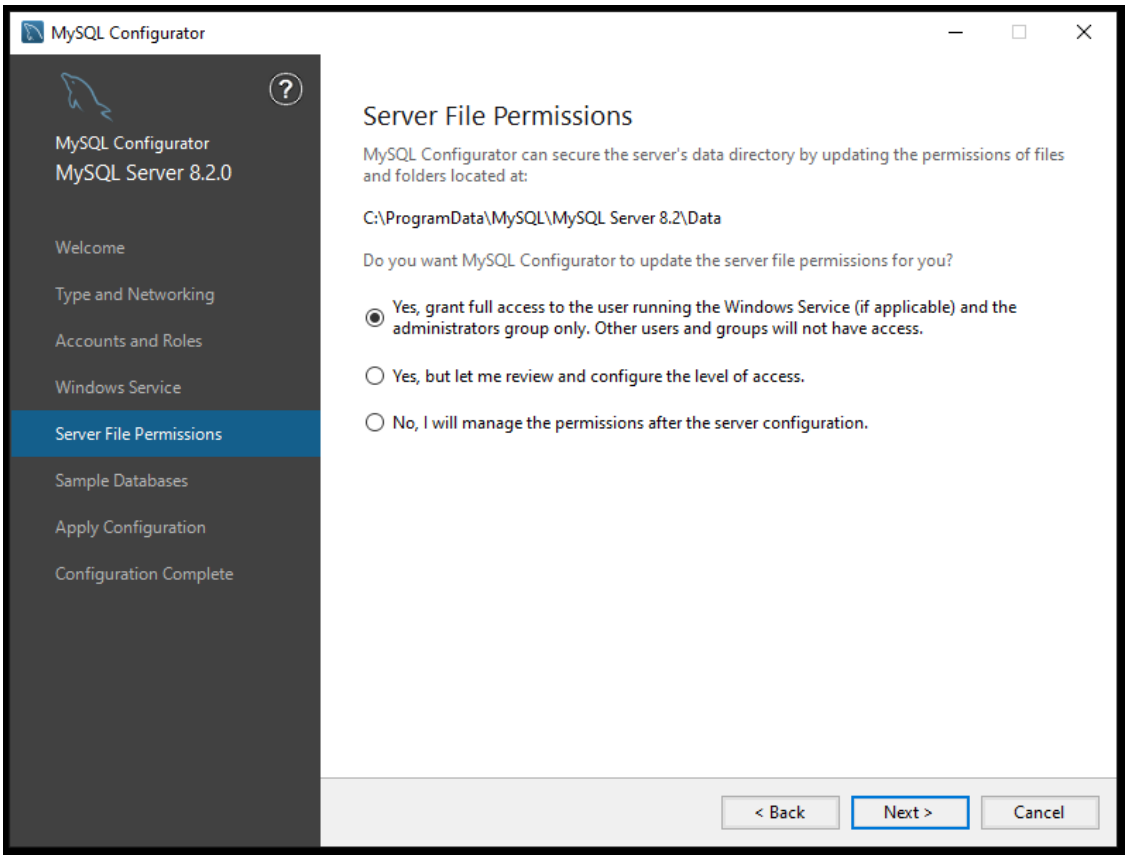
Click on finish

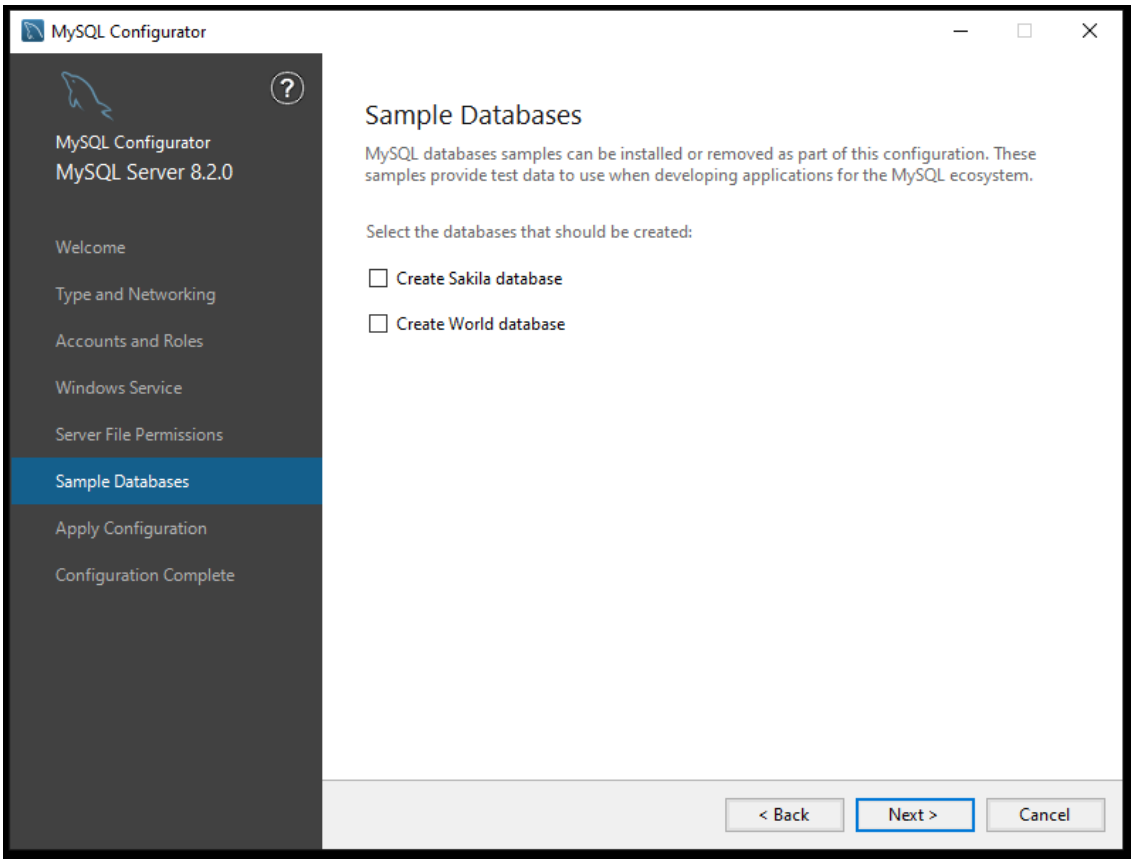


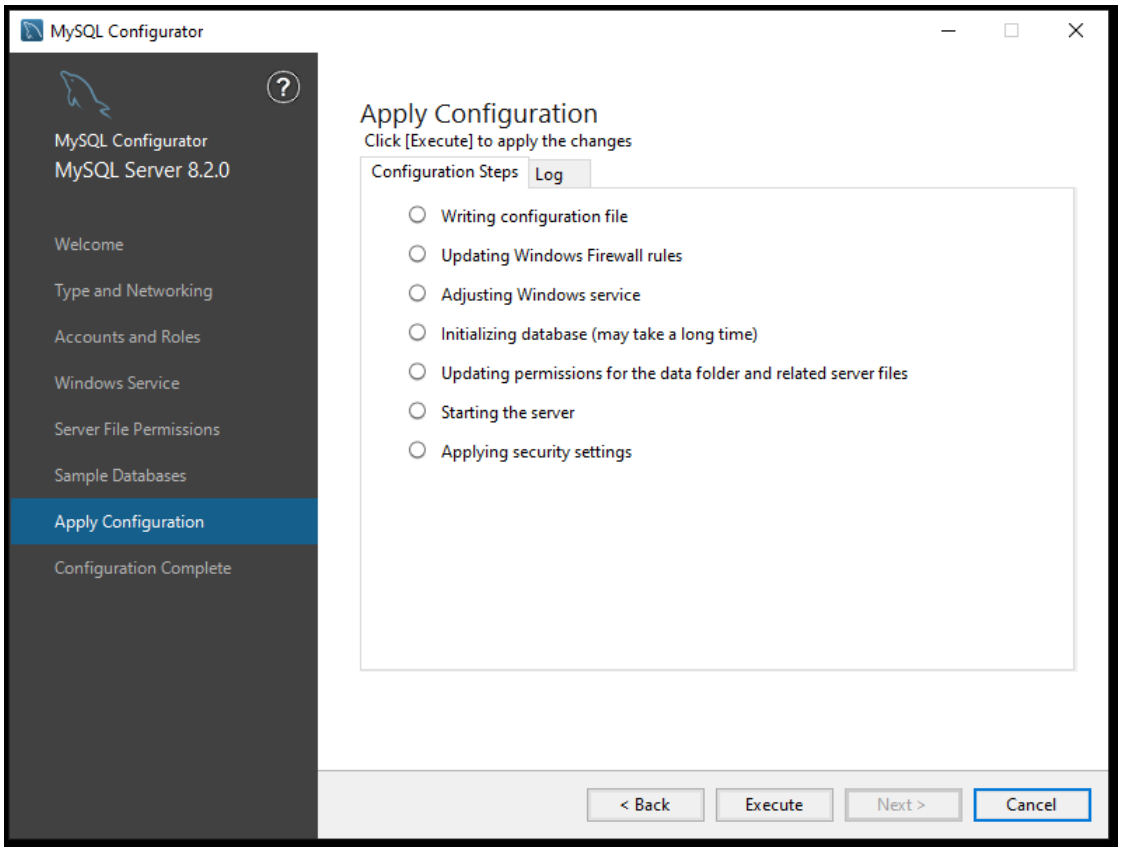

MySQL Configurator
MySQL Configurator
MySQL Server 8.2.0
Welcome
Type and Networking
Accounts and Roles
Windows Service
Server File Permissions
Sample Databases
Apply Configuration
Configuration Complete
Type and Networking
Server Configuration Type
Choose the correct server configuration type for this MySQL Server installation. This setting will define how much system resources are assigned to the MySQL Server instance.
Config Type: Development Computer
Connectivity
Use the following controls to select how you would like to connect to this server.
TCP/IP
Port: 3306
X Protocol Port: 33060
Open Windows Firewall ports for network access
Named Pipe
Pipe Name: MYSQL
Shared Memory
Memory Name: MYSQL
Advanced Configuration
Select the check box below to get additional configuration pages where you can set advanced and logging options for this server instance.
Show Advanced and Logging Options
< Back
Next >
Cancel

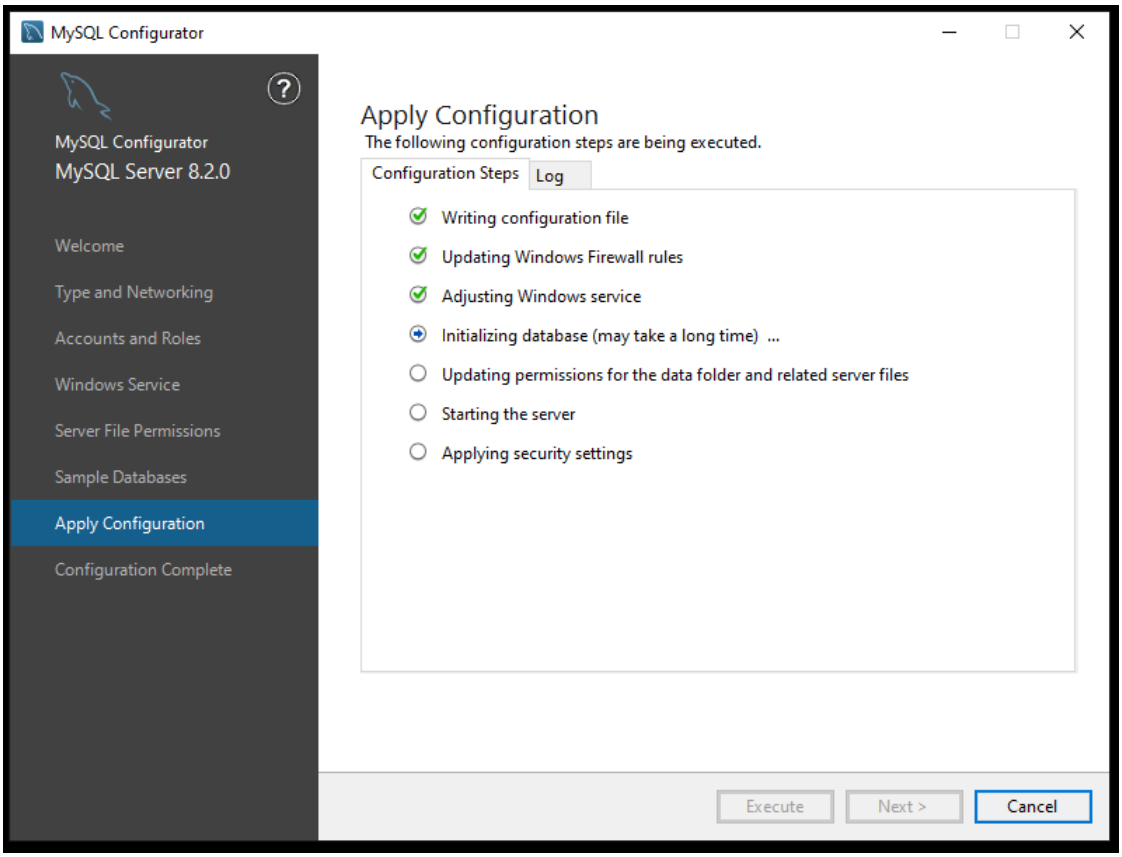

MySQL Configurator
MySQL Configurator
MySQL Server 8.2.0
Welcome
Type and Networking
Accounts and Roles
Windows Service
Server File Permissions
Sample Databases
Apply Configuration
Configuration Complete
Accounts and Roles
Root Account Password
Enter the password for the root account. Please remember to store this password in a secure place.
MySQL Root Password:
Repeat Password:
Password strength: Medium
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

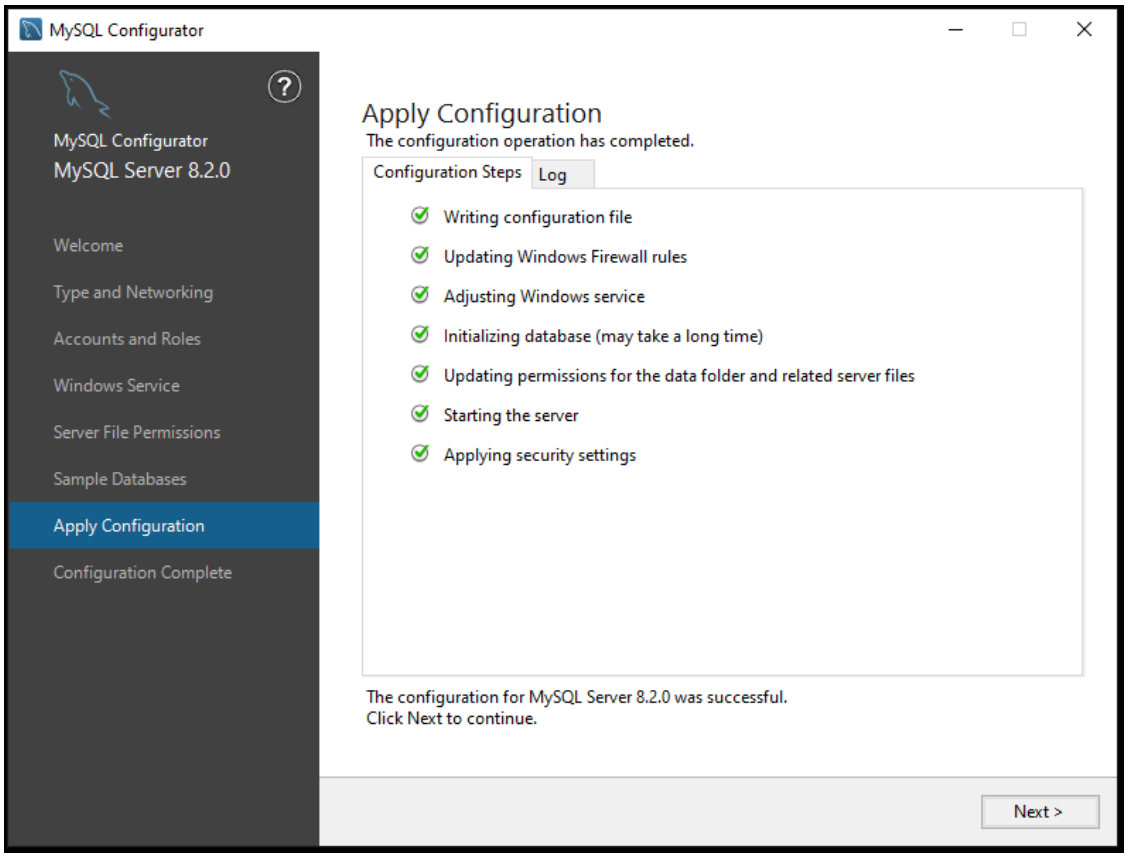

MySQL Configurator
MySQL Configurator
MySQL Server 8.2.0
Welcome
Type and Networking
Accounts and Roles
Windows Service
Server File Permissions
Sample Databases
Apply Configuration
Configuration Complete
Windows Service
Configure MySQL Server as a Windows Service
Windows Service Details
Please specify a Windows Service name to be used for this MySQL Server instance.
A unique name is required for each instance.
Windows Service Name:
MySQL82
Start the MySQL Server at System Startup
Run Windows Service as ...
The MySQL Server needs to run under a given user account. Based on the security requirements of your system you need to pick one of the options below.
Standard System Account
Recommended for most scenarios.
Custom User
An existing user account can be selected for advanced scenarios.
< Back
Next >
Cancel

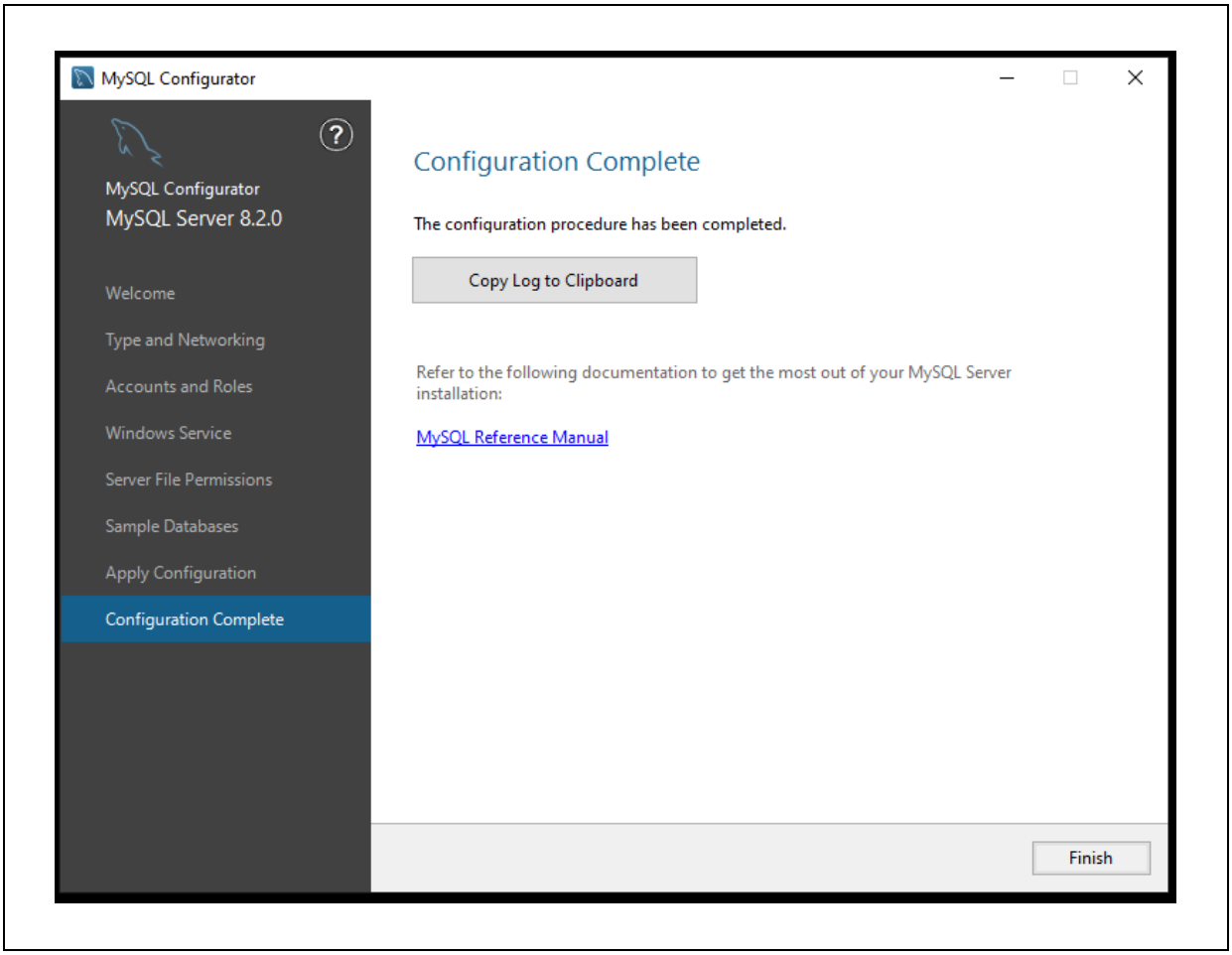

MySQL Configurator
MySQL Configurator
MySQL Server 8.2.0
Welcome
Type and Networking
Accounts and Roles
Windows Service
Server File Permissions
Sample Databases
Apply Configuration
Configuration Complete
Server File Permissions
MySQL Configurator can secure the server's data directory by updating the permissions of files and folders located at:
C:\ProgramData\MySQL\MySQL Server 8.2\Data
Do you want MySQL Configurator to update the server file permissions for you?
Yes, grant full access to the user running the Windows Service (if applicable) and the administrators group only. Other users and groups will not have access.
Yes, but let me review and configure the level of access.
No, I will manage the permissions after the server configuration.
< Back
Next >
Cancel


MySQL Configurator
MySQL Configurator
MySQL Server 8.2.0
Welcome
Type and Networking
Accounts and Roles
Windows Service
Server File Permissions
Sample Databases
Apply Configuration
Configuration Complete
Sample Databases
MySQL databases samples can be installed or removed as part of this configuration. These samples provide test data to use when developing applications for the MySQL ecosystem.
Select the databases that should be created:
Create Sakila database
Create World database
< Back
Next >
Cancel

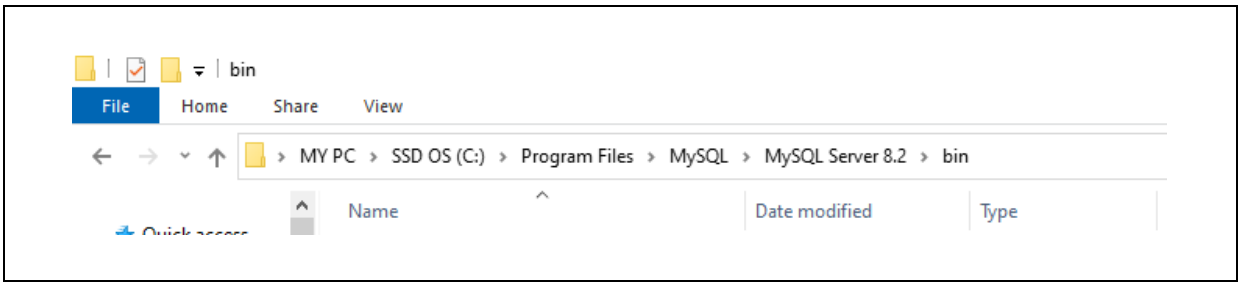

MySQL Configurator
MySQL Configurator
MySQL Server 8.2.0
Welcome
Type and Networking
Accounts and Roles
Windows Service
Server File Permissions
Sample Databases
Apply Configuration
Configuration Complete
Apply Configuration
Click [Execute] to apply the changes
Configuration Steps
Log
Writing configuration file
Updating Windows Firewall rules
Adjusting Windows service
Initializing database (may take a long time)
Updating permissions for the data folder and related server files
Starting the server
Applying security settings
< Back
Execute
Next >
Cancel

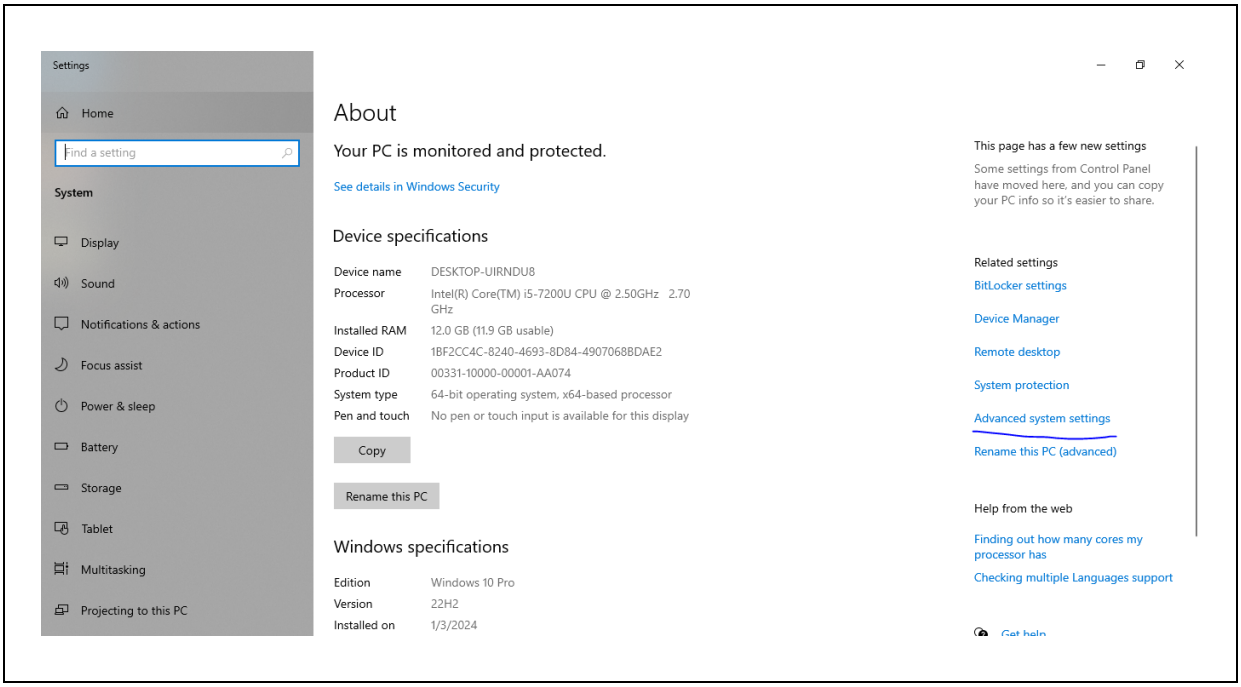

MySQL Configurator
MySQL Configurator
MySQL Server 8.2.0
Welcome
Type and Networking
Accounts and Roles
Windows Service
Server File Permissions
Sample Databases
Apply Configuration
Configuration Complete
Apply Configuration
The following configuration steps are being executed.
Configuration Steps
Log
Writing configuration file
Updating Windows Firewall rules
Adjusting Windows service
Initializing database (may take a long time) ...
Updating permissions for the data folder and related server files
Starting the server
Applying security settings
Execute
Next >
Cancel

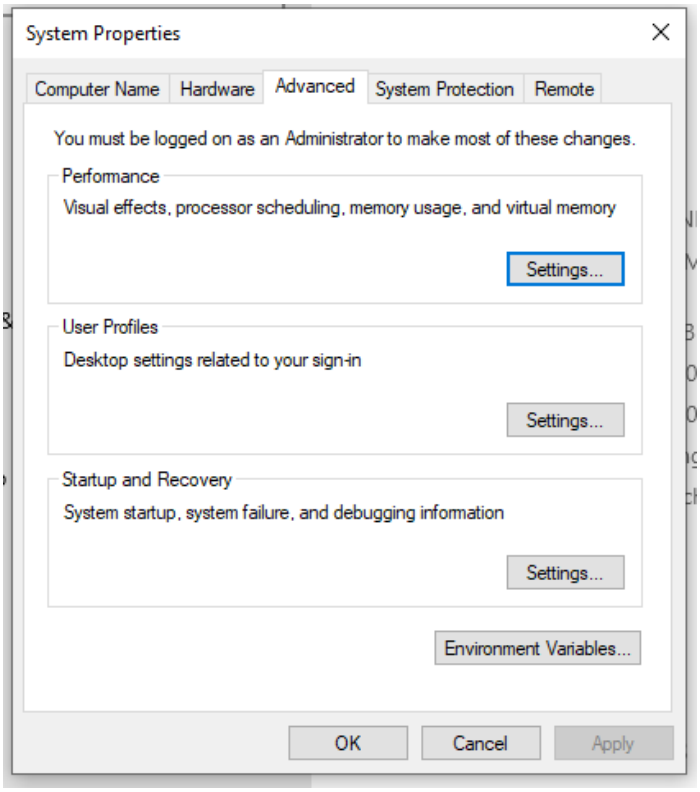

MySQL Configurator
MySQL Configurator
MySQL Server 8.2.0
Welcome
Type and Networking
Accounts and Roles
Windows Service
Server File Permissions
Sample Databases
Apply Configuration
Configuration Complete
Apply Configuration
The configuration operation has completed.
Configuration Steps
Log
Writing configuration file
Updating Windows Firewall rules
Adjusting Windows service
Initializing database (may take a long time)
Updating permissions for the data folder and related server files
Starting the server
Applying security settings
The configuration for MySQL Server 8.2.0 was successful.
Click Next to continue.
Next >

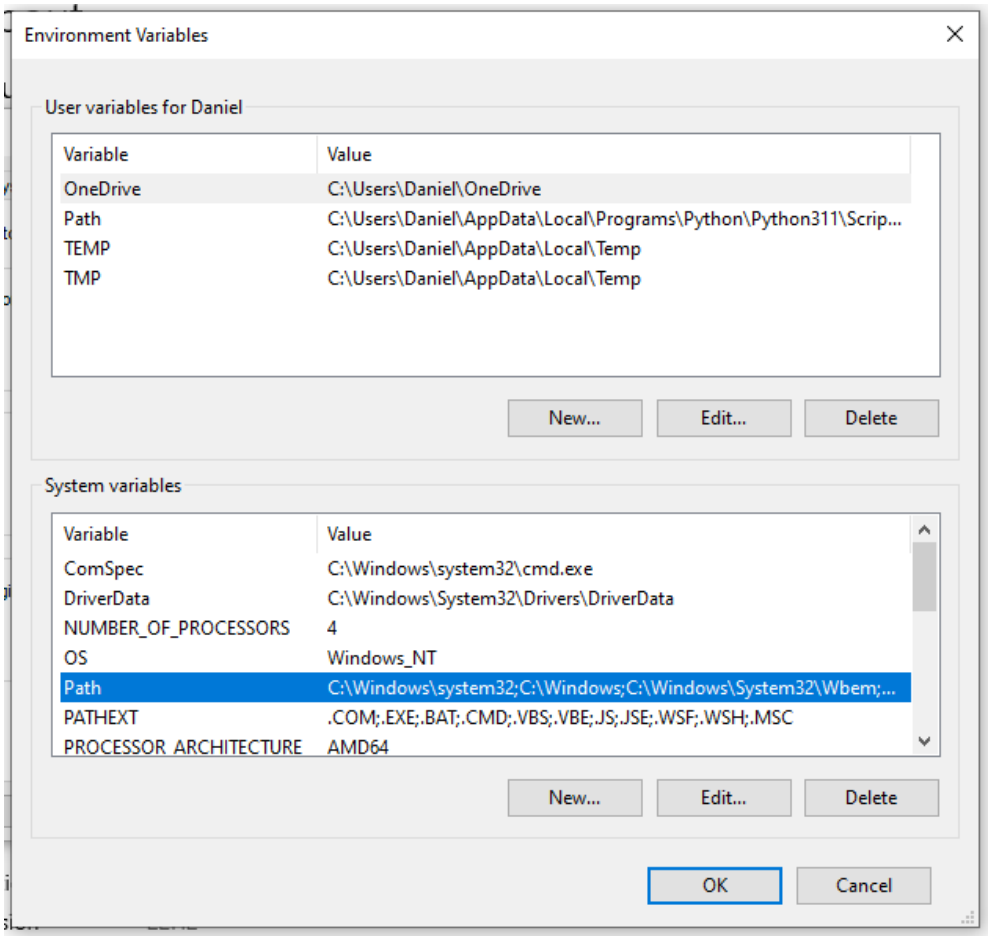


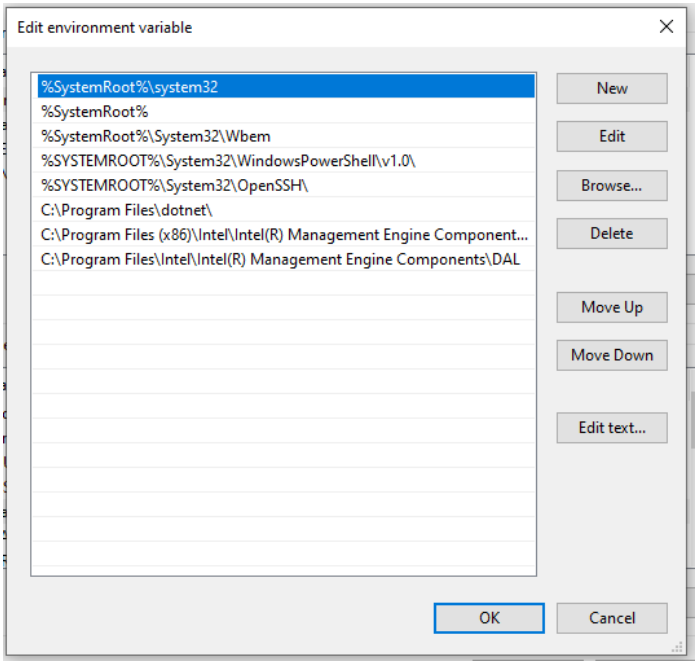
Setting class path

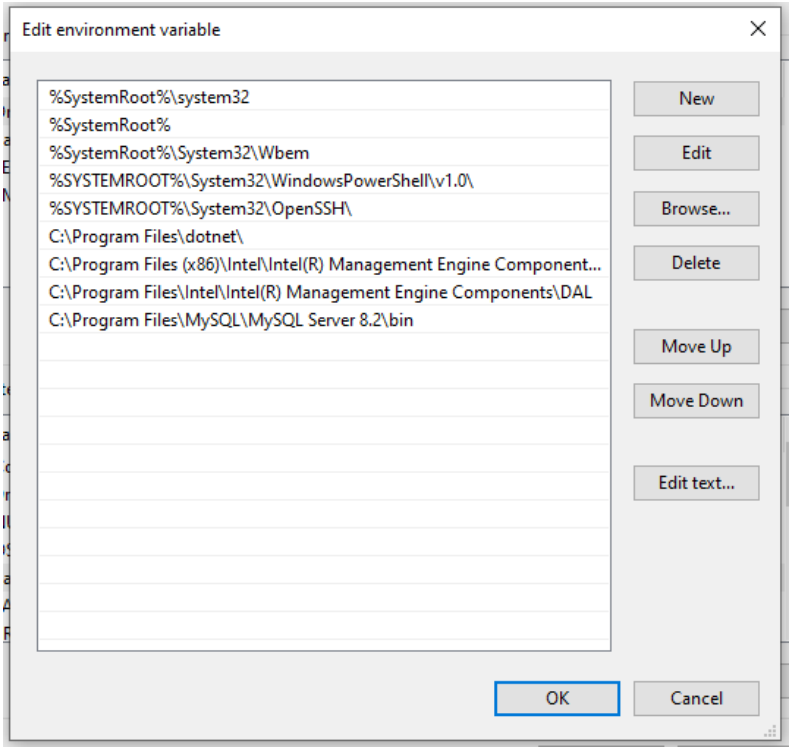



Settings
Home
Find a setting
System
Display
Sound
Notifications & actions
Focus assist
Power & sleep
Battery
Storage
Tablet
Multitasking
Projecting to this PC
About
Your PC is monitored and protected.
See details in Windows Security
Device specifications
Device name DESKTOP-UIRNDU8
Processor Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz 2.70 GHz
Installed RAM 12.0 GB (11.9 GB usable)
Device ID 1BF2CC4C-8240-4693-8D84-4907068BDAE2
Product ID 00331-10000-00001-AA074
System type 64-bit operating system, x64-based processor
Pen and touch No pen or touch input is available for this display
Copy
Rename this PC
Windows specifications
Edition Windows 10 Pro
Version 22H2
Installed on 1/3/2024
This page has a few new settings
Some settings from Control Panel have moved here, and you can copy your PC info so it's easier to share.
Related settings
BitLocker settings
Device Manager
Remote desktop
System protection
Advanced system settings
Rename this PC (advanced)
Help from the web
Finding out how many cores my processor has
Checking multiple Languages support
Get help


System Properties
Computer Name
Hardware
Advanced
System Protection
Remote
You must be logged on as an Administrator to make most of these changes.
Performance
Visual effects, processor scheduling, memory usage, and virtual memory
Settings...
User Profiles
Desktop settings related to your sign-in
Settings...
Startup and Recovery
System startup, system failure, and debugging information
Settings...
Environment Variables...
OK
Cancel
Apply


Environment Variables
User variables for Daniel
Variable
Value
OneDrive
C:\Users\Daniel\OneDrive
Path
C:\Users\Daniel\AppData\Local\Programs\Python\Python311\Scrip...
TEMP
C:\Users\Daniel\AppData\Local\Temp
TMP
C:\Users\Daniel\AppData\Local\Temp
New...
Edit...
Delete
System variables
Variable
Value
ComSpec
C:\Windows\system32\cmd.exe
DriverData
C:\Windows\System32\Drivers\DriverData
NUMBER_OF_PROCESSORS
4
OS
Windows_NT
Path
C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;...
PATHEXT
.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTURE
AMD64
New...
Edit...
Delete
OK
Cancel


Edit environment variable
%SystemRoot%\system32
%SystemRoot%
%SystemRoot%\System32\Wbem
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\
%SYSTEMROOT%\System32\OpenSSH\
C:\Program Files\dotnet\
C:\Program Files (x86)\Intel\Intel(R) Management Engine Component...
C:\Program Files\Intel\Intel(R) Management Engine Components\DAL
New
Edit
Browse...
Delete
Move Up
Move Down
Edit text...
OK
Cancel


Edit environment variable
%SystemRoot%\system32
%SystemRoot%
%SystemRoot%\System32\Wbem
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\
%SYSTEMROOT%\System32\OpenSSH\
C:\Program Files\dotnet\
C:\Program Files (x86)\Intel\Intel(R) Management Engine Component...
C:\Program Files\Intel\Intel(R) Management Engine Components\DAL
C:\Program Files\MySQL\MySQL Server 8.2\bin
New
Edit
Browse...
Delete
Move Up
Move Down
Edit text...
OK
Cancel

1. MySql – Introduction

Contents

1. MySql – Introduction

1. MySql

- ✓ MySQL is a relational database management system
- ✓ This is an open-source
- ✓ We can use for both small and large applications
- ✓ It is very fast, reliable, scalable, and easy to use
- ✓ It was first released in 1995
- ✓ MySQL is developed, distributed, and supported by Oracle Corporation
- ✓ MySQL is software, but SQL is a database language.

2. SQL

- ✓ SQL stands for **S**tructured **Q**uery **L**anguage.
- ✓ SQL is a standard language for accessing and manipulating **database**s.

3. Database

- ✓ A Database is an organized collection of structured data stored in a computer.
- ✓ A database is usually controlled by a Database Management System (DBMS).

4. DBMS

- ✓ **D**ata **B**ase **M**anagement **S**ystem is a software which is used to manage the database.

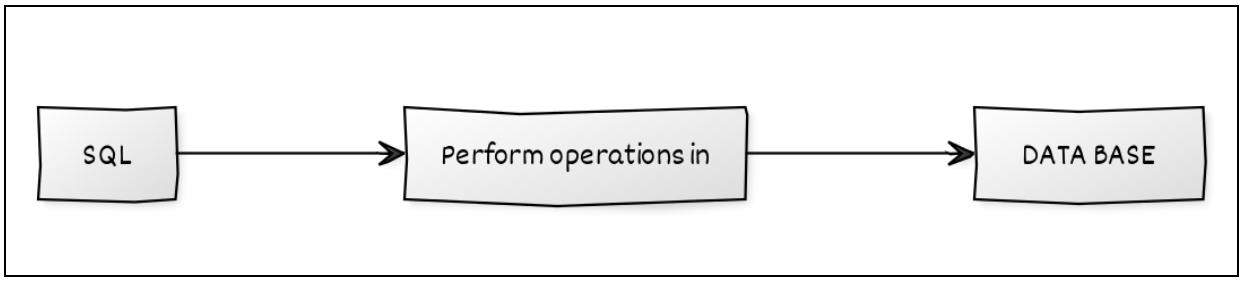
5. RDBMS

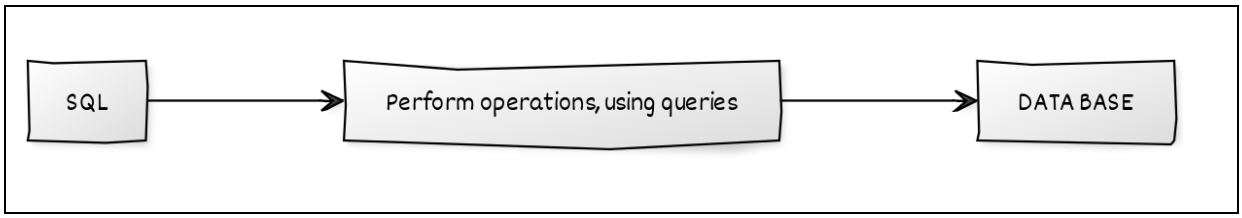
- ✓ **R**elational **D**ata **B**ase **M**anagement **S**ystem is a software which is used to manage the database.
- ✓ RDBMS is the basis for SQL, and for all modern database systems such as Oracle, **MySql**, IBM DB2 & etc.
- ✓ In RDBMS the data used to be stored in the form of tables.
    - Table contains rows and columns.

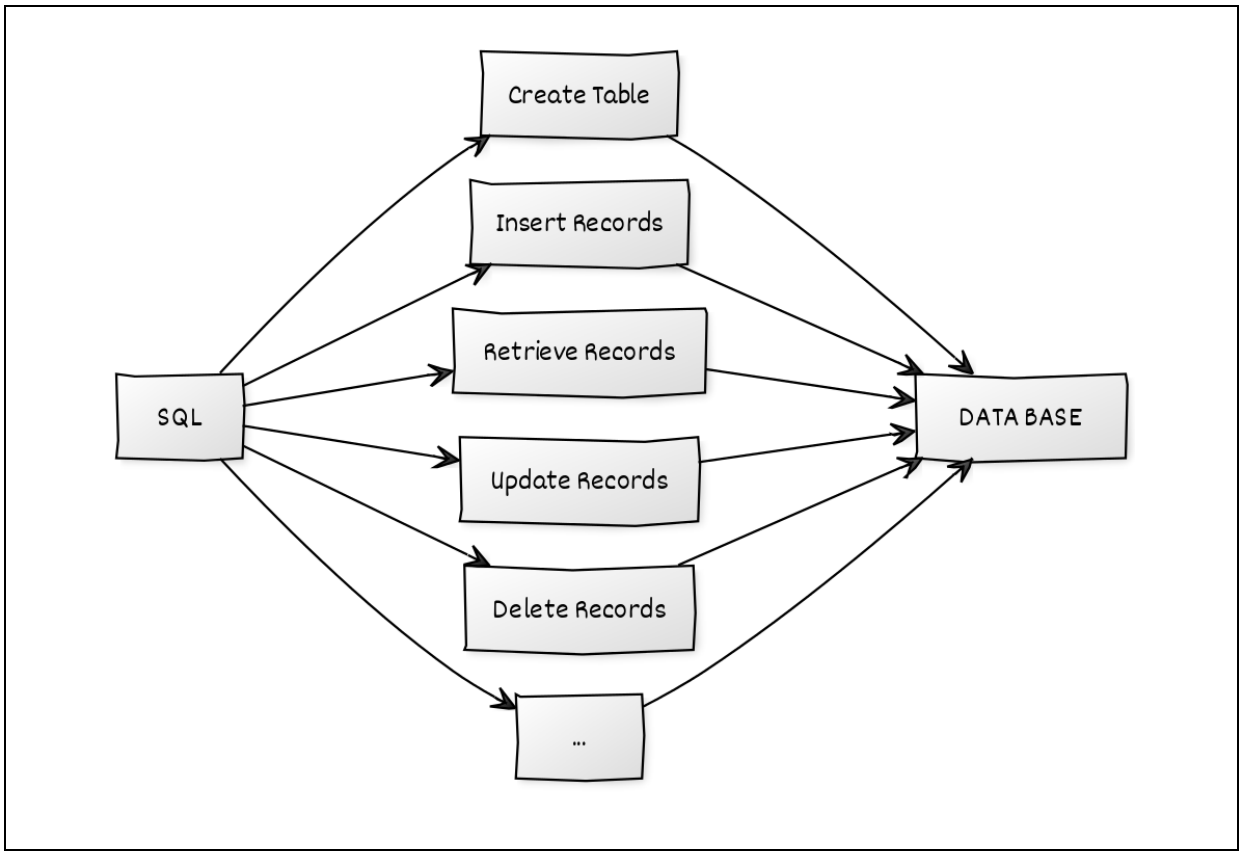
6. Query

- ✓ A query is a request for data from a database table(s)

7. SQL Usage







8. Important SQL Commands

✓ CREATE DATABASE	-	Creates a new database.
✓ CREATE TABLE	-	Creates a new table.
✓ INSERT INTO	-	Inserts new data into a database.
✓ SELECT	-	Extracts data from a database.
✓ UPDATE	-	Updates data in a database.
✓ DROP TABLE	-	Deletes a table.
✓ DELETE	-	Deletes data from a database.
✓ ALTER DATABASE	-	Modifies a database etc.

2. MySql – Database

Contents

2. MySql – Database

1. Database

- ✓ A Database having one or more tables.
- ✓ Table having data in the form of rows and column.

2. Table Example

- ✓ Below is the employees table.

Number	Name	Salary
101	Ranjan	10000
102	Akshay	20000
103	Daniel	30000
104	Veeru	40000

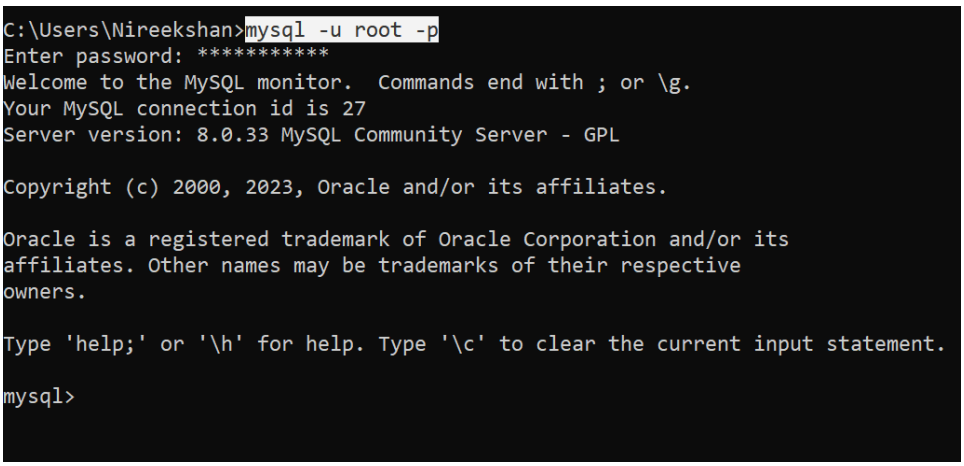
Columns and Rows are

- ✓ Columns are
  - First column name is : Number
  - Second column name is : Name
  - Third column name is : Salary
- ✓ Rows are
  - First row data is : 101 Ranjan 10000
  - Second row data is : 102 Akshay 20000
  - Third row data is : 103 Daniel 10000
  - Forth row data is : 104 Veeru 10000

3. Login to MYSQL

✓ Open a command prompt and enter below command to login into mysql.

Login	Login with valid credentials
Command	mysql –u root -p



Explanation

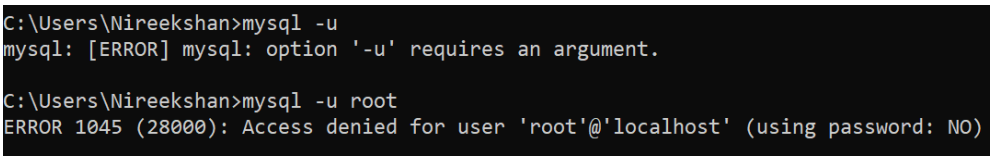
Its successfully connected to MYSQL

Login If login with invalid credentials
Command mysql –u root -p

Explanation

If we enter wrong password then we will get above response

Login If login with invalid command
Command mysql –u
Command mysql –u root



Explanation

Kindly use right command to login mysql

4. Exit from MYSQL

- ✓ We can exit from mysql by using exit command.

Login	Login with valid credentials
Query	To display existing databases.
	mysql> exit
Output	

```
C:\Users\Nireekshan>mysql -u root -p
Enter password: ***********
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 41
Server version: 8.0.33 MySQL Community Server - GPL

Copyright (c) 2000, 2023, 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> exit
Bye
```

5. Show Databases

- ✓ Once successfully login into mysql then we can check total existing databases by using SQL statements.
- ✓ By default, few databases existing in mysql.

Login — Login with valid credentials

Query — To display existing databases.

mysql> show databases;

Output

```
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
4 rows in set (0.00 sec)
```

6. Show schemas

- ✓ We can still use schemas command alternatively to check the databases.

Login Login with valid credentials
Query To display existing databases.

mysql> show schemas;

Output

```
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
4 rows in set (0.00 sec)
```

7. Semicolon

- ✓ Semicolon is the standard way to separate each SQL statement in database systems.
- ✓ By using this we can execute more than one SQL statements.

8. SQL is not case sensitive

- ✓ SQL keywords are not case sensitive.
- ✓ **select** is the same as **SELECT**
- ✓ Both are valid like,
  - show databases; == SHOW DATABASES;

Login Login with right credentials
Query To display existing databases.

mysql> SHOW DATABASES;

Output

```
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
4 rows in set (0.00 sec)
```

9. Create Database

- ✓ Once successfully login into mysql then we can create a database by using SQL statements.

Login	Login with valid credentials
Purpose	Creating danialdb1 Database.
	mysql> create database danieldb1**;** Query OK, 1 row affected (0.01 sec)
Output	danialdb1 database is created successfully

Login	Login with valid credentials
Purpose	Checking all Database.
	mysql> show databases**;**
Output	

```
+--------------------+
| Database           |
+--------------------+
| danieldb1          |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.00 sec)
```

10. Can't create existing database

- ✓ We should not create existing database.
- ✓ If we are creating existing database then we will get an error like,
  - ERROR 1007 (HY000): Can't create database 'danieldb1'; database exists.

Login	Login with valid credentials
Purpose	Creating existing Database.

mysql> create database danieldb1**;**
ERROR 1007 (HY000): Can't create database 'danieldb1'; database exists

Output

11. Creating two databases in one line

- ✓ We can create two databases in one line.
- ✓ We can run two SQL queries by separating semi colon symbol.

Login Login with valid credentials
Purpose Creating existing Database.

mysql> create database danieldb2;create database danieldb3;

Output

```
mysql> create database danieldb2;create database danieldb3;
Query OK, 1 row affected (0.00 sec)

Query OK, 1 row affected (0.00 sec)
```

Login Login with valid credentials
Purpose Creating existing Database.

mysql> show databases;

Output

```
+--------------------+
| Database           |
+--------------------+
| danieldb1          |
| danieldb2          |
| danieldb3          |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
7 rows in set (0.00 sec)
```

12. Drop Database

- ✓ We can delete or drop existing database.

Login Login with valid credentials
Purpose Creating Database.

mysql> drop database danieldb3;
Query OK, 1 row affected (0.01 sec)

Output

```
+--------------------+
| Database           |
+--------------------+
| danieldb1          |
| danieldb2          |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
6 rows in set (0.00 sec)
```

13. Use Database

- ✓ Once successfully created database then we can use the database.

Login	Login with valid credentials
Purpose	Use the created Database.

mysql> use danieldb1**;**
Database changed

Output

```
mysql> use danieldb1;
Database changed
```

3. MySql – Data types

Contents

3. MySql – Data types

1. Data

- ✓ Data is collection of facts.
- ✓ Facts can be values like numbers, strings, alphanumeric, symbols.
- ✓ So, every value having different data type

2. Data type

- ✓ We can create a table in database.
- ✓ Table having data in the form of rows and column.
- ✓ Each column in a table is required to have a name and a data type.
- ✓ Every column can hold the different type of data like, integer, float, character, strings, date and time etc.

3. String data types

- ✓ A group of characters is called as a String.
- ✓ To represent group of characters, String data type helps.

Data Type	Description
✓ CHAR	✓ It can store group or characters in the form of letters, numbers, and special characters as well. ✓ The column lenght is 0 to 255.
✓ VARCHAR	✓ It can store group or characters in the form of letters, numbers, and special characters as well. ✓ The column lenght is 0 to 65535.
✓ We do have other string data types but not much important as part of our sessions. Thanks for understanding.	

4. Numeric data types

- ✓ There are two types of numeric data types.
  - Int/integer : Value without decimal
  - Double : Value with decimal

Data Type	Description
✓ INT	✓ A number without decimal values. ✓ The column lenght is from - 2147483648 to + 2147483647
✓ FLOAT	✓ A number with decimal values.
✓ We do have other numeric data types but not much important as part of our sessions.	

5. Date and Time data types

- ✓ By using these data types we can store Date and time values in table columns.

Data Type	Description
✓ DATE	✓ A date. Format: YYYY-MM-DD. ✓ The supported range is from '1000-01-01' to '9999-12-31'
✓ DATETIME	✓ A date and time combination. Format: YYYY-MM-DD hh:mm:ss. ✓ The supported range is from '1000-01-01 00:00:00' to '9999-12-31 23:59:59'.

4. MySql – DDL and DML

Contents

4. MySql – DDL and DML

1. DDL

- ✓ The full form of DDL is **D**ata **D**efinition **L**anguage.
- ✓ By using this we can create the database structure or schema.
- ✓ DDL command are like,
  - CREATE
  - ALTER
  - DROP
  - TRUNCATE
  - COMMENT
  - RENAME, etc.
- ✓ DDL commands are auto-committed. So, the changes are saved in the database permanently.
- ✓ DDL statements affect the whole table.

2. DML

- ✓ The full form of DDL is **D**ata **M**anipulation **L**anguage.
- ✓ By using this we can change the data which is stored in the database.
- ✓ DML command are like,
  - INSERT
  - UPDATE
  - DELETE
  - MERGE & etc.
- ✓ DML commands used to populate and manipulate database
- ✓ It adds or updates the row of the table
- ✓ DML effects one or more rows.

5. MySql – Table

Contents

5. MySql – Table

1. Table

- ✓ We can create table in mysql.
- ✓ The purpose of creating table to store the data.
- ✓ Table having group of rows and columns.
- ✓ Below is the employees table.

2. Table Example

- ✓ Below is the employees table.

Number	Name	Salary
101	Ranjan	10000
102	Akshay	20000
103	Daniel	30000
104	Veeru	40000

3. Create a table

- ✓ We can create a table by using create table command
- ✓ While creating table we need to specify the name of the table.
- ✓ The table name should be unique in a database.
- ✓ We can use IF NOT EXISTS command (optional) while creating table.
- ✓ This command checks if the table exists in the database or not, if table exists then MySQL ignore to create new table.

Login Login with valid credentials

Query To create table in danialdb1 database.

```
mysql> create table Persons(
        PersonID int,
        LastName varchar(255),
        FirstName varchar(255),
        Address varchar(255),
        City varchar(255)
);
```

Output

```
mysql> CREATE TABLE Persons(
    ->     PersonID int,
    ->     LastName varchar(255),
    ->     FirstName varchar(255),
    ->     Address varchar(255),
    ->     City varchar(255)
    -> );
Query OK, 0 rows affected (0.02 sec)
```

Explanation

- ✓ The PersonID column is of type int.
- ✓ The LastName, FirstName, Address, and City columns are varchar type and the maximum length for these fields is 255 characters.
- ✓ Currently **Persons** table is **empty** table

4. Show tables

✓ We can check created tables in database by using show tables command.

Login — Login with valid credentials

Query — To display the tables in danialdb1 database.

```
mysql> show tables;
```

Output

```
+---------------------+
| Tables_in_danialdb1 |
+---------------------+
| persons             |
+---------------------+
1 row in set (0.00 sec)
```

5. Drop table

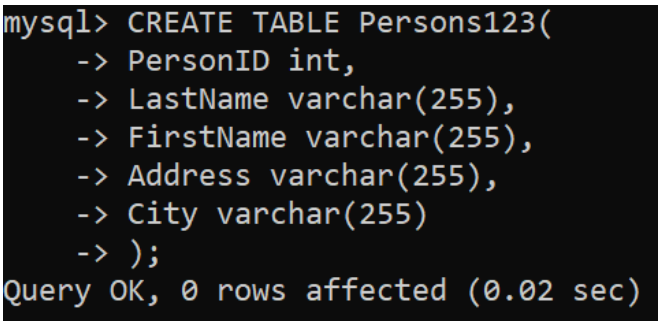
- ✓ We can drop/delete a table by using drop table command.
- ✓ Once we deleted the table then, that table is not available in database.
- ✓ Let's create a dummy table like Persons123 and will drop the same.

Login Login with valid credentials

Query To create table in danialdb1 database.

```
mysql> create table Persons123(
        PersonID int,
        LastName varchar(255),
        FirstName varchar(255),
        Address varchar(255),
        City varchar(255)
);
```

Output



Login Login with valid credentials
Query Display all tables in danialdb1 database.

mysql> show tables;

Output

```
+---------------------+
| Tables_in_danialdb1 |
+---------------------+
| persons             |
| persons123          |
+---------------------+
2 rows in set (0.00 sec)
```

Login Login with valid credentials
Query To drop Persons123 table from danialdb1 database.

mysql> drop table Persons123;
Query OK, 0 rows affected (0.01 sec)

mysql> show tables;

Output

```
+---------------------+
| Tables_in_danialdb1 |
+---------------------+
| persons             |
+---------------------+
1 row in set (0.00 sec)
```

6. Rename the table

- ✓ We can rename the existing table by using rename or alter commands.
- ✓ Let's create a dummy table like Persons111 and will rename the same.

Login Login with valid credentials
Query To create Persons111 table in danialdb1 database.

```
mysql> create table Persons111(
        PersonID int,
        LastName varchar(255)
);
```

Output

```
mysql> CREATE TABLE Persons111(
    ->     PersonID int,
    ->     LastName varchar(255)
    -> );
Query OK, 0 rows affected (0.01 sec)
```

Login Login with valid credentials
Query Display all tables in danialdb1 database.

```
mysql> show tables;
```

Output

```
+---------------------+
| Tables_in_danialdb1 |
+---------------------+
| persons             |
| persons111          |
+---------------------+
2 rows in set (0.00 sec)
```

Login Login with valid credentials

Query Rename table and display all tables in danialdb1 database.

```
mysql> rename table Persons111 to Persons222;
Query OK, 0 rows affected (0.01 sec)

mysql> show tables;
```

Output

```
+---------------------+
| Tables_in_danialdb1 |
+---------------------+
| persons             |
| persons222          |
+---------------------+
2 rows in set (0.00 sec)
```

Login Login with valid credentials

Query Alter table and display all tables in danialdb1 database.

```
mysql> alter table Persons222 rename Persons333;
Query OK, 0 rows affected (0.01 sec)

mysql> show tables;
```

Output

```
+---------------------+
| Tables_in_danialdb1 |
+---------------------+
| persons             |
| persons333          |
+---------------------+
2 rows in set (0.00 sec)
```

7. Insert data into table

- ✓ Once we created the table then we can insert data into the table.
  - We can insert data into table by specifying column names and values.
  - Without specifying column names also we can insert the data into a table. We ensure that the **order of** the values and columns should be same.

Login Login with valid credentials

Query To insert data into Persons table in danialdb1 database.

```
mysql> insert into Persons(PersonID, LastName, FirstName,
Address, City) values(101, 'Danial', 'K', 'Near to Data Science Area',
'Hyderabad');

Query OK, 1 row affected (0.00 sec)

mysql> insert into Persons(PersonID, LastName, FirstName,
Address, City) values(102, 'Nireekshan', 'D', 'Near to AI Theatre',
'Bangalore');

Query OK, 1 row affected (0.00 sec)

mysql> insert into Persons values(103, 'Ranjan', 'M', 'Near to
Python Theatre', 'Hyderabad');

Query OK, 1 row affected (0.00 sec)

mysql> insert into Persons values(104, 'Prasad', 'K', 'Near to
Python Road', 'Hyderabad');

Query OK, 1 row affected (0.00 sec)
```

Output

Successfully data inserted into Persons table.

8. Select the table

- ✓ We can see the data in a table by using select command.
- ✓ While using select command, table name and column names should be match otherwise we will get an error.

Login — Login with valid credentials

Query — To select the data from Persons table in danialdb1 database.

```
mysql> select * from Persons;
```

Output

```
+----------+------------+-----------+--------------------------+-----------+
| PersonID | LastName   | FirstName | Address                  | City      |
+----------+------------+-----------+--------------------------+-----------+
|      101 | Danial     | K         | Near to Data Science Area | Hyderabad |
|      102 | Nireekshan | D         | Near to AI Theatre       | Bangalore |
|      103 | Ranjan     | M         | Near to Python Theatre   | Hyderabad |
|      104 | Prasad     | K         | Near to Python Road      | Hyderabad |
+----------+------------+-----------+--------------------------+-----------+
4 rows in set (0.00 sec)
```

Login — Login with valid credentials

Query — Ensure that table name should exists

```
mysql> select * from Persons444;
```

Output

```
ERROR 1146 (42S02): Table 'danialdb1.persons444' doesn't exist
```

8.1. Select few columns from the table

✓ We can see the data in a table by using select command.

Login — Login with valid credentials

Query — To select few columns from Persons table in danialdb1 database.

```
mysql> select PersonID, LastName from Persons;
```

Output

```
+----------+------------+
| PersonID | LastName   |
+----------+------------+
|      101 | Danial     |
|      102 | Nireekshan |
|      103 | Ranjan     |
|      104 | Prasad     |
+----------+------------+
4 rows in set (0.00 sec)
```

Login Login with valid credentials
Query To select few columns from Persons table in danialdb1 database.

mysql> select PersonID, LastName, City from Persons;

Output

```
+----------+------------+-----------+
| PersonID | LastName   | City      |
+----------+------------+-----------+
|      101 | Danial     | Hyderabad |
|      102 | Nireekshan | Bangalore |
|      103 | Ranjan     | Hyderabad |
|      104 | Prasad     | Hyderabad |
+----------+------------+-----------+
4 rows in set (0.00 sec)
```

Login Login with valid credentials
Query Ensure that column name should exists in table

mysql> select PersonID, LastName, City111 from Persons;

Output

ERROR 1054 (42S22): Unknown column 'city111' in 'field list'

9. Select Distinct

- ✓ By default select statement will returns all values including duplicates.
- ✓ If we wanted to display unique values then we can use select distinct statement.

Login Login with valid credentials
Query To select city column values from Persons table

mysql> select city from Persons;

Output

```
+-----------+
| city      |
+-----------+
| Hyderabad |
| Bangalore |
| Hyderabad |
| Hyderabad |
+-----------+
4 rows in set (0.00 sec)
```

Login Login with valid credentials
Query To select distinct values from city column in Persons table

mysql> select distinct city from persons;

Output

```
+-----------+
| city      |
+-----------+
| Hyderabad |
| Bangalore |
+-----------+
```

10. Alter table

- ✓ Once we created the table then we can apply do modifications based on requirement on existing table.
- ✓ We can add, delete, or modify columns in an existing table.
- ✓ We can add and drop various constraints on an existing table.

10.1. Add column in table

✓ We can add column to existing table.

Login Login with valid credentials
Query Add email column to Persons table

```
mysql> alter table persons
    -> add email varchar(255);
```

Output

```
Query OK, 0 rows affected (0.06 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

Login Login with valid credentials
Query Display the persons table

```
mysql> select * from persons;
```

Output

```
+----------+------------+-----------+---------------------------+-----------+-------+
| PersonID | LastName   | FirstName | Address                   | City      | email |
+----------+------------+-----------+---------------------------+-----------+-------+
|      101 | Danial     | K         | Near to Data Science Area | Hyderabad | NULL  |
|      102 | Nireekshan | D         | Near to AI Theatre        | Bangalore | NULL  |
|      103 | Ranjan     | M         | Near to Python Theatre    | Hyderabad | NULL  |
|      104 | Prasad     | K         | Near to Python Road       | Hyderabad | NULL  |
+----------+------------+-----------+---------------------------+-----------+-------+
4 rows in set (0.00 sec)
```

10.2. Drop column from table

✓ Based on requirement we can drop column from table.

Login Login with valid credentials
Query Drop email column from Persons table

```
mysql> alter table persons
    -> drop column email;
```

Output

```
Query OK, 0 rows affected (0.01 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

Login Login with valid credentials
Query Display persons table.

```
mysql> select * from persons;
```

Output

```
+----------+------------+-----------+---------------------------+-----------+
| PersonID | LastName   | FirstName | Address                   | City      |
+----------+------------+-----------+---------------------------+-----------+
|      101 | Danial     | K         | Near to Data Science Area | Hyderabad |
|      102 | Nireekshan | D         | Near to AI Theatre        | Bangalore |
|      103 | Ranjan     | M         | Near to Python Theatre    | Hyderabad |
|      104 | Prasad     | K         | Near to Python Road       | Hyderabad |
+----------+------------+-----------+---------------------------+-----------+
4 rows in set (0.00 sec)
```

10.3. Modify column name in table

✓ We can modify column name in existing table.

Login Login with valid credentials
Query Modify city column to location in persons table

```
mysql> alter table persons
    -> rename column city to location;
```

Output

```
Query OK, 0 rows affected (0.01 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

Login Login with valid credentials
Query Display persons table.

```
mysql> select * from persons;
```

Output

```
+----------+------------+-----------+--------------------------+-----------+
| PersonID | LastName   | FirstName | Address                  | location  |
+----------+------------+-----------+--------------------------+-----------+
|      101 | Danial     | K         | Near to Data Science Area | Hyderabad |
|      102 | Nireekshan | D         | Near to AI Theatre        | Bangalore |
|      103 | Ranjan     | M         | Near to Python Theatre    | Hyderabad |
|      104 | Prasad     | K         | Near to Python Road       | Hyderabad |
+----------+------------+-----------+--------------------------+-----------+
4 rows in set (0.00 sec)
```

6. MySql – Where clause, Operators

Contents

6. MySql – Where clause, Operators

1. Where clause

- ✓ Based on condition we can filter and get matched rows from table.

Login — Login with valid credentials

Query — Get matching rows by using where clause.

```
mysql> select * from persons
    -> where LastName = 'Danial';
```

Output

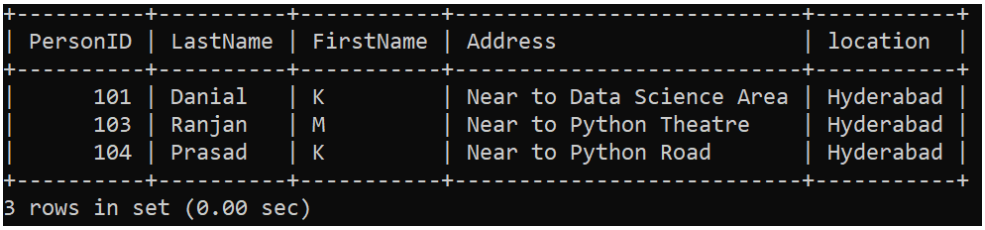
```
+----------+----------+-----------+--------------------------+-----------+
| PersonID | LastName | FirstName | Address                  | location  |
+----------+----------+-----------+--------------------------+-----------+
|      101 | Danial   | K         | Near to Data Science Area | Hyderabad |
+----------+----------+-----------+--------------------------+-----------+
1 row in set (0.00 sec)
```

Login Login with valid credentials

Query where clause with equal operator

```
mysql> select * from persons
    -> where location = 'hyderabad';
```

Output



```
+----------+----------+-----------+--------------------------+-----------+
| PersonID | LastName | FirstName | Address                  | location  |
+----------+----------+-----------+--------------------------+-----------+
|      101 | Danial   | K         | Near to Data Science Area | Hyderabad |
|      103 | Ranjan   | M         | Near to Python Theatre    | Hyderabad |
|      104 | Prasad   | K         | Near to Python Road       | Hyderabad |
+----------+----------+-----------+--------------------------+-----------+
3 rows in set (0.00 sec)
```

2. Where clause with operators

✓ Below operators we can use with where clause.

Operator	Description
=	Equal
>	Greater than
<	Less than
>=	Greater than or Equal
<=	Less than or Equal
<>	Not Equal, as !=
between	Between a certain range
in	To check multiple possible values in column

2.1. Where clause with greater than operator

- ✓ We can apply where clause with greater than operator on specific column.
- ✓ This returns the matched records from the table.

Login Login with valid credentials

Query where clause with greater than operator

```
mysql> select * from persons
    -> where PersonID > 101;
```

Output

```
+----------+------------+-----------+------------------------+-----------+
| PersonID | LastName   | FirstName | Address                | location  |
+----------+------------+-----------+------------------------+-----------+
|      102 | Nireekshan | D         | Near to AI Theatre     | Bangalore |
|      103 | Ranjan     | M         | Near to Python Theatre | Hyderabad |
|      104 | Prasad     | K         | Near to Python Road    | Hyderabad |
+----------+------------+-----------+------------------------+-----------+
3 rows in set (0.00 sec)
```

2.2. Where clause with between and operators

- ✓ We can apply where clause with between, and operators.
- ✓ This returns within range of values.
- ✓ Here start and end values are included.

Login Login with valid credentials

Query where clause with between and operators

```
mysql> select * from persons
    -> where PersonID  between 101 and 104;
```

Output

```
+----------+------------+-----------+--------------------------+-----------+
| PersonID | LastName   | FirstName | Address                  | location  |
+----------+------------+-----------+--------------------------+-----------+
|      101 | Danial     | K         | Near to Data Science Area | Hyderabad |
|      102 | Nireekshan | D         | Near to AI Theatre        | Bangalore |
|      103 | Ranjan     | M         | Near to Python Theatre    | Hyderabad |
|      104 | Prasad     | K         | Near to Python Road       | Hyderabad |
+----------+------------+-----------+--------------------------+-----------+
4 rows in set (0.00 sec)
```

2.3. Where clause with like operator

- ✓ We can apply where clause with like operators.
- ✓ By using this we can search specific pattern in a column
  - 'h%' it returns starts with h values
  - '%h' it returns ends with h values

Login Login with valid credentials

Query where clause with like operator

```
mysql> select * from persons
    -> where location like 'h%';
```

Output

```
+----------+----------+-----------+--------------------------+-----------+
| PersonID | LastName | FirstName | Address                  | location  |
+----------+----------+-----------+--------------------------+-----------+
|      101 | Danial   | K         | Near to Data Science Area | Hyderabad |
|      103 | Ranjan   | M         | Near to Python Theatre    | Hyderabad |
|      104 | Prasad   | K         | Near to Python Road       | Hyderabad |
+----------+----------+-----------+--------------------------+-----------+
3 rows in set (0.00 sec)
```

2.4. Where clause with in operator

- ✓ We can apply where clause with in operator.
- ✓ By using this we can get the matched records from the table.

Login Login with valid credentials

Query where clause with in operator

```
mysql> select * from persons
    -> where Lastname in('Nireekshan', 'Danial');
```

Output

```
+----------+------------+-----------+--------------------------+-----------+
| PersonID | LastName   | FirstName | Address                  | location  |
+----------+------------+-----------+--------------------------+-----------+
|      101 | Danial     | K         | Near to Data Science Area | Hyderabad |
|      102 | Nireekshan | D         | Near to AI Theatre       | Bangalore |
+----------+------------+-----------+--------------------------+-----------+
2 rows in set (0.00 sec)
```

Login Login with valid credentials

Query where clause with not in operator

```
mysql> select * from persons
    -> where Lastname not in('Nireekshan', 'Danial');
```

Output

```
+----------+----------+-----------+------------------------+-----------+
| PersonID | LastName | FirstName | Address                | location  |
+----------+----------+-----------+------------------------+-----------+
|      103 | Ranjan   | M         | Near to Python Theatre | Hyderabad |
|      104 | Prasad   | K         | Near to Python Road    | Hyderabad |
+----------+----------+-----------+------------------------+-----------+
2 rows in set (0.00 sec)
```

3. Where clause with AND, OR, NOT operators

- ✓ We can use below operators along with where clause
  - And operator
  - Or operators
  - Not operator

3.1. Where clause with and operator

- ✓ If both conditions are true then AND operator returns the matched records.

Login — Login with valid credentials

Query — where clause with and operator

```
mysql> select * from Persons
    -> where LastName = 'Danial' and location = 'Hyderabad';
```

Output

```
+----------+----------+-----------+--------------------------+-----------+
| PersonID | LastName | FirstName | Address                  | location  |
+----------+----------+-----------+--------------------------+-----------+
|      101 | Danial   | K         | Near to Data Science Area | Hyderabad |
+----------+----------+-----------+--------------------------+-----------+
1 row in set (0.00 sec)
```

3.2. Where clause with or operator

- ✓ If at least one condition is matching from multiple conditions then or operator returns the corresponding records

Login — Login with valid credentials

Query — where clause with and operator

```
mysql> select * from Persons
    -> where LastName = 'Prasad' or location = 'UK';
```

Output

```
+----------+----------+-----------+---------------------+-----------+
| PersonID | LastName | FirstName | Address             | location  |
+----------+----------+-----------+---------------------+-----------+
|      104 | Prasad   | K         | Near to Python Road | Hyderabad |
+----------+----------+-----------+---------------------+-----------+
1 row in set (0.00 sec)
```

3.3. Where clause with not operator

✓ Not operator displays the records if the condition is not true.

Login Login with valid credentials

Query where clause with and operator

```
mysql> select * from Persons
    -> where not location = 'Bangalore';
```

Output

```
+----------+----------+-----------+---------------------------+-----------+
| PersonID | LastName | FirstName | Address                   | location  |
+----------+----------+-----------+---------------------------+-----------+
|      101 | Danial   | K         | Near to Data Science Area | Hyderabad |
|      103 | Ranjan   | M         | Near to Python Theatre    | Hyderabad |
|      104 | Prasad   | K         | Near to Python Road       | Hyderabad |
+----------+----------+-----------+---------------------------+-----------+
3 rows in set (0.00 sec)
```

7. MYSQL – Constraints

Contents

7. MySql – Constraints

1. Constraints

- ✓ SQL constraints are used to specify rules for data in a table.
- ✓ We can specify constraints during table creation of the table or after table created with alter table.
- ✓ By using this we can restricts the type of the data to the column.
- ✓ This ensures the accuracy and reliability of the data in the table.
- ✓ We can apply constraints on column and table level as well.

2. Constraints table

Name	Description
✓ Not null	✓ Ensures that a column cannot have a NULL value
✓ Primary key	✓ Uniquely identifies each row in a table
✓ Check	✓ Ensures that the values in a column satisfies a specific condition

3. Not null constraint

- ✓ By default, a column values can store null values.
- ✓ If we apply not null constraint over column then columns will not allow null values.
- ✓ It means, columns should have a value instead of null.

Login Login with valid credentials

Query Creating a table

```
mysql> create table Persons10(
        id int,
        lastname varchar(255),
        firstname varchar(255),
        age int
);
```

Output

Successfully table created

Login Login with valid credentials

Query Inserting data into table

```
mysql> insert into Persons10(id, lastname, firstname, age)
        values(NULL, 'daniel', '', 16);
```

Output

```
mysql> select * from persons10;
+------+----------+-----------+------+
| id   | lastname | firstname | age  |
+------+----------+-----------+------+
| NULL | daniel   |           |   16 |
+------+----------+-----------+------+
1 row in set (0.00 sec)
```

Login Login with valid credentials

Query not null constraint

```
mysql> create table Persons11(
        id int not null,
        lastname varchar(255) not null,
        firstname varchar(255) not null,
        age int
);
```

Output

Successfully table created

Login Login with valid credentials

Query not null constraint

```
mysql> insert into Persons11(id, lastname, firstname, age)
        values(NULL, 'daniel', '', 16);
```

Output

```
ERROR 1048 (23000): Column 'id' cannot be null
```

4. Primary key constraint

- ✓ The primary key constraint uniquely identifies each record in a table.
- ✓ Primary keys must contain UNIQUE values, and cannot contain null values.

Login Login with valid credentials
Query Primary key constraint

```
mysql> create table Persons12(
        id int not null,
        lastname varchar(255) not null,
        firstname varchar(255) not null,
        age int,
        primary key(id)
);
```

Output

Successfully table created

Login Login with valid credentials
Query not null constraint

```
mysql> insert into Persons12(id, lastname, firstname, age)
        values(101, 'daniel', 'K', 16);
```

Output

Query OK, 1 row affected (0.01 sec)

Login	Login with valid credentials
Query	not null constraint
	mysql> insert into Persons12(id, lastname, firstname, age) values(101, 'daniel', 'K', 16);
Output	
	ERROR 1062 (23000): Duplicate entry '101' for key 'persons12.PRIMARY'

5. Check constraint

- ✓ The check constraint is used to limit the value range that can be placed in a column.
- ✓ If we define a check constraint on a column it will allow only certain values for this column.

Login — Login with valid credentials

Query — check constraint

```
mysql> create table Persons13(
        id int not null,
        lastname varchar(255) not null,
        firstname varchar(255) not null,
        age int,
        check (age >= 18)
);
```

Output

Successfully table created

Login — Login with valid credentials

Query — not null constraint

```
mysql> insert into Persons13(id, lastname, firstname, age)
        values(101, 'daniel', 'K', 16);
```

Output

```
ERROR 3819 (HY000): Check constraint 'persons13_chk_1' is
violated
```

8. MySql – Imp functions

Contents

8. MySql – Imp functions

1. Functions

- ✓ In mysql there are predefined functions.
- ✓ These functions helps us to finish basic requirements.

2. min(column name) function

- ✓ The min(column) is a predefined function in mysql.
- ✓ By using this function we can get minimum value from specific column

Login Login with valid credentials
Query Creating a table

mysql> select * from persons;

Output

```
mysql> select * from persons;
+----------+------------+-----------+--------------------------+-----------+
| PersonID | LastName   | FirstName | Address                  | location  |
+----------+------------+-----------+--------------------------+-----------+
|      101 | Danial     | K         | Near to Data Science Area | Hyderabad |
|      102 | Nireekshan | D         | Near to AI Theatre        | Bengaluru |
|      103 | Ranjan     | M         | Near to Python Theatre    | Hyderabad |
|      104 | Prasad     | K         | Near to Python Road       | Hyderabad |
+----------+------------+-----------+--------------------------+-----------+
4 rows in set (0.00 sec)
```

Login Login with valid credentials
Query Creating a table

mysql> select min(personid) from persons;

Output

```
+----------------+
| min(personid) |
+----------------+
|            101 |
+----------------+
```

3. max(column name) function

- ✓ The max(column) is a predefined function in mysql.
- ✓ By using this function we can get maximum value from specific column

Login Login with valid credentials
Query Creating a table

mysql> select * from persons;

Output

```
mysql> select * from persons;
+----------+------------+-----------+--------------------------+-----------+
| PersonID | LastName   | FirstName | Address                  | location  |
+----------+------------+-----------+--------------------------+-----------+
|      101 | Danial     | K         | Near to Data Science Area | Hyderabad |
|      102 | Nireekshan | D         | Near to AI Theatre        | Bengaluru |
|      103 | Ranjan     | M         | Near to Python Theatre    | Hyderabad |
|      104 | Prasad     | K         | Near to Python Road       | Hyderabad |
+----------+------------+-----------+--------------------------+-----------+
4 rows in set (0.00 sec)
```

Login Login with valid credentials
Query Creating a table

mysql> select max(personid) from persons;

Output

```
+---------------+
| max(personid) |
+---------------+
|           104 |
+---------------+
1 row in set (0.00 sec)
```

4. count(column name) function

- ✓ The count(column) is a predefined function in mysql.
- ✓ The count(column name) function returns the number of rows from the table.
- ✓ This function even returns the number of rows that matches a specified criterion.

Login Login with valid credentials
Query Creating a table

mysql> select * from persons;

Output

```
mysql> select * from persons;
+----------+-----------+-----------+--------------------------+-----------+
| PersonID | LastName  | FirstName | Address                  | location  |
+----------+-----------+-----------+--------------------------+-----------+
|      101 | Danial    | K         | Near to Data Science Area | Hyderabad |
|      102 | Nireekshan | D        | Near to AI Theatre       | Bengaluru |
|      103 | Ranjan    | M         | Near to Python Theatre   | Hyderabad |
|      104 | Prasad    | K         | Near to Python Road      | Hyderabad |
+----------+-----------+-----------+--------------------------+-----------+
4 rows in set (0.00 sec)
```

Login Login with valid credentials
Query Creating a table

mysql> select count(location)
from persons;

Output

```
+-----------------+
| count(location) |
+-----------------+
|               4 |
+-----------------+
```

Login Login with valid credentials

Query Creating a table

```
mysql> select count(location)
       from persons
       where location =  'hyderabad';
```

Output

```
+-----------------+
| count(location) |
+-----------------+
|               3 |
+-----------------+
1 row in set (0.00 sec)
```

5. avg(column name) function

- ✓ The avg(column) is a predefined function in mysql.
- ✓ The avg(column name) function returns average value from the values

Login Login with valid credentials
Query Creating a table

```
mysql> create table products(
        prodname varchar(255),
        prodprice int
        );
```

Output

Table created successfully

Login Login with valid credentials
Query Insert the data into table

```
mysql> insert into products(prodname, prodprice)
        values ('samsung', 12000);

Query OK, 1 row affected (0.01 sec)

mysql> insert into products(prodname, prodprice)
        values ('iphone 14', 75000);

Query OK, 1 row affected (0.01 sec)

mysql> insert into products(prodname, prodprice)
        values ('nokia', 10000);

Query OK, 1 row affected (0.01 sec)
```

Output

Data inserted successfully into the products table

Login Login with valid credentials
Query Select table

mysql> select * from products;

Output

```
+-----------+-----------+
| prodname  | prodprice |
+-----------+-----------+
| samsung   |     12000 |
| iphone 14 |     75000 |
| nokia     |     10000 |
+-----------+-----------+
```

Login Login with valid credentials
Query Apply avg(column) function

mysql> select avg(prodprice)
from products;

Output

```
+---------------+
| avg(prodprice) |
+---------------+
|     32333.3333 |
+---------------+
```

6. sum(column name) function

- ✓ The sum(column) is a predefined function in mysql.
- ✓ The sum(column name) function returns sum of all values in column

Login Login with valid credentials
Query Select table

mysql> select * from products;

Output

```
+-----------+-----------+
| prodname  | prodprice |
+-----------+-----------+
| samsung   |     12000 |
| iphone 14 |     75000 |
| nokia     |     10000 |
+-----------+-----------+
```

Login Login with valid credentials
Query Apply sum(column) function

mysql> select sum(prodprice)
from products;

Output

```
+-----------------+
| sum(prodprice) |
+-----------------+
|           97000 |
+-----------------+
```

9. MySql – Pandas - DataFrame

Contents

9. MySql – Pandas - DataFrame

1. DataFrame

- ✓ Being a Data Scientist, we are very familiar with Pandas DataFrame
- ✓ We can create DataFrame from,
  - Csv file
  - Excel file
  - Json file
  - Mysql table
  - etc

2. Install the required libraries

- ✓ pip install **sqlalchemy**
- ✓ pip install **pymysql**
- ✓ pip install **cryptography**

Program Name: Accessing table from mysql
demo1.py

```
print("Step 1: Importing libraries")
import pandas as pd
from urllib.parse import quote_plus
from sqlalchemy import create_engine


print("Step 2: Preparing database connection string")
db_connection_str =
        'mysql+pymysql://root:%s@localhost/danieldb1' %
        quote_plus("D@1234niel#")


print("Step 3: Create database connection")
db_connection = create_engine(db_connection_str)


print("Step 4: Accessing read_sql function")
df = pd.read_sql('SELECT * FROM persons', con = db_connection)


print("Step 5: Accessing DataFrame")

print()
print(df)
```

Output

```
   PersonID    LastName FirstName                    Address   location
0       101     Dania l         K  Near to Data Science Area   Hyderabad
1       101  Nireekshan         D          Near to AI theater  Bangalore
2       103      Ranjan         M      Near to Python theater  Hyderabad
3       104      Prasad         K         Near to Python Road  Hyderabad
```

Generative AI - LLM

Table of Contents

Generative AI

1. Generative AI

Generative AI = Generative + AI

- ✓ Generative AI is a type of Artificial Intelligence technology.
- ✓ It is capable to **generate** the **data** or new content.
- ✓ Data means,
  - Text.
  - Image.
  - Audio.
  - Video.
  - Code & etc.

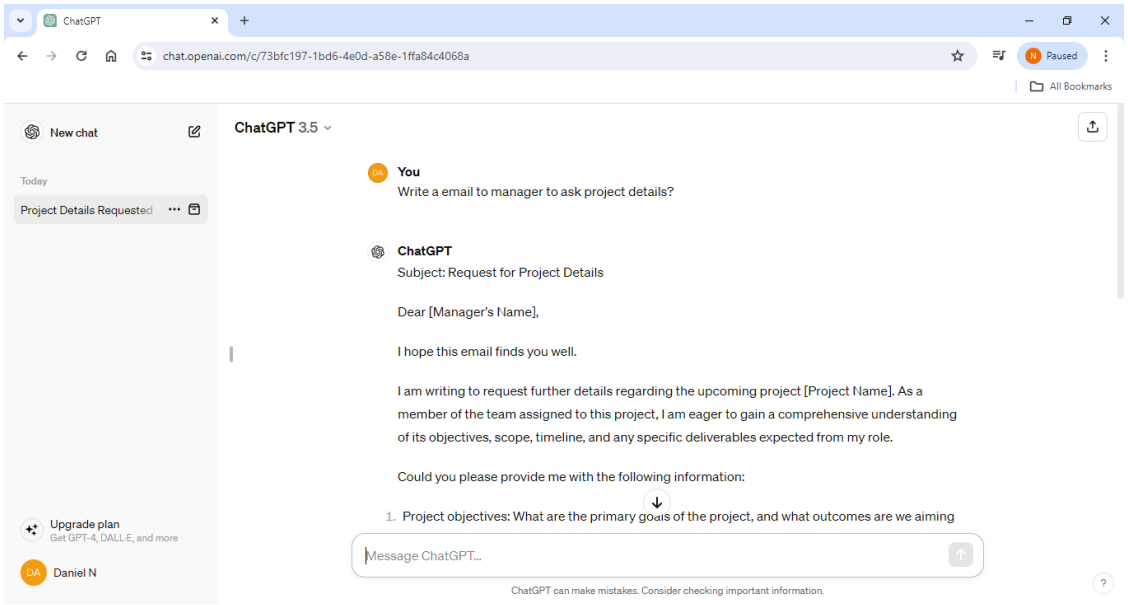
Generative AI	**=**	**Generative + AI**
Generative	**->**	**Generate the content.**
AI	**->**	**Using Artificial Intelligence**

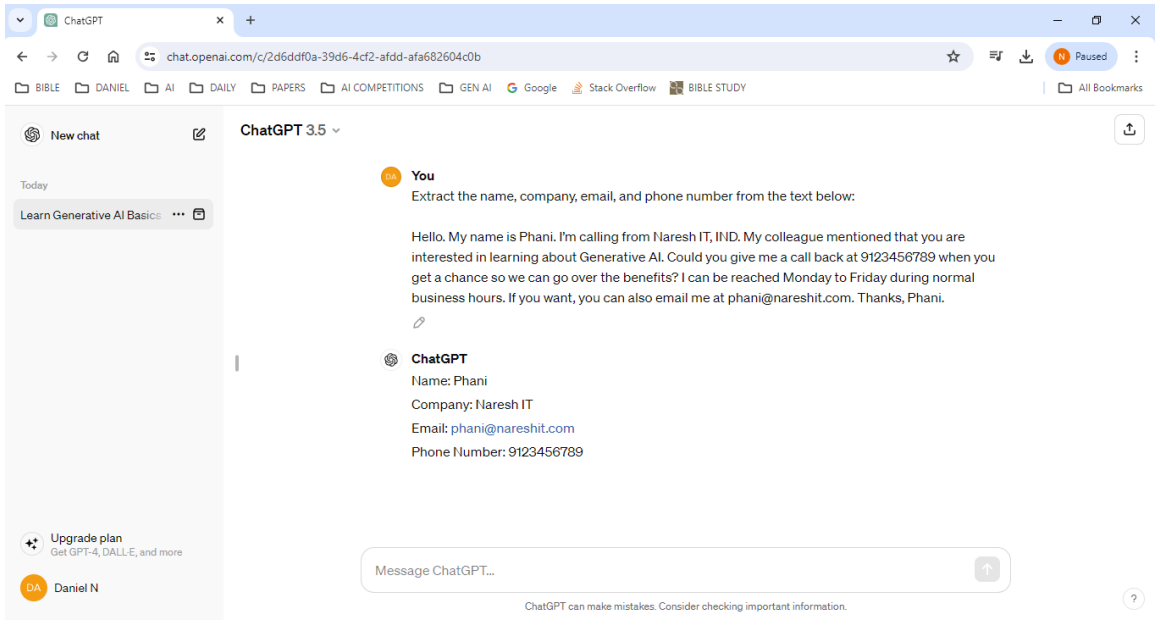
Kind note

- ✓ Generative AI **full form** is Generative Artificial Intelligence.
- ✓ Generative AI **short form** is Gen AI.

1.1. Generate the Text

- ✓ By using Generative AI tool, we can generate the **Text**.
- ✓ One of the Gen AI tools is,
  - ChatGPT





1.2. Generate the Images

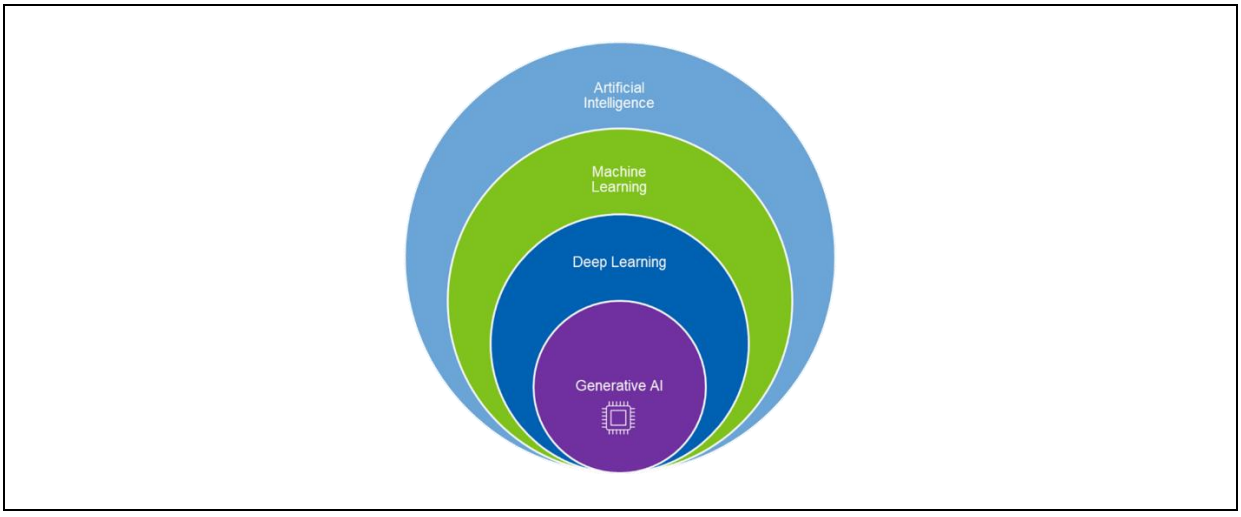
- ✓ By using Generative AI tool, we can generate the **Images**.
- ✓ One of the Gen AI tools is,
  - dall-e-2

A photo of Michelangelo's sculpture of David wearing headphones djing

1.3. Generate the Video

- ✓ By using Generative AI tool, we can generate the **Images**.
- ✓ One of the Gen AI tools is,
  - www.videogen.io

2. Generative AI Diagram



3. Generative AI is generating: Text, Images, Video, Code & etc

3.1. Text

- ✓ Creating a realistic text like news articles, blog posts.
- ✓ Many platforms are using this like,
  - Generating content for websites and social media
  - Creating personalized marketing materials etc.

3.2. Images

- ✓ Creating a realistic image of people, objects and scenes that do not exists in the real world.
  - Generating realistic product images for e-commerce.
  - Creating training data for other AI models & etc

3.3. Videos

- ✓ Creating videos that do not exist in real world.
  - Creating special effects for movies and TV shows.
  - Creating personalized video content for marketing and advertising.

6.4. Code

- ✓ Generative AI models generate the code for different programming languages.
- ✓ This can helpful to programmers/developers.

3.5. Music

- ✓ Generative AI models are being used to create new music.
  - Creating music for movies and TV shows.
  - Generating personalized playlists & etc.

4. Artificial Intelligence

- ✓ Artificial Intelligence is the **ability** for a computer to think, learn and do tasks.
  - Problem solving.
  - Understanding Language.
  - Making decisions & etc.

- ✓ AI having **below** topics like,
  - Machine Learning.
  - Deep Learning.
  - Natural Language Processing.
  - Computer Vision.

- ✓ By using above topics we can do,
  - Prediction.
  - Classification.
  - Sentiment analysis & etc.

- ✓ AI can **enable** machines to mimic human.
- ✓ Artificial Intelligence short form is **AI.**

5. AI Model

- ✓ AI model is a **program**.
- ✓ This program **analyses** datasets to find **patterns** and make **predictions**.

6. Generative AI Models

- ✓ In Generative AI, models generate the data.

6.1. Generative Pre-trained Transformer (GPT)

- ✓ **GPT** is Large Language Model.
- ✓ This is developed by OpenAI.
- ✓ It's trained on a massive dataset of text and code.
- ✓ It is Capable of,
  - Generating text.
  - Translating languages.
  - Writing different creative content.
  - Answering your questions.
- ✓ GPT - 4 is the latest version at the time of this writing.

6.2. Llama 2

- ✓ **Llama 2** is Large Language Model.
- ✓ This is developed by Meta.
- ✓ Llama 2 is second version of a natural large language model.

6.3. Claude

- ✓ Claude is Large Language Model.
- ✓ This is developed by startup company called Anthropic.
- ✓ This is like a ChatGPT, it can,
  - Generate text
  - Write code
  - Summarize & etc.

6.4. Gemini

- ✓ Gemini is Generative AI model.
- ✓ This is developed by Google company.
- ✓ It's a Google's new multi-modal model.
- ✓ This can understand,
  - Text.
  - Images.
  - Videos.
  - Audio.

- ✓ It will be available in different sizes (Ultra, Pro, and Nano), each with different capabilities.

6.5. PaLM2

- ✓ PaLM is Large Language Model.
- ✓ The full form of PaLM is Pathway Language Model.
- ✓ It is a multi-modal model
- ✓ This is developed by Google company.
- ✓ This can process,
  - Text.
  - Code.
  - Images.

6.6. DALL-E

- ✓ DALL-E is Visual AI model.
- ✓ This is developed by OpenAI.
- ✓ It can create,
  - Realistic images from text prompts.

6.7. Stable Diffusion

- ✓ Stable Diffusion is an image generation model.
- ✓ This is developed by OpenAI.
- ✓ It can,
  - Generate detailed images.
  - Text descriptions.
  - Inpainting and out painting.
  - Generate image-to-image translations.
- ✓ This model generates these are by prompt as input.

6.8. Midjourney

- ✓ Midjourney is an image generation model.
- ✓ This is developed by startup called Midjourney, Inc.
- ✓ This is like DALL-E and Stable Diffusion.

6.9. CodeWhisperer

- ✓ CodeWhisperer is a **code** generation model.
- ✓ This is developed by AWS.
- ✓ This can generate the,
  - Code in several programming languages. (Python, Java, JavaScript, TypeScript & etc)

6.10. CodeLlama

- ✓ CodeLlama is a large language model.
- ✓ This is built on Llama 2.
- ✓ This model specifically trained on code.
- ✓ It also comes in various sizes and supports multiple popular programming languages.

6.11. Codex

- ✓ Codex is a large language model.
- ✓ This is a **code** generation model.
- ✓ This model specifically trained on code.
- ✓ This can generate the,
    - Code in several programming languages. (Python, C#, Java, JavaScript, SQL, Go, PHP, and Shell).

7. Use cases of Generative AI

- ✓ In Generative AI, models generate the data.

7.1. Content Generation

- ✓ Generative AI helpful to generate the content,
    - Blogs,
    - Reports,
    - e-mails
    - Social media posts.

- ✓ This content helpful to business for marketing.

7.2. Personalized marketing

- ✓ Generative AI can create personalized marketing content,
    - e-mails
    - Landing pages.
    - Social media posts.

- ✓ This content helpful to businesses to reach their target audience more effectively and increase conversion rates.

7.3. Customer service

- ✓ Generative AI can be used to create chatbots that can answer customer questions and resolve issues.
- ✓ This is really great advantage like, free human customer service.

7.4. Risk management

- ✓ Generative AI can identify and predict risks,
  - Fraud.
  - Cyberattacks
  - Supply chain disruptions.

- ✓ This will helpful to businesses to protect their assets.

7.5. Compliance

- ✓ Generative AI can,
  - Generate compliant documents,
    - Contracts.
    - Reports.
    - Disclosures.

- ✓ This can help businesses to save time and money and reduce the risk of non-compliance.

7.6. Software Development

- ✓ Generative AI can,
  - Generate new code
  - Provide code snippets, or even write simple software.
  - Potentially saving time and reducing errors.

- ✓ In addition, it also helps document code, refactor, generate test cases, and optimize existing code.

7.7. Data Augmentation

- ✓ Generative AI can create synthetic data for Data Science projects if needed.

8. Below domains are using the Generative AI

8.1. Financial

- ✓ Generative AI can help with,
  - Decision-making.
  - Risk model assessment.
  - Development of new financial products and services.

- ✓ Customer operations to enhance services and resolutions for each client based on transactions and history.

8.2. Healthcare

- ✓ Generative AI is used to develop,
  - New drugs and treatments.
  - Design medical devices.
  - Create personalized patient treatment plans.
  - Generate patient documentation on instructions, risks, and drug interactions.

8.3. Manufacturing

- ✓ Generative AI is used to develop,
  - Design new products.
  - Optimize manufacturing processes.
  - Improve quality control.

8.4. Retail and Consumer Packaged Goods

- ✓ Generative AI is used to,
  - Personalize shopping experiences.
  - Recommend products.
  - Manage inventory.
  - Accelerate consumer research.
  - Enhance the supply chain & etc.

8.4. Marking and Sales

- ✓ Generative AI is helping enhance,
  - Understand real-time customer trends.
  - Personalized outreaches embedded into virtual assistants.
  - Dynamic customer journeys & etc

9. OpenAI's Introduction

- ✓ OpenAI is an artificial intelligence research organization and technology company founded in December 2015.
- ✓ Its mission is to ensure that artificial general intelligence (AGI) benefits all of humanity.
- ✓ OpenAI develops AI models and technologies, including the famous language models like GPT (Generative Pre-trained Transformer).

10. Installation

- ✓ pip install openai

Hello World example

11. OpenAI's Hello World Program

- ✓ Let's write and run the basic example by using OpenAI's API.

Kind note

- ✓ To run below program, we should set API key first.

Program Name: OpenAI's API Hello World Program
demo1.py

```
from openai import OpenAI

client = OpenAI()

response = client.completions.create(
        model = "gpt-3.5-turbo-instruct",
        prompt = "How are you."
)

print(response.choices[0].text)
```

Output

```
I am an AI and do not have the capability to feel emotions.
```

12. Understanding the Hello World Program

- ✓ **openai** is library, we are importing.
- ✓ **OpenAI** is predefined class, we are importing from openai library.
- ✓ **client** is an object to OpenAI class.
- ✓ **completion** is an object, accessing by using client.
- ✓ **create** is method, accessing by using client.completions.
- ✓ **create** method having two parameters,
  - **model**
  - **prompt**

13. Hugging Face

- ✓ Hugging Face is a leading company in the field of natural language processing (NLP) and machine learning.
- ✓ Founded in 2016, it has become well-known for its contributions to the AI community, particularly through its open-source libraries.

13.1. transformers library

- ✓ pip install transformers
    - It Provides a wide range of pre-trained models for various NLP tasks, such as text generation, translation, summarization, and classification.
    - Supports models like BERT, GPT, and many others.

Program Name: Text Classification
demo1.py

```
from transformers import pipeline

# Load a pre-trained model for text classification
classifier = pipeline("text-classification")

# Classify a text
a = "I love using Hugging Face's transformers library!"
result = classifier(a)
print(result)
```

Output

```
[{'label': 'POSITIVE', 'score': 0.9978122711181641}]
```

Program Name: Text Generation
demo2.py

```
from transformers import pipeline

# Load a pre-trained model for text generation
generator = pipeline("text-generation")

# Generate text based on a prompt
a = "Once upon a time, in a land far away,"
result = generator(a, max_length = 50)
print(result)
```

Output

```
[{'generated_text': 'Once upon a time, in a land far away, the land of the dead, and the land of a living creature, the human race had become. In them died the dead, and in them dwelt men, who in these times had never'}]
```

Program Name: Question Answering
demo3.py

```
from transformers import pipeline

# Load a pre-trained model for question answering
qa = pipeline("question-answering")

# Answer a question based on a context
context = "Hugging Face is an AI company based in New York.
They are known for their work in natural language processing."
question = "Where is Hugging Face based?"

result = qa(question=question, context=context)
print(result)
```

Output

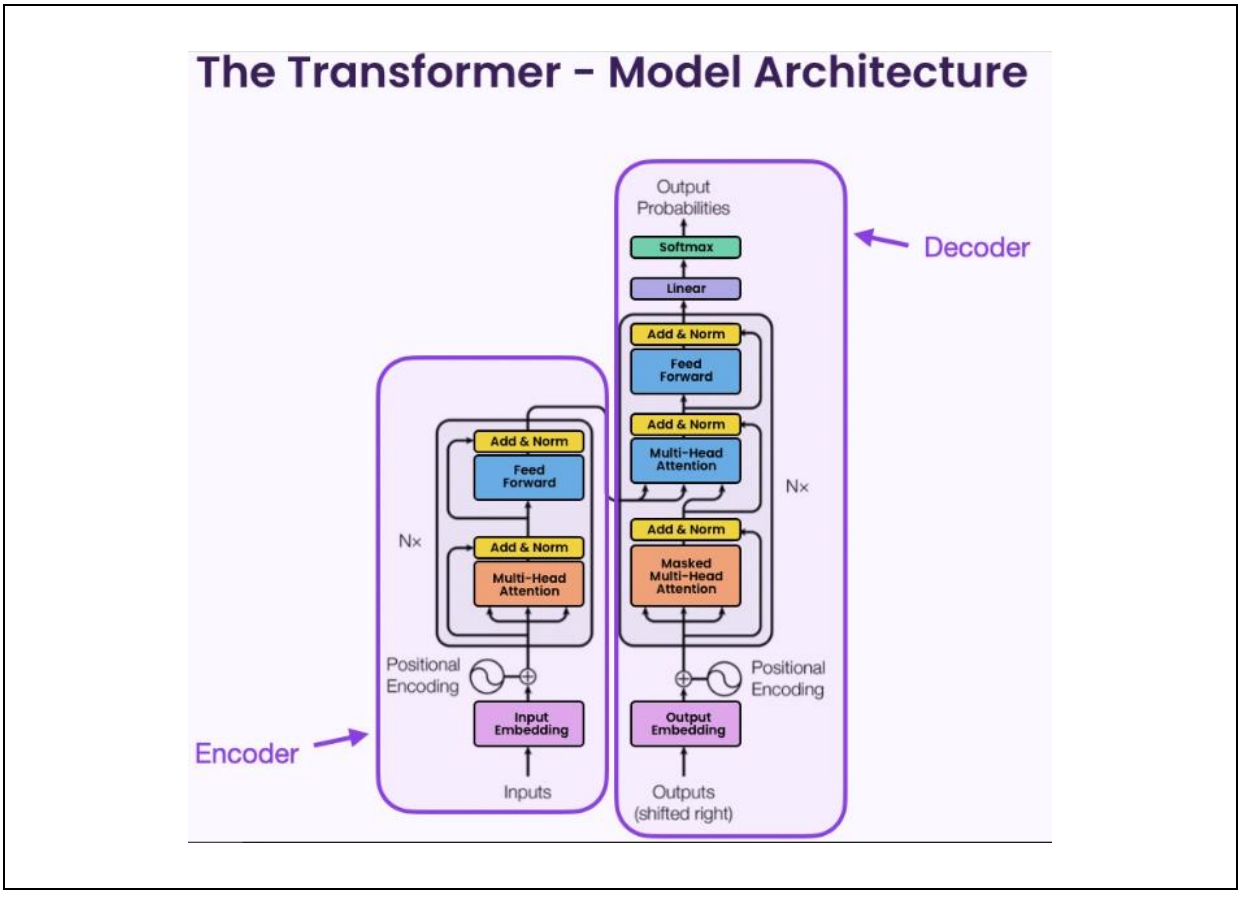
```
{'score': 0.9903117418289185, 'start': 39, 'end': 47, 'answer': 'New
York'}
```

14. Large Language Model (LLM)

- ✓ LLM stands for "Large Language Model".
- ✓ It is a type of AI trained model on text data,
  - To understand human text
  - To generate human-like text
  - Handling tasks like answering questions
  - Summarizing text data
  - Translating the text data
  - Creating new content.

15. Large Language Model Architecture

- ✓ Large Language Models (LLMs) use the Transformer neural network architecture.
- ✓ It is introduced in the 2017 paper "Attention is All You Need" by Vaswani et al.
- ✓ Here we can find effectively processing large-scale data and capturing complex language patterns.



15.1. Input Embedding:

- ✓ Purpose: Each word in the input sequence is represented as a high-dimensional vector, capturing semantic meaning.

15.2. Positional Encoding:

- ✓ Purpose: Adds information about the position of each token. Positional encoding is added to the Transformer model to provide information about the order of tokens

15.3. Add & Norm:

- ✓ Purpose: Combines the output of a sub-layer with the input and normalizes it. The "Add" operation is a residual connection that helps the model avoid vanishing gradients, while "Norm" refers to Layer Normalization, which stabilizes and speeds up training.

15.4. Multi-Head Attention:

- ✓ Purpose: Multi-head attention uses several attention mechanisms (heads) in parallel. Each head processes the input differently, allowing the model to capture various relationships in the data.

15.5. Feed Forward:

- ✓ Purpose: After attention is applied, a feed-forward network processes the data, adding non-linearity and allowing the model to capture more complex patterns.

15.6. Encoder Block (Left Side):

- ✓ Purpose: Consists of several identical layers (denoted as Nx), each containing Multi-Head Attention, Add & Norm, and Feed Forward sub-layers. Processes the input sequence to create a context-aware representation of the input tokens.

15.7. Masked Multi-Head Attention:

- ✓ Purpose: In the decoder, this layer masks future positions to ensure that predictions for a token depend only on previous tokens, not on future ones.

15.8. Decoder Block (Right Side):

- ✓ Purpose: Similar to the encoder, it consists of several identical layers (Nx), but with an additional masked attention mechanism. Generates the output sequence by attending to both the encoder's output and the previously generated tokens.

15.9. Linear + Softmax:

- ✓ Purpose: The linear layer reduces the dimensionality of the decoder's output, and the softmax function converts this into a probability distribution, predicting the next token in the sequence.

15.10. Output Probabilities:

- ✓ Purpose: The final prediction of the model. The output probabilities are used to determine the next token in the sequence, forming the basis for tasks like translation, summarization, or text generation.