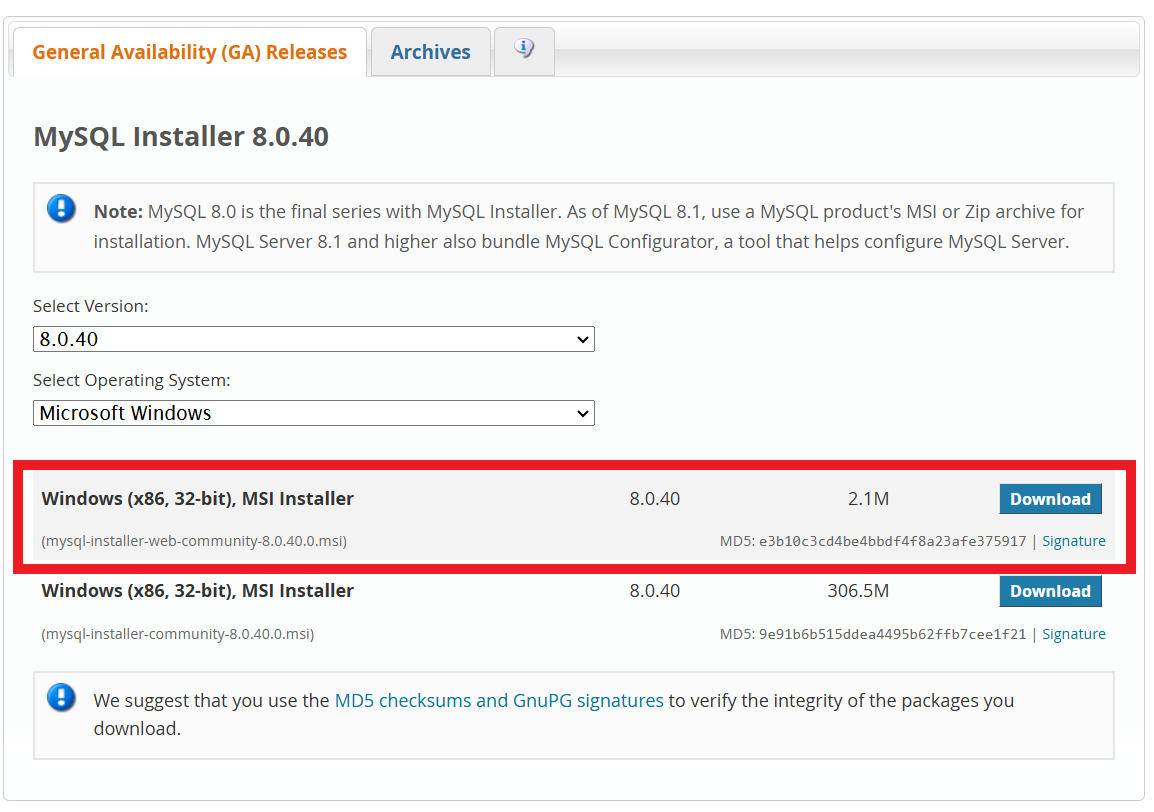
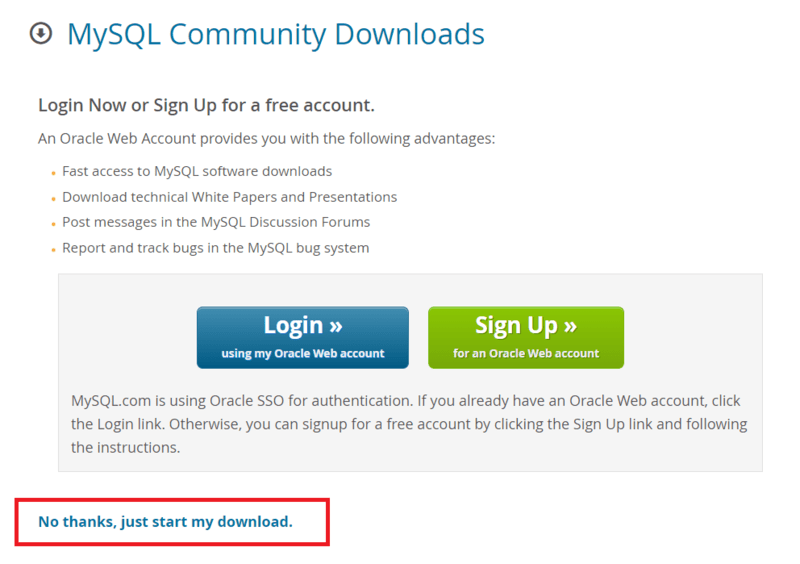
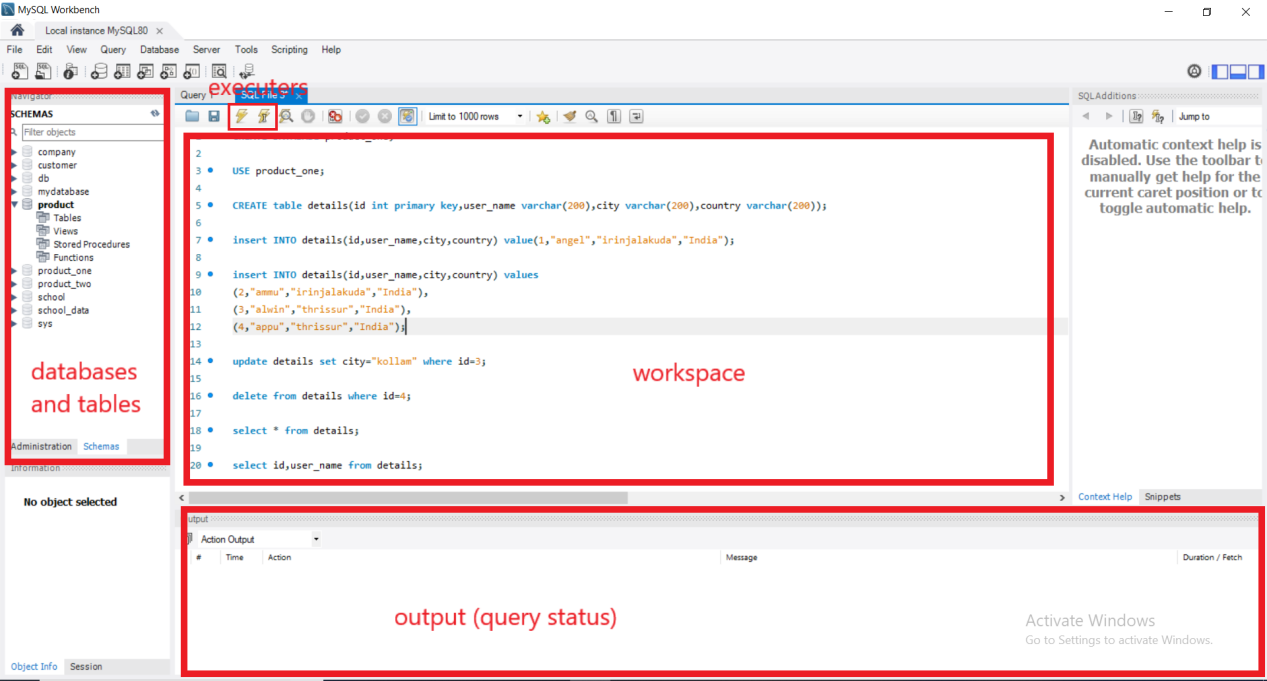
## Download and Install MySQL for Windows Steps

### Visit the Official MySQL Website

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







**Download mysql in python**

Step 1: open command prompt

Step 2: pip install mysql-connector-python

Step 3:import package using cmd

Import mysql.connector

Step 4:create connection variable using connect()

import mysql.connector # Establishing the connection

connection = mysql.connector.connect(

host='localhost', # Your MySQL server address user='your\_username', # Your MySQL username password='your\_password', # Your MySQL password database='your\_database' # Your database name

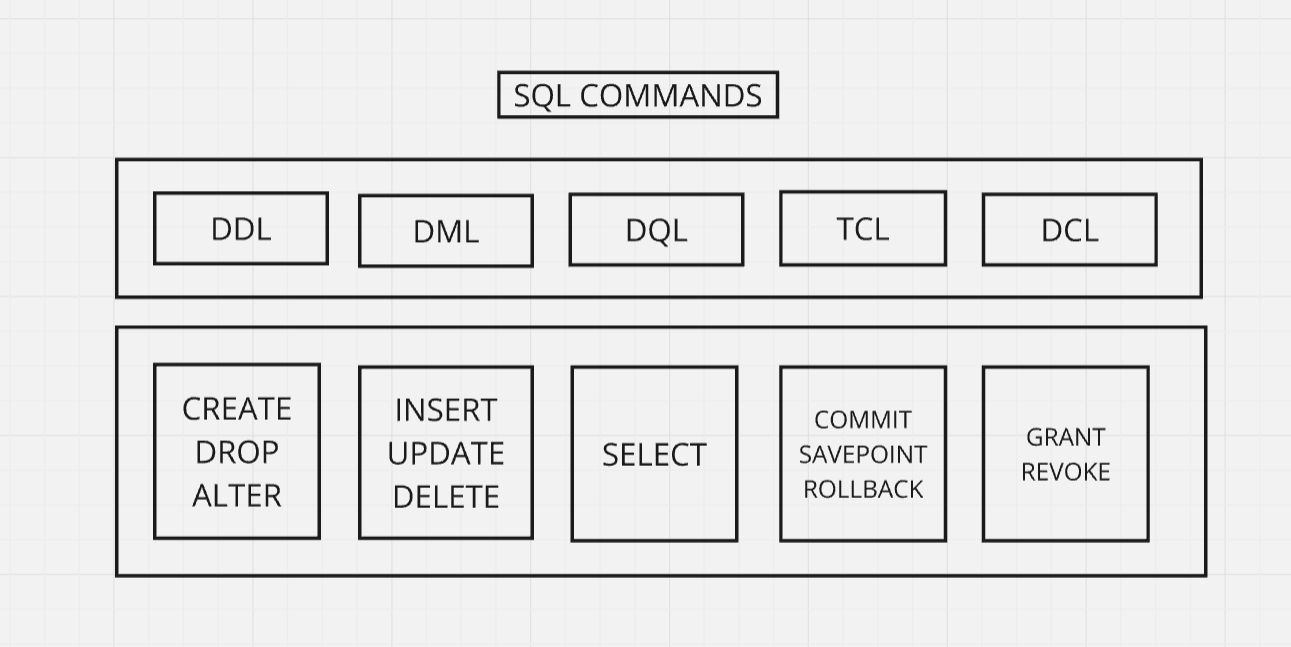
) # Check if the connection was successful

if connection.is\_connected():

print("Connected to MySQL Server")

connection.close()

# SQL Commands



## ****DDL (Data Definition Language) USING MySql****

1. **CREATE**
2. **Database creation:**
   1. **Synatx:** CREATE DATABASE databasename;
   2. Example : CREATE DATABASE college;

1. **Table creation:**
   1. **Syntax :USE *databasename*;**

CREATE TABLE table\_name (  
    column1 datatype,  
    column2 datatype,  
    column3 datatype,

);

* 1. Example :

#select the college database

**USE college;**

#create table called dept

**CREATE TABLE Dept(dept\_id int,dept\_name varchar(200));**

1. **DROP**
   1. **DROP is used to delete entire database and table.**

DROP TABLE table\_name;

DROP DATABASE db\_name;

* 1. The TRUNCATE TABLE statement is used to delete the data inside a table, but not the table itself.

TRUNCATE TABLE table\_name;

1. **ALTER**

The ALTER TABLE statement is used to add, delete, or modify columns in an existing table.

The ALTER TABLE statement is used for:

* **Adding Columns**: You can add new columns to a table.
* **Dropping Columns**: You can remove existing columns from a table.
* **Modifying Columns**: You can change the data type or properties of existing columns.
* **Renaming Columns**: You can rename existing columns in a table.
* **Adding/Dropping Constraints**: You can add or remove various constraints on the table.

|  |  |  |
| --- | --- | --- |
| **Description** | **syntax** | **Example** |
| **Add a Column** | ALTER TABLE table\_name ADD column\_name datatype; | ALTER TABLE Customers ADD Email VARCHAR(255); |
| **Drop a Column** | ALTER TABLE table\_name DROP COLUMN column\_name; | ALTER TABLE Customers DROP COLUMN Email; |
| **Modify a Column** | ALTER TABLE table\_name MODIFY COLUMN column\_name datatype; | ALTER TABLE Persons MODIFY COLUMN DateOfBirth YEAR; |
| **Rename a Column** | ALTER TABLE table\_name RENAME COLUMN old\_name TO new\_name; | ALTER TABLE old\_table\_name RENAME TO new\_table\_name; |
| Adding Constraints | ALTER TABLE table\_name ADD CONSTRAINT constraint\_name constraint\_type (column\_name); | ALTER TABLE Employees ADD CONSTRAINT PK\_EmployeeID PRIMARY KEY (EmployeeID); |
| Dropping Constraints | ALTER TABLE table\_name DROP CONSTRAINT constraint\_name; | ALTER TABLE Employees DROP CONSTRAINT PK\_EmployeeID; |

**constraints in mysql with synatx and example**

In MySQL, constraints are rules that enforce data integrity and consistency within a table. They can be applied at the column level or the table level.

1. NOT NULL CONSTRAINT

* Ensures that a column cannot have NULL values.
* Syntax:

**CREATE TABLE table\_name ( column\_name datatype NOT NULL );**

* Example

**CREATE TABLE Customers (**

**CustomerID INT NOT NULL,**

**CustomerName VARCHAR(50) NOT NULL,**

**Age INT );**

1. UNIQUE CONSTRAINT

* Ensures that all values in a column are unique.
* Syntax :

**CREATE TABLE table\_name ( column\_name datatype UNIQUE );**

* Example :

**CREATE TABLE Users ( UserID INT NOT NULL UNIQUE,**

**Username VARCHAR(50) NOT NULL,**

**Email VARCHAR(100) UNIQUE )**;

1. PRIMARY KEY CONSTRAINT

* A combination of NOT NULL and UNIQUE. It uniquely identifies each row in a table.
* Syntax :

**CREATE TABLE table\_name ( column\_name datatype, PRIMARY KEY (column\_name) );**

* Example :

**CREATE TABLE Employees ( EmployeeID INT NOT NULL,**

**Name VARCHAR(50),**

**PRIMARY KEY (EmployeeID) );**

1. FOREIGN KEY CONSTRAINT

* Prevents actions that would destroy links between tables by enforcing referential integrity.
* Syntax:

**CREATE TABLE table\_name ( column\_name datatype,**

**FOREIGN KEY (column\_name)**

**REFERENCES other\_table(column\_name) );**

* Example :

**CREATE TABLE Orders ( OrderID INT NOT NULL,**

**CustomerID INT,**

**FOREIGN KEY (CustomerID)**

**REFERENCES Customers(CustomerID) );**

1. CHECK CONSTRAINT

* Ensures that the values in a column satisfy a specific condition.
* Syntax :

**CREATE TABLE table\_name ( column\_name datatype,**

**CHECK (condition) );**

* Example:

**CREATE TABLE Persons ( PersonID INT NOT NULL,**

**Age INT, CHECK (Age >= 18) );**

1. DEFAULT CONSTRAINT

* Sets a default value for a column if no value is specified during insertion.
* Syntax:

**CREATE TABLE table\_name ( column\_name datatype DEFAULT default\_value );**

* Example:

**CREATE TABLE Products ( ProductID INT NOT NULL,**

**ProductName VARCHAR(50),**

**Price DECIMAL(10, 2) DEFAULT 0.00 );**

## Adding Constraints to Existing Tables

## Syntax for Adding Constraints:

**ALTER TABLE table\_name ADD CONSTRAINT constraint\_name constraint\_type (column\_name);**

## Example of Adding a Foreign Key Constraint:

**ALTER TABLE Orders**

**ADD CONSTRAINT FK\_CustomerID FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID);**

## ****DDL (Data Definition Language) USING PYTHON****

1. ****CREATE**** 
   1. ****Check database exist****

import mysql.connector

mydb = mysql.connector.connect(  
  host="localhost",  
   user="yourusername",  
  password="yourpassword"  
 )

mycursor = mydb.cursor()

mycursor.execute("SHOW DATABASES")

for x in mycursor:

print(x)

* 1. ****Create database****

import mysql.connector  
  
mydb = mysql.connector.connect(  
  host="localhost",  
  user="yourusername",  
  password="yourpassword"  
)  
  
mycursor = mydb.cursor()  
  
mycursor.execute("CREATE DATABASE mydatabase")

* 1. ****Check table****

import mysql.connector  
  
mydb = mysql.connector.connect(  
  host="localhost",  
  user="yourusername",  
  password="yourpassword",  
  database="mydatabase"  
)  
  
mycursor = mydb.cursor()  
  
mycursor.execute("SHOW TABLES")  
  
for x in mycursor:  
  print(x)

* 1. ****Create table****

import mysql.connector  
  
mydb = mysql.connector.connect(  
  host="localhost",  
  user="yourusername",  
  password="yourpassword",  
  database="mydatabase"  
)  
  
mycursor = mydb.cursor()  
  
mycursor.execute("CREATE TABLE customers (name VARCHAR(255), address VARCHAR(255))")

1. **DROP**
   1. **Check and drop**

import mysql.connector  
  
mydb = mysql.connector.connect(  
  host="localhost",  
  user="yourusername",  
  password="yourpassword",  
  database="mydatabase"  
)  
  
mycursor = mydb.cursor()  
  
sql = "DROP TABLE IF EXISTS customers"  
  
mycursor.execute(sql)

* 1. **Drop**

import mysql.connector  
  
mydb = mysql.connector.connect(  
  host="localhost",  
  user="yourusername",  
  password="yourpassword",  
  database="mydatabase"  
)  
  
mycursor = mydb.cursor()  
  
sql = "DROP TABLE customers"  
  
mycursor.execute(sql)