**1. Install mysql-connector-python Package**

To interact with MySQL in Python, install the mysql-connector-python package:

pip install mysql-connector-python

**2. Create a Database Connection**

To establish a connection with the MySQL server, use the following:

import mysql.connector

# Establish connection

conn = mysql.connector.connect(

host="localhost", # Change to your MySQL server

user="root", # Change to your MySQL username

password="password" # Change to your MySQL password

)

# Check connection

if conn.is\_connected():

print("Connected to MySQL Server")

# Close connection

conn.close()

**3. Create a Database**

Once connected, create a database:

import mysql.connector

# Establish connection

conn = mysql.connector.connect(

host="localhost",

user="root",

password="password"

)

cursor = conn.cursor()

# Create database

cursor.execute("CREATE DATABASE IF NOT EXISTS MyDatabase")

print("Database Created Successfully")

# Close connection

cursor.close()

conn.close()

**4. DDL (Data Definition Language) Statements**

DDL commands include CREATE, ALTER, DROP, and TRUNCATE.

**Creating a Table**

conn = mysql.connector.connect(

host="localhost",

user="root",

password="password",

database="MyDatabase"

)

cursor = conn.cursor()

# Create a table

cursor.execute("""

CREATE TABLE IF NOT EXISTS Users (

id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(100),

email VARCHAR(100) UNIQUE,

age INT

)

""")

print("Table Created Successfully")

cursor.close()

conn.close()

**Altering a Table**

cursor = conn.cursor()

cursor.execute("ALTER TABLE Users ADD COLUMN phone VARCHAR(15)")

print("Column Added Successfully")

cursor.close()

**Dropping a Table**

cursor = conn.cursor()

cursor.execute("DROP TABLE IF EXISTS Users")

print("Table Dropped Successfully")

cursor.close()

**5. DML (Data Manipulation Language) Statements**

DML includes INSERT, UPDATE, DELETE, and SELECT.

**Insert Data**

conn = mysql.connector.connect(

host="localhost",

user="root",

password="password",

database="MyDatabase"

)

cursor = conn.cursor()

# Insert data

sql = "INSERT INTO Users (name, email, age) VALUES (%s, %s, %s)"

values = [

("Alice", "alice@example.com", 25),

("Bob", "bob@example.com", 30)

]

cursor.executemany(sql, values)

conn.commit()

print(cursor.rowcount, "Records Inserted")

cursor.close()

conn.close()

**Update Data**

cursor = conn.cursor()

sql = "UPDATE Users SET age = %s WHERE name = %s"

values = (28, "Alice")

cursor.execute(sql, values)

conn.commit()

print(cursor.rowcount, "Record Updated")

cursor.close()

**Delete Data**

cursor = conn.cursor()

sql = "DELETE FROM Users WHERE name = %s"

values = ("Bob",)

cursor.execute(sql, values)

conn.commit()

print(cursor.rowcount, "Record Deleted")

cursor.close()

**Select Data**

cursor = conn.cursor()

cursor.execute("SELECT \* FROM Users")

# Fetch all records

records = cursor.fetchall()

for row in records:

print(row)

cursor.close()

**6. TCL (Transaction Control Language) Statements**

TCL includes COMMIT, ROLLBACK, and SAVEPOINT.

**Using COMMIT**

conn = mysql.connector.connect(

host="localhost",

user="root",

password="password",

database="MyDatabase"

)

cursor = conn.cursor()

try:

sql = "INSERT INTO Users (name, email, age) VALUES (%s, %s, %s)"

values = ("Charlie", "charlie@example.com", 35)

cursor.execute(sql, values)

conn.commit() # Commit the transaction

print("Transaction Committed")

except:

conn.rollback() # Rollback if error occurs

print("Transaction Rolled Back")

cursor.close()

conn.close()

**Using SAVEPOINT & ROLLBACK**

cursor = conn.cursor()

# Start Transaction

conn.start\_transaction()

try:

cursor.execute("INSERT INTO Users (name, email, age) VALUES ('David', 'david@example.com', 40)")

conn.commit()

cursor.execute("SAVEPOINT sp1") # Create savepoint

cursor.execute("INSERT INTO Users (name, email, age) VALUES ('Eve', 'eve@example.com', 32)")

# Rollback to savepoint

conn.rollback() # Rolls back the second insert but keeps the first

print("Rolled Back to Savepoint")

except:

conn.rollback()

print("Transaction Rolled Back Completely")

cursor.close()

conn.close()

**Summary**

* **Database Connection:** Establish connection with MySQL.
* **Create Database:** Use CREATE DATABASE to create a new database.
* **DDL Statements:** Define and manage schema (CREATE, ALTER, DROP).
* **DML Statements:** Manipulate data (INSERT, UPDATE, DELETE, SELECT).
* **TCL Statements:** Manage transactions (COMMIT, ROLLBACK, SAVEPOINT).