**MYSQL :**

**1. Create connection :**

Import mysql.connector

Mydb = mysql.connector.connect(

host= “localhost”,

user = “root”,

passwd= “\*\*\*\*\*”,

**database="mydatabase"**

)

Print(mydb)

**2**. **Create data base :**

mycursor = mydb.cursor()

mycursor.execute(“CREATE DATABASE mydatabase\_name”)

**3**. **Show data base :**

mycursor.execute(“SHOW DATABASES”)

for x in mycursor:

print(x)

**4**. **Create Table :**

mycursor.execute(“CREATE TABLE customers (name VARCHAR(255) , address VARCHAR(255))”)

**5**. **Check if Table exists :**

mycursor.execute("SHOW TABLES")  
  
for x in mycursor:  
  print(x)

**6. Create Table with Primary Key :**

mycursor.execute(“CREATE TABLE customers (id INT AUTO\_INCREMENT PRIMARY KEY, name VARCHAR(255) , address VARCHAR(255))”)

**7. Adding Primary Key to the existing table:**

mycursor.execute(“ALTER TABLE customers ADD COLUMN id INT AUTO\_INCREMENT PRIMARY KEY” )

**8. Insert into the Table :**

Sql = “INSERT INTO customers (name, address) VALUES (%s, %s) ”)

val = (“John”, “HighWay 21”)

mycursor.execute(sql, val)

mydb.commit()

print(mycursor.rowcount, “record inserted…”)

**9. Insert multiple values into the Table :**

Sql = “INSERT INTO customers (name, address) VALUES (%s, %s) ”)

val = [  
  ('Peter', 'Lowstreet 4'),  
  ('Amy', 'Apple st 652'),  
  ('Hannah', 'Mountain 21'),

]

mycursor.executemany(sql, val)

mydb.commit()

print(mycursor.rowcount, “record inserted…”)

**10. Get Inserted ID of Last Inserted Row :**

print("1 record inserted, ID:", mycursor.lastrowid)

**11. Select all from the Table :**

mycursor.execute("SELECT \* FROM customers")  
  
myresult = mycursor.fetchall()  
  
for x in myresult:  
  print(x)

**12. Select columns from the Table :**

mycursor.execute("SELECT name, address FROM customers")  
  
myresult = mycursor.fetchall()  
  
for x in myresult:  
  print(x)

**13. Select only First Row from the Table :**

mycursor.execute("SELECT \* FROM customers"

myresult = mycursor.fetchone()

print(myresult)

**14. Select with a filter using WHERE :**

Sql = SELECT \* FROM customers WHERE address= ‘Park Lane’ ”

mycursor.execute(sql)

myresult = mycursor.fetchall()

for x in myresult:

print(x)

**15. Select with a filter using WHERE and wildcard character :**

Sql = “SELECT \* FROM customers WHERE address LIKE ‘%way%’ ”

**16. Select with a filter and Prevent SQL Injection:**

sql = "SELECT \* FROM customers WHERE address = %s"  
adr = ("Yellow Garden 2", )

mycursor.execute(sql, adr)

**17.** **SORT the Result using ORDER BY () ascending order :**

sql = "SELECT \* FROM customers ORDER BY name"

**18.** **SORT the Result using ORDER BY () descending order :**

sql = "SELECT \* FROM customers ORDER BY name DESC"

**16. Delete Record :**

sql = "DELETE FROM customers WHERE address = 'Highway' "

**17. Delete Record and prevent SQL Injection :**

sql "DELETE FROM customers WHERE address = %s"

adr = ("highway 2",)

**18. Delete or Drop Table :**

sql = "DROP TABLE customers"

**19. Delete or Drop Table only if exists :**

sql = "DROP TABLE IF EXISTS customers"

**20. Update Table :**

sql= "UPDATE customers SET address = 'Promenade' WHERE address = 'D-Piazza' "

**21. Update Table and Prevent SQL: Injection :**

sql= "UPDATE customers SET address = %s WHERE address = %s"

val = ("1-world", "1-sky")

mycursor..execute(sql,val)

**22. Limit the Result :**

sql = "SELECT \* FROM customers LIMIT 5"

**23. Limit the Result but start from another position instead of starting :**

sql = "SELECT \* FROM customers LIMIT 5 OFFSET 2"

**24. Join two or more Tables :**

sql = "SELECT \

users.name AS user, \

products.fav AS favourite \

FROM users \

INNER JOIN products ON users.fav = products.id"

**25. LEFT Join :**

sql = "SELECT \

users.name AS user, \

products.fav AS favourite \

FROM users \

LEFT JOIN products ON users.fav = products.id"

**26. RIGHT Join :**

sql = "SELECT \

users.name AS user, \

products.fav AS favourite \

FROM users \

RIGHT JOIN products ON users.fav = products.id"