**Types of SQL Commands**

**DDL (Data Definition Language):** Create, alter, rename, truncate & drop.

**DQL (Data Query Language):** Select

**DML (Data Manipulation Language):** insert, update & delete

**DCL (Data Control Language):** grant & revoke permission to uses

**TCL (Transaction Control Language):** start transaction, commit & rollback

**List of Query**

1. **CREATE QUERY**

Create database db\_name;

Create database if not exists db\_name;

Create table table\_name(

Column\_name1 datatype constraint,

Column\_name1 datatype constriant

);

1. **SHOW QUERY**

Show databases;

Show tables;

1. **DROP QUERY**

Drop database db\_name;

Drop table table\_name;

1. **USE QUERY**

Use College;

1. **INSERT QUERY**

Insert intostudents(id, name, age) values(01, “Haroon”, 22);

1. **SELECT QUERY**

Select \* from table\_name;

Select col1,col2 from table\_name;

Select distinct col\_name from table\_name; remove duplicate values

1. **WHERE CLAUSE**

Select \* from table\_name where marks > 80;

Select \* from table\_name where city = “Faisalabad”;

**Comparison Operators**

1. **(=) EQUAL TO**

Select \* from table\_name where marks = 82;

1. **(!=) NOT EQUAL TO**

Select \* from table\_name where marks != 82;

1. **(>) GREATER THAN**

Select \* from table\_name where marks > 70;

1. **(<) LESS THAN**

Select \* from table\_name where marks < 80;

1. **(>=) GREATER THAN EQUAL TO**

Select \* from table\_name where marks >= 80;

1. **(<=) LESS THAN EQUAL TO**

Select \* from table\_name where marks <= 85;

**Logical Operators**

1. **AND STATEMENT**

Select \* from table\_name where marks > 80 and city = “Faisalabad”;

1. **OR STATEMENT**

Select \* from table\_name where marks > 80 or city = “Faisalabad”;

1. **BETWEEN STATEMENT**

Select \* from table\_name where marks between 70 and 90;

1. **IN STATEMENT**

Select \* from table\_name where city in (“Faisalabad”, “Lahore”);

1. **NOT STATEMENT (invert)**

Select \* from table\_name where city not in (“Faisalabad”, “Lahore”);

1. **LIMIT CLAUSE**

Select \* from table\_name limit 3;

Select \* from table\_name where marks > 75 limit 3;

1. **ORDER BY CLAUSE**

Select \* from table\_name order by marks ASC;

Select \* from table\_name order by marks DESC;

**Aggregate Functions**

1. **MAX () STATEMENT**

Select max(marks) from students;

1. **MIN () STATEMENT**

Select min(marks) from students;

1. **AVG () STATEMENT**

Select avg(marks) from students;

1. **SUM ()** STATEMENT

Select sum(marks) from students;

1. **COUNT () STATEMENT**

Select count(marks) from students;

1. **GROUP BY CLAUSE**

Select city, count(name) from students group by city;

1. **HAVING CLAUSE**

Select city, max(marks) from students group by city having max(marks) > 90;

**Joins**

1. **INNER JOIN**

Select \*/ col from tableA inner join tableB on tableA.col\_name = tableB.col\_name;

1. **LEFT JOIN**

Select \*/ col from tableA left join tableB on tableA.col\_name = tableB.col\_name;

1. **RIGHT JOIN**

Select \*/ col from tableA right join tableB on tableA.col\_name = tableB.col\_name;

1. **FULL JOIN**

Select \*/ col from tableA left join tableB on tableA.col\_name = tableB.col\_name

union

Select \*/ col from tableA right join tableB on tableA.col\_name = tableB.col\_name;

1. **LEFT EXCLUSIVE JOIN**

Select \*/ col from tableA left join tableB on tableA.col\_name = tableB.col\_name

Where tableB.col\_name is null;

1. **RIGHT EXCLUSIVE JOIN**

Select \*/ col from tableA right join tableB on tableA.col\_name = tableB.col\_name

Where tableA.col\_name is null;

1. **FULL EXCLUSIVE JOIN**

Select \*/ col from tableA left join tableB on tableA.col\_name = tableB.col\_name

Where tableB.col\_name is null

Union

Select \*/ col from tableA right join tableB on tableA.col\_name = tableB.col\_name

Where tableA.col\_name is null;

1. **UNION**

Select col from tableA

Union

Select col from tableB

1. **UNION ALL**

Select col from tableA

Union all

Select col from tableB

1. **UPDATE QUERY**

Update students set grade = “O” where grade = “A”;

1. **SAFE QUERY**

Set sql\_safe\_updates = 0; or 1

1. **DELETE QUERY**

Delete from table\_name ;

Delete from table\_name where marks < 33; deletes rows data

1. **TRUNCATE QUERY**

Truncate table table\_name ;

1. **ALTER QUERY**
2. **ADD COLUMN**

Alter table table\_name add column column\_name int;

1. **DROP COLUMN**

Alter table table\_name drop column column\_name;

1. **RENAME TABLE**

Alter table table\_name rename to new\_table\_name;

1. **CHANGE COLUMN (rename)**

Alter table table\_name change column old\_name new\_name new\_datatype int;

1. **MODIFY COLUMN (modify datatype)**

Alter table table\_name modify column column\_name new\_datatype;

1. **SUB QUERY**

Select column(s) from table\_name where col\_name operator (subquery);

* **FOREIGN KEY**

Id int,

Foreign key (id) references table\_name (primary key)

* **CASCADING FOR FK**

Id int,

Foreign key (id) references table\_name (primary key)

On delete cascade

On update cascade

* **MYSQL VIEWS**

Create view view\_name as

select col1, col2, col3… able\_name from table\_name;

select \* from view\_name;

**Python-Mysql Database Connectivity**

**CREATE DATABASE**

*import* mysql.connector

cnx = mysql.connector.connect(*user*='root', *password*='1056',

*host*='127.0.0.1')

db\_cursor = cnx.cursor()

db\_cursor.execute("create database database\_name")

cnx.close()

**CREATE TABLE**

*import* mysql.connector

cnx = mysql.connector.connect(*user*='root', *password*='1056',

*host*='127.0.0.1',

*database*='testfile')

db\_cursor = cnx.cursor()

db\_cursor.execute("create table Emp(id int, name varchar(100))")

cnx.close()

**SELECT QUERY**

*import* mysql.connector

cnx = mysql.connector.connect(*user*='root', *password*='1056',

*host*='127.0.0.1',

*database*='testfile')

db\_cursor = cnx.cursor()

query= "select \* from emp"

db\_cursor.execute(query)

db\_select = db\_cursor.fetchall()

print(db\_select)

cnx.close()

**INSERT QUERY**

*import* mysql.connector

cnx = mysql.connector.connect(*user*='root', *password*='1056',

*host*='127.0.0.1',

*database*='testfile')

db\_cursor = cnx.cursor()

db\_cursor.execute("insert into emp(id, name) values(%s, %s)", (1, "Haroon"))

cnx.commit()

cnx.close()

**MUL ITEM INSERT**

*import* mysql.connector

cnx = mysql.connector.connect(*user*='root', *password*='1056',

*host*='127.0.0.1',

*database*='testfile')

db\_cursor = cnx.cursor()

query = "insert into emp(id, name) values(%s, %s)"

insert\_list = [(2,"Haroon"), (3, "Awais")]

db\_cursor.executemany(query,insert\_list)

cnx.commit()

cnx.close()

**UPDATE QUERY/delete**

*import* mysql.connector

cnx = mysql.connector.connect(*user*='root', *password*='1056',

*host*='127.0.0.1',

*database*='testfile')

db\_cursor = cnx.cursor()

query = "update emp set id = %s where name = %s"

update = (4,"Awais")

db\_cursor.execute(query, update)

cnx.close()

**TRUNCATE QUERY**