**List of SQL Queries:**

1. mysql -u root -p : To login to MySQL.
2. use db1 : To change the database.
3. show databases; : To see all existing databases.
4. show tables; : To see all the existing tables.
5. create table *table\_name*(); : To create table.

For eg.,

CREATE TABLE EMPLOYEE (

empId INTEGER PRIMARY KEY,

name TEXT NOT NULL,

dept TEXT NOT NULL

);

1. insert into *table\_name* values (val1, val2, val3s); : To insert the values into table.

For eg.,

INSERT INTO EMPLOYEE VALUES (0001, 'Clark', 'Sales');

INSERT INTO EMPLOYEE VALUES (0002, 'Dave', 'Accounting');

INSERT INTO EMPLOYEE VALUES (0003, 'Ava', 'Sales');

1. select \* from *table\_name* : To view the table.
2. desc *table\_name* : To describe the table i.e. to see all the columns and its data types and all.
3. rename table *old\_name* to *new\_name* : Used to rename the table.
4. alter table *table\_name* add *column\_name* *datatype* : Used to add any column.
5. alter table *table\_name* drop *column\_name* : Used to delete any column.
6. alter table *table\_name* modify *column\_name* *new\_datatype* : Used to change the datatype of any column.
7. alter table *table\_name* rename column *old\_name* to *new\_name* : Used to rename any column.
8. delete from *table\_name* where *column\_name*=*specified\_value*; : Used to delete the row of the table whose specific value of any column is known.
9. update *table\_name* set *column\_name*=*any\_value* where *column\_name2*=existing*\_value*; : To update the existing row by adding some value to another column of that particular row.

**SQL :**

1. DDL (Data Definition Language) : Works with the structure of table
2. Create -
3. Alter - To modify the table.
4. Drop - To delete the structure and data, free up the storage space.
5. Truncate -
6. DML (Data Manipulation Language) :
7. Insert -
8. Update -
9. Delete -
10. DCL (Data Control Language) :
11. TCL (Transaction Control Language) :
12. DQL (Data Query Language) :

**SQL Constraints:**

1. NOT NULL - If it is applied then that particular attribute cannot have null value.
2. PRIMARY KEY - If this constraint is applied then, neither NULL nor duplicate data allowed.
3. UNIQUE - No duplicate data allowed.
4. CHECK - Used to check the condition

For eg. CHECK(AGE>180)

CHECK(SALARY>25000)

1. FOREIGN KEY -

* Example of these constraints:

create table Customers

(

id int PRIMARY KEY,

last\_name varchar(40) NOT NULL,

first\_name varchar(40) NOT NULL,

city varchar(40),

FOREIGN KEY (id) references Persons(p\_id)

);

**SQL Operators:**

1. Arithmetic Operators: +, -, \*, /
2. Relational Operators: >, >=, <, <=
3. Logical Operators: AND, OR, BETWEEN, LIKE, IN, NOT IN

For eg.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Id | Name | Age | Country | Dept |
| 1 | abc | 18 | IN | CSE |
| 2 | bcd | 20 | EN | IT |
| 3 | cde | 21 | USA | CSE |
| 4 | def | 19 | IN | IT |
| 5 | efg | 17 | AUS | CSE |
| 6 | fgh | 18 | IN | ECE |

For getting details of student who are from India and CSE Dept.

select \* from Student where Country=’IN’ AND Dept=’CSE’;

For getting details of student who are from India or America.

select \* from Student where Country=’IN OR Country=’USA’;

For getting details of student whose age lies between 18 to 20.

select \* from Student where age BETWEEN 18 AND 20;

For getting details of student whose name ends with ‘g’.

select \* from Student where NAME LIKE ’%g’;

For getting details of student whose name starts with ‘a’.

select \* from Student where NAME LIKE ’a%’;

For getting details of student whose name starts with ‘e’ and ends with ‘g’.

select \* from Student where NAME LIKE ’e%g’;

For getting details of student whose name contains ‘ef’.

select \* from Student where NAME LIKE ’%ef%’;

LIKE has two types:

1. LIKE with % : For 0, 1 or multiple characters
2. LIKE with \_ : For single character matching

To get distinct value of any attribute i.e. to ignore same value:

select DISTINCT Name from STUDENT;

To get data is ascending order:

select name from Student ORDER BY name;

To get data is descending order:

select name from Student ORDER BY name DESC;

For getting name of students who belongs to CSE Department.

select name from Student where dept IN (‘CSE’) ;

For getting name of students who belongs to any department.

select name from Student where dept IN (‘CSE’, ‘IT’, ‘ECE’) ;

For getting details of student whose name is in the of form \_e\_.

select \* from Student where name LIKE ‘\_e\_’ ;

To increase the salary of the employees by 1.5 times from table EMPLOYEE:

select Salary\*1.5 as total\_sal from EMPLOYEE;

Here Salary is old name of that column and then after updating the values, column name is also updated to total\_sal.