What is SQL?

* SQL stands for Structured Query Language
* SQL lets you access and manipulate databases
* SQL became a standard of the American National Standards Institute (ANSI) in 1986, and of the International Organization for Standardization (ISO) in 1987

What Can SQL do?

* SQL can execute queries against a database
* SQL can retrieve data from a database
* SQL can insert records in a database
* SQL can update records in a database
* SQL can delete records from a database
* SQL can create new databases
* SQL can create new tables in a database
* SQL can create stored procedures in a database
* SQL can create views in a database
* SQL can set permissions on tables, procedures, and views
* SELECT - extracts data from a database
* UPDATE - updates data in a database
* DELETE - deletes data from a database
* INSERT INTO - inserts new data into a database
* CREATE TABLE - creates a new table
* ALTER TABLE - modifies a table
* DROP TABLE - deletes a table

1. Sql select statement

The SELECT statement is used to select data from a database.

The data returned is stored in a result table, called the result-set.

**select \* from test;**

2.Select with Distinct statement

The following SQL statement selects only the DISTINCT values from the "Country" column in the "Customers" table:

**select distinct city from test;**

3. Sql Where clause

The SQL WHERE Clause

The WHERE clause is used to filter records.

It is used to extract only those records that fulfill a specified condition.

**SELECT \* FROM test WHERE city='Srinagar';**

4.Sql and, or not

The WHERE clause can be combined with AND, OR, and NOT operators.

The AND and OR operators are used to filter records based on more than one condition:

* The AND operator displays a record if all the conditions separated by AND are TRUE.
* The OR operator displays a record if any of the conditions separated by OR is TRUE.

The NOT operator displays a record if the condition(s) is NOT TRUE.

4.1 And

The following SQL statement selects all fields from "test" where city is "Srinagar" AND salary is "80000":

**SELECT \* FROM test WHERE city='Srinagar' AND salary='80000';**

4.2 Or

The following SQL statement selects all fields from "test" where city is "Srinagar" OR "Panjim":

**SELECT \* FROM test WHERE city='Srinagar' or city='Panjim';**

4.3 Not

The following SQL statement selects all fields from "test" where country is NOT "Srinagar":

**SELECT \* FROM test WHERE NOT city='Srinagar';**

5 The SQL ORDER BY Keyword

The ORDER BY keyword is used to sort the result-set in ascending or descending order.

The ORDER BY keyword sorts the records in ascending order by default. To sort the records in descending order, use the DESC keyword.

**SELECT \* FROM test ORDER BY city ASC;**

6 insert

The INSERT INTO statement is used to insert new records in a table.

INSERT INTO Syntax

It is possible to write the INSERT INTO statement in two ways:

1. Specify both the column names and the values to be inserted:

**insert into test (id,name,city,salary,Mobile\_no) values(5,"Rakesh","Pune","100000","546321098");**

2. Specify only values to be inserted:

**insert into test values(6,'Bhavesh','Sikkim','40000','9638527410');**

7 Update

The UPDATE statement is used to modify the existing records in a table.

**update test set city='Banglore',salary='80000',Mobile\_no='9996321450' where id=6;**

8 delete

The DELETE statement is used to delete existing records in a table.

**delete from test where id=6;**

if you not specified the where id=6 then it will delete all record

**delete from test;**

9 min and max

The MIN() function returns the smallest value of the selected column.

**SELECT MIN(salary)**

**FROM test;**

The MAX () function returns the largest value of the selected column.

**SELECT MAX(salary)**

**FROM test;**

10 Count,avg and sum

The COUNT() function returns the number of rows that matches a specified criterion.

**SELECT count(salary) FROM test;**

The AVG() function returns the average value of a numeric column.

**SELECT avg(salary) FROM test;**

The SUM() function returns the total sum of a numeric column.

**SELECT sum(salary) FROM test;**

11 Join

A JOIN clause is used to combine rows from two or more tables, based on a related column between them.  
Here are 2 table employee and emppost

**SELECT employee.id, employee.empname,emppost.post**

**FROM employee**

**JOIN emppost ON employee.id=emppost.id;**

11.1 Inner join

The INNER JOIN keyword selects records that have matching values in both tables

**SELECT employee.id, employee.empname,emppost.post**

**FROM employee**

**INNER JOIN emppost ON employee.id=emppost.id;**

11.2 left join

The LEFT JOIN keyword returns all records from the left table (table1), and the matching records from the right table (table2). The result is 0 records from the right side, if there is no match

**SELECT employee.id, employee.empname,emppost.post**

**FROM employee**

**left JOIN emppost ON employee.id=emppost.id;**

11.3 right join

The RIGHT JOIN keyword returns all records from the right table (table2), and the matching records from the left table (table1). The result is 0 records from the left side, if there is no match

**SELECT employee.id, employee.empname,emppost.post**

**FROM employee**

**left JOIN emppost ON employee.id=emppost.id;**

11.4 Full join

The FULL OUTER JOIN keyword returns all records when there is a match in left (table1) or right (table2) table records

**SELECT employee.id, employee.empname,emppost.post**

**FROM employee**

**full outer JOIN emppost ON employee.id=emppost.id;**

12 Create table

The CREATE DATABASE statement is used to create a new SQL database.

The following example creates a table called "Persons" that contains five columns: PersonID, LastName, FirstName, Address, and City:

CREATE TABLE Persons (

**PersonID int,**

**LastName varchar(255),**

**FirstName varchar(255),**

**Address varchar(255),**

**City varchar(255)**

**);**

13 Drop table

The DROP TABLE statement is used to drop an existing table in a database.

**drop table persons;**

14 Alter table

**ALTER TABLE persons**

**ADD Email varchar(255)**

15 Primary key

The following SQL creates a PRIMARY KEY on the "ID" column when the "Persons" table is created:

**CREATE TABLE Person (**

**ID int NOT NULL,**

**LastName varchar(255) NOT NULL,**

**FirstName varchar(255),**

**Age int,**

**PRIMARY KEY (ID)**

**);**

16 Check key

The CHECK constraint is used to limit the value range that can be placed in a column.

If you define a CHECK constraint on a column it will allow only certain values for this column.

If you define a CHECK constraint on a table it can limit the values in certain columns based on values in other columns in the row.

**CREATE TABLE Persons (**

**ID int NOT NULL,**

**LastName varchar(255) NOT NULL,**

**FirstName varchar(255),**

**Age int CHECK (Age>=18)**

**);**

17 AUTO INCREMENT Field

Auto-increment allows a unique number to be generated automatically when a new record is inserted into a table.

Often this is the primary key field that we would like to be created automatically every time a new record is inserted

**CREATE SEQUENCE seq\_person**

**MINVALUE 1**

**START WITH 1**

**INCREMENT BY 1**

**CACHE 10;**