**Important SQL Queries:**

* GUI client: User Interface
* Command Line client: Command Prompt.

**Commands**

* Show databases -> list out all databases.
* Use <databasename> -> point towards particular database.
* Describe >tablename> -> displays structure of table.

**DISTINCT:**  returns only unique values from specified column.

* select DISTINCT region from country;

**WHERE:** is used to filter records.

* select <columnname> from <tablename> WHERE condition;

select \* from country WHERE name='India'; [ single quote for character].

Used with =, >, <, >=, <=, <>, BETWEEN, LIKE, IN operators.

**AND, OR, NOT operator:** is used to filter records based on more than one condition.

* select \* from <tablename> WHERE condition1 **AND** condition2;
* select \* from <tablename> WHERE condition1 **OR** condition2;
* select \* from <tablename> WHERE **NOT** condition;

**LIMIT:** used to limit the records to display number of rows.

* select \* from country **LIMIT** 10**;** -> displays 10 records.;

**ORDER BY:** used to sort result-set in ascending or descending order.

* select \* from country ORDER BY <columnname> ASC/DESC;

select \* from country ORDER BY localname DESC;

select \* from country ORDER BY localname DESC, name ASC;

**IS NULL, IS NOT NULL:**  used to test for empty values. [ not possible with **comparison operators**.]

* select region from country IS NULL; -> returns null values records.
* select \* from country IS NOT NULL; -> returns not null records.

**TOP, ROWNUM:** limits number of records. Similar to LIMIT.

* select TOP 5 \* from <tablename> -> retrieves top 5 records.
* select \* from <tablename> WHERE ROWNUM =5; -> retrieves top 5 records.
* Select TOP 3 \* from <tablename> WHERE country=’Germany’; -> top 3 records of cond.

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

* select MIN(columnname) from <tablename> where cond. – returns smallest value record.

**MAX()** functionreturns the largest value of selected column.

* select MAX(columnname) from <tablename> where cond. – returns largest value record.

**COUNT()** function: returns the number of rows that matched a condiotnon-

* select COUNT(cloumnname) from <tablename> where marks<=50;

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

* select AVG(cloumnname) from <tablename> WHERE condition;

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

* select SUM(cloumnname) from <tablename> WHERE condition;

**LIKE** operator: used to search for specified pattern in the column.

* WHERE <columnname> LIKE ‘a%’ -> finds any values start with ‘a’.
* WHERE <columnname> LIKE ‘%a’ -> finds any values end with ‘a’.
* WHERE <columnname> LIKE ‘%a%’ -> finds any values have ‘a’ at any position.
* WHERE <columnname> LIKE ‘a\_%’ -> finds any values start with ‘a’ and are 3 chars.
* WHERE <columnname> LIKE ‘a%o’ -> finds any values start with ‘a’ and ends with ‘o’.

**Wildcards Characters:** used to substitute one or more characters in a String.

* \* -> represents zero or more characters. (bl\* finds all bl chars in String).
* ? -> represent a single character. (h?t finds hot, hat and hit).
* [] -> represents any single character within brackets. ( h[oa] finds hot, hat, but not hit).
* ! -> represents any character not in brackets. (h[oa] finds hit, but not hat and hot).
* - > represents a range of character. ( c[a-b]t finds cat and cbt).
* # -> represents any single numeric character. (2#5 finds 202,215.225 etc.).

**IN**, **NOT IN** operator: allows to specify multiple values in a WHERE clause.

* Select <columnname> from <tablename> WHERE columnname IN ( val1, val2);

SELECT \* FROM Customers WHERE Country IN (SELECT Country FROM Suppliers);

**BETWEEN** operator: selects values within a given range.

* SELECT \* FROM Products WHERE Price BETWEEN 10 AND 20;
* SELECT \* FROM Products WHERE Price BETWEEN 10 AND 20 AND NOT CategoryID IN (1,2,3);

**Aliases**: used to give a table or column in table a temporary name. (exists for duration of the query).

* SELECT <columnname> AS <aliasname> FROM tablename;
* SELECT <columnname(s)> FROM <tablename> AS <aliasname>;

**GROUP BY**: is a statement groups rows that have the same values into summary rows,

* SELECT COUNT(CustomerID), Country FROM Customers GROUP BY Country;
* SELECT COUNT(CustomerID), Country FROM Customers GROUP BY Country ORDER BY COUNT(CustomerID) DESC;

**HAVING**: WHERE keyword could not be used with aggregate functions.

* SELECT column\_name(s) FROM table\_name WHERE condition GROUP BY column\_name(s) HAVING condition ORDER BY column\_name(s);

**EXISTS**: used to test for the existence of any record in a subquery. Return true if condition is meet.

**ANY, ALL**: return true, if ANY subquery values meet condition whereas ALL of subquery meets

**JOIN**: is used to combine rows from two or more tables based on a related column between them.

INNER JOIN: returns all records that having matching values in both tables.

* SELECT Orders.OrderID, Customers.CustomerName, Orders.OrderDateFROM Orders

INNER JOIN Customers ON Orders.CustomerID=Customers.CustomerID;

LEFT (OUTER) JOIN: returns all records from the left table and the matched records for right table.

* SELECT column\_name(s) FROM table1 LEFT JOIN table2 ON table1.column\_name = table2.column\_name;

RIGHT (OUTER) JOIN: returns all records from the right table and matched records for left table.

* SELECT column\_name(s) FROM table1 RIGHT JOIN table2 ON table1.column\_name = table2.column\_name;

FULL (OUTER) JOIN: returns all records when there is a match in either left or right table.

* SELECT column\_name(s) FROM table1 FULL OUTER JOIN table2 ON table1.column\_name = table2.column\_name WHERE condition;

SELF JOIN: is a regular join but the table is joined with itself

* SELECT column\_name(s) FROM table1 T1, table1 T2 WHERE condition;

UNION operator: is used to combine the result of two or more select statements.

* SELECT column\_name(s) FROM table1 UNION SELECT column\_name(s) FROM table2;

**Views:** view is a virtual table based on the result-set of an SQL statement.

**Most common SQL Interview Questions:**

Q. Return the employee with the highest salary?

Select \* from emp where salary = (select max(salary) from emp)

Q. Return the highest salary of the employee?

Select max(salary) from emp

Q. Return 2nd highest salary from employee?

Select max(salary) from emp where salary Not In (Select max(salary) from emp)

Or

Select mac(salary) from emp where salary <(select max(salary) from emp)

Or

Select salary from (select DISTINCT TOP 2 salary from emp ORDER BY salary Desc ) Result Order by salary

Q. Return the range of employee based on EMP\_ID?

Select \* from emp where emp\_id between 2001 and 2004

Q. Return emp\_name, highest salary and department (emp table and department table)?

Select e.first\_name, e.last\_name, e.salary, d.depatment\_name from Emp e Inner Join Department d ON e.department\_id =d.department\_id) where salary IN (select max(salary) from emp)

Q. Return emp\_name, salary department\_name of each department?

Select e.first\_name, e.last\_name, e.salary, d.depatment\_name from Emp e Inner Join Department d ON e.department\_id =d.department\_id) where salary IN (select max(salary) from emp GRPUP BY department\_id)

Q. Return nth highest salary of employee?

Select TOP 1 salary from (select DISTINCT TOP ***5*** salary from emp ORDER BY salary Desc ) Result Order by salary