**SQL**

**Q1->What is SQL?**

SQL is Structured Query Language, which is a computer language for storing, manipulating and retrieving data stored in a relational database.

SQL is the standard language for Relational Database System. All the Relational Database Management Systems (RDMS) like MySQL, MS Access, Oracle, Sybase, Informix, Postgres and SQL Server use SQL as their standard database language.

SQL is widely popular because it offers the following advantages −

* Allows users to access data in the relational database management systems.
* Allows users to describe the data.
* Allows users to define the data in a database and manipulate that data.
* Allows to embed within other languages using SQL modules, libraries & pre-compilers.
* Allows users to create and drop databases and tables.
* Allows users to create view, stored procedure, functions in a database.
* Allows users to set permissions on tables, procedures and views.

**Q2->What is Join ?**

Join is a query that is used to combine rows from two or more tables, views, or materialized views. It retrieves data from multiple tables and creates a new table.

Join Conditions

There may be at least one join condition either in the FROM clause or in the WHERE clause for joining two tables. It compares two columns from different tables and combines pair of rows, each containing one row from each table, for which join condition is true.

Types of Joins

1. Inner Joins (Simple Join)
2. Outer Joins
3. Left Outer Join (Left Join)
4. Right Outer Join (Right Join)
5. Full Outer Join (Full Join)
6. Equijoins
7. Self Joins
8. Cross Joins (Cartesian Products)
9. Antijoins
10. Semijoins

**Syntax of inner join**

SELECT columns

FROM table1

INNER JOIN table2

ON table1.column = table2.column;

**Syntax of outer join**

SELECT columns

FROM table1

LEFT [OUTER] JOIN table2

ON table1.column = table2.column;

**Syntax of EQUI JOIN**

SELECT column\_list

FROM table1, table2....

WHERE table1.column\_name =

table2.column\_name;

**Q3->What are some common clauses used with SELECT query in SQL?**

Some common SQL clauses used in conjuction with a SELECT query are as follows:

* WHERE clause in SQL is used to filter records that are necessary, based on specific conditions.
* ORDER BY clause in SQL is used to sort the records based on some field(s) in ascending (ASC) or descending order (DESC).

SELECT \*

FROM myDB.students

WHERE graduation\_year = 2019

ORDER BY studentID DESC;

* GROUP BY clause in SQL is used to group records with identical data and can be used in conjunction with some aggregation functions to produce summarized results from the database.
* HAVING clause in SQL is used to filter records in combination with the GROUP BY clause. It is different from WHERE, since the WHERE clause cannot filter aggregated records.

SELECT COUNT(studentId), country

FROM myDB.students

WHERE country != "INDIA"

GROUP BY country

HAVING COUNT(studentID) > 5;

**Q4-> What are UNION, MINUS and INTERSECT commands?**

The UNION operator combines and returns the result-set retrieved by two or more SELECT statements.

The MINUS operator in SQL is used to remove duplicates from the result-set obtained by the second SELECT query from the result-set obtained by the first SELECT query and then return the filtered results from the first.

The INTERSECT clause in SQL combines the result-set fetched by the two SELECT statements where records from one match the other and then returns this intersection of result-sets.

Certain conditions need to be met before executing either of the above statements in SQL -

Each SELECT statement within the clause must have the same number of columns

The columns must also have similar data types

The columns in each SELECT statement should necessarily have the same order

SELECT name FROM Students

UNION

SELECT name FROM Contacts;

SELECT name FROM Students

UNION ALL

SELECT name FROM Contacts;

SELECT name FROM Students

MINUS

SELECT name FROM Contacts;

SELECT name FROM Students

INTERSECT

SELECT name FROM Contacts;

**Q5->what is Transaction ?**

A transaction is a logical, atomic unit of work that contains one or more SQL statements.

A transaction groups SQL statements so that they are either all committed, which means they are applied to the database, or all rolled back, which means they are undone from the database. Oracle Database assigns every transaction a unique identifier called a transaction ID.

All Oracle transactions obey the basic properties of a database transaction, known as ACID properties. ACID is an acronym for the following:

**Atomicity**

All tasks of a transaction are performed or none of them are. There are no partial transactions. For example, if a transaction starts updating 100 rows, but the system fails after 20 updates, then the database rolls back the changes to these 20 rows.

**Consistency**

The transaction takes the database from one consistent state to another consistent state. For example, in a banking transaction that debits a savings account and credits a checking account, a failure must not cause the database to credit only one account, which would lead to inconsistent data.

**Isolation**

The effect of a transaction is not visible to other transactions until the transaction is committed. For example, one user updating the hr.employees table does not see the uncommitted changes to employees made concurrently by another user. Thus, it appears to users as if transactions are executing serially.

**Durability**

Changes made by committed transactions are permanent. After a transaction completes, the database ensures through its recovery mechanisms that changes from the transaction are not lost.