**JOINS**

Joins are used to combine rows from two or more tables based on a related column between them. Joins are essential for querying and retrieving data from multiple tables simultaneously. There are several types of joins in SQL, each serving a different purpose. The common types of joins are:

***INNER JOIN:***

The INNER JOIN returns only the rows that have matching values in both tables. It filters out any rows where there is no match between the specified columns in the tables being joined.

***LEFT JOIN:***

The LEFT JOIN returns all the rows from the left table and the matching rows from the right table. If there is no match in the right table, the result will contain NULL values for the columns of the right table.

***RIGHT JOIN:***

The RIGHT JOIN is similar to the LEFT JOIN, but it returns all the rows from the right table and the matching rows from the left table. If there is no match in the left table, the result will contain NULL values for the columns of the left table.

***FULL JOIN:***

The FULL JOIN returns all the rows from both tables, including the unmatched rows from both tables. If there is no match for a row in one table, the corresponding columns from the other table will contain NULL values.

***CROSS JOIN:***

The CROSS JOIN returns the Cartesian product of the two tables, meaning it combines each row from the first table with every row from the second table. It does not require any matching column.

**Uses of Joins:**

1. Retrieving Related Data: Joins are commonly used to fetch related data from different tables. For example, if you have a "customers" table and an "orders" table, you can use a join to get information about orders placed by each customer.

2. Creating Views: Joins are often used to create views that encapsulate complex queries involving multiple tables. Views allow you to simplify queries and provide a more straightforward interface for end-users.

3. Aggregating Data: Joins can be used in combination with aggregate functions (e.g., SUM, COUNT, AVG) to perform calculations across multiple related tables. For example, you can calculate the total sales per product by joining the "products" and "order\_items" tables.

4. Data Cleansing: Joins can be used to identify and clean inconsistent data. For instance, if you have a table with customer information and another table with customer orders, you can use joins to find records with missing or incorrect data.

5. Business Reporting: In business intelligence and reporting, joins are used to combine data from various tables to generate comprehensive reports. For example, you can join tables containing sales data, product information, and customer details to create a sales report.

6. Data Integration: Joins are essential when integrating data from multiple sources. When you have data distributed across different tables or databases, joins help unify the data for analysis and processing.

7. Entity-Relationship Modeling: In database design, joins help establish relationships between different entities and ensure data integrity by enforcing referential integrity constraints.

8. Identifying Orphaned Records: Joins can be used to identify orphaned records in the database, which are records in one table with no matching records in another table. This can be useful for data validation and cleanup.

9. Filtering Data: Joins can be combined with filtering conditions to extract specific subsets of data from related tables. For example, you can join tables and filter results based on dates, categories, or other criteria.

10. Complex Data Manipulation: When working with complex queries involving multiple tables and data transformations, joins play a crucial role in assembling the data in the required format.