Step 1:

A relational database management system's information can be managed using the special-purpose programming language SQL (RDBMS). The essential word here is "relational," which denotes that the database management system is set up in a way that makes it evident how various sets of data are related to one another. Relational database systems are relational by virtue of join.

Step 2:

With just one SELECT statement, joins let you combine data from two or more tables into a single query result.

You must use the JOIN command to combine rows from several tables in a single query. The following should assist illustrate how joins differ from one another since there are several different kinds of them.

Using a single SELECT statement, joins enable you to combine data from two or more tables into a single query result.

A SQL SELECT query that has more than one table listed after the FROM keyword is known as a "Join."

For example:

SEARCH CONDITIONS: SELECT "list-of-columns" FROM table 1, table 2, AND

It is simpler to understand joins if you show what would happen if you only dealt with one table and weren't allowed to use "joins." A "flat table" is another name for this single-table database.

Step 3:

A related field between two or more tables is used to merge rows from those tables using the JOINS SQL statement. They are typically employed when a user is attempting to extract data from tables that include one-to-many or many-to-many connections.

There are various join types in SQL.

You mostly need to understand four different sorts of joins. As follows:

INNER JOIN

FULL JOIN

LEFT JOIN

RIGHT JOIN

Step 4:

INNER JOIN

Records with matching values in both tables are returned by this kind of join. Therefore, all the tuples with matching values in both tables will be output if you run an INNER join operation between the Employee table and the Projects table.

Syntax:

SELECT Table1.Column1,Table1.Column2,Table2.Column1,....

FROM Table1

INNER JOIN Table2

ON Table1. MatchingColumnName = Table2.MatchingColumnName;

FULL JOIN

A FULL JOIN, as contrast to an INNER JOIN, returns all the rows from both connected tables, whether or not they match. Consequently, a FULL JOIN is also known as a FULL OUTER JOIN. A FULL JOIN returns both the overlap between the two tables and the mismatched rows from both tables.

Syntax

FULL JOIN SELECT columns FROM table1 name ON table2 Table1 is the first table, Table2 is the second table, and column x and column y are the columns for executing a Left Join, followed by a semicolon. SELECT, FULL JOIN, and ON are the keywords, columns are the list of columns, table1 is the first table, and table2 is the second table.

LEFT JOIN

All of the rows from the left table and the matched rows from the right table are returned by the LEFT JOIN statement. If there is no match, the result is NULL on the right side.

All records from the left table (table1) and the matched records from the right table are returned via the LEFT JOIN keyword (table2). If there is no match, there are no records displayed on the right side.

The following is the general syntax of LEFT JOIN:

**SELECT** column\_list  **FROM** table\_name1

LEFT JOIN table\_name2

**ON** column\_name1 = column\_name2

**WHERE** join\_condition

OUTER JOIN

All records with values in either the left table or the right table are returned using the FULL OUTER JOIN (also known as OUTER JOIN). For instance, a full outer join between a table of customers and a table of orders may result in the return of all customers, including those who have no orders.

Unmatched rows in one or both tables may be returned in an outer join. A few different kinds of outer joins include: Only mismatched rows from the left table are returned by an LEFT JOIN. Only mismatched rows from the right table are returned by RIGHT JOIN. Unmatched rows from both tables are returned by FULL OUTER JOIN.

syntax : SELECT column\_name(s) FROM table1 LEFT JOIN Table2 ON Table1. Column\_Name=table2.