What are joins in SQL? State its types.

In SQL, joins are used to combine rows from two or more tables based on a related column between them. Joins are fundamental for querying relational databases as they allow you to retrieve related data spread across multiple tables.

Types of Joins

1. **INNER JOIN:**
   - **Description:** Returns only the rows that have matching values in both tables. It excludes rows that do not have a match in either table.
   - **Syntax:**

     ```
     SELECT columns
     FROM table1
     INNER JOIN table2
     ON table1.column = table2.column;
     ```

   - **Example:**

     ```
     SELECT e.first_name, d.dept_name
     FROM employee e
     INNER JOIN department d
     ON e.dept_id = d.id;
     ```

2. **LEFT JOIN (or LEFT OUTER JOIN):**
   - **Description:** Returns all rows from the left table and the matched rows from the right table. If there is no match, NULL values are returned for columns from the right table.
   - **Syntax:**

     ```
     SELECT columns
     FROM table1
     LEFT JOIN table2
     ON table1.column = table2.column;
     ```

   - **Example:**

     ```
     SELECT e.first_name, d.dept_name
     FROM employee e
     LEFT JOIN department d
     ON e.dept_id = d.id;
     ```

3. **RIGHT JOIN (or RIGHT OUTER JOIN):**

- **Description:** Returns all rows from the right table and the matched rows from the left table. If there is no match, NULL values are returned for columns from the left table.
- **Syntax:**

```
SELECT columns
FROM table1
RIGHT JOIN table2
ON table1.column = table2.column;
```

- **Example:**

```
SELECT e.first_name, d.dept_name
FROM employee e
RIGHT JOIN department d
ON e.dept_id = d.id;
```

4. **FULL JOIN (or FULL OUTER JOIN):**
   - **Description:** Returns all rows when there is a match in one of the tables. It returns NULL for rows that do not have a match in the other table.
   - **Syntax:**

```
SELECT columns
FROM table1
FULL JOIN table2
ON table1.column = table2.column;
```

   - **Example:**

```
SELECT e.first_name, d.dept_name
FROM employee e
FULL JOIN department d
ON e.dept_id = d.id;
```

5. **CROSS JOIN:**
   - **Description:** Returns the Cartesian product of the two tables, i.e., every combination of rows from the first table with every row from the second table.
   - **Syntax:**

```
SELECT columns
FROM table1
CROSS JOIN table2;
```

   - **Example:**

```
SELECT e.first_name, d.dept_name
FROM employee e
CROSS JOIN department d;
```

6. **SELF JOIN:**
    - **Description:** A regular join but the table is joined with itself. It's useful for querying hierarchical data or comparing rows within the same table.
    - **Syntax:**

```
SELECT a.columns, b.columns
FROM table a
INNER JOIN table b
ON a.column = b.column;
```

    - **Example:**

```
SELECT a.first_name AS Employee, b.first_name AS Manager
FROM employee a
INNER JOIN employee b
ON a.manager_id = b.id;
```

Define Cross join.

A CROSS JOIN (also known as a Cartesian Join) is a type of join in SQL that produces the Cartesian product of the two tables involved. This means that it returns every combination of rows from the first table with every combination of rows from the second table.