Joins

- In MySQL, joins are used to combine data from two or more tables in a relational database based on a related column between them.
- A join is performed using the JOIN clause in SQL queries, which allows you to retrieve data from multiple tables in a single query by specifying how the tables are related. There are several types of joins in MySQL:

1. INNER JOIN

- Also known as a simple join, an inner join returns only the rows that have matching values in both tables. It combines the rows from two or more tables based on a common column.
- Syntax :-

1) SELECT column1, column2, ...
2) FROM table1
3) JOIN table2 ON table1.column_name = table2.column_name;

2. LEFT JOIN

- A left join 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 the right table's columns.
- Syntax :-

1) SELECT column1, column2, ...
2) FROM table1
3) LEFT JOIN table2 ON table1.column_name = table2.column_name;

3. RIGHT JOIN

- A right join 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 the left table's columns.
- Syntax :-

1. SELECT column1, column2, ...
2. FROM table1
3. RIGHT JOIN table2 ON table1.column_name = table2.column_name;

4. FULL JOIN

- A full join returns all rows from both tables and NULL values for columns where there is no match.
- Syntax :-
- SELECT column1, column2, ...
- FROM table1
- FULL JOIN table2 ON table1.column_name = table2.column_name;

5. CROSS JOIN

- A cross join returns the Cartesian product of two or more tables, meaning it returns all possible combinations of rows from the joined tables.
- Syntax :-
- SELECT column1, column2, ...
- FROM table1
- CROSS JOIN table2;

❖ **Example**

1. CREATE DATABASE account;
2. CREATE TABLE customers (customer_id INT, customer_name VARCHAR(50), customer_email VARCHAR(50));
3. CREATE TABLE orders (order_id INT, customer_id INT, product_name VARCHAR(50), order_date DATE);
4. INSERT INTO customers VALUES
   (1, 'John Smith', 'john@example.com'),
   (2, 'Jane Doe', 'jane@example.com'),
   (3, 'Mark Johnson', 'mark@example.com');
5. INSERT INTO orders VALUES (1001, 1, 'Product A', '2023-04-10'),
   (1002, 1, 'Product B', '2023-04-11'),
   (1003, 2, 'Product C', '2023-04-12'),
   (1004, 3, 'Product A', '2023-04-13');

1. INNER JOIN

1) SELECT customers.customer_id, customers.customer_name, orders.order_id, orders.product_name
2) FROM customers
3) JOIN orders ON customers.customer_id = orders.customer_id;

2. LEFT JOIN

1) SELECT customers.customer_id, customers.customer_name, orders.order_id, orders.product_name
2) FROM customers
3) LEFT JOIN orders ON customers.customer_id = orders.customer_id;

3. RIGHT JOIN

1) SELECT customers.customer_id, customers.customer_name, orders.order_id, orders.product_name
2) FROM customers
3) RIGHT JOIN orders ON customers.customer_id = orders.customer_id;

4. FULL JOIN

1) SELECT customers.customer_id, customers.customer_name, orders.order_id, orders.product_name
2) FROM customers
3) FULL JOIN orders ON customers.customer_id = orders.customer_id;

5. CROSS JOIN

1) SELECT customers.customer_id, customers.customer_name, orders.order_id, orders.product_name
2) FROM customers
3) CROSS JOIN orders;