Tasks Assigned Today

27-02-2025
Suneetha.V

Joins:

1. INNER JOIN
2. LEFT JOIN (or LEFT OUTER JOIN)
3. RIGHT JOIN (or RIGHT OUTER JOIN)
4. FULL JOIN (or FULL OUTER JOIN)
5. CROSS JOIN
6. SELF JOIN

1. INNER JOIN

The `INNER JOIN` returns only the rows that have matching values in both tables. If there is no match, the row will not appear in the result.

Syntax:

```
SELECT columns

FROM table1

INNER JOIN table2

ON table1.column_name = table2.column_name;
```

Example:

```
SELECT Employees.EmployeeID, Employees.FirstName,
Employees.LastName, Departments.DepartmentName FROM Employees INNER
JOIN Departments ON Employees.DepartmentID =
Departments.DepartmentID;
```

Tasks Assigned Today

2. LEFT JOIN (LEFT OUTER JOIN)

The LEFT JOIN returns all the rows from the left table (the first table), and the matched rows from the right table (the second table). If there is no match, NULL values are returned for columns of the right table.

Syntax:

```
SELECT columns

FROM table1

LEFT JOIN table2

ON table1.column_name = table2.column_name;
```

Example:

```
SELECT Employees.EmployeeID, Employees.FirstName,
Employees.LastName, Departments.DepartmentName

FROM Employees

LEFT JOIN Departments

ON Employees.DepartmentID = Departments.DepartmentID;
```

3. RIGHT JOIN (RIGHT OUTER 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, NULL values are returned for columns of the left table.

Tasks Assigned Today

Syntax:

```
SELECT columns

FROM table1

RIGHT JOIN table2

ON table1.column_name = table2.column_name;
```

Example:

```
SELECT Employees.EmployeeID, Employees.FirstName,
Employees.LastName, Departments.DepartmentName

FROM Employees

RIGHT JOIN Departments

ON Employees.DepartmentID = Departments.DepartmentID;
```

4. FULL JOIN (FULL OUTER JOIN)

MySQL does not directly support FULL JOIN. However, you can achieve similar results by combining LEFT JOIN and RIGHT JOIN using UNION.

- A FULL JOIN returns all records when there is a match in either the left (table1) or right (table2) table. If there is no match, the result will contain NULL for columns from the table without a match.

Tasks Assigned Today

Syntax (Using UNION in MySQL):

```
SELECT columns

FROM table1

LEFT JOIN table2

ON table1.column_name = table2.column_name

UNION

SELECT columns

FROM table1

RIGHT JOIN table2

ON table1.column_name = table2.column_name;
```

Example:

```
SELECT Employees.EmployeeID, Employees.FirstName,
Employees.LastName, Departments.DepartmentName

FROM Employees

LEFT JOIN Departments

ON Employees.DepartmentID = Departments.DepartmentID

UNION

SELECT Employees.EmployeeID, Employees.FirstName,
Employees.LastName, Departments.DepartmentName

FROM Employees

RIGHT JOIN Departments
```

```
ON Employees.DepartmentID = Departments.DepartmentID;
```

5. CROSS JOIN

A CROSS JOIN returns the Cartesian product of the two tables. This means it will return every possible combination of rows from both tables.

Syntax:

```
SELECT columns

FROM table1

CROSS JOIN table2;
```

Example:

```
SELECT Employees.EmployeeID, Employees.FirstName,
Departments.DepartmentName

FROM Employees

CROSS JOIN Departments;
```

6. SELF JOIN

A Self Join is when a table is joined with itself. This is useful for hierarchical or recursive relationships within the same table.

Tasks Assigned Today

Syntax:

```
SELECT columns

FROM table1 AS alias1

JOIN table1 AS alias2

ON alias1.column_name = alias2.column_name;
```

Example:

```
SELECT e1.EmployeeID, e1.FirstName AS Employee, e2.FirstName
AS Manager

FROM Employees e1

JOIN Employees e2

ON e1.ManagerID = e2.EmployeeID;
```