**SQL JOINS**

A **join** is an operation in SQL that combines rows from two or more tables based on a related column between them. It allows for the retrieval of data from multiple tables simultaneously, based on a specified condition or relationship.

**Types of joins**

1. Inner join

2. Left join

3. Right join

5. Full join

6. Cross join

**Example:**

Suppose we have two tables: "Employees" and "Departments".

Employees Table: Departments Table:

|  |  |  |
| --- | --- | --- |
| EmployeeID | EmployeeName | DepartmentID |
| 1 | John | 101 |
| 2 | Mary | 102 |
| 3 | David | 101 |
| 4 | Sarah | 103 |

|  |  |
| --- | --- |
| DepartmentID | DepartmentName |
| 101 | HR |
| 102 | Marketing |
| 103 | Finance |
| 104 | IT |

Now, let's explore each type of join using these tables:

**Inner Join:** Retrieves only the matching records between the two tables.

sql code:

SELECT Employees.EmployeeID, Employees.EmployeeName, Departments.DepartmentName

FROM Employees

INNER JOIN Departments ON Employees.DepartmentID = Departments.DepartmentID;

This query will return the following result:

|  |  |  |
| --- | --- | --- |
| EmployeeID | EmployeeName | DepartmentName |
| 1 | John | HR |
| 2 | Mary | Marketing |
| 3 | David | HR |
| 4 | Sarah | Finance |

**Left Join:** Retrieves all records from the left table (Employees) and the matching records from the right table (Departments).

sql code:

SELECT Employees.EmployeeID, Employees.EmployeeName, Departments.DepartmentName

FROM Employees

LEFT JOIN Departments ON Employees.DepartmentID = Departments.DepartmentID;

This query will return the following result:

|  |  |  |
| --- | --- | --- |
| EmployeeID | EmployeeName | DepartmentName |
| 1 | John | HR |
| 2 | Mary | Marketing |
| 3 | David | HR |
| 4 | Sarah | Finance |

**Right Join:** Retrieves all records from the right table (Departments) and the matching records from the left table (Employees).

sql code:

SELECT Employees.EmployeeID, Employees.EmployeeName, Departments.DepartmentName

FROM Employees

RIGHT JOIN Departments ON Employees.DepartmentID = Departments.DepartmentID;

This query will return the following result:

|  |  |  |
| --- | --- | --- |
| EmployeeID | EmployeeName | DepartmentName |
| 1 | John | HR |
| 2 | Mary | Marketing |
| 3 | David | HR |
| 4 | Sarah | Finance |
| NULL | NULL | IT |

Note: Since there is no matching DepartmentID for EmployeeID NULL in the Employees table, the EmployeeID and EmployeeName are displayed as NULL.

**Full Join:** Retrieves all records from both tables, including the unmatched records.

sql code:

SELECT Employees.EmployeeID, Employees.EmployeeName, Departments.DepartmentName

FROM Employees

FULL JOIN Departments ON Employees.DepartmentID = Departments.DepartmentID;

This query will return the following result:

|  |  |  |
| --- | --- | --- |
| EmployeeID | EmployeeName | DepartmentName |
| 1 | John | HR |
| 2 | Mary | Marketing |
| 3 | David | HR |
| 4 | Sarah | Finance |
| NULL | NULL | IT |

Note: The unmatched record from the right table (DepartmentID 104) is displayed with NULL values for EmployeeID and EmployeeName.

**Cross Join:** Retrieves the Cartesian product of two tables, meaning it combines each row from the first table with every row from the second table.

sql code:

SELECT Employees.EmployeeID, Employees.EmployeeName, Departments.DepartmentName

FROM Employees

CROSS JOIN Departments;

This query will return the following result:

|  |  |  |
| --- | --- | --- |
| EmployeeID | EmployeeName | DepartmentName |
| 1 | John | HR |
| 1 | John | Marketing |
| 1 | John | Finance |
| 1 | John | IT |
| 2 | Mary | HR |
| 2 | Mary | Marketing |
| 2 | Mary | Finance |
| 2 | Mary | IT |
| 3 | David | HR |
| 3 | David | Marketing |
| 3 | David | Finance |
| 3 | David | IT |
| 4 | Sarah | HR |
| 4 | Sarah | Marketing |
| 4 | Sarah | Finance |
| 4 | Sarah | IT |

Note: The Cross Join combines each row from the Employees table (4 rows) with every row from the Departments table (4 rows), resulting in a total of 16 rows in the output.