A **FULL OUTER JOIN** in SQL combines the results of both a **LEFT JOIN** and a **RIGHT JOIN**. It returns all rows from both tables, with matching rows from both sides where available. If there is no match, the result will contain NULL in the columns for the table without a match.

**Key Features of FULL OUTER JOIN:**

1. **All Rows from Both Tables**: It includes every row from both tables, even if there is no match.
2. **NULLs for Non-Matching Rows**: For rows in one table without a matching row in the other, the result will include NULL values in the columns of the unmatched table.

**Syntax:**

SELECT

column\_list

FROM

table1

FULL OUTER JOIN

table2

ON

table1.common\_column = table2.common\_column;

**Example:**

Consider two tables, Employees and Departments:

**Employees Table:**

| **EmployeeID** | **EmployeeName** | **DepartmentID** |
| --- | --- | --- |
| 1 | Alice | 101 |
| 2 | Bob | 102 |
| 3 | Charlie | NULL |

**Departments Table:**

| **DepartmentID** | **DepartmentName** |
| --- | --- |
| 101 | HR |
| 103 | IT |

**Query:**

SELECT

Employees.EmployeeID,

Employees.EmployeeName,

Employees.DepartmentID,

Departments.DepartmentName

FROM

Employees

FULL OUTER JOIN

Departments

ON

Employees.DepartmentID = Departments.DepartmentID;

**Result:**

| **EmployeeID** | **EmployeeName** | **DepartmentID** | **DepartmentName** |
| --- | --- | --- | --- |
| 1 | Alice | 101 | HR |
| 2 | Bob | 102 | NULL |
| 3 | Charlie | NULL | NULL |
| NULL | NULL | 103 | IT |

**When to Use:**

* Use a **FULL OUTER JOIN** when you want to retrieve all rows from both tables, regardless of whether there is a match.
* It is particularly useful when analyzing data for relationships or when combining datasets to ensure no data is excluded.

**Notes:**

* Some databases may require enabling FULL OUTER JOIN explicitly or use workarounds if it's not natively supported.