TOPS TECHNOLOGY

Module 4 – Introduction to DBMS

Presented By :

Nandni Vala

SQL Joins

1.Explain the concept of JOIN in SQL. What is the difference between INNER JOIN, LEFT JOIN, RIGHT JOIN, and FULL OUTER JOIN?

- SQL, a **JOIN** is used to combine rows from two or more tables based on a related column between them. This operation is typically performed to retrieve related data spread across different tables in a relational database.
- The most common types of joins are **INNER JOIN**, **LEFT JOIN**, **RIGHT JOIN**, and **FULL OUTER JOIN**.
- **INNER JOIN**
- **Definition**: The **INNER JOIN** returns only the rows where there is a match in both tables.
- It retrieves records that have matching values in both tables.
- Use when you want to get only the records that have corresponding matches in both tables.

- **Example**:
- SELECT employees.name, departments.name
- FROM employees
- INNER JOIN departments ON employees.department_id = departments.department_id;
- **LEFT JOIN (or LEFT OUTER JOIN)**
- **Definition**: The **LEFT JOIN** returns all rows from the **left table** (the first table in the query), and the matching rows from the **right table** (the second table). If there is no match, NULL values are returned for columns from the right table.
- Use when you want to get all rows from the left table and the matched rows from the right table, including rows from the left table with no matching rows in the right table.
  - **Example**:
- SELECT employees.name, departments.name
- FROM employees
- LEFT JOIN departments ON employees.department_id = departments.department_id;

- **RIGHT JOIN (or RIGHT OUTER JOIN)**
- **Definition**: The **RIGHT JOIN** returns all rows from the **right table**, and the matching rows from the **left table**. If there is no match, NULL values are returned for columns from the left table.
- Use when you want to get all rows from the right table and the matched rows from the left table, including rows from the right table with no matching rows in the left table.
  - **Example**:
- SELECT employees.name, departments.name
- FROM employees
- RIGHT JOIN departments ON employees.department_id = departments.department_id;
- **FULL OUTER JOIN**
- **Definition**: The **FULL OUTER JOIN** returns all rows when there is a match in **either** the left or the right table. It combines the results of both **LEFT JOIN** and **RIGHT JOIN**, returning all rows from both tables, with NULL for non-matching rows.

- Use when you want to get all rows from both tables, with NULL values for unmatched rows.
  - **Example**:
- SELECT employees.name, departments.name
- FROM employees
- FULL OUTER JOIN departments ON employees.department_id = departments.department_id;

2. How are joins used to combine data from multiple tables?

- **Joins** are used in SQL to combine data from multiple tables based on a related column. They allow you to retrieve data that is stored in different tables and present it together in a single result set.
- Types of Joins:
- **INNER JOIN**: Returns only the rows with matching values in both tables.
- **LEFT JOIN**: Returns all rows from the left table, and matched rows from the right table. If no match, NULL values are returned for the right table's columns.

- **RIGHT JOIN**: Returns all rows from the right table, and matched rows from the left table. If no match, NULL values are returned for the left table’s columns.
- **FULL OUTER JOIN**: Returns all rows when there is a match in either the left or right table. Unmatched rows will contain NULL values for the non-matching table's columns.
  - **Example**: To combine employees and departments tables based on a department_id, you would use a **JOIN** like this:
- SELECT employees.name, departments.name
- FROM employees
- INNER JOIN departments ON employees.department_id = departments.department_id;