Sql-MERGE

Here's a summary of the `MERGE` statement in SQL, along with an example:

What is MERGE?

The `MERGE` statement is a data manipulation language (DML) command that combines two tables based on a common column. It allows you to update or insert new rows into one table from another, while preserving the existing data.

Basic syntax:

```
 MERGE INTO target_table AS T
USING source_table AS S
ON condition_column = T.column
WHEN MATCHED THEN
  UPDATE SET ...
WHEN NOT MATCHED THEN
  INSERT (columns) VALUES ();
```

Let's break it down:

- `target_table` is the table where you want to update or insert data.
- `source_table` is the table that contains the new data you want to merge with `target_table`.
- `condition_column` is the column that links rows between `source_table` and `target_table`.
- The `WHEN MATCHED THEN` clause specifies what action to take when a row in `source_table` matches an existing row in `target_table`. Typically, this involves updating one or more columns in `target_table`.
- The `WHEN NOT MATCHED THEN` clause specifies what action to take when a row in `source_table` does not match any row in `target_table`. Typically, this involves inserting the new row into `target_table`.

Example:

Suppose we have two tables:

Table 1 (`employees`):

id (primary key)	name	department
101	John Smith	Sales
102	Jane Doe	Marketing

Table 2 (`new_employees`):

id (foreign key to employees.id)	name	department
103	Bob Johnson	IT

We want to merge the `new_employees` table with the `employees` table, updating any existing rows in `employees` and inserting new rows from `new_employees`.

Here's the `MERGE` statement:

```
 MERGE INTO employees AS E
USING new_employees AS N
ON E.id = N.id
WHEN MATCHED THEN
  UPDATE SET E.department = N.department
WHEN NOT MATCHED THEN
  INSERT (id, name, department)
VALUES (N.id, N.name, N.department);
```

After running the `MERGE` statement, the resulting `employees` table would be:

id	name	department
101	John Smith	Sales
102	Jane Doe	Marketing
103	Bob Johnson	IT

The new row from `new_employees` was inserted into `employees`, while the existing row for employee 101 in `employees` was updated to reflect the same department as the new row.

Curated by Brajesh Kumar