Table Joins in SQL

Hia Al Saleh

November 18th, 2024

Contents

1 Introduction to Relational Databases

Relational databases create relationships between two or more tables. This is useful when storing related information. For example:

- We want to store information about a product and its supplier.
- A simple product table can include attributes like `prod_name`, `qty`, and `supplier`.
- As data grows, storing all supplier contact details in the same table can become cumbersome and redundant.

2 Creating Tables

Below are examples of how to create tables for products and suppliers:

2.1 Product Table

```
CREATE TABLE product(
    id INTEGER NOT NULL AUTO_INCREMENT PRIMARY KEY,
    prod_name VARCHAR(30) NOT NULL,
    qty SMALLINT NOT NULL,
    supplier VARCHAR(10)
) ENGINE=INNODB;
```

2.2 Extended Product Table with Supplier Information

```
CREATE TABLE product(
    id INTEGER NOT NULL AUTO_INCREMENT PRIMARY KEY,
    prod_name VARCHAR(30) NOT NULL,
    qty SMALLINT NOT NULL,
    supplier VARCHAR(10),
    contactName VARCHAR(50),
    contactPhone CHAR(10),
    contactEmail VARCHAR(30),
    contactPosition VARCHAR(15)
) ENGINE=INNODB;
```

2.3 Supplier Table

```
CREATE TABLE supplier(
    id INTEGER NOT NULL AUTO_INCREMENT PRIMARY KEY,
    supplierName VARCHAR(30),
    contactName VARCHAR(50),
    contactPhone CHAR(10),
    contactEmail VARCHAR(30),
    contactPosition VARCHAR(15)
);
```

3 Inserting Data into Tables

To insert data into these tables, the following SQL statements are used:

3.1 Inserting Data into the Supplier Table

```
INSERT INTO supplier
(supplierName, contactName, contactPhone, contactEmail, contactPosition)
VALUES
('Farm Chicken Supplier', 'John Doe', '555-6656', 'jdoe@email.com', 'Buyer'),
('Cow Farms', 'Mike Smith', '666-9656', 'msmith@email.com', 'Manager');
```

3.2 Inserting Data into the Product Table

```
INSERT INTO product
(prod_name, qty, supplier)
VALUES
('Chicken', 2, 1),
('Turkey', 14, 1),
('Beef', 22, 2);
```

4 Retrieving Information

- Use the `SELECT` statement to retrieve information from tables.
- Information can be pulled from multiple tables using joins.

5 Types of Joins

Joins allow tables to be related row by row, satisfying specific conditions.

5.1 INNER JOIN

An `INNER JOIN` returns only the rows that satisfy the condition specified in the `ON` clause.

```
SELECT column1, column2, column3
FROM table_a INNER JOIN table_b
ON column_x = column_y;
```

Example using the product and supplier tables:

```
SELECT *
FROM product INNER JOIN supplier
ON product.supplier = supplier.id;
```

5.2 OUTER JOIN

An `OUTER JOIN` returns rows from at least one of the tables, even if there is no match in the other table.

5.2.1 LEFT OUTER JOIN

Includes all rows from the left table and the matched rows from the right table.

```
SELECT a.au_fname, a.au_lname, p.pub_name
FROM authors AS a LEFT OUTER JOIN publishers AS p
ON a.city = p.city;
```

5.2.2 RIGHT OUTER JOIN

Includes all rows from the right table and the matched rows from the left table.

```
SELECT a.au_fname, a.au_lname, p.pub_name
FROM authors AS a RIGHT OUTER JOIN publishers AS p
ON a.city = p.city;
```

6 Querying with Conditions

Use conditions to filter and join tables:

- Use `WHERE` to specify filtering conditions.
- Use `ORDER BY` to sort the results.

Example:

```
SELECT products.productName, suppliers.companyName
FROM products INNER JOIN suppliers
ON products.SupplierID = suppliers.SupplierID
WHERE country = 'United States'
ORDER BY suppliers.companyName;
```

7 Group Data with Aggregations

To group data and perform aggregations, the `GROUP BY` clause is useful.

Example:

```
SELECT s.companyName, COUNT(p.productName) AS numberOfProducts
FROM suppliers AS s INNER JOIN products AS p
ON s.SupplierID = p.SupplierID
GROUP BY s.companyName
ORDER BY numberOfProducts;
```

8 Summary

- Joins are a fundamental concept in SQL to combine data from multiple tables.
- `INNER JOIN` retrieves matching rows, while `OUTER JOIN` includes non-matching rows.
- Understanding joins is essential for efficient data querying and manipulation.