How to Copy Data from One Table to Another

To copy data from one table to another in SQL, you can use the `INSERT INTO ... SELECT` statement. This statement allows you to copy data from one table and insert it into another, either by selecting specific columns or all columns. Here are the different ways you can achieve this:

1. Copy All Data from One Table to Another

If both tables have the same structure (i.e., the same columns and data types), you can copy all rows from the source table to the destination table.

```
INSERT INTO destination_table

SELECT * FROM source_table;
```

- Explanation: This will copy all rows and columns from the `source_table` to the `destination_table`.

2. Copy Specific Columns from One Table to Another

If the tables have different structures, you can specify which columns to copy from the source table to the destination table.

```
INSERT INTO destination_table (column1, column2, column3)

SELECT column1, column2, column3

FROM source_table;
```

- Explanation: This will copy the specified columns (`column1`, `column2`, `column3`) from the `source_table` and insert them into the corresponding columns in the `destination_table`.

3. Copy Data with a Condition

You can also copy data based on a specific condition. For example, you might want to copy only rows where a certain condition is true.

```
INSERT INTO destination_table (column1, column2, column3)

SELECT column1, column2, column3

FROM source_table

WHERE condition;
```

- Explanation: This will copy only the rows that satisfy the `WHERE` condition from the `source_table` into the `destination_table`.

4. Copy Data and Add New Values

You can copy data from one table to another and also add new values for columns that don't exist in the source table or need to be different.

```
INSERT INTO destination_table (column1, column2, column3, new_column)

SELECT column1, column2, column3, 'new_value'

FROM source_table;
```

- Explanation: This copies data from `source_table` and adds a constant value (e.g., `'new_value'`) for `new_column` in the `destination_table`.

5. Copy Data into a New Table

If the destination table doesn’t exist, you can create a new table and insert the data simultaneously.

```
CREATE TABLE new_table AS
SELECT * FROM source_table;
```

- Explanation: This will create a new table `new_table` with the same structure as `source_table` and copy all its data.

Practical Example: Copying Customer Data in E-commerce

Scenario:

You want to copy customer data from an old `customers_old` table to a new `customers_new` table but only for customers who have placed more than 5 orders.

```
INSERT INTO customers_new (CustomerID, CustomerName, Contact, OrdersCount)
SELECT CustomerID, CustomerName, Contact, OrdersCount
FROM customers_old
WHERE OrdersCount > 5;
```

- Explanation: This will copy customer data from `customers_old` to `customers_new` only for those customers who have placed more than 5 orders.

Practical Example: Copying Product Data in Retail

Scenario:

You need to copy only active products from an `old_products` table to a `new_products` table.

```
INSERT INTO new_products (ProductID, ProductName, Price)
SELECT ProductID, ProductName, Price
FROM old_products
WHERE Status = 'active';
```

- Explanation: This query copies only the active products from the `old_products` table to the `new_products` table.

Summary

- Simple Copy: `INSERT INTO ... SELECT *` copies all data.

- Selective Copy: `INSERT INTO ... SELECT column1, column2 ...` copies specific columns.

- Conditional Copy: Add a `WHERE` clause to filter the rows.

- Copy with New Values: Add additional columns or constant values while copying.

This process is useful across industries, whether it's copying customer data, financial transactions, or product details.