

LAB - TRUNCATE

In this lab, you will learn how to use the SQL Server `TRUNCATE TABLE` statement to remove all rows from a table faster and more efficiently.

Sometimes, you want to delete all rows from a table. In this case, you typically use the `DELETE` statement without a `WHERE` clause.

The following example creates a new table named `customer_groups` and inserts some rows into the table:

```
CREATE TABLE sales.customer_groups (  
    group_id INT PRIMARY KEY IDENTITY,  
    group_name VARCHAR (50) NOT NULL  
);  
  
INSERT INTO sales.customer_groups (group_name)  
VALUES  
    ('Intercompany'),  
    ('Third Party'),  
    ('One time');
```

To delete all rows from the `customer_groups` table, you use the `DELETE` statement as follows:

```
DELETE FROM sales.customer_groups;
```

Besides the `DELETE FROM` statement, you can use the `TRUNCATE TABLE` statement to delete all rows from a table.

The following illustrates the syntax of the `TRUNCATE TABLE` statement:

```
TRUNCATE TABLE [database_name.][schema_name.]table_name;
```

In this syntax, first, you specify the name of the table from which you want to delete all rows. Second, the database name is the name of the database in which the table was created. The database name is optional. If you skip it, the statement will delete the table in the currently connected database.

The following statements first insert some rows into the `customer_groups` table and then delete all rows from it using the `TRUNCATE TABLE` statement:

```
INSERT INTO sales.customer_groups (group_name)
VALUES
    ('Intercompany'),
    ('Third Party'),
    ('One time');

TRUNCATE TABLE sales.customer_groups;
```

The `TRUNCATE TABLE` is similar to the `DELETE` statement without a `WHERE` clause. However, the `TRUNCATE` statement executes faster and uses a fewer system and transaction log resources.

TRUNCATE vs. DELETE

The `TRUNCATE TABLE` has the following advantages over the `DELETE` statement:

1) Uses less transaction log

The `DELETE` statement removes rows one at a time and inserts an entry in the transaction log for each removed row. On the other hand, the `TRUNCATE TABLE` statement deletes the data by deallocating the data pages used to store the table data and inserts only the page deallocations in the transaction logs.

2) Uses fewer locks

When the `DELETE` statement is executed using a row lock, each row in the table is locked for removal. The `TRUNCATE TABLE` locks the table and pages, not each row.

3) Identity reset

If the table to be truncated has an identity column, the counter for that column is reset to the seed value when data is deleted by the `TRUNCATE TABLE` statement but not the `DELETE` statement.

In this lab, you have learned how to use the `TRUNCATE TABLE` statement to delete all rows from a table faster and more efficiently.