DELETE

`DELETE` is a DML (Data Manipulation Language) command. This command removes records from a table. It is used only for deleting data from a table, not to remove the table from the database.

You can delete **all records** with the syntax:

```
DELETE FROM name_table;
```

Or you can delete a **group of records** using the WHERE clause:

```
DELETE FROM name_table WHERE col=value;
```

If you'd like to remove all records from a given table, use `DELETE` FROM followed by the table name. Notice that there aren't any column names in this syntax; you're removing all records, not the data in selected columns.

If you want to remove specific records, use `WHERE` with filtering conditions, as in the second example.

Let's use these commands in an example. Here's the table **`product`** :

id	name	price
1	milk	2.40
2	bread	3.68
3	butter	5.55
4	sugar	2.88

This query ...

```
DELETE FROM product;
```

… removes all the data in the table `product` . After this query, the table `product` will be empty.

But the `WHERE` query ...

```
DELETE FROM product WHERE price<2.90;
```

… deletes only the records for milk and sugar, because their prices are lower than $2.90. (Milk is $2.40 and sugar is $2.88.)

Now the table `product` only has records with prices higher than $2.90:

id	name	price
2	bread	3.68
3	butter	5.55

How does DELETE work?

If you don't want to remove table structure or you're only deleting specific rows, use the `DELETE` command. It can remove one, some, or all rows in a table. `DELETE` returns the number of rows removed from the table.

However, `DELETE` uses a row lock during execution and can be rolled back. Every deleted row is locked, so it will require a lot of locks if you're working in a large table.

`DELETE` also keeps the auto-increment ID in the table. If you remove the last record in the table with ID=20 and then add a new record, this record will have ID=21 – even though the record immediately before it will be ID=19.

`DELETE` can be executed by triggers. A trigger can be called before, after, or instead of the `DELETE` operation. It can be executed for any row change or when all rows are removed. Removing rows in another table can also trigger `DELETE` .

Of course, to use the `DELETE` command you need `DELETE` permission for that table.

TRUNCATE TABLE

`TRUNCATE TABLE` is similar to `DELETE` , but this operation is a DDL (Data Definition Language) command. It also deletes records from a table without removing table structure, but it doesn't use the `WHERE` clause. Here's the syntax:

```
TRUNCATE TABLE table_name;
```

If you use this command, all rows in this table will be removed. The following query ...

```
TRUNCATE TABLE product;
```

… deletes all records stored in the table **`product`** .

How does TRUNCATE TABLE work?

Be careful using this command. `TRUNCATE` transactions can be rolled back in database engines like SQL Server and PostgreSQL, but not in MySQL and Oracle.

`TRUNCATE` is faster than `DELETE` , as it doesn't scan every record before removing it. `TRUNCATE TABLE` locks the whole table to remove data from a table; thus, this command also uses less transaction space than `DELETE` .

Unlike `DELETE` , `TRUNCATE` does not return the number of rows deleted from the table. It also resets the table auto-increment value to the starting value (usually 1). If you add a record after truncating the table, it will have ID=1. **Note:** In **PostgreSQL**, you can choose to restart or continue the auto-increment value. **Oracle** uses a sequence to increment values, which is not reset by `TRUNCATE` .

Of course, you need permission to use `TRUNCATE TABLE` . In PostgreSQL, you need the privilege `TRUNCATE` ; in SQL Server, the minimum permission is `ALTER` table; in MySQL, you need the `DROP` privilege. Finally, Oracle requires the `DROP ANY TABLE` system privilege to use this command.

You can learn more in the course "The Basics of Creating Tables in SQL", which is part of our Data Engineering Path.

DROP TABLE

The `DROP TABLE` is another DDL (Data Definition Language) operation. But it is not used for simply removing data from a table; it deletes the table structure from the database, along with any data stored in the table.

Here is the syntax of this command:

```
DROP TABLE table_name;
```

All you need after `DROP TABLE` is the name of the table you want to delete. For example, if you'd like to remove the entire **`product`** table from the database, you'd write:

```
DROP TABLE product;
```

This removes all data in the table **`product`** and the structure of the table.

How does DROP TABLE work?

The `DROP TABLE` operation removes the table definition and data as well as the indexes, constraints, and triggers related to the table.

This command frees the memory space.

No triggers are fired when executing `DROP TABLE` .

This operation cannot be rolled back in MySQL, but it can in Oracle, SQL Server, and PostgreSQL.

In SQL Server, `DROP TABLE` requires `ALTER` permission in the schema to which the table belongs; MySQL requires the DROP privilege; Oracle the requires the `DROP ANY TABLE` privilege. In PostgreSQL, users can drop their own tables.

DROP TABLE vs. DELETE TABLE vs. TRUNCATE TABLE in SQL

Which cases call for `DROP TABLE` ? When should you use `TRUNCATE` or opt for a simple `DELETE` ? We've prepared the table below to summarize the properties of each command:

	DELETE	TRUNCATE	DROP
Type	DML	DDL	DDL
Uses a lock	Row lock	Table lock	Table lock
Works in WHERE	Yes	No	No
Removes ...	One, some, or all rows in a table.	All rows in a table.	Entire table structure: data, privileges, indexes, constraints, triggers.
Resets ID auto-increment	No	MySQL: Yes Oracle: No PostgreSQL: User decides SQL Server: Yes	Doesn’t apply
Rollback	Yes	MySQL: No Oracle: No PostgreSQL: Yes SQL Server: Yes	MySQL: No Oracle: Yes PostgreSQL: Yes SQL Server : Yes
Transaction logging	Each row	Whole table (minimal)	Whole table (minimal)
Works with indexed views	Yes	No	No
Space requirements	More space	Less space	More space
Fires triggers	Yes	No	No
Speed	Slow	Fastest	Faster

Which operation is best for which use case?

- To remove specific rows, use `DELETE` .

- To remove all rows from a large table and leave the table structure, use `TRUNCATE TABLE` . It’s faster than `DELETE` .
- To remove an entire table, including its structure and data, use `DROP TABLE` .