SQL DROP TABLE

A SQL DROP TABLE statement is used to delete a table definition and all data from a table.

This is very important to know that once a table is deleted all the information available in the table is lost forever, so we have to be very careful when using this command.

- This action is **irreversible** which means that once a table is **dropped**, it cannot be recovered unless there is a **backup**.

```
DROP TABLE table_name;
```

Important Points About SQL DROP TABLE

- The SQL DROP TABLE statement is used to delete tables in a database, along with all associated
  data, indexes, triggers, constraints and permission specifications.
- The table will be permanently disable, so use this query with caution.
- Use **DROP TABLE IF EXISTS** query to prevent errors when dropping a table that does not exist
- When dropping a partitioned table, the DROP TABLE statement removes the table definition, all partitions, all data stored in those partitions, and all partition definitions.
- The DROP TABLE statement can be used to drop temporary tables by including the **TEMPORARY keyword.**
- To verify if a table is dropped, you can use the DESC command.

Step 1: Create a Table

```
CREATE TABLE employees (
    employee_id INT PRIMARY KEY,
    first_name VARCHAR(50),
    last_name VARCHAR(50),
    email VARCHAR(100) UNIQUE,
    department_id INT
);
```

Example 1: Drop the Table

```
DROP TABLE employees;
```

Effect: This command deletes the employees table along with all its data. After executing this command, the employees table no longer exists in the database

Example 2: Drop a Table If It Exists

```
DROP TABLE IF EXISTS employees;
```

Effect: This command drops the employees table if it exists. If the table does not exist, no error is thrown.

Example 3: Drop Multiple Tables

```
DROP TABLE IF EXISTS employees, departments;
```

Effect: This command attempts to drop both the employees and departments tables. If one or both tables do not exist, the command will still execute without errors for the non-existent tables.

Example 4: Drop a Table with CASCADE Constraints (Advanced)

If the table you're dropping is referenced by foreign keys in other tables, the DROP TABLE command might fail unless you specify the CASCADE option.

```
DROP TABLE employees CASCADE;
```

Effect: This command will drop the employees table and automatically drop any foreign key constraints in other tables that reference this table. Use this option with caution, as it can have widespread effects.

Example 5: Drop a Table and its Dependent Objects (Views, Indexes)

Some SQL dialects allow you to drop a table along with other objects that depend on it, like views and indexes.

```
DROP TABLE employees CASCADE CONSTRAINTS;
```

Effect: This command not only drops the employees table but also removes associated constraints, indexes, and views that depend on it.

Summary

- **Basic**: DROP TABLE employees; - Removes the table and its data.
- **IF EXISTS**: DROP TABLE IF EXISTS employees; - Avoids errors if the table doesn't exist.
- **Multiple Tables**: DROP TABLE IF EXISTS employees, departments; - Drops more than one table.
- **CASCADE**: DROP TABLE employees CASCADE; - Drops the table and any referencing foreign keys.
- **CASCADE CONSTRAINTS**: DROP TABLE employees CASCADE CONSTRAINTS; - Drops the table and all dependent objects like views and indexes.