**Promt :** EXPLAIN DELETE COMMAND IN POSTGRESQL IN DETALIED WITH 2 EXAMPLES AS A BEGGINER

**What is DELETE in PostgreSQL?**

The DELETE command in PostgreSQL is used to remove one or more rows from a table based on a given condition.  
Unlike TRUNCATE (which wipes the whole table instantly), DELETE allows you to target specific rows using a WHERE clause.

If you don’t use WHERE, it will remove all rows — but still, it deletes them one-by-one and logs each deletion, so it’s slower for large tables compared to TRUNCATE.

The DELETE command in PostgreSQL is used to remove existing rows from a table. It's a fundamental part of a database's data manipulation language (DML) and allows for precise control over which records are removed from a table. Unlike the TRUNCATE command, DELETE is transactional and can be used with a WHERE clause to specify which rows to remove.

**Syntax**

The basic syntax for the DELETE command is as follows:

SQL

DELETE FROM table\_name

WHERE condition;

* **DELETE FROM table\_name**: This part of the command specifies which table you want to delete rows from. The FROM keyword is optional in some versions of PostgreSQL, but it's good practice to include it for clarity.
* **WHERE condition**: This is a crucial part of the command. The WHERE clause filters the rows and specifies which ones to delete. If you omit the WHERE clause, **all rows** in the table will be deleted. The condition is a boolean expression that evaluates to true for each row to be deleted.

**How DELETE Works**

The DELETE command operates on a row-by-row basis. For each row that matches the WHERE condition, PostgreSQL removes it from the table. This process is slower than TRUNCATE for large tables because each deletion is logged and processed individually. However, this granular control is what makes DELETE so powerful.

Because DELETE is a transactional operation, you can wrap it in a BEGIN and COMMIT block. This means you can reverse the changes using a ROLLBACK command if you make a mistake before committing the transaction.

**Examples**

**Example 1: Deleting a Specific Record**

Let's say you have a table named employees and you need to remove the record for an employee with the ID 105.

SQL

-- Assume the 'employees' table exists with columns like employee\_id, first\_name, last\_name.

-- This command will delete the single row where employee\_id is 105.

DELETE FROM employees

WHERE employee\_id = 105;

**Explanation:** In this example, the WHERE employee\_id = 105 clause acts as a filter. PostgreSQL scans the employees table, finds the row where the employee\_id column's value is 105, and removes only that row. The rest of the rows in the table remain untouched.

**Example 2: Deleting Multiple Records**

Suppose you have an orders table and you need to delete all orders that were placed before January 1, 2025.

SQL

-- Assume the 'orders' table exists with columns like order\_id, customer\_id, order\_date.

-- This command will delete all rows from the 'orders' table where the order\_date is before '2025-01-01'.

DELETE FROM orders

WHERE order\_date < '2025-01-01';

**Explanation:** Here, the WHERE clause uses a comparison operator (<) to specify a range of records to delete. PostgreSQL will identify and remove every row where the value in the order\_date column is earlier than the specified date. This shows how a single DELETE statement can remove multiple rows that meet a specific condition.

**Key Points About DELETE**

* Deletes rows one-by-one (row-level operation).
* WHERE clause lets you delete specific data.
* Fires DELETE triggers if they exist.
* Can be rolled back if in a transaction.
* Slower than TRUNCATE for huge datasets because it logs each deleted row.

**DELETE vs TRUNCATE (Quick Recap)**

| **Feature** | **DELETE** | **TRUNCATE** |
| --- | --- | --- |
| WHERE clause | ✅ Yes | ❌ No |
| Speed | Slower (logs each row) | Very fast |
| Triggers | ✅ Fires DELETE triggers | ❌ Does not fire DELETE triggers |
| Resets ID | ❌ No (unless manually reset) | ✅ With RESTART IDENTITY |
| Logging | Fully logged | Minimal logging |

**When to Use DELETE**

* When you need fine control over which rows are removed.
* When you need triggers to run.
* When you want to log each deletion.
* When the dataset is small to medium**.**