**TEMPORARY TABLE**

A temporary table in PostgreSQL is a table that exists temporarily and is automatically dropped at the end of a database session or transaction, depending on how it is created.

Temporary tables are often used for storing intermediate data during complex queries or for session-specific operations.

Key Features of Temporary Tables:

* Session-Specific: Each session has its own instance of the temporary table. Other sessions cannot access it.
* Automatic Cleanup: The table is automatically dropped when the session ends.
* Isolation: Temporary tables with the same name can exist in multiple sessions simultaneously without conflict.

Syntax For Creation Of Temporary Table:

create temp table table\_name (

column\_name data\_type constraints,

column\_name data\_type constraints,

);

**VIEW**

In PostgreSQL, a view is a virtual table that is defined by a SQL query. Unlike a physical table, a view does not store data itself; instead, it retrieves data dynamically from the underlying tables whenever it is queried.

Key Features of Views:

* Dynamic Query Results: A view always reflects the latest data from the underlying tables.
* Simplify Complex Queries: Views can encapsulate complex SQL logic, making it easier to reuse and manage.
* Data Security: Views can restrict access to specific columns or rows of a table.

Syntax For Creation Of View:

create view view\_name as

select columns from table\_name

where conditions;

* Materialized Views

In PostgreSQL, you can also use materialized views, which store the query results physically, improving performance for complex queries at the cost of real-time data updates.

Syntax For Creation Of Materialized View:

create materialized view materialized\_view\_name as

select columns from table\_name

where conditions;

Key Features of Materialized Views:

* Persistent Data: Materialized views store the query results on disk.
* Improved Performance: Since the results are precomputed, querying a materialized view is much faster than running the original query.
* Refreshable: The data in a materialized view can be updated using the REFRESH MATERIALIZED VIEW statement.

**CTE (Common Table Expression) In Sql**

In PostgreSQL, CTE stands for Common Table Expression. It is a temporary result set that can be referred to within a SELECT, INSERT, UPDATE, or DELETE statement. CTEs make complex queries easier to write and understand by breaking them into modular parts.

Syntax For CTE (Common Table Expression):

WITH cte\_name AS (

SELECT col1, AVG(col\_name) AS average\_col\_name

FROM table\_name

GROUP BY col1

)

SELECT \* FROM cte\_name

WHERE cte\_name > value;