**Temporary Table :**

* A temporary table in SQL is an important tool for maintaining intermediate results during query execution. They help **store** **temporary data** without affecting the underlying permanent tables.
* A temporary table in SQL is a special type of table that is created and stored in the system’s temporary database (such as **TempDB** in SQL Server). This table is primarily used to store and generate important mediation results when executing a query, stored procedure, or session.
* Temporary tables are **automatically deleted**when the session or transaction that created them ends, making them perfect for temporary or intermediate data storage.

**View :**

* In SQL, a view is a virtual table based on the result-set of an SQL statement.
* A view contains rows and columns, just like a real table. The fields in a view are fields from one or more real tables in the database.
* You can add SQL statements and functions to a view and present the data as if the data were coming from one single table.
* A view is created with the**CREATE VIEW statement.**

**TYPES OF VIEW :**

* **Simple View:** A view based on only a single table, which doesn't contain GROUP BY clause and any functions.
* **Complex View:** A view based on multiple tables, which contain GROUP BY clause and functions.
* **Inline View:** A view based on a subquery in FROM Clause, that subquery creates a temporary table and simplifies the complex query.
* **Materialized View:** A view that stores the definition as well as data. It creates replicas of data by storing it physically.

**Difference between Simple View & Materialized View :**

| **View** | **Materialised View** |
| --- | --- |
| **This view of the table is a virtual table created with the help of “select query.” To get the view, we have to fire the same query again, as this view does not get stored in the physical memory.** | **This view is also a virtual table, and it is also created with the help of “select query,” but these views are stored in physical memory, so we don’t have to run the query again and again.** |
| **When the data is updated in the actual table, it is automatically updated in the view.** | **In materialized view, the user must manually update the data after updating it in the actual table.** |
| **This view does not cost extra memory as the table is not stored in the database.** | **The materialized view is stored in the physical memory, costing some memory in the database.** |
| **When the particular data is used infrequently, we must use this view.** | **When we have to use data frequently, we should use the Materialized view.** |
| **This view responds very slowly, which deceives the performance of the query.** | **This view is faster as compared to the normal view.** |

**CTE in SQL ( Common Table Expression ) :**

* A Common Table Expression (CTE) in SQL is a temporary result set that can be referenced within **a SELECT, INSERT, UPDATE, or DELETE statement**.
* CTEs are defined using the **WITH** keyword and allow you to create a named, reusable subquery within your SQL statement.
* They provide a way to simplify complex queries and make them more readable.

**Use Cases :**

* Simplifying complex queries by breaking them down into smaller, reusable parts.
* Reusing subqueries within a single SQL statement to avoid duplication.
* Performing recursive operations, such as navigating hierarchical data structures.
* Organizing SQL statements for better readability and maintainability.