Answer the following questions.

1. What is the difference between View and Table .

Feature	View	Table
Definition	A virtual table based on the result of a query.	A physical storage of data in rows and columns.
Storage	Does not store data physically.	Stores data physically in the database.
Data Update	Indirectly reflects changes in the underlying tables.	Directly stores and updates data.
Usage	Used to simplify complex queries and to provide security by restricting access to certain data.	Used to store raw data.
Performance	May be slower due to query execution on the fly.	Generally faster for direct data access.
Modification	Generally read-only, though some can be updatable.	Fully updatable with Insert, Update, and Delete operations.
Dependence	Dependent on the underlying tables.	Independent, stands alone as a data structure.
Schema Binding	Can be schema-bound to the underlying tables (optional).	Schema-bound, representing a fundamental data structure.
Indexes	Cannot have indexes directly, but underlying tables can be indexed.	Can have indexes to speed up query performance.

2. What is view? State the types of View.

A **View** in a database is a virtual table that is created by a query and presents data from one or more underlying tables. It doesn't store data physically; instead, it stores the query definition. When a view is queried, the underlying query is executed to retrieve the data.

Types of Views

1. **Simple View**
   - **Definition**: A view that is based on a single table without any functions, aggregations, or groupings.
   - **Characteristics**:
     - Directly corresponds to a single table.
     - Can be updatable if it follows certain conditions.
     - Often used to simplify data retrieval from a single table.
2. **Complex View**
   - **Definition**: A view that is based on multiple tables, often involving joins, aggregations, and functions.
   - **Characteristics**:
     - May not always be updatable.
     - Can include various SQL clauses like JOIN, GROUP BY, HAVING, etc.

        - Useful for summarizing data from multiple sources.
3. **Materialized View**
    - **Definition**: A view that stores the result set of the query physically. It is periodically refreshed to update the data.
    - **Characteristics**:
        - Improves performance by avoiding re-execution of complex queries.
        - Requires storage space since it physically holds data.
        - Used for performance optimization in scenarios where data doesn't change frequently.
4. **Indexed View**
    - **Definition**: A type of view that has an index created on it, which stores the data physically to improve performance.
    - **Characteristics**:
        - Often used in SQL Server.
        - The underlying tables must meet certain criteria for the view to be indexed.
        - Provides performance benefits similar to materialized views.
5. **Partitioned View**
    - **Definition**: A view that allows data to be divided across multiple tables in different databases, but accessed as a single view.
    - **Characteristics**:
        - Can be local (within a single database) or distributed (across multiple databases).
        - Helps manage large datasets by partitioning them across different tables.
        - Used in scenarios where horizontal partitioning of data is beneficial.

3. What is the difference between Simple View and Complex View.

Feature	Simple View	Complex View
Definition	A view based on a single table without any complex SQL operations.	A view based on multiple tables, often involving joins, aggregations, or other complex SQL operations.
Underlying Tables	Based on a single table.	Can involve multiple tables.
SQL Operations	No complex operations like joins, aggregates, or groupings.	Includes complex operations such as joins, aggregates, subqueries, and groupings.
Updatability	Generally updatable if certain conditions are met (e.g., no GROUP BY, DISTINCT).	May not be updatable due to the complexity of the query.
Performance	Typically faster since it involves simpler queries.	May be slower due to the complexity of the underlying query.
Use Case	Used for simplifying access to a single table or for renaming columns.	Used for combining data from multiple tables or summarizing data.

Example	CREATE VIEW SimpleView AS SELECT column1, column2 FROM TableName;	CREATE VIEW ComplexView AS SELECT t1.column1, t2.column2 FROM Table1 t1 JOIN Table2 t2 ON t1.id = t2.id;
Data Aggregation	Typically does not involve data aggregation.	Often involves data aggregation using functions like SUM, COUNT, AVG, etc.
Data Redundancy	Minimal to none, as it deals with a single table.	Possible data redundancy due to joins and multiple tables.