SQL Views – Research & Implementation Task

✅ Part 1: Research & Documentation

🔹 1. Types of Views in SQL Server

Type	What It Is	Key Differences	Use Cases	Limitations / Performance
Standard View	A virtual table defined by a `SELECT` query.	Most common; not stored physically.	Hiding columns, simplifying joins	Cannot have indexes; no performance boost
Indexed View	A materialized view stored physically with an index.	Increases performance; requires schema-bound view.	Financial dashboards, reporting systems	Limited to deterministic functions; strict rules
Partitioned View	Combines data from multiple tables using `UNION ALL`.	Spans tables with the same structure; horizontal partitioning	National bank systems combining branches	All tables must have same schema, limited DML

🔹 2. Can We Use DML on Views?

Question	Answer
Can you use `INSERT/UPDATE/DELETE` on views?	✅ Yes, but only on **updatable views**
Which types allow DML?	**Standard Views** (with simple logic, 1 base table)
What are the restrictions?	❌ No joins, ❌ no GROUP BY, ❌ no aggregates or subqueries
Example:	Updating an `HR.EmployeeView` where `Salary` can be changed safely

```sql
CREATE VIEW HRView AS
SELECT EmployeeID, Name, Salary FROM Employee;

UPDATE HRView
SET Salary = 1200
WHERE EmployeeID = 101;
```

3. How Views Simplify Complex Queries

- Views **hide JOINs** and filters from repeated use
- Easier for **non-technical users** to access data securely
- Example: Combine Customer and Account info

```sql
CREATE VIEW CustomerAccountView AS
SELECT
  C.FullName,
  C.Email,
  A.AccountID,
  A.Balance
FROM Customer C
JOIN Account A ON C.CustomerID = A.CustomerID;
```