# Views in SQL Server:

## An Overview:

**Views in SQL** are virtual tables compiled at runtime. The data associated with views are not physically stored in the view but in the base tables of the view.

## What is a View in SQL?

- A View is just an SQL statement, and the associated data is not physically stored in the view but in the base tables of it. It can be made over one or more database tables. We make views for security purposes since it restricts the user to view some columns/fields of the table(s).
- A T-SQL view is a virtual table and is created by a select composed of one or more tables.
- A T-SQL view is composed of an SQL query.
- Syntax to Create a View in SQL

```
CREATE VIEW view_name AS
SELECT column1, column2, ...
FROM table_name
WHERE condition;
```

Create a View (1,1)

# Types of Views:

- ✓ Basic user-defined view
- ➢ **A standard view** is a **stored SELECT query** that does **not store data itself.**
  Each time the view is queried, SQL Server runs the underlying SELECT statement.
- ➢ a query from one or more tables or from other views.

## ★ Key Differences

| Feature | Standard View |
|---|---|
| Data stored | ❌ No |
| Performance | Same as base query |
| Index allowed | ❌ No |
| DML support | ✅ Yes (with restrictions) |

Key Differences (1,2)

## ★ Real-Life Use Case

☺ University System

- ❖ Show students and their enrolled courses
- ❖ Hide sensitive columns (e.g., GPA, salary)

☺ Banking

- ❖ Display customer accounts without exposing balance details

## ★ Limitations & Performance

- ▪ No performance improvement by itself
- ▪ Complex joins → slower queries
- ▪ Cannot be indexed
- ▪ Depends completely on base tables

✓ Indexed Views

- An **indexed view** physically stores the result of the query on disk using a **clustered index**.
- are views that have been materialized.

★ Key Differences

| Feature | Indexed View |
|---|---|
| Data stored | ✅ Yes |
| Performance | 🚀 Very fast for reads |
| Index allowed | ✅ Yes |
| DML overhead | ⚠️ Higher |

Key Differences (1,3)

★ Real-Life Use Case

☺ **E-Commerce**

- ❖ Daily sales summary
- ❖ Fast dashboards & reports

☺ Banking

- ❖ Total transactions per day
- ❖ Account balance summaries

★ Limitations & Performance

- ▪ Aggregate functions are limited
- ▪ Slower INSERT/UPDATE on base tables (extra maintenance)

✓ Partitioned Views

- A **partitioned view** combines multiple tables using UNION ALL, usually split by **time or region**.
- are views that have partitioned data from a set of member tables across one or more servers.

## ★ Key Differences

| Feature | Partitioned View |
|---|---|
| Data stored | ❌ No |
| Used for | Horizontal partitioning |
| UNION required | ✅ UNION ALL |
| DML support | ⚠️ Limited |

Key Differences (1,4)

## ★ Real-Life Use Case

☺ **E-Commerce**

❖ Orders split by year

❖ Faster access to recent data

☺ **Banking**

❖ Transactions per branch or region

## ★ Limitations & Performance

- Requires **CHECK constraints**
- Complex setup
- DML must target correct partition
- Poor design → slow queries

✓ System Views are stored in Object Catalog Views and used to return information about the SQL Server database instance, or the objects defined.

# Can We Use DML (INSERT, UPDATE, DELETE) on Views?

| View Type | INSERT | UPDATE | DELETE |
|---|---|---|---|
| Standard View | ✅ | ✅ | ✅ |
| Indexed View | ❌ (directly) | ❌ | ❌ |
| Partitioned View | ⚠️ Yes (restricted) | | |

## Restrictions for DML on Views

- View references **one base table**
- No GROUP BY, DISTINCT, HAVING
- No aggregate functions (SUM, AVG)
- No UNION
- No calculated columns

☺ Aaish AlSaidi