**# Task 7 - SQL Views – Interview Questions with Answers**

**## 1. What is a SQL View?**

A view is a virtual table that is defined by a SQL `SELECT` statement. It does not store data itself but retrieves it from the underlying tables whenever the view is queried. Views are useful for abstracting complex queries, simplifying user access to specific data, and enhancing security by hiding unnecessary details.

**## 2. Why are views used in SQL?**

Views are used to simplify data retrieval and enhance security. They allow developers to encapsulate complex joins, calculations, or filter logic into a single object. Views can also be used to present different data perspectives to different users without altering the base tables. For example, a view might expose only specific columns or rows from a table to certain users.

**## 3. Can we update data through a view?**

Yes, but only if the view is **\*\*simple\*\***. A simple view is one that is based on a single table and does not include any of the following: GROUP BY, aggregation functions, subqueries, DISTINCT, LIMIT, or UNION. Complex views cannot be updated directly. Also, updates through views might be restricted depending on database engine rules.

**## 4. What is a materialized view?**

A materialized view stores the result of the SQL query **\*\*physically on disk\*\***, unlike a regular view which is virtual. Materialized views can improve performance for complex queries because the data is pre-computed. However, the data in a materialized view may not always reflect the current state of the base tables unless it is refreshed manually or on schedule. (Note: Materialized views are **\*\*not available in MySQL\*\*** but are supported in Oracle and PostgreSQL.)

**## 5. Can a view contain JOINs?**

Yes. Views can include JOINs to combine data from multiple tables. This is one of the primary benefits of using views — you can write a complex JOIN query once and reuse it many times by querying the view. For example, a view could join a members table with a loans table to show current loan activity.

**## 6. What is the difference between a view and a table?**

- **\*\*Table\*\***: A real object that stores data physically in the database.

- **\*\*View\*\***: A virtual table created from a query; does not store data.

Tables are used to insert, update, and delete data, whereas views are primarily used for reading and displaying data. Views always return real-time data (unless materialized).

**## 7. Can views improve data security?**

Yes. Views are often used to **\*\*restrict user access\*\*** to specific columns or rows in a table. For example, if a table contains sensitive information like salaries or passwords, a view can be created that omits those columns. This way, users can query the view without having access to sensitive data.

**## 8. What is `WITH CHECK OPTION` in a view?**

`WITH CHECK OPTION` is used when creating a view to ensure that any `INSERT` or `UPDATE` operation through the view complies with the view’s WHERE clause. It prevents users from updating or inserting rows that do not satisfy the condition defined in the view. This maintains logical consistency.

**## 9. Can we create indexed views in MySQL?**

No. MySQL does not support indexed or materialized views. All views in MySQL are virtual and execute their underlying query each time they are accessed. However, in databases like SQL Server, indexed views (also called materialized views) are supported under strict rules.

**## 10. How do you drop a view?**

To delete a view from the database, use the `DROP VIEW` statement:

SQL

DROP VIEW view\_name;

This will remove the view definition, but it won’t affect the underlying base tables or data.