Questions

1. What is a view?

A view is a virtual table created from a SQL query. It does not store data physically but displays data from one or more underlying tables. It is used for simplifying complex queries, abstraction, reporting, and security.

---

2. Can we update data through a view?

Yes, we can update data through a view only if:

- It is based on a single table
- It does not contain aggregate functions, GROUP BY, DISTINCT, UNION, or JOIN
- It does not include a calculated column

If the view contains complex operations, then updates may not be allowed.

---

3. What is a materialized view?

A materialized view stores the result of a query physically unlike a normal view, which is virtual. It needs manual or automatic refresh and is used for performance optimization in large data systems (e.g., data warehouses).

---

4. Difference between view and table?

View	Table
Virtual table	Physical table that stores data
Does not store data, only query definition	Stores actual data
Fetches data from base tables each time	Data stored permanently
Can be used for security & abstraction	Used for storage and manipulation

---

5. How to drop a view?

DROP VIEW view_name;

Example:

DROP VIEW EmpView;

---

6. Why use views?

- To simplify complex SQL queries
- To provide data security by restricting sensitive columns
- To improve data abstraction
- To present customized logical data to different users
- To enable reusability of queries

---

7. Can we create indexed views?

Yes, Indexed views (also called materialized views in some DBMS) can be created in databases like SQL Server and Oracle to improve performance. MySQL and SQLite do not support indexed views directly.

---

8. How to secure data using views?

By creating a view that hides sensitive columns (e.g., salary, password) and giving access only to that view instead of the underlying table.

Example:

CREATE VIEW PublicEmployees AS

SELECT name, department FROM Employees;

Users accessing the view cannot see confidential data.

---

9. What are limitations of views?

- Cannot always be updated (e.g., joins, aggregates)
- Performance may be slow for complex queries
- No storage, so data is computed every time (normal views)
- Cannot apply indexes in some DBMS
- Dependent on base table changes

---

10. How does WITH CHECK OPTION work?

WITH CHECK OPTION ensures that rows inserted or updated through a view must satisfy the view's condition. It prevents invalid data from being inserted.

Example:

```
CREATE VIEW HighSalary AS

SELECT * FROM Employees

WHERE salary > 50000

WITH CHECK OPTION;
```

Now if a user tries:

```
UPDATE HighSalary

SET salary = 40000

WHERE emp_id = 101;
```

It will be rejected, because it violates the filter condition.