

No: _____ Date: ____/____/____

Now user 1 gives privileges to user 2 WITH GRANT OPTION so that user 2 can delegate privileges further.

GRANT INSERT ON sample.customer TO 'usr2' @'localhost'
WITH GRANT OPTION;

- If there are multilevel privileges granted, $A \rightarrow B \rightarrow C$. Then revoking privileges from user B also may cause revoking privileges from user C, as well. Then here we use 'RESTRICT' keyword with the REVOKING query at the end.

Database Views

In a database a view is a **virtual table** that is based on the result of a SELECT query. Views are stored in the database but do not contain data themselves. Instead, they provide a way to access and organize data from one or more tables in a database.

- 1) Views are defined using SQL statements, specially the SELECT statement.
- 2) Views can include joins, aggregations and other operations that you can perform on tables.
- 3) Views can be used to simplify complex queries and make them more readable by abstracting away some of the details.
- 4) Views can be updated just like tables, but there are some restrictions.

- views cannot be indexed.

- You cannot insert data into a view, if

• It used any aggregated functions,

• It contains subqueries or derived tables.

- If view based on multiple tables
 - Select statement should not use 'Distinct' keyword
- syntax to create a view:

```
CREATE VIEW <view name> AS
<select query>;
```

example:

```
CREATE VIEW customer_order AS
SELECT c.customer_id, c.name, o.order_id, o.order_date,
o.total_amount FROM customers c INNER JOIN
orders o ON c.customer_id = o.order_id;
```

Selecting data from a view;

```
SELECT * FROM customer_order
WHERE name = 'John Smith';
```

Views can be used in a variety of ways to simplify queries, control data access and organize data.

1. **Simplify Complex queries:** Views can be used to simplify complex queries by encapsulating the details of the underlying tables. Instead of writing a long, complicated query that involves multiple joins and subqueries, you can create a view that performs the necessary operations and then query the view instead.
2. **Restrict Access to data:** Views can be used to restrict access to sensitive data by limiting the columns and rows that are visible to certain users & roles.
Ex: You can create a view that shows a subset of the columns in a table or that only includes rows that meets certain criteria.

3. Organize data :- Views can be used to organize data in a way that makes sense for your application. Example? You could create a view that combines data from multiple tables into a single view that represents data in organized manner.

4. Provide a consistent interface :- Views can be used to provide a consistent interface to your database, even if the underlying schema changes over time. For example? You can create a view that represents a table entity even if underlying tables & columns change.

Drop a view.

`DROP VIEW <view name>;`