33231

CLASSMATE
Date :
Page :

ASSIGNMENT 11

Aim: Write DDL statements to create views on single and multiple tables ~~abov~~ from above DB.

Problem Statement: Write DDL statements to create VIEWS on single and multiple tables from above DB. Do the following operation to demonstrate the use of view :-

- Update the base table
- Insert new record in base table
- Delete record in the base table
- DML on view

What are the restrictions applicable while creating or modifying views? Demonstrate using suitable queries.

Objective:
- To understand the use of views
- To understand implementation of different types of views
- To implement and analyze various operations on views

Theory:

View :- Database view is known as a "virtual table" that allows you to query the data in it. Understanding the database views and using them correctly are very important. A database view is defined as a SQL SELECT query with joins. Because a database view is similar to a database table, which consists of rows and columns, so you can query data against it. Most database management systems, including MYSQL allow you to update data in the underlying tables through the database view with some prerequisites.

A database view is dynamic because it is not related to the physical schema. The database system stores database views ~~reflects~~ as a SQL SELECT statement with

joins. When the data of the tables change the view reflects the changes as well.

Advantages of views :-

1) allows you to simplify complex queries
2) It helps limit data access to specific users
3) It provides an extra security layer
4) It enables computed columns.
5) It enables backward compatibility

Disadvantages of views :-

1) Performances querying data from a database view can be slow especially if the view is created based on other views.
2) Tables Dependency : you create a view based on underlying tables of the database. Whenever you change the structure of those tables that view associated with, you have to change the view as well

CREATE VIEW statement :

To create a new view in MYSQL, you use the CREATE NEW statement. The syntax of creating a view is as follows:

→ create
[algorithm = { merge | temptable | undefined }]
view [database_name][view_name]
as
[select statement]

View processing algorithms : The algorithm attribute allows you to control which mechanism MYSQL uses when creating

the view. MYSQL provides 3 algorithms :

- MERGE - MYSQL first combines the input query with the SELECT statement which defines the view into a single query and then executes the combined query to return the result set
- TEMPTABLE - MYSQL first creates a temporary table based on the select statement that defines the view and then executes the input query against this temporary table.
- UNDEFINED - It is the default algorithm when you create a view without specifying an explicit algorithm.

View Name

Within a database, views and tables share the same namespace therefore, a view and a table cannot have the same name. In addition, the name of a view must follow the table's naming rules.

SELECT statement

In select statement, you can query data from any table or view that exists in database. There are several rules that the SELECT statement must follow :

- It can contain a subquery in WHERE clause but not in the FROM clause.
- It cannot refer to any variables including local variables, user variables and session variables
- It cannot refer to the parameters of prepared statements.

Creating simple views

```
create view SalePreOrder as select
    orderNumber, SUM(quantity * price) total from
```

33231

CLASSMATE
Date :
Page :

orderdetails group by orderNumber
Order by total desc ;

Creating a view based on another view
create view BigSalesOrder as
Select orderNumber , Round(total , 2) as total
from saleOrder
where total > 60000 ;

Show view definition
show create view view_name

Modifying views
MYSQL provides two statements that allow you to modify an existing view : ALTER VIEW
CREATE OR REPLACE VIEW

Removing views
Once a view is created , you can remove it using the DROP VIEW statement . The following illustrates the syntax :
drop view [if exists] [db_name] . [view_name]

Conclusion :
Understood the concept of PL/SQL block by implementing all types of views on DB .