Pre-Joining Topics

Week 2: MYSQL ADVANCED

Views

Views in MySQL are indeed **"virtual tables"** that are used to view data from one or more tables. Views do not have their data but rather store data virtually, consisting of rows and columns. Views are very helpful in restricting access to your application's critical data to third-party users. Views in MySQL can be created by selecting some/all columns and some/all rows of a table by filtering out the rows based on some condition(s).

Views help particularly in the following ways:

1. **Simplicity:** Instead of writing complex joins & queries, views provide a way of writing simple SELECT statements.
2. **Enhanced Security:** Views expose only the data to the third-party apps and hide the internal details like table structure, attributes, etc, thus adding extra security.
3. **Consistency:** By writing views instead of common queries, we can write a view that avoids multiple declarations & definitions of the same queries and eventually provides a centralized way.

Just like the normal tables, you can perform operations like create, update, drop, etc., on the views. We will look at their syntax along with examples now.

Create View in MySQL

A view in MySQL can be created based on a single table or multiple tables. The **CREATE VIEW** statement is used to create a view in MYSQL.

Create a View Based On Single Table

Syntax:

CREATE VIEW <view_name> AS

SELECT <column1>, <column2>....., <columnN>

FROM <table-name>

WHERE [conditions];

Example:

Let's us create a view named "**IITHyderabadStudentsView**" from the StudentDetails table. This view selects the students from the StudentDetails table who study in "IIT Hyderabad" university and outputs their details like student id, name, and age.

```
CREATE VIEW IITHyderabadStudentsView AS
SELECT sid, sname, age
FROM StudentDetails
WHERE university = "IIT Hyderabad";
```

Now to view the tuples in the IITHyderabadStudentsView, we will query the view the just like how we query a normal table.

```
SELECT * FROM IITHyderabadStudentsView;
```

Update View in MySQL

There are certain conditions that need to be satisfied to update a view. If any one of these conditions is not met, then we are not allowed to update the view.

1. The View should be created from a single table. If the view is created using multiple tables then we will not be allowed to update the view.
2. The View should not be created using nested queries or complex queries.
3. The View should have all **NOT NULL** values.
4. The **SELECT** statement should not have the **DISTINCT** keyword.
5. The SELECT statement which is used to create the view should not include **GROUP BY** clause or **ORDER BY** clause.

1. Update View Definition/Structure

To update the view for adding or remove columns and rows by changing WHERE clause condition, we can use **CREATE OR REPLACE VIEW** statement.

Syntax:

CREATE OR REPLACE VIEW <view_name> AS

SELECT <column1>, <column2>, ..., <columnN>

FROM <table_name>

WHERE [condition];

Example:

Suppose we want to update the view IITHyderabadStudentsView we created above and delete the column sid from this view from StudentDetails table, we can do this as follows:

```
CREATE OR REPLACE VIEW IITHyderabadStudentsView AS
SELECT sname, age
FROM StudentDetails
WHERE university = "IIT Hyderabad";
```

2. Insert Into View

To insert the new row into the view, we can do it in a similar way just like how we do it for normal tables.

Syntax:

```
INSERT INTO <view_name>(<column1>, <column2>, <column3>,.....)
VALUES(<value1>, <value2>, <value3>,.....);
```

Example:

Let us insert a new row in the view IITHyderabadStudentsView which we have created above in the example of "Create View-based On Single Table".

```
INSERT INTO IITHyderabadStudentsView(sid, sname, age)
VALUES(7, "Tenali Rama", 26);

SELECT * FROM IITHyderabadStudentsView;
SELECT * FROM StudentDetails;
```

3. Delete From View

To insert the new row into the view, we can do it in a similar way just like how we do it for normal tables. The syntax is:

```
DELETE FROM <view_name> WHERE [condition];
```

Example:

Let us delete a row in the view IITHyderabadStudentsView which we have created above in the example of "Create View-based On Single Table". Let us delete the details of student whose name is "Tenali Rama".

```
DELETE FROM IITHyderabadStudentsView WHERE sname = "Tenali Rama";
```

Drop View in MySQL

Suppose now there is no need of the created view anymore? So, we want to delete it now. MySQL allows to deletion an already existing view. We can drop a view using the DROP statement.

Syntax:

DROP VIEW <view_name>;

Example:

Let's suppose we want to delete the view named "PythonEnrolledView" that we created above, which can be done as follows:

DROP VIEW PythonEnrolledView;