CREATE VIEW

MySQL views are a type of virtual tables. They are stored in the database with an associated name. They allow users to do the following −

- Structure data in a way that users or classes of users find natural or intuitive.
- Restrict access to the data in such a way that a user can see and (sometimes) modify exactly what they need and no more.
- Summarize data from various tables which can be used to generate reports.

A view can be created from one or more tables, containing either all or selective rows from them. Unless indexed, a view does not exist in a database.

MySQL Create View Statement

Creating a view is simply creating a virtual table using a query. A view is an SQL statement that is stored in the database with an associated name. It is actually a composition of a table in the form of a predefined SQL query.

Syntax

Following is the syntax of the CREATE VIEW Statement –

```
CREATE VIEW view_name AS select_statements FROM table_name;
```

Example

Assume we have created a table using the SELECT statement as shown below –

```
CREATE TABLE CUSTOMERS (
  ID INT NOT NULL,
  NAME VARCHAR(15) NOT NULL,
  AGE INT NOT NULL,
  ADDRESS VARCHAR(25),
  SALARY DECIMAL(10, 2),
  PRIMARY KEY(ID)
);
```

```
INSERT INTO CUSTOMERS VALUES
(1, 'Ramesh', '32', 'Ahmedabad', 2000),
(2, 'Khilan', '25', 'Delhi', 1500),
(3, 'Kaushik', '23', 'Kota', 2500),
(4, 'Chaitali', '26', 'Mumbai', 6500),
(5, 'Hardik','27', 'Bhopal', 8500),
(6, 'Komal', '22', 'MP', 9000),
(7, 'Muffy', '24', 'Indore', 5500);
```

Following query creates a view based on the above create table –

```
CREATE VIEW first_view AS SELECT * FROM CUSTOMERS;
```

```
SELECT * FROM first_view;
```

With REPLACE and IF NOT EXISTS Clauses

Usually, if you try to create a view with the name same as an existing view an error will be generated as shown as –

```
CREATE VIEW first_view AS SELECT * FROM CUSTOMERS;
```

As the view already exists, following error is raised –

```
ERROR 1050 (42S01): Table 'first_view' already exists
```

So, you can use the REPLACE clause along with CREATE VIEW to replace the existing view.

```
CREATE OR REPLACE VIEW first_view AS SELECT * FROM CUSTOMERS;
```

With WHERE Clause

We can also create a view using the where clause as shown below –

```
CREATE VIEW test_view AS SELECT * FROM CUSTOMERS WHERE SALARY>3000;
```

Following are the contents of the above created view –

ID	NAME	AGE	ADDRESS	SALARY
4	Chaitali	25	Mumbai	6500.00
5	Hardik	27	Bhopal	8500.00
6	Komal	22	Hyderabad	4500.00
7	Muffy	24	Indore	10000.00

The With Check Option

The WITH CHECK OPTION is an option used with CREATE VIEW statement. The purpose of this WITH CHECK OPTION is to ensure that all UPDATE and INSERT statements satisfy the condition(s) in the query. If they do not satisfy the condition(s), the UPDATE or INSERT returns an error.

Syntax

Following is the syntax –

```
CREATE VIEW view_name
AS SELECT column_name(s)
FROM table_name
WITH CHECK OPTION;
```

Example

In the following example, we are creating a view using CREATE VIEW statement along with the WITH CHECK OPTION –

```
CREATE VIEW NEW_VIEW

AS SELECT * FROM CUSTOMERS

WHERE NAME IS NOT NULL

WITH CHECK OPTION;
```

The view is created as follows –

Creating a MySQL View Using Client Program

```
create_view_query = "CREATE VIEW view_name AS Select_statements FROM table"
cursorObj.execute(create_view_query);
```

```
import mysql.connector
#establishing the connection
connection = mysql.connector.connect(
  host='localhost',
  user='root',
  password='password',
  database='tut'
)
cursorObj = connection.cursor()
create_view_query = """
CREATE VIEW tutorial_view AS
SELECT tutorial_id, tutorial_title, tutorial_author, submission_date
FROM tutorials_tbl
WHERE submission_date >= '2023-01-01'
"""
cursorObj.execute(create_view_query)
connection.commit()
print("View created successfully.")
cursorObj.close()
connection.close()
```