Update View

The MySQL UPDATE statement is used on various database objects to update the existing data in them. This is a DML (Data Manipulation language) command.

We need to be careful while using the UPDATE statement as it can modify all the records in an object, if not selected beforehand. To avoid losing or re-inserting correct data, we use clauses to filter the records that need to be updated. This way, we can update either a single row or multiple rows selectively.

MySQL UPDATE View Statement

In MySQL, a view is a database object that can contain rows (all or selected) from an existing table. It can be created from one or many tables which depends on the provided SQL query to create a view.

There is no direct statement to update a MySQL view. We use the UPDATE statement to modify all or selective records in a view. The results are reflected back in the original table as well.

Syntax

The basic syntax of the UPDATE query with a WHERE clause is as follows −

```
UPDATE view_name
SET column1 = value1, column2 = value2...., columnN = valueN
WHERE [condition];
```

Note: We can combine N number of conditions using the AND or the OR operators.

Example

First of all, let us create a table with the name CUSTOMERS using the following query –

```
CREATE TABLE CUSTOMERS(
   ID int NOT NULL,
   NAME varchar(20) NOT NULL,
   AGE int NOT NULL,
   ADDRESS varchar(25),
   SALARY decimal(18, 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);
```

Creating a view –

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

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

```
SELECT * FROM CUSTOMERS_VIEW;
```

Updating this view –

Now, through the view we created, we are trying to update the age of Ramesh to 35 in the original CUSTOMERS table, using the following query –

```
UPDATE CUSTOMERS_VIEW SET AGE = 35 WHERE name = 'Ramesh';
```

Updating Multiple Rows and Columns

In MySQL, we can update multiple rows and columns of a table using the UPDATE statement. To update multiple rows, specify the condition in a WHERE clause such that only the required rows would satisfy it.

To update multiple columns, set the new values to all the columns that need to be updated. In this case, using the WHERE clause would narrow down the records of the table and not using the clause would change all the values in these columns.

Syntax

Following is the syntax to update multiple rows and columns –

```
UPDATE table_name

SET column_name1 = new_value, column_name2 = new_value...

WHERE condition(s)
```

Example

In the following query, we are trying to modify the NAME and AGE column values in the CUSTOMERS table for WHERE ID = 3:

```
UPDATE CUSTOMERS_VIEW
SET NAME = 'Kaushik', AGE = 24
WHERE ID = 3;
```

Example

If we want to modify all the records of AGE column in the CUSTOMERS table, we can use the following query –

```
UPDATE CUSTOMERS_VIEW SET AGE = 24;
```

Updated a View Using a Client Program

We have learned how to update a view using the SQL UPDATE query. In addition to it, we can also perform the update operation on a view using another client program.

```
update_view_query = "UPDATE tutorial_view SET tutorial_title = 'New Title' WHERE tutorial_id = 2"
cursorObj.execute(update_view_query)
```

```
import mysql.connector
#establishing the connection
connection = mysql.connector.connect(
    host='localhost',
    user='root',
    password='password',
    database='tut'
)
cursorObj = connection.cursor()
update_view_query = """
UPDATE tutorial_view
SET tutorial_title = 'New Title'
WHERE tutorial_id = 2
"""
```

```
cursorObj.execute(update_view_query)
connection.commit()
print("View updated successfully.")
cursorObj.close()
connection.close()
```