UPDATE VIEW in SQL

If you want to update the existing data within the view, use the **UPDATE** statement.

Syntax

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

Note: Not all views can be updated using the UPDATE statement.

If you want to update the view definition without affecting the data, use the **CREATE OR REPLACE VIEW** statement. you can use this syntax.

```
CREATE OR REPLACE VIEW view_name AS
SELECT column1, column2, ...
FROM table_name
WHERE condition;
```

Rules to Update Views in SQL

Certain conditions need to be satisfied to update a view. If any of these conditions are not met, the view can not be updated.

- The SELECT statement which is used to create the view should not include GROUP BY clause or ORDER BY clause.
- The SELECT statement should not have the DISTINCT keyword.
- The View should have all NOT NULL values.
- The view should not be created using nested queries or complex queries.
- 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.

Examples

Let’s look at different use cases for updating a view in SQL. We will cover these use cases with examples to get a better understanding.

Example 1: Update View to Add or Replace a View Field

We can use the **CREATE OR REPLACE VIEW** statement to add or replace fields from a view.

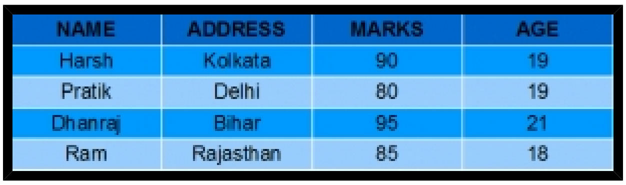
If we want to update the view **MarksView** and add the field AGE to this View from **StudentMarks Table**, we can do this by:

```
CREATE OR REPLACE VIEW MarksView AS
SELECT StudentDetails.NAME, StudentDetails.ADDRESS, StudentMarks.MARKS,
StudentMarks.AGE
FROM StudentDetails, StudentMarks
WHERE StudentDetails.NAME = StudentMarks.NAME;
```

If we fetch all the data from MarksView now as:

```
SELECT * FROM MarksView;
```

Output



NAME	ADDRESS	MARKS	AGE
Harsh	Kolkata	90	19
Pratik	Delhi	80	19
Dhanraj	Bihar	95	21
Ram	Rajasthan	85	18

Example 2: Update View to Insert a row in a view

We can insert a row in a View in the same way as we do in a table. We can use the INSERT INTO statement of SQL to insert a row in a View.

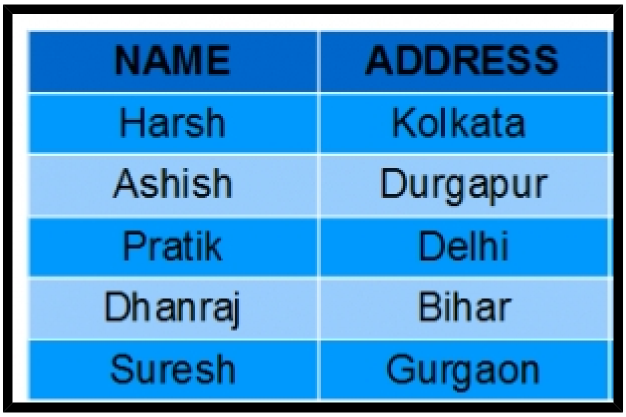
In the below example, we will insert a new row in the View DetailsView which we have created above in the example of “creating views from a single table”.

```
INSERT INTO StudentDetails
VALUES
   ('Suresh', 'Gurgaon');
```

If we fetch all the data from DetailsView now as,

```
SELECT * FROM DetailsView;
```

Output



NAME	ADDRESS
Harsh	Kolkata
Ashish	Durgapur
Pratik	Delhi
Dhanraj	Bihar
Suresh	Gurgaon

Example 3: Deleting a row from a View

Deleting rows from a view is also as simple as deleting rows from a table. We can use the DELETE statement of SQL to delete rows from a view. Also deleting a row from a view first deletes the row from the actual table and the change is then reflected in the view.

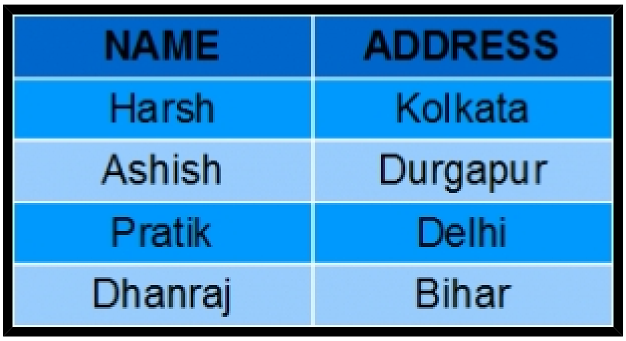
In this example, we will delete the last row from the view DetailsView which we just added in the above example of inserting rows.

```
DELETE FROM DetailsView
WHERE NAME = "Suresh";
```

If we fetch all the data from DetailsView now as,

```
SELECT * FROM DetailsView;
```

Output



NAME	ADDRESS
Harsh	Kolkata
Ashish	Durgapur
Pratik	Delhi
Dhanraj	Bihar

WITH CHECK OPTION Clause

The **WITH CHECK OPTION** clause in SQL is a very useful clause for views. It applies to an updatable view.

The WITH CHECK OPTION clause is used to prevent data modification (using INSERT or UPDATE) if the condition in the WHERE clause in the CREATE VIEW statement is not satisfied.

If we have used the WITH CHECK OPTION clause in the CREATE VIEW statement, and if the UPDATE or INSERT clause does not satisfy the conditions then they will return an error.

WITH CHECK OPTION Clause Example

In the below example, we are creating a View SampleView from the StudentDetails Table with a WITH CHECK OPTION clause.

```
CREATE VIEW SampleView AS
SELECT S_ID, NAME
FROM StudentDetails
WHERE NAME IS NOT NULL
WITH CHECK OPTION;
```

In this view, if we now try to insert a new row with a null value in the NAME column then it will give an error because the view is created with the condition for the NAME column as NOT NULL. For example, though the View is updatable then also the below query for this View is not valid:

```
INSERT INTO SampleView
VALUES(6);
```

NOTE: The default value of NAME column is null.