SQL Views

- Views in SQL are a kind of virtual table.
- A view also has rows and columns like tables, but a view doesn’t store data on the disk like a table.
- A View defines a customized query that retrieves data from one or more tables, and represents the data as if it was coming from a single source.
- We can create a view by selecting fields from one or more tables present in the database.
- A View can either have all the rows of a table or specific rows based on certain conditions.

Creating a View

- To create a view, you use the CREATE VIEW statement. Here’s the syntax:

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

- Parameters:

view_name: Name for the View

table_name: Name of the table

condition: Condition to select rows

- **Example 1: Creating View from a single table:**

```
CREATE VIEW DetailsView AS
SELECT NAME, ADDRESS
FROM StudentDetails
WHERE S_ID < 5;
```

- **To see the data in the View, you query the view in the same manner as we query a table.**

```
SELECT * FROM DetailsView;
```

- **Example 2: Creating View from multiple tables**

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

- We can list View using the SHOW FULL TABLES statement or using the information_schema table.
- Syntax :

```
USE "database_name";
SHOW FULL TABLES WHERE table_type LIKE "%VIEW";
```

- We can delete or drop View using the DROP statement.

  Syntax : DROP VIEW view_name;

- You can also update data through a view, but there are some restrictions. The view must be updatable, which generally means:
  - The view should be based on a single table.
  - The view should not contain any aggregated data (like SUM, AVG, etc.).
  - The view should not contain any joins, subqueries, or set operations like UNION.
  - All columns being updated must come from the same base table.

- 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.

```
INSERT INTO DetailsView(NAME, ADDRESS)
VALUES("Suresh","Gurgaon");
```

- We can use the CREATE OR REPLACE VIEW statement to add or replace fields from a view.
- For example in the query below we update the view MarksView and add the field **AGE** to this View from StudentMarks Table,

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

- 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.
- Deleting a row from a view first deletes the row from the actual table and the change is then reflected in the view.
- For example:

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