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. 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.

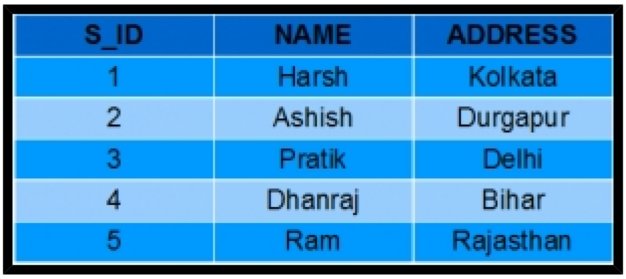
We can create a view by selecting fields from one or more tables present in the database. A View can either have all the rows of a table or specific rows based on certain conditions.

In this article, we will learn about creating, updating, and deleting views in SQL.

Demo SQL Database

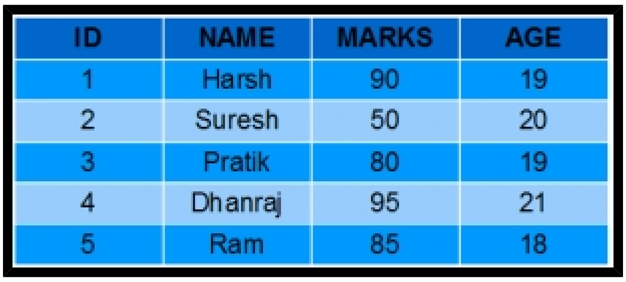
We will be using these **two SQL tables** for examples.

StudentDetails



S_ID	NAME	ADDRESS
1	Harsh	Kolkata
2	Ashish	Durgapur
3	Pratik	Delhi
4	Dhanraj	Bihar
5	Ram	Rajasthan

StudentMarks



ID	NAME	MARKS	AGE
1	Harsh	90	19
2	Suresh	50	20
3	Pratik	80	19
4	Dhanraj	95	21
5	Ram	85	18

You can create these tables on your system by writing the following SQL query:

```
-- Create StudentDetails table
CREATE TABLE StudentDetails (
    S_ID INT PRIMARY KEY,
    NAME VARCHAR(255),
    ADDRESS VARCHAR(255)
);
INSERT INTO StudentDetails (
    S_ID, NAME, ADDRESS
) VALUES
    (1, 'Harsh', 'Kolkata'),
    (2, 'Ashish', 'Durgapur'),
    (3, 'Pratik', 'Delhi'),
    (4, 'Dhanraj', 'Bihar'),
    (5, 'Ram', 'Rajasthan');
```

```
-- Create StudentMarks table
CREATE TABLE StudentMarks (
    ID INT PRIMARY KEY,
    NAME VARCHAR(255),
    Marks INT,
    Age INT
);
INSERT INTO StudentMarks (
    ID, NAME, Marks, Age
) VALUES
    (1, 'Harsh', 90, 19),
    (2, 'Suresh', 50, 20),
    (3, 'Pratik', 80, 19),
    (4, 'Dhanraj', 95, 21),
    (5, 'Ram', 85, 18);
```

CREATE VIEWS in SQL

We can create a view using **CREATE VIEW** statement. A View can be created from a single table or multiple tables.

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

SQL CREATE VIEW Statement Examples

Let's look at some examples of CREATE VIEW Statement in SQL to get a better understanding of how to create views in SQL.

Example 1: Creating View from a single table

In this example, we will create a View named DetailsView from the table StudentDetails.

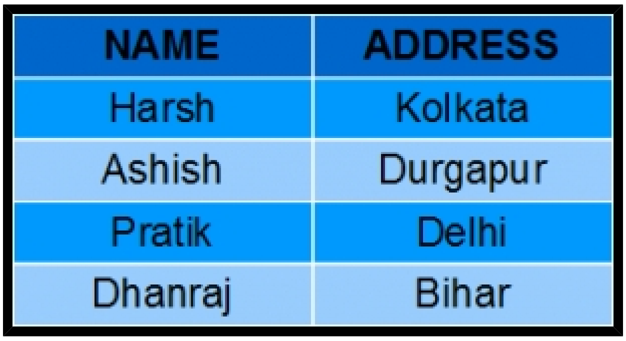
Query

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

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

```
SELECT *
FROM DetailsView;
```

Output



NAME	ADDRESS
Harsh	Kolkata
Ashish	Durgapur
Pratik	Delhi
Dhanraj	Bihar

Example 2: Create View From Table

In this example, we will create a view named StudentNames from the table StudentDetails.

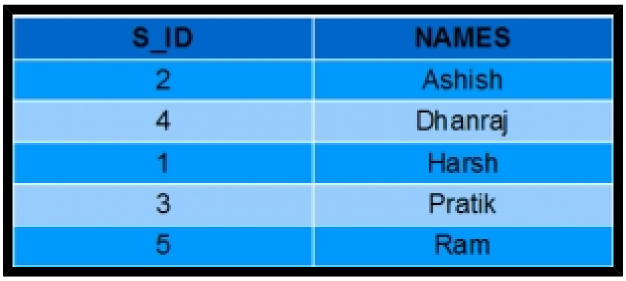
Query

```
CREATE VIEW StudentNames AS
SELECT S_ID, NAME
FROM StudentDetails
ORDER BY NAME;
```

If we now query the view as,

```
SELECT * FROM StudentNames;
```

Output



S_ID	NAMES
2	Ashish
4	Dhanraj
1	Harsh
3	Pratik
5	Ram

Example 3: Creating View from multiple tables

In this example we will create a View named MarksView from two tables StudentDetails and StudentMarks. To create a View from multiple tables we can simply include multiple tables in the SELECT statement.

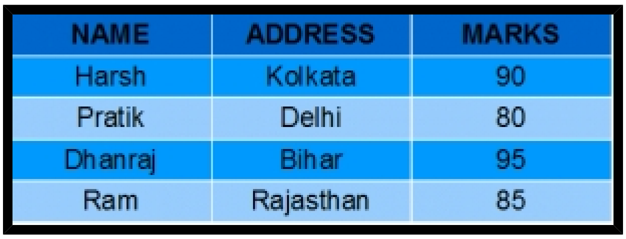
```
CREATE VIEW MarksView AS
SELECT StudentDetails.NAME, StudentDetails.ADDRESS, StudentMarks.MARKS
FROM StudentDetails, StudentMarks
```

```
WHERE StudentDetails.NAME = StudentMarks.NAME;
```

To display data of View MarksView:

```
SELECT * FROM MarksView;
```

Output



NAME	ADDRESS	MARKS
Harsh	Kolkata	90
Pratik	Delhi	80
Dhanraj	Bihar	95
Ram	Rajasthan	85

LISTING ALL VIEWS IN A DATABASE

We can list View using the **SHOW FULL TABLES** statement or using the **information_schema table**. A View can be created from a single table or multiple tables.

Syntax

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

Using information_schema

```
SELECT table_name
FROM information_schema.views
WHERE table_schema = 'database_name';
```

OR

```
SELECT table_schema, table_name, view_definition
FROM information_schema.views
WHERE table_schema = 'database_name';
```

Uses of a View

A good database should contain views for the given reasons:

- **Restricting data access** – Views provide an additional level of table security by restricting access to a predetermined set of rows and columns of a table.
- **Hiding data complexity** – A view can hide the complexity that exists in multiple joined tables.
- **Simplify commands for the user** – Views allow the user to select information from multiple tables without requiring the users to actually know how to perform a join.

- **Store complex queries** – Views can be used to store complex queries.
- **Rename Columns** – Views can also be used to rename the columns without affecting the base tables provided the number of columns in view must match the number of columns specified in a select statement. Thus, renaming helps to hide the names of the columns of the base tables.
- **Multiple view facility** – Different views can be created on the same table for different users.

Key Takeaways About SQL Views

- Views in SQL are a kind of virtual table.
- The fields in a view can be from one or multiple tables.
- We can create a view using the CREATE VIEW statement and delete a view using the DROP VIEW statement.
- We can update a view using the CREATE OR REPLACE VIEW statement.
- WITH CHECK OPTION clause is used to prevent inserting new rows that do not satisfy the view's filtering condition.