SQL | Views

Views in SQL are kind of virtual tables. A view also has rows and columns as they are in a real table in the database. 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 condition.

1. CREATING VIEWS

We can create 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;
```

StudentDetails

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

* create table sd (sid int,sname varchar(10),sadd varchar(10));

* insert into sd values(1,'Harish','kolkatta'),(2,'Ashish','Durgapur'),(3,'Pratik','Delhi'),
  (4,'Dhanraj','Bihar'),(5,'Ram','Rajesthan');

* select * from sd;

```
+-----+---------+-----------+
| sid | sname   | sadd      |
+-----+---------+-----------+
|   1 | Harish  | kolkata   |
|   2 | Ashish  | Durgapur  |
|   3 | Pratik  | Delhi     |
|   4 | Dhanraj | Bihar     |
|   5 | Ram     | Rajesthan |
+-----+---------+-----------+
```

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

* create table sm (sid int,sname varchar(10),marks int,age int);

* insert into sm values(1,'Harish',90,19),(2,'Suresh',50,20),(3,'Pratik',80,19),(4,'Dhanraj',95,21),
(5,'Ram',85,18);

* mysql> select * from sd;

```
+-----+---------+-----------+
| sid | sname   | sadd      |
+-----+---------+-----------+
|   1 | Harish  | kolkata   |
|   2 | Ashish  | Durgapur  |
|   3 | Pratik  | Delhi     |
|   4 | Dhanraj | Bihar     |
|   5 | Ram     | Rajesthan |
+-----+---------+-----------+
```

Creating View from a single table:

* create view vsd as select sid , sname from sd where sid<4;

* mysql> select * from vsd;

```
+--------+----------+
| sname  | sadd     |
+--------+----------+
| Harish | kolkata  |
| Ashish | Durgapur |
| Pratik | Delhi    |
+--------+----------+
```

NAME	ADDRESS
Harsh	Kolkata
Ashish	Durgapur
Pratik	Delhi
Dhanraj	Bihar

Creating View from multiple tables:

* create view vsdm as select sd.sname,sd.sadd,sm.marks from sd,sm where sd.sname=sm.sname;

```
mysql> select * from vsdm;

+---------+-----------+-------+

| sname   | sadd      | marks |

+---------+-----------+-------+

| Harish  | kolkata   |    90 |

| Pratik  | Delhi     |    80 |

| Dhanraj | Bihar     |    95 |

| Ram     | Rajesthan |    85 |

+---------+-----------+-------+
```

Output:

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

2. DELETING VIEWS

Syntax:

DROP VIEW view_name;

view_name: Name of the View which we want to delete.

3. UPDATING VIEWS

There are certain conditions needed to be satisfied to update a view. If any one of these conditions is **not** met, then we will not be allowed to update the view.

1. The SELECT statement which is used to create the view should not include GROUP BY clause or ORDER BY clause.
2. The SELECT statement should not have the DISTINCT keyword.
3. The View should have all NOT NULL values.
4. The view should not be created using nested queries or complex queries.
5. 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.

CREATE OR REPLACE VIEW

Syntax:

CREATE OR REPLACE VIEW view_name AS
SELECT column1,coulmn2,..
FROM table_name
WHERE condition;

4.INSERTING A NEW ROW INTO VIEW

Syntax:

INSERT INTO view_name(column1, column2 , column3,..) VALUES(value1, value2, value3..);

Example:

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

SELECT * FROM DetailsView;

Output:

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

5. DETETING A ROW FROM A VIEW

Syntax:

DELETE FROM view_name
WHERE condition;

Example:

DELETE FROM DetailsView WHERE NAME="Suresh";

- SELECT * FROM DetailsView;

Output:

NAME	ADDRESS
Harsh	Kolkata
Ashish	Durgapur
Pratik	Delhi
Dhanraj	Bihar

6. WITH CHECK OPTION (For Insert / Update Purpose Only)

- The WITH CHECK OPTION clause is used to prevent the insertion of rows in the view where the condition in the WHERE clause in CREATE VIEW statement is not satisfied.

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

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

query for this View is not valid:

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