Frame view VIEW in SQL

In SQL, a VIEW view is a virtual table in a database whose content is defined through a certain SQL statement.

In SQL, **a VIEW view** is a virtual table in a database whose content is defined through a certain SQL statement. A VIEW consists of rows and columns like a real table. Fields in a view are fields from one or more real tables in Database.

The difference between VIEW view and table is that VIEW is not considered a data storage structure that exists in the database. Essentially the observed data in VIEW is taken from tables through data query statements and used to restrict database access or to hide complex data.

In this article, Quantrimang will give you detailed instructions on how to use VIEW in SQL with syntax and specific examples to make it easier to visualize and capture commands.

Create VIEW in SQL

VIEW is created by **CREATE VIEW statement** . VIEWs can be created from a single table, multiple tables or from other VIEW.

The basic syntax of CREATE VIEW command in SQL is as follows:

```
CREATE VIEW ten_view AS
SELECT cot1, cot2.
FROM ten_bang
WHERE [dieu_kien];
```

You can add multiple tables in the SELECT statement similar to using them in a normal SQL SELECT query.

Examples of VIEW in SQL

Suppose the *NHANVIEN* table has the following records:

```
+----+----------+-----+-----------+----------+ | ID | TEN |TUOI | DIACHI | LUONG
```

The following is an example to create a VIEW from the *NHANVIEN* table . This VIEW is used to get the name and age from the *NHANVIEN* table .

```
SQL > CREATE VIEW NHANVIEN_VIEW AS
SELECT ten, tuoi
FROM NHANVIEN;
```

Now, you can query *NHANVIEN_VIEW* in the same way as you query the actual table, for example:

```
SQL > SELECT * FROM NHANVIEN_VIEW;
```

The above command returns the result:

```
+----------+-----+ | ten | tuoi| +----------+-----+ | Thanh | 32 | | Loan | 25
```

WITH CHECK OPTION option in SQL

WITH CHECK OPTION is an option of CREATE VIEW command. The purpose of WITH CHECK OPTION is to ensure that all UPDATE and INSERT satisfy the conditions in the VIEW definition.

If they do not satisfy the conditions, UPDATE and INSERT will return an error.

The following example creates the view *NHANVIEN_VIEW* with the WITH CHECK OPTION option.

```
CREATE VIEW NHANVIEN_VIEW AS
SELECT ten, tuoi
FROM NHANVIEN
WHERE tuoi IS NOT NULL
WITH CHECK OPTION;
```

In this case, if you try to UPDATE or INSERT *NHANVIEN_VIEW* with the value *tuoi = null* , an error will occur, but if another NULL, UPDATE or INSERT will succeed.

UPDATE a VIEW in SQL

A VIEW can be updated under the following specific conditions:

1. The SELECT clause cannot contain the keyword DISTINCT.
2. SELECT clause cannot contain sum functions.
3. SELECT clause cannot contain aggregation functions.
4. The SELECT clause cannot contain calculation expressions.
5. SELECT clause must not contain the ORDER BY clause.
6. The FROM clause cannot contain multiple tables.
7. The WHERE clause cannot contain subqueries.
8. The query does not contain GROUP BY or HAVING.
9. The estimated columns cannot be updated.
10. All NOT NULL columns from the original table must be selected in the VIEW to trigger INSERT queries.

So if a VIEW satisfies all of the above rules, you can use **the UPDATE statement** for that VIEW. The following example updates for an employee named *Thanh.*

```
SQL > UPDATE NHANVIEN_VIEW
 SET AGE = 35
 WHERE ten = 'Thanh';
```

Finally, the original *NHANVIEN* table was updated and accordingly VIEW was updated. Now, try to query the original table and the SELECT statement will produce the result:

```
+----+----------+-----+----------+----------+ | ID | TEN |TUOI | DIACHI | LUONG
```

Insert rows into VIEW in SQL

Data rows can be inserted in a VIEW. UPDATE similar rules also apply to **INSERT statements** .

Here, it is not possible to insert rows into *NHANVIEN_VIEW* because we do not select all NOT NULL columns from the original table in VIEW. We insert rows into a VIEW in the same way when you insert them into a table.

Delete rows from VIEW in SQL

Data rows may be deleted from a VIEW. The same UPDATE and INSERT rules also apply to **DELETE statements** in SQL.

The following example will delete a row with *TUOI = 22* :

```
SQL > DELETE FROM NHANVIEN_VIEW
 WHERE tuoi = 22;
```

The result of a row in the original *NHANVIEN* table will be deleted and the result is similar to that VIEW itself. Now, try querying the original table, and the SELECT statement will produce the result:

```
+----+----------+-----+----------+----------+ | ID | TEN |TUOI | DIACHI | LUONG
```

Delete VIEW in SQL

You can delete VIEW if it is no longer needed. The syntax is as follows:

```
DROP VIEW ten_view;
```

Example delete view *NHANVIEN_VIEW* from the original table:

```
DROP VIEW NHANVIEN_VIEW;
```

In the next section, we will learn **how to use the HAVING clause in SQL** , remember to follow it.

Last lesson: TRUNCATE TABLE command in SQL

Next lesson: HAVING clause in SQL

You finished reading the article "**Frame view VIEW in SQL**" edited by the **TipsMake** team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.