

# **SQL Interview Question asked in**

The Oracle logo is displayed within a solid red rectangular box. The word "ORACLE" is written in a white, bold, sans-serif typeface. A registered trademark symbol (®) is positioned at the top right of the letter "E".

**ORACLE®**

# **Problem Statement**

**What is a view in SQL, and how does it differ from a table?**

**Can you perform DML operations (Update, Insert, Delete) on views in SQL?**

**How do views impact the underlying tables in a database?**

## **What is a view?**

- A view in SQL is a result set returned by a SELECT query which is defined at the view creation time .
- Unlike physical tables, a view does not store the data itself. instead, it represents a set of data derived from one or more underlying tables or other views.
- Views as used to hide complex business logic from end user.

# Can you perform DML operations (Update, Insert, Delete) on views?

- Interestingly ,answer is both yes and no . It depends on definition/complexity of the view.
- If view definition is based on single table then we can perform DML operations on view.
- If view definition is based on multiple tables using join then also we can perform DML operations.
- Complex views involving aggregations, subqueries, or functions may not be directly updatable.
- Views that use DISTINCT, GROUP BY, or HAVING clauses might not be updatable.

# Understanding things practically

## Approach 1: view with simple logic

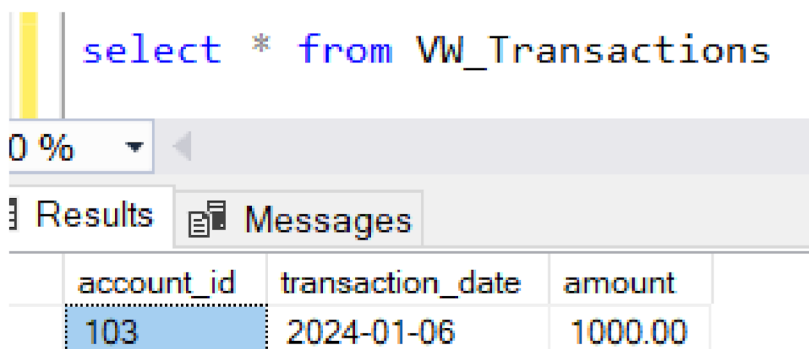
### Transactions Table

account_id	transaction_date	amount
103	2024-01-05	1000.00
103	2024-01-06	1000.00

### Transactions View

#### View Creation script:

```
CREATE VIEW VW_Transactions
AS
SELECT account_id,transaction_date,amount
FROM transactions
WHERE transaction_date='2024-01-06'
```



The screenshot shows a database client interface. At the top, a SQL query is entered in a text area: `select * from VW_Transactions`. Below the query area, there is a progress bar showing 0% completion. Underneath the progress bar, there are two tabs: 'Results' and 'Messages'. The 'Results' tab is active, displaying a table with the following data:

account_id	transaction_date	amount
103	2024-01-06	1000.00

We have applied where clause in view definition just to observe what happens with underlying table when DML operation is performed.

# Approach 1 continued

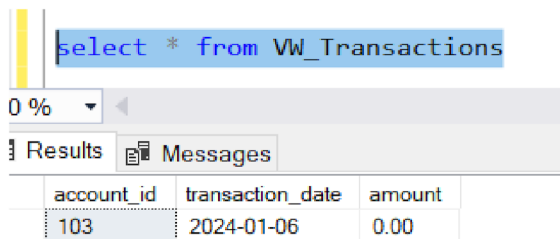
To start with, We will update amount column of view

Script:

```
update VW_Transactions  
set amount=0.0
```

Let's fetch our view and underlying table respectively.

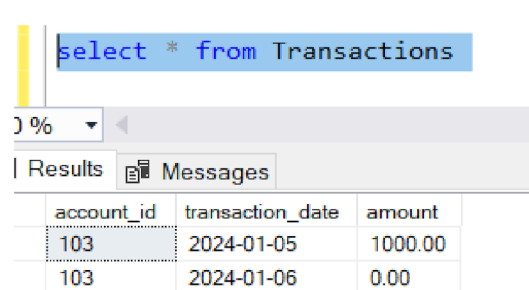
View post update



The screenshot shows a SQL query window with the query `select * from VW_Transactions`. Below the query, the 'Results' tab is active, displaying a table with three columns: `account_id`, `transaction_date`, and `amount`. The table contains one row with values 103, 2024-01-06, and 0.00.

account_id	transaction_date	amount
103	2024-01-06	0.00

Table post update



The screenshot shows a SQL query window with the query `select * from Transactions`. Below the query, the 'Results' tab is active, displaying a table with three columns: `account_id`, `transaction_date`, and `amount`. The table contains two rows: one with values 103, 2024-01-05, and 1000.00, and another with values 103, 2024-01-06, and 0.00.

account_id	transaction_date	amount
103	2024-01-05	1000.00
103	2024-01-06	0.00

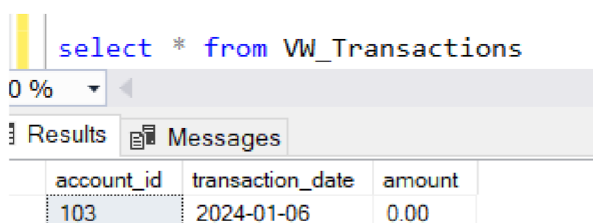
If you closely observe here, view got updated as expected and underlying table's only rows got updated which we have used in view definition. Other rows are unchanged.

Similarly we can perform delete operation and insert operation on view.

Insertion Script :

```
insert into VW_Transactions  
values (104,'2024-01-07',3000.00)
```

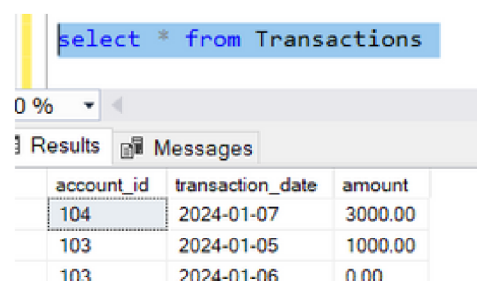
View post Insert



The screenshot shows a SQL query window with the query `select * from VW_Transactions`. Below the query, the 'Results' tab is active, displaying a table with three columns: `account_id`, `transaction_date`, and `amount`. The table contains one row with values 103, 2024-01-06, and 0.00.

account_id	transaction_date	amount
103	2024-01-06	0.00

Table post insert



The screenshot shows a SQL query window with the query `select * from Transactions`. Below the query, the 'Results' tab is active, displaying a table with three columns: `account_id`, `transaction_date`, and `amount`. The table contains three rows: one with values 104, 2024-01-07, and 3000.00, one with values 103, 2024-01-05, and 1000.00, and another with values 103, 2024-01-06, and 0.00.

account_id	transaction_date	amount
104	2024-01-07	3000.00
103	2024-01-05	1000.00
103	2024-01-06	0.00

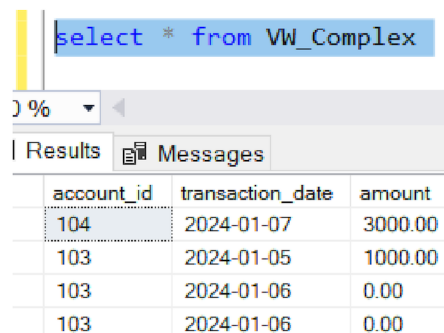
**Tell me in the comments section why inserted row is not appearing in view but it is there in underlying table.**

# Approach 2: Complex Logic View

Here we will use our previously used 2 objects i.e. Transaction table and VW\_Transaction view to create another view

## View Creation script:

```
CREATE VIEW VW_Complex
AS
SELECT account_id,transaction_date,amount FROM transactions
UNION all
SELECT account_id,transaction_date,amount FROM VW_Transactions
```



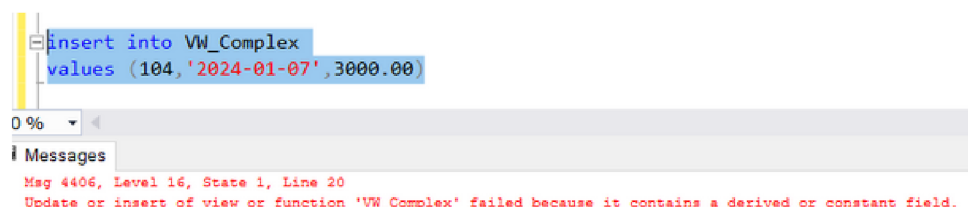
The screenshot shows a SQL query editor with the query `select * from VW_Complex` and its results. The results are displayed in a table with three columns: `account_id`, `transaction_date`, and `amount`. The data is as follows:

account_id	transaction_date	amount
104	2024-01-07	3000.00
103	2024-01-05	1000.00
103	2024-01-06	0.00
103	2024-01-06	0.00

Let's try to insert one row in newly created view.

Insertion Script :

```
insert into VW_Complex
values (104,'2024-01-07',3000.00)
```



In an attempt to insert a new row we got an error. As we used UNION ALL operator in view definition, SQL failed to figure out exact underlying table.