View in MS SQL

**\_\_ View \_\_**

* **View is Virtual (Imaginary) table.**
* **Its shows data of 1 or more table**
* **This doesn’t store data itself , it takes data from actual tables.**

**\_\_ Banifit \_\_**

* **View hides Actual Complex Query inside it. Give us Simple Query.**
* **Views useful because they make it easier to work with complex data.**

---- Create View ----------

CREATE VIEW view\_name

AS

SELECT column1, column2, ...

FROM table\_name

WHERE condition;

---- Select View ----------

select \* from view\_name

---- Drop View ----------

drop view view\_name

---- Alter View ----------

ALTER VIEW view\_name

AS

SELECT new\_column1, new\_column2, ...

FROM new\_table\_name

WHERE new\_conditions;

**--location of View in Explorer**

**(Database >> Views)**

**Example 1 : Basic Example**

To create a basic view that joins these two tables and shows the customer name and order total for each order, you can use the following SQL code:

create table customers(

customer\_id int primary key identity,

customer\_name varchar(25),

customer\_email varchar(25)

)

INSERT INTO customerss

VALUES ( 'John Smith', 'john@example.com'), ( 'ABC', 'Abc@example.com'), ('Jazz Cahs', 'Jazz@example.com'),

('Billal', 'Billal@example.com'), ('Saqib', 'Saqib@example.com'), ('John Smith', 'john@example.com')

create table orders(

order\_id int primary key identity,

customer\_id int,

totalOrder int,

order\_date date

)

INSERT INTO ordersss

VALUES ( 1, 90.00, '2023-03-01'),(1, 100.00, '2023-03-01'),( 2, 40.00, '2023-03-01'),

(2, 25.00, '2023-03-01'),(2, 50.00, '2023-03-01'),(3, 90.00, '2023-03-01'),(3, 10.00, '2023-03-01'),

(3, 11.00, '2023-03-01'),(1, 30.00, '2023-03-01'),(2, 70.00, '2023-03-01');

---\_\_\_\_\_\_\_\_\_\_\_ 1. WithOut View \_\_\_\_\_\_\_\_\_\_\_

--Let ager mujhaa 2 tables ka data 🡪 Daily base par Get krnaa ha kisi Company me to 🡪 kya main Rozaa Querian lkhtaa rhon gaa 🡪 NO 🡪 NO

select

order\_id , c.customer\_id , customer\_name,order\_date,customer\_email,totalOrder

from orders o

join customers c on c.customer\_id = o.customer\_id

---\_\_\_\_\_\_\_\_\_\_\_ 2. By View \_\_\_\_\_\_\_\_\_\_\_

-- will create View 🡪 use this View on Daily Bases..

create view customerOrdersDetail

as

select

order\_id , c.customer\_id , customer\_name,order\_date,customer\_email,totalOrder

from orders o

join customers c on c.customer\_id = o.customer\_id

SELECT \* FROM customerOrdersDetail

**Example 2 : Basic Example**

**View** that shows the average order total for each customer. Jus include a **GROUP BY** clause and the **AVG** function:

create View CustomerOrderAvgSummery

as

--- Let i want to use this Long Query again agian so the i will use Viewf

select

c.customer\_name ,

avg(o.totalOrder) avrg\_order

from customerss c

join Ordersss o on c.customer\_id = o.customer\_id

group by c.customer\_name

select \* from CustomerOrderAvgSummery

**Example 3 : shows the top 3 customers by average order total**

INSERT INTO ordersss

VALUES ( 4, 90.00, '2023-03-01'),(5, 100.00, '2023-03-01'),( 2, 40.00, '2023-03-01'),

(5, 25.00, '2023-03-01'),(5, 50.00, '2023-03-01'),

( 6, 90.00, '2023-03-01'),(6, 100.00, '2023-03-01'),( 6, 40.00, '2023-03-01'),

(6, 25.00, '2023-03-01'),(6, 50.00, '2023-03-01')

--\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 1st Way \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

create View Top3CustomerByOrder\_orderBy

as

SELECT top 3 customer\_name, avg\_order\_total

FROM (

SELECT c.customer\_name,

AVG(o.totalOrder) as avg\_order\_total

FROM Customerss c

JOIN Ordersss o ON c.customer\_id = o.customer\_id

GROUP BY c.customer\_name

) as subQuery

order by avg\_order\_total desc

--\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2nd Way \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

create View Top3CustomerByOrder\_over

as

SELECT customer\_name, avg\_order\_total

FROM (

SELECT c.customer\_name,

AVG(o.totalOrder) as avg\_order\_total,

ROW\_NUMBER() OVER (ORDER BY AVG(o.totalOrder) DESC) as row\_num

FROM Customerss c

JOIN Ordersss o ON c.customer\_id = o.customer\_id

GROUP BY c.customer\_name

) as subquery

WHERE row\_num <= 3