Create a new table called Sales\_order\_info as a derived table from the above then it should have Product\_ID ,Category, Sub\_Category ,Product\_Name ,Sales ,Quantity , per\_quantity\_price , sales\_type ,Discount ,Profit , Loss

**CODE:**

**CREATE TABLE Sales\_order\_info AS**

**SELECT Product\_ID ,Category, Sub\_Category ,Product\_Name ,Sales ,Quantity ,Discount ,Profit FROM sales\_purchase\_data;**

**ALTER table Sales\_order\_info Add column (per\_quantity\_price int, sales\_type varchar(20), Loss int);**

1) product id should be in this format

ex FUR-BO-10001798 to 10001798

**CODE:**

**UPDATE Sales\_order\_info**

**SET Product\_ID = SUBSTRING\_INDEX(Product\_ID, '-', -1)**

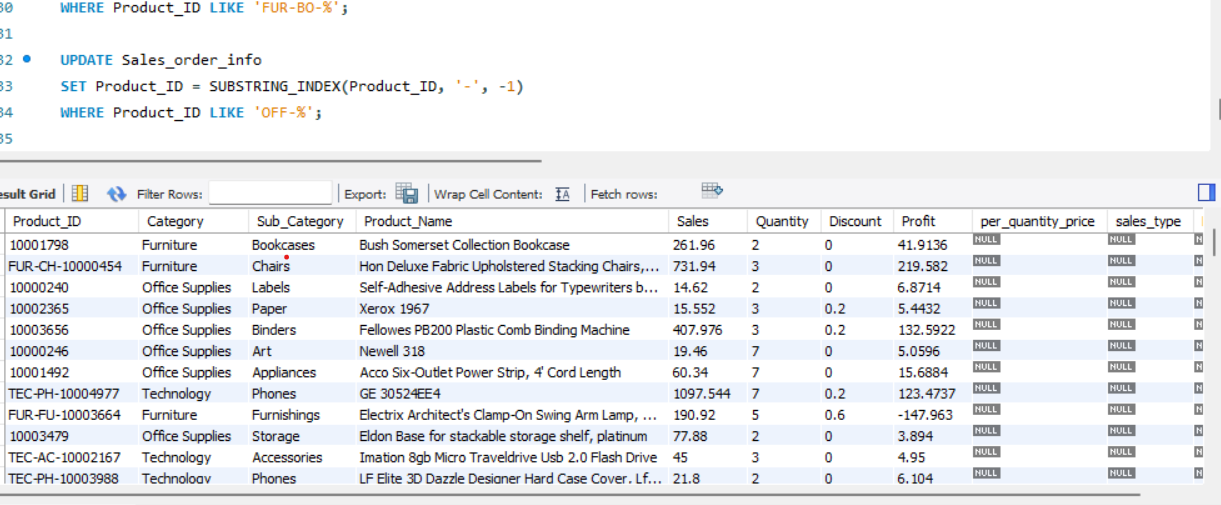
**WHERE Product\_ID LIKE 'FUR-BO-%';**

**UPDATE Sales\_order\_info**

**SET Product\_ID = SUBSTRING\_INDEX(Product\_ID, '-', -1)**

**WHERE Product\_ID LIKE 'OFF-%';**

**OUTPUT:**

****

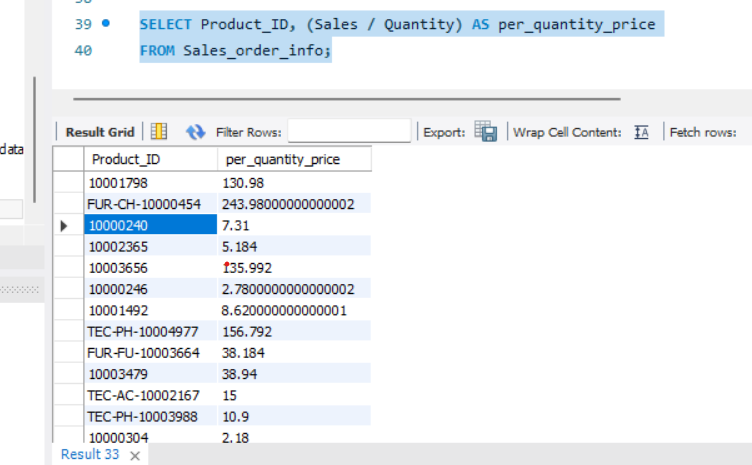
2) per\_quantity\_price should be output of Sales / Quantity of each customer records

**CODE:**

**SELECT Product\_ID, (Sales / Quantity) AS per\_quantity\_price**

**FROM Sales\_order\_info;**

**OUTPUT:**

****

3) sales\_type should be 3 categories

if sales are higher than 1000 then 'Super'

if sales are higher than 400 but lesser than 1000 then 'Average'

if sales are lesser than 400 then 'low'

**CODE:**

**SELECT Sales,**

**CASE**

**WHEN Sales > 1000 THEN 'Super'**

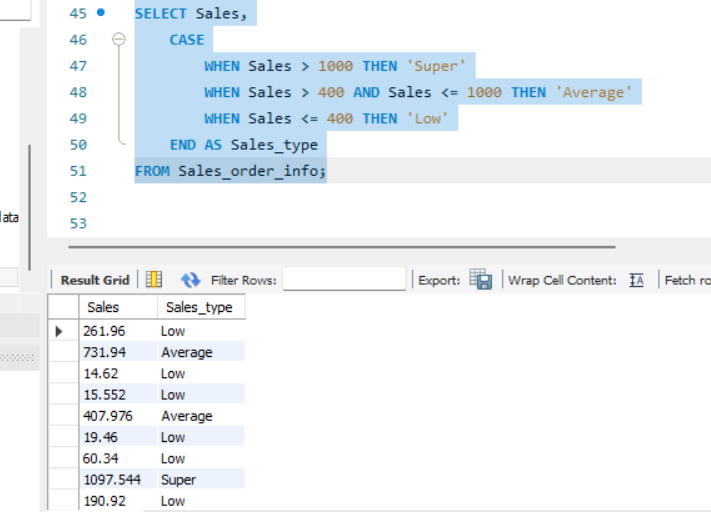
**WHEN Sales > 400 AND Sales <= 1000 THEN 'Average'**

**WHEN Sales <= 400 THEN 'Low'**

**END AS Sales\_type**

**FROM Sales\_order\_info;**

**OUTPUT:**

****

4)Loss will be 1 if profit is negative

**CODE:**

**SELECT profit,**

**CASE**

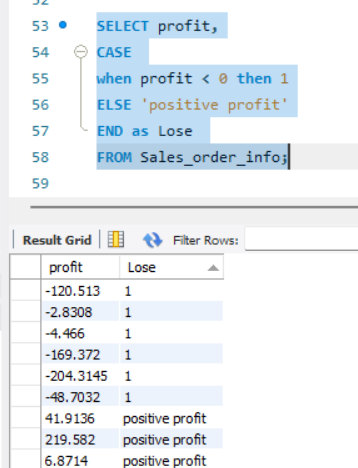
**when profit < 0 then 1**

**ELSE 'positive profit'**

**END as Lose**

**FROM Sales\_order\_info;**

**OUTPUT:**

****

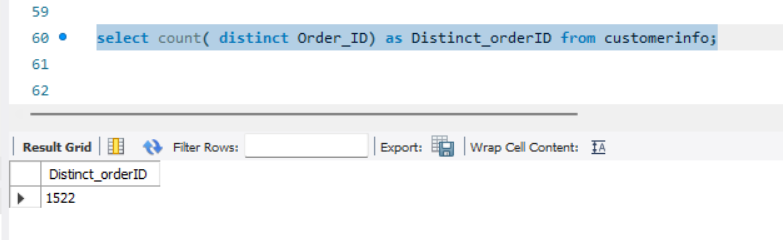
------Use single select statements to execute the below

5) Count of distinct order\_ids in the sales\_purchase\_data\_updated table

**CODE:**

**select count( distinct Order\_ID) as Distinct\_orderID from customerinfo;**

**OUTPUT:**

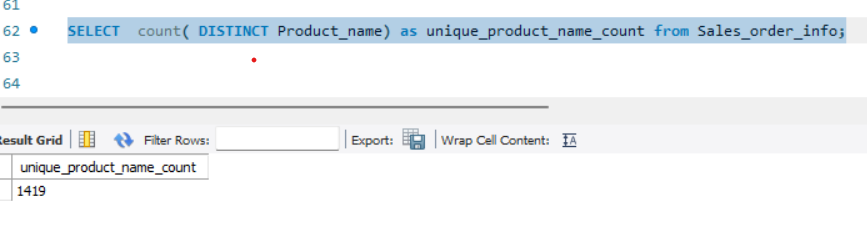
****

6) Count of Unique product names in Sales\_order\_info

**CODE:**

**SELECT count( DISTINCT Product\_name) as unique\_product\_name\_count from Sales\_order\_info;**

**OUTPUT:**

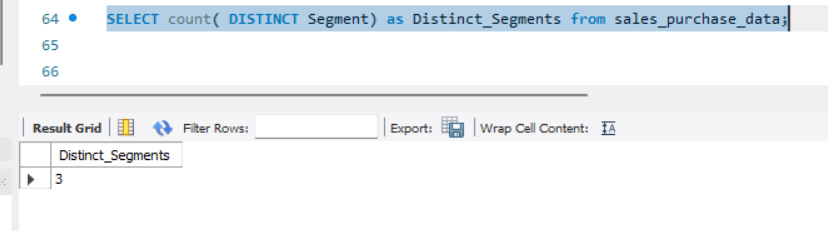


7) Count of distinct Segments in the sales\_purchase\_data\_updated table

**CODE:**

**SELECT count( DISTINCT Segment) as Distinct\_Segments from sales\_purchase\_data;**

**OUTPUT:**

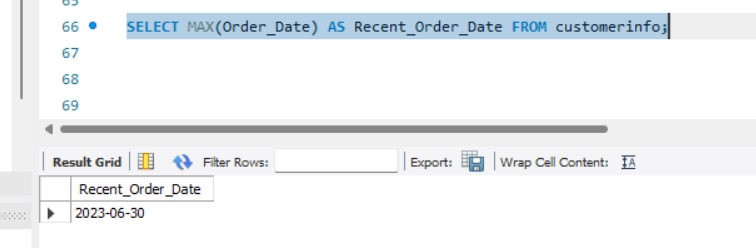
****

8) Recent order date in sales\_purchase\_data\_updated table

**CODE:**

**SELECT MAX(Order\_Date) AS Recent\_Order\_Date FROM customerinfo;**

**OUTPUT:**

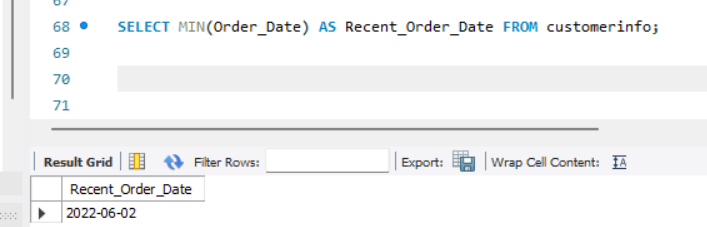
****

9)Old order date in sales\_purchase\_data\_updated table

**CODE:**

**SELECT MIN(Order\_Date) AS Recent\_Order\_Date FROM customerinfo;**

**OUTPUT:**

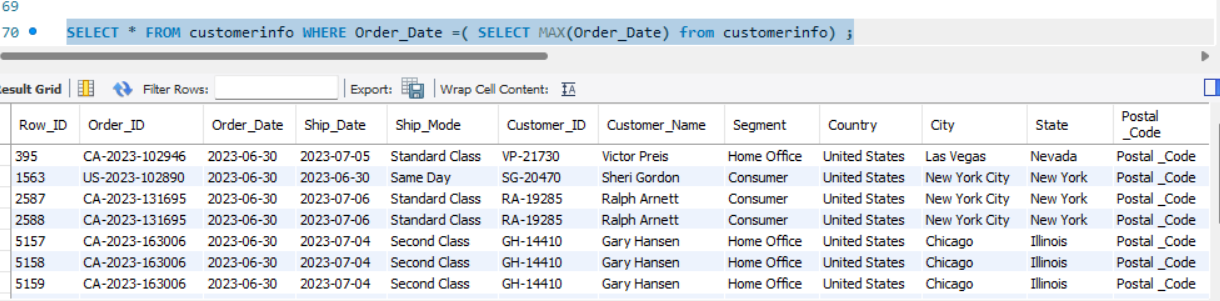


10) Customer info of all the columns for the maximum order date

**CODE:**

**SELECT \* FROM customerinfo WHERE Order\_Date =( SELECT MAX(Order\_Date) from customerinfo) ;**

**OUTPUT:**

****

11) No .of Unique Customers from Texas and Newyork

o/p No\_of\_cust\_texas | No\_of\_cust\_New\_york

**CODE:**

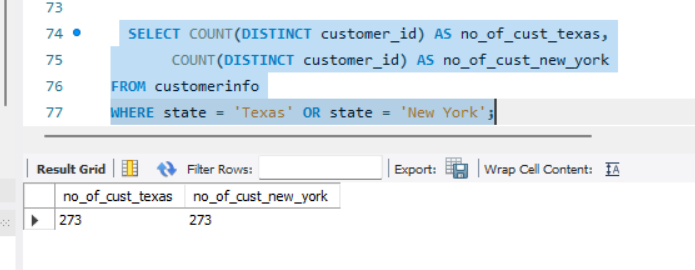
**SELECT COUNT(DISTINCT customer\_id) AS no\_of\_cust\_texas,**

**COUNT(DISTINCT customer\_id) AS no\_of\_cust\_new\_york**

**FROM customerinfo**

**WHERE state = 'Texas' OR state = 'New York';**

**OUTPUT:**

****