

----- E-COMMERCE DATA PROJECT -----

--> Create a view to combine ORDERS_DETAILS AND ORDERS_LIST tables

```
CREATE VIEW COMBINED_ORDERS AS
SELECT O.ORDER_ID, O.AMOUNT, O.PROFIT, O.QUANTITY, O.CATEGORY, O.SUB_CATEGORY, L.ORDER_DATE, L.CUSTOMERNAME,
L.STATE, L.CITY
FROM ORDER_DETAILS AS O
INNER JOIN
ORDERS_LIST AS L
ON O.ORDER_ID = L.ORDER_ID
```

--> Find the number of orders, customers, cities, and states.

```
SELECT COUNT(DISTINCT ORDER_ID) AS TOTAL_ORDERS,
COUNT(DISTINCT CUSTOMERNAME) AS TOTAL_CUSTOMERS,
COUNT(DISTINCT CITY) AS TOTAL_CITIES,
COUNT(DISTINCT STATE) AS TOTAL_STATES
FROM COMBINED_ORDERS
```

--> Find the top 10 profitable states & cities so that the company can expand its business.

Determine the number of products sold and the number of customers in these top 10 profitable states & cities.

```
SELECT TOP 10 STATE, CITY, COUNT(DISTINCT CUSTOMERNAME) AS TOTAL_CUSTOMERS,
SUM(PROFIT) AS TOTAL_PROFIT,
SUM(QUANTITY) AS TOTAL_QUANTITY
FROM COMBINED_ORDERS
GROUP BY STATE, CITY
ORDER BY TOTAL_PROFIT DESC
```

--> Find the new customers who made purchases in the year 2019. Only shows the top 5 new customers and their respective cities and states. Order the result by the amount they spent.

```
SELECT TOP 5 CUSTOMERNAME, CITY, STATE, SUM(AMOUNT) AS SALES
FROM COMBINED_ORDERS
WHERE CUSTOMERNAME NOT IN
( SELECT DISTINCT CUSTOMERNAME
FROM COMBINED_ORDERS
WHERE YEAR (TRY_CONVERT(DATE,ORDER_DATE,105))=2018)
AND YEAR(TRY_CONVERT(DATE,ORDER_DATE,105)) = 2019
GROUP BY CUSTOMERNAME, CITY, STATE
ORDER BY SALES DESC
```

--> Find the total sales, total profit, and total quantity sold for each category and sub-category. Return the maximum cost and maximum price for each sub-category too.

```
SELECT CATEGORY, SUB_CATEGORY, SUM(AMOUNT) AS SALES, SUM(PROFIT) AS PROFIT, SUM(QUANTITY) AS QUANTITY
FROM ORDER_DETAILS
GROUP BY CATEGORY, SUB_CATEGORY
ORDER BY CATEGORY
```

--> Find the Maximum cost per unit and maximum price per unit for each category and sub-category

```
SELECT CATEGORY, SUB_CATEGORY, MAX(COST_PER_UNIT) AS MAX_COST, MAX(PRICE_PER_UNIT) AS MAX_PRICE FROM
(SELECT *, ROUND((AMOUNT-PROFIT)/QUANTITY,2) AS COST_PER_UNIT, ROUND((AMOUNT/QUANTITY),2) AS PRICE_PER_UNIT
FROM ORDER_DETAILS) M
GROUP BY CATEGORY, SUB_CATEGORY
ORDER BY MAX_COST DESC
```

--> Determine the number of orders, the total sales and the total profit for different days of the week

\\For easy execution create a view which gives you the day_of_the_week

```
CREATE VIEW ORDERS_DAY AS(
SELECT *,CASE DATEPART(DW,TRY_CONVERT(DATE,ORDER_DATE,105))
WHEN 1 THEN 'SUNDAY'
WHEN 2 THEN 'MONDAY'
WHEN 3 THEN 'TUESDAY'
WHEN 4 THEN 'WEDNESDAY'
WHEN 5 THEN 'THURSDAY'
WHEN 6 THEN 'FRIDAY'
WHEN 7 THEN 'SATURDAY'
END AS DAYNAME
FROM COMBINED_ORDERS )
```

```
SELECT DAYNAME, SUM(AMOUNT) AS SALES, SUM(PROFIT) AS PROFIT, SUM(QUANTITY) AS QUANTITY_SOLD
FROM ORDERS_DAY GROUP BY DAYNAME
ORDER BY SALES DESC
```