Q.1) Find out the top 3 Products (“productLine”) for each month that generated the highest profit.

A.) Which “productline” generated the highest profit in the June Month?

B.) Submit the 2nd highest profit amount for the month of October.

C.) Select the “productline” which is not in the top 3 lists in the month of November.

Answer :

Select \*

From (Select month\_name,

productline,

profit,

Rank() over(partition by month\_name Order By profit DESC) As Ranks

From (Select productline,

Monthname(o. orderdate) As Month\_Name,

Sum((od.priceeach - p. buyprice)\*od.quantityordered) As profit

From orderdetails od

Join orders o

ON od.ordernumber = o.ordernumber

Join products p

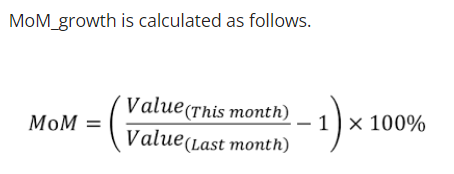
ON p.productcode = od.productcode

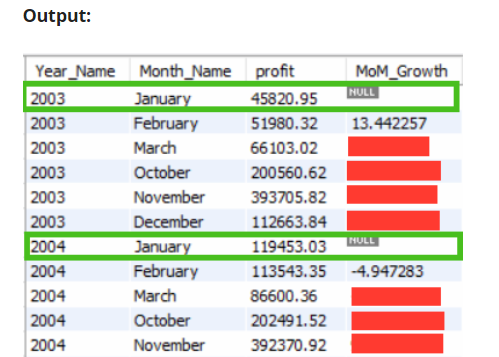
Group by month\_name

productline) a) b

Where ranks <=3

Q.2) Find the Month on Month growth in profit for each year.





1. What is the MoM for Feb 2005?
2. What is the MoM for June 2004?
3. For the year 2004, select which month shows negative MoM growth?

Answer :

Select year\_name,

month\_name,

profit,

((profit/Lag(profit, 1, Null) over(partition by year\_name) ) -1) \*100 as MOM\_Growth

From (Select Year(o. orderdate) As Year\_Name,

Monthname(o. orderdate) As Month\_Name,

Month(o. orderdate) As Month\_num,

Sum((od.priceeach - p.buyprice)\*od.quantityordered) as profit

From orderdetails od

Join orders o

ON od. ordernumber = o. ordernumber

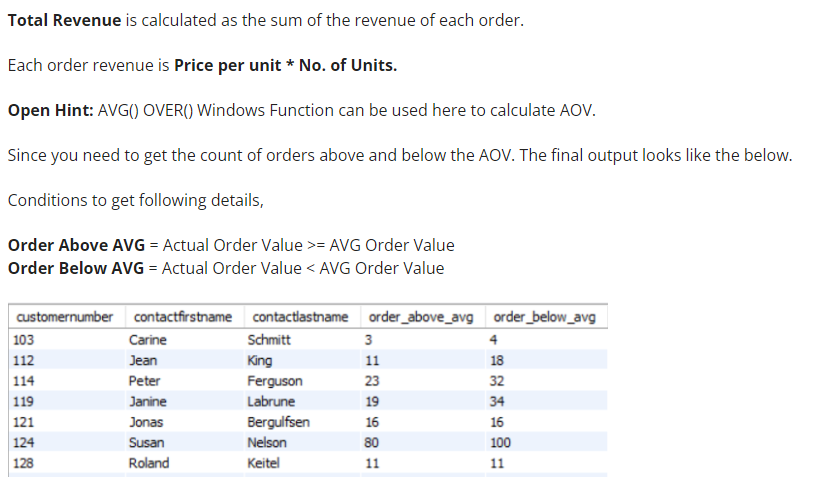
Join products p

On p.productcode = od.productcode

Group by month\_name,

year\_name) a

Q.3) For each customer i.e (customerNumber) count the number of orders above and below the average order value.



1. What is the Average Order Value? Submit the value in the answer.
2. Number of orders above AOV
3. Number of orders below AOV

Answer :

Select customernumber,

contactfirstname,

contactlastname,

Sum(order\_value >= avg\_order\_value) order\_above\_avg,

Sum(order\_value < avg\_order\_value) order\_below\_avg

From (Select c.customernumber,

c.contactfirstname,

c.contactlastname,

o. ordernumber,

od. quantityordered,

od. priceeach,

priceeach\*quantityordered As order\_value,

Round(Avg(priceeach\*quantityordered) over(),2) avg\_order\_value

From orderdetails od

join orders o

On od. ordernumber = o.ordernumber

join customers c

on c. customernumber = o. customernumber) a

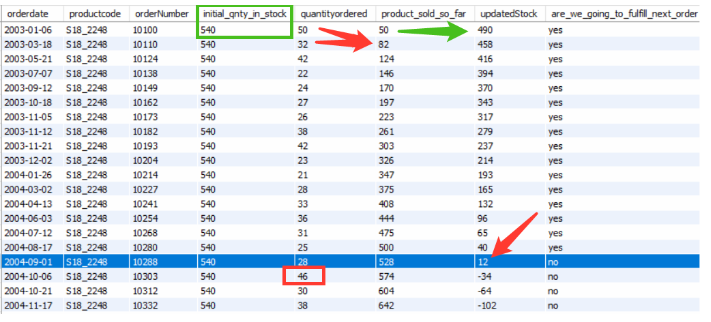
Group by customernumber,

contactfirstname,

Contactlastname

Q.4) In the business, customer order the product in bulk and it is the utmost priority of the business to fulfill the order requirement as to when required. Hence a business has to keep checking their stock regularly to give uninterrupted delivery.

To make that happen, create a report as shown below to get the updated stock and status to check if we are running out of stock (or are we going to fulfill the next customer order?)



Consider the first row as an example where product code **“S18\_2248”**. Its initial stock quantity was **540**. After the first order i.e orderNumber 10100 from the customer i.e **order quantity of 50**, the updated stock value would be (540 - 50) 490. Now, the status needs to be updated based on the next order quantity, for example, the Next order is of quantity 32 and we have 490 products in stock hence we can easily serve the next order request (Status is **Yes**)

But if you look at the blue highlighted row then, in that case, “updatedStock” value is 12 and next order is of quantity 46 hence the status would **No**.

Select the list of productCodes that will be getting out of Stock (i.e “are\_we\_going\_to\_fulfill\_next\_order” value is **No**)

Select orderdate,

productcode,

ordernumber,

quantityordered,

product\_sold\_so\_far,

quantityinstock - product\_sold\_so\_far As updatedStock,

IF(quantityinstock - product\_sold\_so\_far - Lead(quantityordered, 1, 0) over(partition by productcode orderby orderdate) > 0,"yes", "no"

As are\_we\_going\_to\_fulfill\_next\_order

From (Select o.orderdate,

o.ordernumber,

p.productcode,

od.quantityordered,

od.priceeach,

p. quantityinstock,

Sum(od. quantityordered) over(partition by p.productcode order by o.orderdate) As product\_sold\_so\_far

From orderdetails od

join products p

on od.productcode = p.productcode

join orders o

on od.ordernumber = o.ordernumber) a