**Analytical SQL Project**

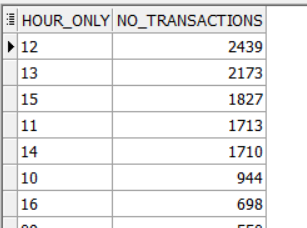
Aliaa Adel

# Question 1

* At what time does most transactions happen

select sub.\*

from(

SELECT distinct(TO\_CHAR(invoicedate, 'HH24')) as hour\_only, count(\*) over(partition by TO\_CHAR(invoicedate, 'HH24')) no\_transactions

FROM tableRetail)sub

order by no\_transactions desc

here it appears that most transactions happen at 12 and 1 pm thus to increase total sales special promotions or flash sales can be offered to customers

* What are the top 10 selling products?

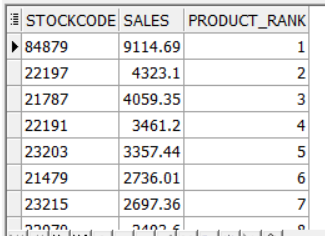
select STOCKCODE, sales ,product\_rank

from (

select STOCKCODE ,sum(price \* quantity) sales,

rank() over (order by sum(price \* quantity) desc) product\_rank

from tableRetail

 group by STOCKCODE

)

where product\_rank <= 10;

the sales of each product is the product of the quantity and price

* What is the percentage difference from previous month’s sales?

with sales\_change as(

select sub.\*, lead(sales,1) over(order by sub.date\_month desc ) prev\_month\_sales from(

select to\_char(invoicedate, 'YYYY-MM') date\_month ,

sum(price \* quantity) over(partition by to\_char(invoicedate, 'YYYY-MM')) sales

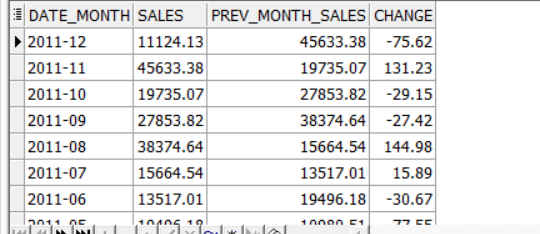
from tableRetail)sub

group by sub.date\_month, sub.sales

)

select cte.\* , round(((sales-prev\_month\_sales)/prev\_month\_sales)\*100,2) change

from sales\_change cte ;

it’s shown that December 2011 has the lowest percentage difference but when looking at the previous month sales it’s approx. 132% increase as it was during black Friday sales so it makes sense

while august has the highest positive percentage difference as it’s the start of summer vacation and a lot of festivals take place during that time

* What are the total sales for weekends and weekdays per month?

select sub.\*

from( select

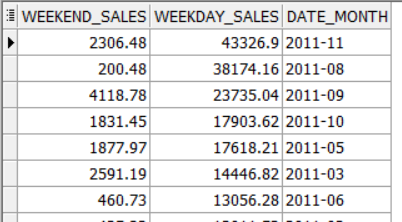
sum(case when to\_char(invoicedate, 'D') in (1, 7) then price \* quantity else 0 end ) over(partition by to\_char(invoicedate, 'YYYY-MM')) weekend\_sales,

sum(case when to\_char(invoicedate, 'D') not in (1, 7) then price \* quantity else 0 end ) over(partition by to\_char(invoicedate, 'YYYY-MM')) weekday\_sales,

to\_char(invoicedate, 'YYYY-MM') date\_month

from tableRetail) sub

group by sub.date\_month,sub.weekend\_sales,sub.weekday\_sales

order by sub.weekday\_sales desc

from this results it’s shown that weekdays sales are much higher than weekends but with respect to each type of sales (weekend /weekday) it’s found that September has the highest weekday sales because of the holiday season and September has the highest weekend sales

knowing this information, we could push ads to the customer for the most selling products during these days in the month that has the highest sales during 12 to 1 pm at which most transactions happen

# Question 2

\*\*\* note that I ran this commands in sql plus first

*--ALTER SESSION SET NLS\_DATE\_FORMAT = 'MM/DD/YYYY HH24:MI';*

Then changed the datatype of invoice date to date

*-- started by craeting the RFM values first*

with RFM as(

select distinct(customer\_id) , count(distinct(invoice)) over(partition by customer\_id) frequency,

round(max(invoicedate) over () - max(invoicedate) over( partition by customer\_id)) recency,

sum(price\*quantity ) over( partition by customer\_id) monetary

from tableretail

order by customer\_id

)

*-- then creating the system by calculating each score*

, sys as(

select customer\_id, frequency,recency,monetary,r\_score, round((f\_score+m\_score)/2) fm\_score

from (

select cte.\* , ntile(5) over(order by frequency) f\_score ,

ntile(5) over(order by recency desc) r\_score,

ntile(5) over(order by monetary) m\_score

from RFM cte)

order by customer\_id

)

*-- segementation of the customers based on the scores*

SELECT

s.customer\_id,

s.r\_score,

s.fm\_score,

CASE CONCAT(r\_score, fm\_score)

WHEN '55' THEN 'Champions'

WHEN '54' THEN 'Champions'

WHEN '45' THEN 'Champions'

WHEN '52' THEN 'Potential Loyalists'

WHEN '42' THEN 'Potential Loyalists'

WHEN '33' THEN 'Potential Loyalists'

WHEN '43' THEN 'Potential Loyalists'

WHEN '53' THEN 'Loyal Customers'

WHEN '44' THEN 'Loyal Customers'

WHEN '35' THEN 'Loyal Customers'

WHEN '34' THEN 'Loyal Customers'

WHEN '51' THEN 'Recent Customers'

WHEN '41' THEN 'Promising'

WHEN '31' THEN 'Promising'

WHEN '32' THEN 'Customers Needing Attention'

WHEN '23' THEN 'Customers Needing Attention'

WHEN '22' THEN 'Customers Needing Attention'

WHEN '25' THEN 'At Risk'

WHEN '24' THEN 'At Risk'

WHEN '13' THEN 'At Risk'

WHEN '15' THEN 'Can''t Lose Them'

WHEN '14' THEN 'Can''t Lose Them'

WHEN '12' THEN 'Hibernating'

 WHEN '11' THEN 'Lost'

ELSE 'other'

END AS customer\_group

from sys s;

# Question 3

Question 3a

with consecutive\_days as (

select sub3.\* ,first\_value(sub3.running\_sum) OVER (PARTITION BY sub3.CUST\_ID,sub3.groups ORDER BY sub3.running\_sum desc )days

from(

select sub2.\*, sum(date\_diff) OVER (PARTITION BY CUST\_ID,groups ORDER BY CALENDAR\_DT) running\_sum

from (

select sub.\* ,sum(case when date\_diff =1 then 0 else 1 end) OVER (PARTITION BY CUST\_ID ORDER BY CALENDAR\_DT) groups

from

(

select CUST\_ID, CALENDAR\_DT , COUNT (\*) OVER (PARTITION BY CUST\_ID ORDER BY CALENDAR\_DT

RANGE BETWEEN INTERVAL '1' DAY PRECEDING AND CURRENT ROW)-1 AS date\_diff

from transactions) sub) sub2)sub3

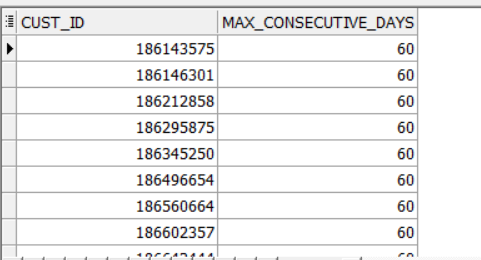
)

select CUST\_ID , max(days) max\_consecutive\_days

from consecutive\_days cte

group by cust\_id

order by max(days) desc;



Question 3b

select distinct(cust\_id) ,first\_value(days\_count) over(PARTITION BY CUST\_ID ORDER BY days\_count desc) max\_days\_till\_250LE

from(

select cust\_id, calendar\_dt,amt\_le,

count(\*) over(PARTITION BY CUST\_ID ORDER BY CALENDAR\_DT) days\_count,

sum(amt\_le) over(PARTITION BY CUST\_ID order by CALENDAR\_DT) total

from transactions) sub

where total <= 250

order by cust\_id;

