

# 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

	HOUR_ONLY	NO_TRANSACTIONS
▶	12	2439
	13	2173
	15	1827
	11	1713
	14	1710
	10	944
	16	698

- 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

	STOCKCODE	SALES	PRODUCT_RANK
▶	84879	9114.69	1
	22197	4323.1	2
	21787	4059.35	3
	22191	3461.2	4
	23203	3357.44	5
	21479	2736.01	6
	23215	2697.36	7

- 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

DATE_MONTH	SALES	PREV_MONTH_SALES	CHANGE
2011-12	11124.13	45633.38	-75.62
2011-11	45633.38	19735.07	131.23
2011-10	19735.07	27853.82	-29.15
2011-09	27853.82	38374.64	-27.42
2011-08	38374.64	15664.54	144.98
2011-07	15664.54	13517.01	15.89
2011-06	13517.01	19496.18	-30.67
2011-05	19496.18	10000.51	77.55

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

WEEKEND_SALES	WEEKDAY_SALES	DATE_MONTH
2306.48	43326.9	2011-11
200.48	38174.16	2011-08
4118.78	23735.04	2011-09
1831.45	17903.62	2011-10
1877.97	17618.21	2011-05
2591.19	14446.82	2011-03
460.73	13056.28	2011-06

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 creating 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
)
```

*-- segmentation 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;
```

	CUSTOMER_ID	R_SCORE	FM_SCORE	CUSTOMER_GROUP
▶	12747	5	5	Champions
	12748	5	5	Champions
	12749	5	5	Champions
	12820	5	4	Champions
	12821	1	1	Lost
	12822	3	3	Potential Loyalists
	12823	2	4	At Risk
	12824	3	2	Customers Needing Attention
	12826	5	5	Champions
	12827	5	3	Loyal Customers

## 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;
```

	CUST_ID	MAX_CONSECUTIVE_DAYS
▶	186143575	60
	186146301	60
	186212858	60
	186295875	60
	186345250	60
	186496654	60
	186560664	60
	186602357	60
	186602357	60

### 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;
```

	CUST_ID	MAX_DAYS_TILL_250LE
▶	26592	4
	45234	16
	54815	11
	60045	8
	66688	3
	113502	6
	145392	6
	150488	28