Q1

WITH CUSTOMER\_COUNTS AS (

SELECT

CUSTOMER\_ID,

COUNT(DISTINCT INVOICE) AS TOTAL\_PURCHASES

FROM

ONLINERETAIL

GROUP BY

CUSTOMER\_ID

)

SELECT

COUNT(\*) AS RETURNING\_CUSTOMERS

FROM

CUSTOMER\_COUNTS

WHERE

TOTAL\_PURCHASES > 1;

2.

SELECT

TO\_CHAR(TO\_DATE(INVOICEDATE, 'MM/DD/YYYY HH24:MI'), 'YYYY-Mon') AS MONTH,

COUNT(DISTINCT CUSTOMER\_ID) AS MONTHLY\_ACTIVE\_USERS

FROM

ONLINERETAIL

GROUP BY

TO\_CHAR(TO\_DATE(INVOICEDATE, 'MM/DD/YYYY HH24:MI'), 'YYYY-Mon')

ORDER BY

MONTHLY\_ACTIVE\_USERS DESC ;

3.

SELECT

TO\_DATE(INVOICEDATE, 'MM/DD/YYYY HH24:MI') AS INVOICE\_DATE,

SUM(PRICE\*QUANTITY) AS DAILY\_REVENUE

FROM

ONLINERETAIL

WHERE

TO\_CHAR(TO\_DATE(INVOICEDATE, 'MM/DD/YYYY HH24:MI'), 'YYYY-MM') = '2011-12'

GROUP BY

TO\_DATE(INVOICEDATE, 'MM/DD/YYYY HH24:MI')

ORDER BY

TO\_DATE(INVOICEDATE, 'MM/DD/YYYY HH24:MI');

4.

SELECT

TO\_CHAR(TO\_DATE(INVOICEDATE, 'MM/DD/YYYY HH24:MI'), 'YYYY-MM') AS MONTH,

COUNT(DISTINCT INVOICE) AS TOTAL\_ORDERS,

SUM(PRICE\*QUANTITY) AS MONTHLY\_REVENUE

FROM

ONLINERETAIL

GROUP BY

TO\_CHAR(TO\_DATE(INVOICEDATE, 'MM/DD/YYYY HH24:MI'), 'YYYY-MM')

ORDER BY

TOTAL\_ORDERS DESC, MONTHLY\_REVENUE DESC;

5.

SELECT

CORR(PRICE, QUANTITY) as CORRELATION

FROM

ONLINERETAIL

6.

select distinct Year\_, quarter,

count(QUANTITY) as Quantity,

SUM(PRICE\*QUANTITY) AS Quarter\_REVENUE

from (

SELECT OnlineRetail.\*,

TO\_CHAR(TO\_DATE(INVOICEDATE, 'MM/DD/YYYY HH24:MI'), 'YYYY') AS Year\_,

NTILE(4) OVER(PARTITION BY EXTRACT(YEAR FROM TO\_DATE(INVOICEDATE, 'MM/DD/YYYY HH24:MI')), ((EXTRACT(MONTH FROM TO\_DATE(INVOICEDATE, 'MM/DD/YYYY HH24:MI')) - 1) / 3) + 1 ORDER BY TO\_DATE(INVOICEDATE, 'MM/DD/YYYY HH24:MI')) AS quarter

FROM OnlineRetail)

Group By quarter,Year\_

order By Quarter\_REVENUE DESC , Quantity DESC

Q2

WITH Data\_ AS (

SELECT DISTINCT CUSTOMER\_ID,

TRUNC(LAST\_VALUE(TO\_DATE(INVOICEDATE, 'MM/DD/YYYY HH24:MI')) OVER (ORDER BY TO\_DATE(INVOICEDATE, 'MM/DD/YYYY HH24:MI') ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING) - LAST\_VALUE(TO\_DATE(INVOICEDATE, 'MM/DD/YYYY HH24:MI')) OVER (PARTITION BY CUSTOMER\_ID ORDER BY TO\_DATE(INVOICEDATE, 'MM/DD/YYYY HH24:MI') RANGE BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING)) AS recency,

COUNT(INVOICE) OVER (PARTITION BY CUSTOMER\_ID ORDER BY INVOICE ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING) AS frequency,

SUM(PRICE \* QUANTITY) OVER (PARTITION BY CUSTOMER\_ID) AS monetary

FROM ONLINERETAIL

),

rfm\_score As (

select Data\_.\* ,

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

Ntile (5) over (order by frequency ) as f\_score,

Ntile(5) over(order by monetary) as m\_score

from Data\_

),

fm\_score As (

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

from rfm\_score)

select customer\_id,recency,frequency,monetary,r\_score, fm\_score,

CASE

WHEN (r\_score = 5 AND fm\_score IN (4, 5)) OR

(r\_score = 4 AND fm\_score = 5) THEN 'Champions'

WHEN (r\_score = 5 AND fm\_score = 2) OR

(r\_score = 4 AND fm\_score in(2,3)) OR

(r\_score = 3 AND fm\_score =3) THEN 'Potential Loyalists'

WHEN (r\_score = 5 AND fm\_score = 3) OR

(r\_score = 4 AND fm\_score = 4) OR

(r\_score = 3 AND fm\_score in(5,4)) THEN 'Loyal Customers'

WHEN r\_score = 5 AND fm\_score = 1 THEN 'Recent Customers'

WHEN (r\_score = 4 AND fm\_score = 1) OR

(r\_score= 3 AND fm\_score=1) THEN 'Promising'

WHEN (r\_score = 3 AND fm\_score = 2) OR

(r\_score = 2 AND fm\_score = 3) OR

(r\_score = 2 AND fm\_score = 2) THEN 'Customers Needing Attention'

WHEN (r\_score = 2 AND fm\_score IN (4, 5)) OR

(r\_score = 1 AND fm\_score = 3) THEN 'At Risk'

WHEN (r\_score = 1 AND fm\_score IN (4, 5)) THEN 'Cant Lose Them'

WHEN r\_score = 1 AND fm\_score = 2 THEN 'Hibernating'

WHEN r\_score = 1 AND fm\_score = 1 THEN 'Lost'

Else 'About to sleep'

END AS cust\_segment

from fm\_score

Q3

1.

WITH cons\_days AS (

SELECT Cust\_Id, Calendar\_Dt, Amt\_LE,

ROW\_NUMBER() OVER (PARTITION BY Cust\_Id ORDER BY Calendar\_Dt) AS rn

FROM CUSTOMER\_PURCHASES

)

Select Cust\_Id,max(rn) as Max\_consecutive\_days

from cons\_days

group by Cust\_Id

order by Max\_consecutive\_days DESC

2.

with count\_days As (SELECT

Cust\_Id,

calendar\_dt,

amt\_le,

min(calendar\_dt)over(PARTITION BY Cust\_Id ORDER BY calendar\_dt) as min\_date,

SUM(amt\_le) OVER (PARTITION BY Cust\_Id ORDER BY calendar\_dt) AS running\_total,

COUNT(\*) OVER (PARTITION BY Cust\_Id ORDER BY calendar\_dt) AS running\_count

FROM CUSTOMER\_PURCHASES),

runnning\_total as (

select Cust\_Id,

calendar\_dt - min\_date date\_,

Amt\_LE,

running\_total,

running\_count

from count\_days

where running\_total >=250),

min\_values as (

select distinct Cust\_Id,

first\_value( running\_total) over (partition by Cust\_Id order by running\_count) as amount\_min\_value,

first\_value (running\_count) over(partition by Cust\_Id order by running\_count ) as count\_min\_value,

first\_value (date\_) over(partition by Cust\_Id order by running\_count ) as count\_days

from runnning\_total)

select avg (count\_min\_value) , avg(count\_days)

from min\_values;