Q1

select distinct country ,SUM (price\*quantity) OVER (

partition by country

) as "total sales for each country"

FROM tableretail

Graphical user interface, text

Description automatically generated with medium confidence

//based on the above query statement , I got the info that the market is based in UK there is no other country, and the total amount of sales is 255718.38.

SELECT distinct stockcode, sum(price\*quantity) OVER (

PARTITION BY stockcode

)as "total revenue for each stock"

from tableretail;

Table

Description automatically generated

// this query for getting the total sales for each stock, that’s why I used partition by stock code.

select stockcode, dense\_rank( )over(order by stockcode)

from tableretail

A picture containing text

Description automatically generated

//then performed the query above to get the best seller items

select distinct customer\_id, avg(price\*quantity) over (partition by customer\_id) as "avg cost for each customer"

from tableretail

Table

Description automatically generated

//from the query above we got the average sales cost for each customer.

select distinct count(invoice) over (partition by invoicedate) as "tatal invoices per date", invoicedate

from tableretail;

Table

Description automatically generated

// the query above is to show the total invoices for each date.

select count(invoice) as invoice\_count,customer\_id

from tableretail

group by customer\_id

order by invoice\_count desc;

Table

Description automatically generated

// the total invoice count for each customer, to identify customers with the most invoices.

Q2:

with c as (

select distinct customer\_id, count(\*) over (partition by customer\_id) as number\_of\_orders,

sum(quantity\*price) over (partition by customer\_id) as sum\_cost\_for\_each\_customer,

round(MAX(TO\_DATE(invoicedate, 'MM/DD/YYYY HH24:MI')) OVER() - MAX(TO\_DATE(invoicedate, 'MM/DD/YYYY HH24:MI')) OVER(PARTITION BY customer\_id)) AS Recency

from tableretail),

c1 as (

select DISTINCT customer\_id,*--*

ntile(5) over (order by number\_of\_orders) as rfm\_frequency,

ntile(5) over (order by sum\_cost\_for\_each\_customer) as rfm\_monetary

,ntile(5) over (order by RECENCY) as r\_score

, (ntile(5) over (order by number\_of\_orders) + ntile(5) over (order by sum\_cost\_for\_each\_customer))/2 AS fm\_score

from c

)

select

DISTINCT C.customer\_ID, C.Recency,C1.rfm\_frequency,C1.rfm\_monetary, C1.r\_score, C1.fm\_score

,CASE

WHEN((R\_SCORE = 5) AND (FM\_SCORE =5 )) or ((R\_SCORE = 5) AND (FM\_SCORE =4 )) or((R\_SCORE = 4) AND (FM\_SCORE =4 ))

THEN 'Champions'

WHEN ( (R\_SCORE = 5) AND (FM\_SCORE =2 ) ) or ( (R\_SCORE = 4) AND (FM\_SCORE =2 )) or ( (R\_SCORE = 3) AND (FM\_SCORE =3 ))or ( (R\_SCORE = 4) 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 =5 )) or ( (R\_SCORE = 3) AND (FM\_SCORE =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 =5 )) or ( (R\_SCORE = 2) AND (FM\_SCORE =4 )) or ( (R\_SCORE = 1) AND (FM\_SCORE =3))

THEN 'At Risk'

WHEN ((R\_SCORE = 1) AND (FM\_SCORE =5 )) or ((R\_SCORE = 1) AND (FM\_SCORE =4))

THEN 'cant lose them'

WHEN (R\_SCORE = 1) AND (FM\_SCORE =5 )

THEN 'Hibernating '

WHEN (R\_SCORE =1) AND (FM\_SCORE =2)

THEN 'Hibernating'

WHEN (R\_SCORE =1) AND (FM\_SCORE =1 )

THEN 'Lost'

else 'other'

END AS rfm\_segment

from c,c1

WHERE C.CUSTOMER\_ID=C1.CUSTOMER\_ID

ORDER BY CUSTOMER\_ID;