* **Query to get the count of invoices , sum of price , average of price per invoice ,number of customers**

SELECT DISTINCT

country,

COUNT (DISTINCT invoice) OVER () AS "total invoices",

ROUND (SUM (price) OVER (), 2) AS "total price ",

COUNT (DISTINCT customer\_id) OVER () AS "number of customers ",

ROUND (AVG (price \* quantity) OVER (), 2)

AS " average price per invoice "

FROM tableretail

* **Query to get sum of sales per 2011 or 2010 , we can do the same query to get the average sales and count of invoices**

WITH tablecte2010

AS (SELECT invoice,

stockcode,

quantity,

price,

customer\_id,

country,

invoicedate,

ROUND (SUM (price) OVER (), 2) AS "total in 2010"

FROM tableretail

WHERE invoicedate LIKE '%2010%'),

tablecte2011

AS (SELECT invoice,

stockcode,

quantity,

price,

customer\_id,

country,

invoicedate,

ROUND (SUM (price) OVER (), 2) AS "total in 2011"

FROM tableretail

WHERE invoicedate LIKE '%2011%')

SELECT tablecte2010.invoice,

tablecte2010.stockcode,

tablecte2010.quantity,

tablecte2010.price,

tablecte2010.customer\_id,

tablecte2010.country,

tablecte2010.invoicedate,

tablecte2010."total in 2010",

tablecte2011."total in 2011"

FROM tablecte2010 JOIN tablecte2011 ON 1 = 1;

* **Query to get most stock code placed as a total**

SELECT stockcode, COUNT(\*) AS "Occurrences",

DENSE\_RANK() OVER (ORDER BY COUNT(\*) DESC) AS "Rank"

FROM tableretail

GROUP BY stockcode

ORDER BY "Occurrences" DESC

* **Query to get most stock code placed in 2010**

SELECT stockcode, COUNT(\*) AS "Occurrences",

DENSE\_RANK() OVER (ORDER BY COUNT(\*) DESC) AS "Rank"

FROM tableretail

where invoicedate like '%2010%'

GROUP BY stockcode

ORDER BY "Occurrences" DESC

* **Query to get most stock code placed in 2011**

SELECT stockcode, COUNT(\*) AS "Occurrences",

DENSE\_RANK() OVER (ORDER BY COUNT(\*) DESC) AS "Rank"

FROM tableretail

where invoicedate like '%2011%'

GROUP BY stockcode

ORDER BY "Occurrences" DESC

* **Query to get most cst placed orders as total**

select customer\_id,count(\*) as " number of invocies",

dense\_rank() over ( order by count(invoice) desc ) as "Rank"

from tableretail

group by customer\_id

* ***Query to get most cst placed orders in 2010 and 2011***

select customer\_id,count(\*) as " number of invocies",

dense\_rank() over ( order by count(invoice) desc ) as "Rank"

from tableretail

where invoicedate like '%2010%'

group by customer\_id

select customer\_id,count(\*) as " number of invocies",

dense\_rank() over ( order by count(invoice) desc ) as "Rank"

from tableretail

where invoicedate like '%2011%'

group by customer\_id

* ***Query to get most cst paid for the orders as a total***

SELECT

customer\_id,

sum(price) AS "Most Sales per cst",

DENSE\_RANK() OVER ( ORDER BY sum(price) DESC) AS "Rank"

FROM

tableretail

GROUP BY

customer\_id

* ***Query to get most cst paid for the orders in 2010 and 2011***

SELECT

customer\_id,

sum(price) AS " Most Sales per cst ",

DENSE\_RANK() OVER ( ORDER BY sum(price) DESC) AS "Rank"

FROM

tableretail

where invoicedate like '%2010%'

GROUP BY

customer\_id

====

SELECT

customer\_id,

sum(price) AS " Most Sales per cst ",

DENSE\_RANK() OVER ( ORDER BY sum(price) DESC) AS "Rank"

FROM

tableretail

where invoicedate like '%2010%'

GROUP BY

customer\_id

* ***Query to get top 3 cst paid for the orders***

with cte as (

SELECT

customer\_id,

sum(price) AS "number of invoices",

DENSE\_RANK() OVER (ORDER BY sum(price) DESC) AS "Rank"

FROM

tableretail

GROUP BY

customer\_id

)

SELECT \*

FROM cte

WHERE "Rank" <= 3;

Q2:

* ***Query to get the last purchase per each customer***

SELECT customer\_id,

MAX(invoicedate) AS " last purchase " ,

FROM tableretail

GROUP BY customer\_id

ORDER BY customer\_id

* ***Query to get the last purchase per cst – number of orders – totals sales***

SELECT t1.customer\_id,

t1.Recency,

COUNT(t1.invoice) AS "Number of Invoices ",

SUM(t1.price\*t1.quantity) AS “Total Price”

FROM (

SELECT customer\_id,

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),0) AS Recency,

invoice,

price,

quantity

FROM tableretail

) t1

GROUP BY t1.customer\_id, t1.Recency

ORDER BY t1.customer\_id

* ***Customer Segmentation***

SELECT t1.customer\_id,

t1.Recency,

COUNT(t1.invoice) AS Frequency,

SUM(t1.price) AS Monetary,

ntile(5) over (order by t1.recency) as r\_score,

ntile(5) over ( order by COUNT(DISTINCT t1.invoice),sum(t1.price)) as fm\_score,

CASE

WHEN ntile(5) over (order by t1.recency) = 5 AND

ntile(5) over ( order by COUNT(DISTINCT t1.invoice),sum(t1.price)) = 5 THEN 'champion'

WHEN ntile(5) over (order by t1.recency) = 4 AND

ntile(5) over ( order by COUNT(DISTINCT t1.invoice),sum(t1.price)) = 4 THEN 'champion'

WHEN ntile(5) over (order by t1.recency) = 5 AND

ntile(5) over ( order by COUNT(DISTINCT t1.invoice),sum(t1.price)) = 4 THEN 'champion'

WHEN ntile(5) over (order by t1.recency) = 4 AND

ntile(5) over ( order by COUNT(DISTINCT t1.invoice),sum(t1.price)) = 5 THEN 'champion'

WHEN ntile(5) over (order by t1.recency) = 5 AND

ntile(5) over ( order by COUNT(DISTINCT t1.invoice),sum(t1.price)) = 1 THEN 'Recent Customer'

WHEN ntile(5) over (order by t1.recency) = 5 AND

ntile(5) over ( order by COUNT(DISTINCT t1.invoice),sum(t1.price)) = 2 THEN 'Potential Loyalists'

WHEN ntile(5) over (order by t1.recency) = 5 AND

ntile(5) over ( order by COUNT(DISTINCT t1.invoice),sum(t1.price)) = 3 THEN 'Potential Loyalists'

WHEN ntile(5) over (order by t1.recency) = 4 AND

ntile(5) over ( order by COUNT(DISTINCT t1.invoice),sum(t1.price)) = 1 THEN 'Promising'

WHEN ntile(5) over (order by t1.recency) = 4 AND

ntile(5) over ( order by COUNT(DISTINCT t1.invoice),sum(t1.price)) = 2 THEN 'Promising'

WHEN ntile(5) over (order by t1.recency) = 4 AND

ntile(5) over ( order by COUNT(DISTINCT t1.invoice),sum(t1.price)) = 3 THEN 'Potential Loyalists'

WHEN ntile(5) over (order by t1.recency) = 3 AND

ntile(5) over ( order by COUNT(DISTINCT t1.invoice),sum(t1.price)) = 5 THEN 'Loyal Customer'

WHEN ntile(5) over (order by t1.recency) = 3 AND

ntile(5) over ( order by COUNT(DISTINCT t1.invoice),sum(t1.price)) = 4 THEN 'Loyal Customer'

WHEN ntile(5) over (order by t1.recency) = 3 AND

ntile(5) over ( order by COUNT(DISTINCT t1.invoice),sum(t1.price)) = 3 THEN 'Potential Loyalists'

WHEN ntile(5) over (order by t1.recency) = 3 AND

ntile(5) over ( order by COUNT(DISTINCT t1.invoice),sum(t1.price)) = 2 THEN 'Customers Needing Attention'

WHEN ntile(5) over (order by t1.recency) = 3 AND

ntile(5) over ( order by COUNT(DISTINCT t1.invoice),sum(t1.price)) = 1 THEN 'Promising'

WHEN ntile(5) over (order by t1.recency) = 2 AND

ntile(5) over ( order by COUNT(DISTINCT t1.invoice),sum(t1.price)) = 5 THEN 'At Risk'

WHEN ntile(5) over (order by t1.recency) = 2 AND

ntile(5) over ( order by COUNT(DISTINCT t1.invoice),sum(t1.price)) = 4 THEN 'At Risk'

WHEN ntile(5) over (order by t1.recency) = 2 AND

ntile(5) over ( order by COUNT(DISTINCT t1.invoice),sum(t1.price)) = 3 THEN 'At Risk'

WHEN ntile(5) over (order by t1.recency) = 2 AND

ntile(5) over ( order by COUNT(DISTINCT t1.invoice),sum(t1.price)) = 2 THEN 'Customers Needing Attention'

WHEN ntile(5) over (order by t1.recency) = 2 AND

ntile(5) over ( order by COUNT(DISTINCT t1.invoice),sum(t1.price)) = 2 THEN 'Customers Needing Attention'

WHEN ntile(5) over (order by t1.recency) =1 AND

ntile(5) over ( order by COUNT(DISTINCT t1.invoice),sum(t1.price)) = 5 THEN 'Cannot Lose Them'

WHEN ntile(5) over (order by t1.recency) =1 AND

ntile(5) over ( order by COUNT(DISTINCT t1.invoice),sum(t1.price)) = 4 THEN 'Cannot Lose Them'

WHEN ntile(5) over (order by t1.recency) =1 AND

ntile(5) over ( order by COUNT(DISTINCT t1.invoice),sum(t1.price)) = 3 THEN 'At Risk'

WHEN ntile(5) over (order by t1.recency) = 2 AND

ntile(5) over ( order by COUNT(DISTINCT t1.invoice),sum(t1.price)) = 1 THEN 'Customers Needing Attention'

WHEN ntile(5) over (order by t1.recency) = 1 AND

ntile(5) over ( order by COUNT(DISTINCT t1.invoice),sum(t1.price)) =2 THEN 'Hibernating'

WHEN ntile(5) over (order by t1.recency) = 1 AND

ntile(5) over ( order by COUNT(DISTINCT t1.invoice),sum(t1.price)) = 1 THEN 'Lost'

END as "Customer Segementation"

FROM (

SELECT DISTINCT customer\_id,

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),0) AS Recency,

invoice,

price

FROM tableretail

) t1

GROUP BY t1.customer\_id, t1.Recency

ORDER BY t1.customer\_id