**Q1- Using Online Retail dataset**

**• write at least 5 analytical SQL queries that tells a story about the data**

**• write small description about the business meaning behind each query**

The data set is about Online Retail store which sales his products all over the world

From the data we can find

1. The store sales in 38 country around the world

-- number of branches over the world

select count(distinct country)

from online\_retail

1. the top country buy from the store

-- ranking countries from highest to lowers

select \* ,dense\_rank() over (order by sort\_countries desc)

from (

select distinct country ,count(\*)over (partition by country) as sort\_countries

from online\_retail

) as counting

1. Top country is “United Kingdom” and the lowest country is “Saudi Arabia”

-- ranking countries from hieghest to lowers

select \* ,dense\_rank() over (order by sort\_countries desc) ,

first\_value(country) over() as top\_country ,

last\_value(country) over() as lowest\_country

from (

select distinct country ,count(\*)over (partition by country) as sort\_countries

from online\_retail

) as counting

1. the top 5 products in the store “WORLD WAR 2 GLIDERS ASSTD DESIGNS, JUMBO BAG RED RETROSPOT, ASSORTED COLOUR BIRD ORNAMENT, POPCORN HOLDER, PACK OF 72 RETROSPOT CAKE CASES”

-- top 5 product

select \* , dense\_rank()over(order by product\_sum desc ) from

(select distinct stockcode,description ,sum(quantity) over(partition by description) as product\_sum

from online\_retail

)as p\_sum

limit 5

1. number of customers in the store over all and how they increase by time

-- total number of customers in the store

select count(distinct customerid)

from online\_retail

-- total number of customers in the store in 2 years separately

select count(distinct customerid)

from(

select distinct customerid, invoicedate

from online\_retail

where invoicedate like '%2010%'

) as datee

select count(distinct customerid)

from(

select distinct customerid, invoicedate

from online\_retail

where invoicedate like '%2011%'

) as date

1. The total income from each country

-- total income from each country

select distinct country ,round(sum(quantity \* unitprice) over(partition by country ) :: numeric,2) as profit

from online\_retail

where stockcode != (stockcode||'c')

order by profit desc

**Q2- After exploring the data now you are required to implement a Monetary model for customers behavior for product purchasing and segment each customer based on the below groups Champions - Loyal Customers - Potential Loyalists – Recent Customers – Promising - Customers Needing Attention - At Risk - Cant Lose Them – Hibernating – Lost**

Answer :  
select customerid ,Recency ,Frequency,Monetary , r\_score ,fm\_score,

--naming groups

case

when(r\_score = 5 and (fm\_score =5 or fm\_score >=4))

or(r\_score = 4 and fm\_score =5 )

then 'Champions'

when(r\_score = 4 and (fm\_score =2 or fm\_score =3))

or(r\_score = 5 and fm\_score =2 )

or(r\_score = 3 and fm\_score =3 )

then 'Potential Loyalists'

when(r\_score = 3 and (fm\_score =4 or fm\_score =5))

or(r\_score = 4 and fm\_score =4 )

or(r\_score = 5 and fm\_score =3 )

then 'Loyal Customers'

when(r\_score = 5 and fm\_score =1 ) then 'Recent Customers'

when((r\_score = 4 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 fm\_score =2))then 'Customers Needing Attention'

when((r\_score = 1 or r\_score = 2) and fm\_score >=3 ) then 'At Risk'

when(r\_score = 1 and fm\_score >=4 ) then 'Cant Lose Them'

when(r\_score = 1 and fm\_score =2 ) then 'Hibernating'

when(r\_score = 1 and fm\_score =1 ) then 'Lost'

end

as group\_name

from

(select \*from

(select customerid ,Recency ,Frequency,Monetary,

-- grouping data into 5 group sorted by latest visits

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

-- average of visits & total units

round(((Frequency + Monetary)/2)::numeric,0) as fm\_score

from

(select distinct customerid,

--diff between last date of data and customer last visit to store

(select max(to\_date(invoicedate,'mm/dd/yy')) from online\_retail) - max(to\_date(invoicedate,'mm/dd/yy'))over(partition by customerid) as Recency,

-- how many time customer visit the store

count(\*)over (partition by customerid) as Frequency,

-- sum of units customer buy

round(sum(unitprice)over (partition by customerid)::numeric,2) as Monetary

from online\_retail

where customerid !=''

) as level\_1

order by r\_score desc

)as level\_2) as finall

**Sample of Data**

