Customer Segmentation using SQL (BigQuery)

Problem Statement:

- To perform RFM Analysis for a retail store.
- The stores need to adjust their marketing budget and have better targeting of customers so they need to know which customers to focus on and how important they are for the business.

R: Recency (More points for customer who made purchase recently

F: Frequency (More points for customer who purchase many times

M: Monetary (More points for customer whose purchase value is larger

Customer Segment	Activity	Actionable Tip
Champions	Bought recently, buy often and spend the most!	Reward them. Can be early adopters for new products. Will promote your brand.
Loyal Customers	Spend good money with us often. Responsive to promotions.	Upsell higher value products. Ask for reviews. Engage them.
Potential Loyalist	Recent customers, but spent a good amount and bought more than once.	Offer membership / loyalty program, recommend other products.
Recent Customers	Bought most recently, but not often.	Provide on-boarding support. Give them early success, start building relationship.
Promising	Recent shoppers, but have not spent much.	Create brand awareness, offer free trials
Customers Needing Attention	Above average recency, frequency and monetary values. May not have bought very recently though.	Make limited time offers, Recommend based on past purchases. Reactivate them.
About To Sleep	Below average recency, frequency and monetary values. Will lose them if not reactivated.	Share valuable resources, recommend popular products/ renewals at discount, reconnect with them.
At Risk	Spent big money and purchased often. But long time ago. Need to bring them back!	Send personalized emails to reconnect, offer renewals, provide helpful resources.
Can't Lose Them	Made biggest purchases, and often. But have not returned for a long time.	Win them back via renewals or newer products, do not lose them to competition, talk to them.
Hibernating	Last purchase was long back, low spenders and low number of orders.	Offer other relevant products and special discounts. Recreate brand value.
Lost	Lowest recency, frequency and monetary scores.	Revive interest with reach out campaign, ignore otherwise.

Dataset:

Query:

```
select * FROM `customersegmentation-414317.Sales_Data.customer_Segmentation`
```

Output:

Row	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
1	571035	21238	RED RETROSPOT CUP	8	2011-10-13 12:50:00 UTC	0.85	12446.0	RSA
2	571035	21243	PINK POLKADOT PLATE	8	2011-10-13 12:50:00 UTC	1.69	12446.0	RSA
3	571035	23240	SET OF 4 KNICK KNACK TINS DOILY	6	2011-10-13 12:50:00 UTC	4.15	12446.0	RSA
4	571035	23209	LUNCH BAG VINTAGE DOILY	10	2011-10-13 12:50:00 UTC	1.65	12446.0	RSA
5	571035	23201	JUMBO BAG ALPHABET	10	2011-10-13 12:50:00 UTC	2.08	12446.0	RSA
6	571035	23205	CHARLOTTE BAG VINTAGE ALPHABE...	10	2011-10-13 12:50:00 UTC	0.85	12446.0	RSA
7	571035	21936	RED RETROSPOT PICNIC BAG	5	2011-10-13 12:50:00 UTC	2.95	12446.0	RSA
8	571035	22620	4 TRADITIONAL SPINNING TOPS	12	2011-10-13 12:50:00 UTC	1.45	12446.0	RSA
9	571035	22619	SET OF 6 SOLDIER SKITTLES	4	2011-10-13 12:50:00 UTC	3.75	12446.0	RSA
10	571035	21889	WOODEN BOX OF DOMINOES	12	2011-10-13 12:50:00 UTC	1.25	12446.0	RSA

Results per page: 50 1 – 50 of 375187

Step1: Calculating Bill Amount for each invoice

We have details for all invoices, for our analysis we need for each invoice what is bill amount

Query:

```
with invoiceVsbill as (
SELECT InvoiceNo, sum(Quantity*UnitPrice) as bill
FROM `customersegmentation-414317.Sales_Data.customer_Segmentation`
group by InvoiceNo)
select * from invoiceVsbill order by bill
```

Output:

Row	InvoiceNo	bill
1	568375	0.001
2	570554	0.38
3	567869	0.4
4	542736	0.55
5	540945	0.85
6	538669	0.95
7	538194	0.95
8	539645	0.95

Step2: Getting details for customers

Query1:

```
with Customer_Invoices as (
SELECT CustomerID, InvoiceNo, min(InvoiceDate) as Invoice_date
FROM `customersegmentation-414317.Sales_Data.customer_Segmentation`
group by CustomerID,InvoiceNo)
select * from Customer_Invoices
```

Output1:

Row	CustomerID	InvoiceNo	Invoice_date
1	12446.0	571035	2011-10-13 12:50:00 UTC
2	12558.0	580158	2011-12-02 10:41:00 UTC
3	12646.0	572215	2011-10-21 12:52:00 UTC
4	12646.0	580553	2011-12-05 10:14:00 UTC
5	12607.0	570467	2011-10-10 16:06:00 UTC
6	12733.0	550644	2011-04-19 16:19:00 UTC
7	14016.0	539421	2010-12-17 14:21:00 UTC
8	14016.0	546569	2011-03-15 10:53:00 UTC
9	14016.0	553210	2011-05-16 09:40:00 UTC
10	14016.0	558684	2011-07-01 11:29:00 UTC

Query2: Joining first two tables to get all details of customer and invoices

```
with for_RFM as (
SELECT ci.*,ib.bill FROM `Sales_Data.Customer_Invoices` ci
left join `Sales_Data.invoiceVsbill` ib on ci.InvoiceNo=ib.InvoiceNo)
select * from for_RFM
```

Output2:

Row	CustomerID	InvoiceNo	Invoice_date	bill
1	13824.0	543008	2011-02-02 13:07:00 UTC	442.88
2	13824.0	549735	2011-04-12 08:42:00 UTC	591.3
3	13824.0	554032	2011-05-20 14:04:00 UTC	130.8
4	13824.0	561387	2011-07-27 09:22:00 UTC	196.86
5	13824.0	574868	2011-11-07 12:41:00 UTC	269.1
6	14336.0	546015	2011-03-08 16:48:00 UTC	534.4200000000...
7	14336.0	551859	2011-05-04 15:37:00 UTC	190.23
8	14336.0	554665	2011-05-25 14:44:00 UTC	349.9399999999...
9	14336.0	578187	2011-11-23 11:40:00 UTC	500.5199999999...
10	15360.0	573418	2011-10-31 09:35:00 UTC	427.9299999999...

Step3: Getting Recency, Frequency and Monetary for each customer

Recency is calculated as difference between ref date and purchase date.
Frequency is calculated as no of purchases/month.
Monetary is calculated as sum of all bills for customer.

Query:

```
with cte1 as (
SELECT CustomerID, min(date(Invoice_date)) as First_Purchase,max(date(Invoice_date)) as
Last_Purchase,
count(InvoiceNo) as Total_purchases, sum(bill) as Monetary
FROM `customersegmentation-414317.Sales_Data.for_RFM` group by CustomerID
order by CustomerID),

cte2 as (
select max(date(Invoice_date)) as ref_date from `customersegmentation-
414317.Sales_Data.for_RFM`),

cte3 as (
select * from cte1,cte2),

RFM as (
select CustomerID, date_diff(ref_date,Last_Purchase,day)+1 as Recency,
Total_purchases/(date_diff(Last_Purchase,First_Purchase,month)+1) as Frequency,
Monetary
from cte3 order by CustomerID)

select * from RFM
```

Output:

Row	CustomerID	Recency	Frequency	Monetary
1	12346.0	326	1.0	77183.6
2	12347.0	3	0.538461538461...	4078.95
3	12348.0	76	0.4	1437.24
4	12349.0	19	1.0	1287.15
5	12350.0	311	1.0	294.4000000000...
6	12352.0	37	0.7	1147.439999999...
7	12353.0	205	1.0	29.30000000000...
8	12354.0	233	1.0	925.95
9	12355.0	215	1.0	414.0
10	12356.0	246	0.5	1911.4

Step4: Dividing R, F and M into 5 quantiles

- We need to divide Recency, frequency and Monetary into 5 parts for segmentation.
- Initially these columns are divided into 100 parts (percentile) and then 5 percentiles (20^{th}, 40^{th}, 60^{th}, 80^{th} and 100^{th}) are calculated for segmentation

Query:

```
with quantile as (
select
r[offset(20)] as r20,r[offset(40)] as r40,r[offset(60)] as r60,r[offset(80)] as r80,r[offset(100)]
as r100,
f[offset(20)] as f20,f[offset(40)] as f40,f[offset(60)] as f60,f[offset(80)] as f80,f[offset(100)] as
f100,
m[offset(20)] as m20,m[offset(40)] as m40,m[offset(60)] as m60,m[offset(80)] as
m80,m[offset(100)] as m100
from (
SELECT approx_quantiles(Recency,100) as r,
approx_quantiles(Frequency,100) as f,
approx_quantiles(Monetary,100) as m,
FROM `customersegmentation-414317.Sales_Data.RFM` ))

select * from `Sales_Data.RFM`,quantile
```

Output:

Row	CustomerID	Recency	Frequency	Monetary	r20	r40	r60	r80	r100	f20	f40	f60	f80	f100	m20	m40	m60	m80
1	12748.0	1	15.692...	29589....	4	11	22	74	374	1.3...	1.5	2.0	2.0	34.0	578...	1437...	33...	6...
2	13777.0	1	2.5384...	25964....	4	11	22	74	374	1.3...	1.5	2.0	2.0	34.0	578...	1437...	33...	6...
3	17389.0	1	2.5833...	19917....	4	11	22	74	374	1.3...	1.5	2.0	2.0	34.0	578...	1437...	33...	6...
4	13263.0	2	2.8333...	7031.2...	4	11	22	74	374	1.3...	1.5	2.0	2.0	34.0	578...	1437...	33...	6...
5	13408.0	2	4.6923...	26583....	4	11	22	74	374	1.3...	1.5	2.0	2.0	34.0	578...	1437...	33...	6...
6	14056.0	2	2.0909...	7778.6...	4	11	22	74	374	1.3...	1.5	2.0	2.0	34.0	578...	1437...	33...	6...
7	14606.0	2	6.9230...	9888.0...	4	11	22	74	374	1.3...	1.5	2.0	2.0	34.0	578...	1437...	33...	6...
8	14646.0	2	5.5384...	26619...	4	11	22	74	374	1.3...	1.5	2.0	2.0	34.0	578...	1437...	33...	6...
9	14911.0	2	15.153...	11971...	4	11	22	74	374	1.3...	1.5	2.0	2.0	34.0	578...	1437...	33...	6...
10	15189.0	2	3.4166...	12590....	4	11	22	74	374	1.3...	1.5	2.0	2.0	34.0	578...	1437...	33...	6...

Step5: Calculating RFM score

- Based on quantiles, we need to give RFM score to each customer from 1 to 5.
- Most recent customers (recency percentile less than 20) will be given 5 points.
- Most frequent buyers (frequency percentile between 80-100) will be given 5 points.
- Most expensive buyers (Monetary percentile between 80-100) will be given 5 points.
- F and M scores are combined to reduce large no of segments.

Query:

```
with rfm_score as (
SELECT CustomerID, Recency, Frequency, Monetary,
case when Recency<=r20 then 5
when Recency>r20 and Recency<=r40 then 4
when Recency>r40 and Recency<=r60 then 3
when Recency>r60 and Recency<=r80 then 2
when Recency>r80 and Recency<=r100 then 1 end as RScore,

case when Frequency<=f20 then 1
when Frequency>f20 and Frequency<=f40 then 2
when Frequency>f40 and Frequency<=f60 then 3
when Frequency>f60 and Frequency<=f80 then 4
when Frequency>f80 and Frequency<=f100 then 5 end as FScore,

case when Monetary<=m20 then 1
when Monetary>m20 and Monetary<=m40 then 2
when Monetary>m40 and Monetary<=m60 then 3
when Monetary>m60 and Monetary<=m80 then 4
when Monetary>m80 and Monetary<=m100 then 5 end as MScore
FROM `customersegmentation-414317.Sales_Data.Quantile`)

select CustomerID, Recency, Frequency, Monetary,
RScore, round((FScore+MScore)/2) as FM_score from rfm_score
```

Output:

Row	CustomerID	Recency	Frequency	Monetary	RScore	FM_score
1	12748.0	1	15.692307692307692	29589.39	5	5.0
2	13777.0	1	2.5384615384615383	25964.409999999996	5	5.0
3	17389.0	1	2.5833333333333335	19917.48	5	5.0
4	13263.0	2	2.8333333333333335	7031.2699999999995	5	5.0
5	13408.0	2	4.6923076923076925	26583.039999999997	5	5.0
6	14056.0	2	2.0909090909090908	7778.6500000000024	5	5.0
7	14606.0	2	6.9230769230769234	9888.0000000000018	5	5.0
8	14646.0	2	5.5384615384615383	266194.22	5	5.0
9	14911.0	2	15.153846153846153	119715.28999999995	5	5.0
10	15189.0	2	3.4166666666666665	12590.640000000003	5	5.0

Step5: Segmentation based on RFM scores

- Based on RFM score, customers are divided into 11 segments as follows

Sr No	Segments
1	Champions
2	Loyal Customers
3	Potential Loyalists
4	Recent Customers
5	Promising
6	Customers Needing Attention
7	About to Sleep
8	At Risk
9	Cant Lose Them
10	Hibernating
11	Lost

- Based on score these segments will be assigned as follows

		FM				
		1	2	3	4	5
R	1	Lost	Hibernating	At Risk	Cant Lose Them	Cant Lose Them
	2	About to Sleep	Customers Needing Attention	Customers Needing Attention	At Risk	At Risk
	3	Promising	Customers Needing Attention	Potential Loyalists	Loyal Customers	Loyal Customers
	4	Promising	Potential Loyalists	Potential Loyalists	Loyal Customers	Champions
	5	Recent Customers	Potential Loyalists	Loyal Customers	Champions	Champions

Query:

```
SELECT
    CustomerID,
    Recency, Frequency, Monetary,
    RScore, FM_score,
    CASE
    WHEN (RScore = 5 AND FM_score = 5)
      OR (RScore = 5 AND FM_score = 4)
      OR (RScore = 4 AND FM_score = 5)
    THEN 'Champions'
    WHEN (RScore = 5 AND FM_score =3)
      OR (RScore = 4 AND FM_score = 4)
      OR (RScore = 3 AND FM_score = 5)
      OR (RScore = 3 AND FM_score = 4)
    THEN 'Loyal Customers'
    WHEN (RScore = 5 AND FM_score = 2)
      OR (RScore = 4 AND FM_score = 2)
      OR (RScore = 3 AND FM_score = 3)
      OR (RScore = 4 AND FM_score = 3)
    THEN 'Potential Loyalists'
    WHEN RScore = 5 AND FM_score = 1 THEN 'Recent Customers'
    WHEN (RScore = 4 AND FM_score = 1)
      OR (RScore = 3 AND FM_score = 1)
    THEN 'Promising'
    WHEN (RScore = 3 AND FM_score = 2)
      OR (RScore = 2 AND FM_score = 3)
      OR (RScore = 2 AND FM_score = 2)
    THEN 'Customers Needing Attention'
    WHEN RScore = 2 AND FM_score = 1 THEN 'About to Sleep'
    WHEN (RScore = 2 AND FM_score = 5)
      OR (RScore = 2 AND FM_score = 4)
      OR (RScore = 1 AND FM_score = 3)
    THEN 'At Risk'
    WHEN (RScore = 1 AND FM_score = 5)
      OR (RScore = 1 AND FM_score = 4)
    THEN 'Cant Lose Them'
    WHEN RScore = 1 AND FM_score = 2 THEN 'Hibernating'
    WHEN RScore = 1 AND FM_score = 1 THEN 'Lost'
    END AS rfm_segment
FROM `customersegmentation-414317.Sales_Data.rfm_score_updated`
```

Output:

Row	CustomerID	Recency	Frequency	Monetary	RScore	FM_score	rfm_segment
107	16745.0	87	2.125	7194.29999...	1	5.0	Cant Lose Them
108	17850.0	373	34.0	4783.45999...	1	5.0	Cant Lose Them
109	18073.0	115	3.0	3817.52000...	1	5.0	Cant Lose Them
110	13066.0	25	2.0	388.65	2	2.0	Customers Needing Attention
111	17357.0	25	2.0	351.289999...	2	2.0	Customers Needing Attention
112	12508.0	27	2.0	398.27	2	2.0	Customers Needing Attention
113	13727.0	29	2.0	333.48	2	2.0	Customers Needing Attention
114	16372.0	34	2.0	316.379999...	2	2.0	Customers Needing Attention
115	17608.0	34	2.0	193.420000...	2	2.0	Customers Needing Attention
116	17078.0	37	2.0	378.200000...	2	2.0	Customers Needing Attention