Segmenting Customers with Google BigQuery

Problem Statement:

We will be performing a RFM analysis for a chain of retail stores that sells a lot of different items and categories.

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.

What is RFM Score?

We all know that valuing customers based on a single parameter is flawed. The biggest value customer may have only purchased once or twice in a year, or the most frequent purchaser may have a value so low that it is almost not profitable to service them.

One parameter will never give you an accurate view of your customer base, and you'll ignore customer lifetime value.

We calculate the RFM score by attributing a numerical value for each of the criteria. The customer gets more points -

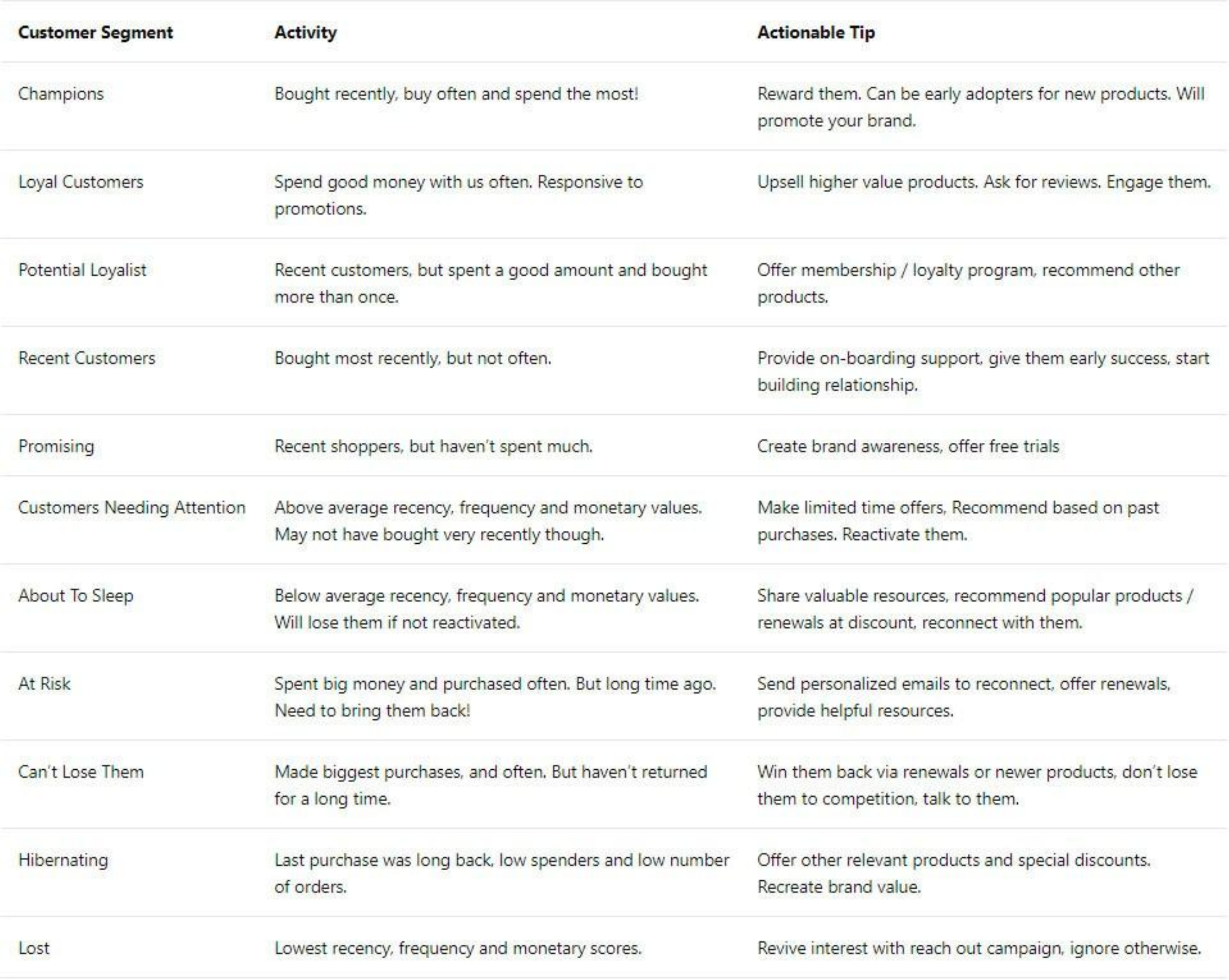
- if they bought in the recent past,
- bought many times or
- if the purchase value is larger.

Combine these three values to create the RFM score.

This RFM score can then be used to segment your customer data platform (CDP).

- Ultimately, we will end up with 5 bands for each of the R, F and M-values, this can be reduced to bands of 3 if the variation of your data values is narrow.
- The larger the score for each value the better it is. A final RFM score is calculated simply by combining individual RFM score numbers.
- There are many different permutations of the R,F & M scores, 125 in total, which is too many to deal with on an individual basis and many will require similar marketing responses.

Analysis of the customer RFM values will create some standard segments.
The **UK Data & Marketing Association (DMA)** laid out 11 segments, and specified marketing strategies according to their respective characteristics:



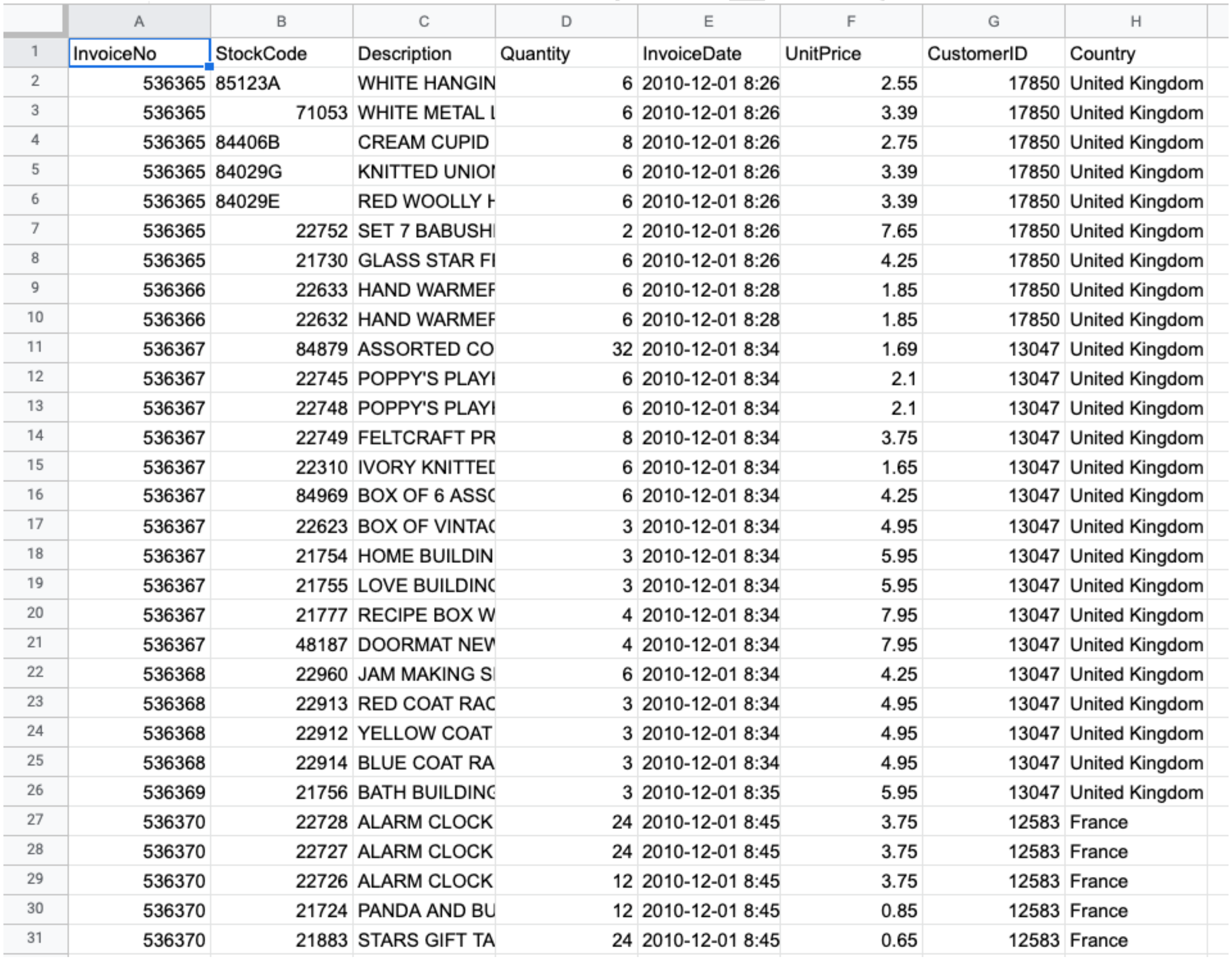
Customer Segment	Activity	Actionable Tip
Champions	Bought recently, buy often and spend the most!	Reward them. Can be early adopters for new products. Will promote your brand.
Loyal Customers	Spend good money with us often. Responsive to promotions.	Upsell higher value products. Ask for reviews. Engage 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 haven't 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 haven't returned for a long time.	Win them back via renewals or newer products, don't 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.

Think about what percentage of our existing customers would be in each of these segments and evaluate how effective the recommended marketing action can be for your business.

Data

Attribute Information:

- **InvoiceNo:** Invoice number. Nominal, a 6-digit integral number uniquely assigned to each transaction. If this code starts with letter 'c', it indicates a cancellation.
- **StockCode:** Product (item) code. Nominal, a 5-digit integral number uniquely assigned to each distinct product.
- **Description:** Product (item) name. Nominal.
- **Quantity:** The quantities of each product (item) per transaction. Numeric.
- **InvoiceDate:** Invoice Date and time. Numeric, the day and time when each transaction was generated.
- **UnitPrice:** Unit price. Numeric, Product price per unit in sterling.
- **CustomerID:** Customer number. Nominal, a 5-digit integral number uniquely assigned to each customer.
- **Country:** Country name. Nominal, the name of the country where each customer resides.



	A	B	C	D	E	F	G	H
1	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
2	536365	85123A	WHITE HANGIN	6	2010-12-01 8:26	2.55	17850	United Kingdom
3	536365	71053	WHITE METAL L	6	2010-12-01 8:26	3.39	17850	United Kingdom
4	536365	84406B	CREAM CUPID	8	2010-12-01 8:26	2.75	17850	United Kingdom
5	536365	84029G	KNITTED UNIOI	6	2010-12-01 8:26	3.39	17850	United Kingdom
6	536365	84029E	RED WOOLLY H	6	2010-12-01 8:26	3.39	17850	United Kingdom
7	536365	22752	SET 7 BABUSHI	2	2010-12-01 8:26	7.65	17850	United Kingdom
8	536365	21730	GLASS STAR FI	6	2010-12-01 8:26	4.25	17850	United Kingdom
9	536366	22633	HAND WARMEF	6	2010-12-01 8:28	1.85	17850	United Kingdom
10	536366	22632	HAND WARMEF	6	2010-12-01 8:28	1.85	17850	United Kingdom
11	536367	84879	ASSORTED CO	32	2010-12-01 8:34	1.69	13047	United Kingdom
12	536367	22745	POPPY'S PLAYI	6	2010-12-01 8:34	2.1	13047	United Kingdom
13	536367	22748	POPPY'S PLAYI	6	2010-12-01 8:34	2.1	13047	United Kingdom
14	536367	22749	FELTCRAFT PR	8	2010-12-01 8:34	3.75	13047	United Kingdom
15	536367	22310	IVORY KNITTEL	6	2010-12-01 8:34	1.65	13047	United Kingdom
16	536367	84969	BOX OF 6 ASSC	6	2010-12-01 8:34	4.25	13047	United Kingdom
17	536367	22623	BOX OF VINTAC	3	2010-12-01 8:34	4.95	13047	United Kingdom
18	536367	21754	HOME BUILDIN	3	2010-12-01 8:34	5.95	13047	United Kingdom
19	536367	21755	LOVE BUILDINC	3	2010-12-01 8:34	5.95	13047	United Kingdom
20	536367	21777	RECIPE BOX W	4	2010-12-01 8:34	7.95	13047	United Kingdom
21	536367	48187	DOORMAT NEV	4	2010-12-01 8:34	7.95	13047	United Kingdom
22	536368	22960	JAM MAKING SI	6	2010-12-01 8:34	4.25	13047	United Kingdom
23	536368	22913	RED COAT RAC	3	2010-12-01 8:34	4.95	13047	United Kingdom
24	536368	22912	YELLOW COAT	3	2010-12-01 8:34	4.95	13047	United Kingdom
25	536368	22914	BLUE COAT RA	3	2010-12-01 8:34	4.95	13047	United Kingdom
26	536369	21756	BATH BUILDINC	3	2010-12-01 8:35	5.95	13047	United Kingdom
27	536370	22728	ALARM CLOCK	24	2010-12-01 8:45	3.75	12583	France
28	536370	22727	ALARM CLOCK	24	2010-12-01 8:45	3.75	12583	France
29	536370	22726	ALARM CLOCK	12	2010-12-01 8:45	3.75	12583	France
30	536370	21724	PANDA AND BU	12	2010-12-01 8:45	0.85	12583	France
31	536370	21883	STARS GIFT TA	24	2010-12-01 8:45	0.65	12583	France

RFM Segmentation in BigQuery

The RFM Segmentation can be executed using these five steps:

1. Data processing
2. Compute for recency, frequency, and monetary values per customer
3. Determine quantiles for each RFM metric
4. Assign scores for each RFM metric
5. Define the RFM segments using the scores in step 4

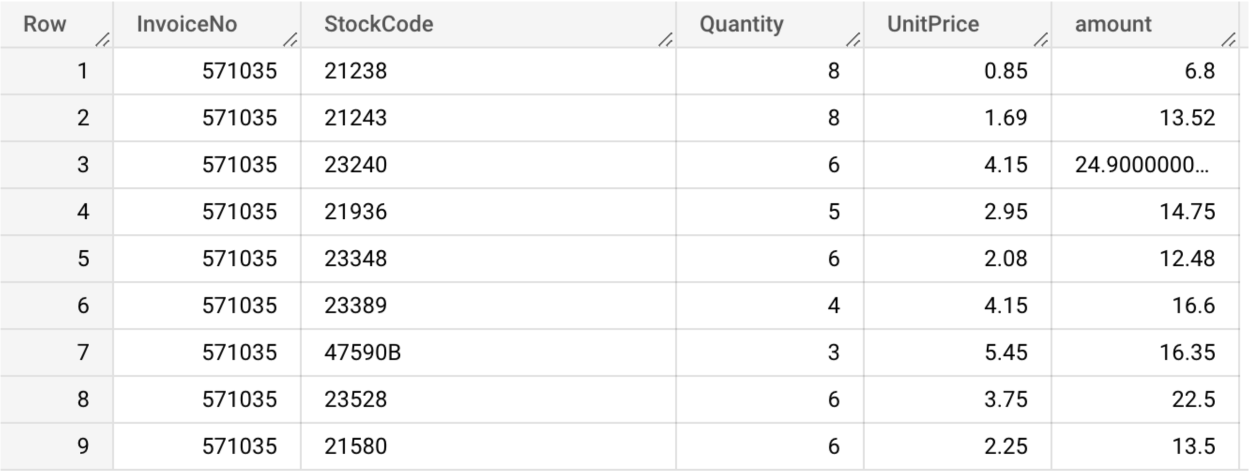
Data Processing

Adding the data to BigQuery:
Create a new dataset and upload ‘sales.csv’ as a new table.
We created a dataset named `retail` in a project customer segmentation and the table name is `sales`.

Now if we look at the data we can see that there are products that have been bought in quantities more than one and we have unit price for those products but we do not have the total cost of that product.

So the first thing we’re gonna do is find the total cost for that product i.e., quantity * unit price -

```
SELECT
  InvoiceNo,StockCode,Quantity,UnitPrice,
  (Quantity*UnitPrice) AS amount
FROM
  `customer-segmentation-373712.retail.sales`
```



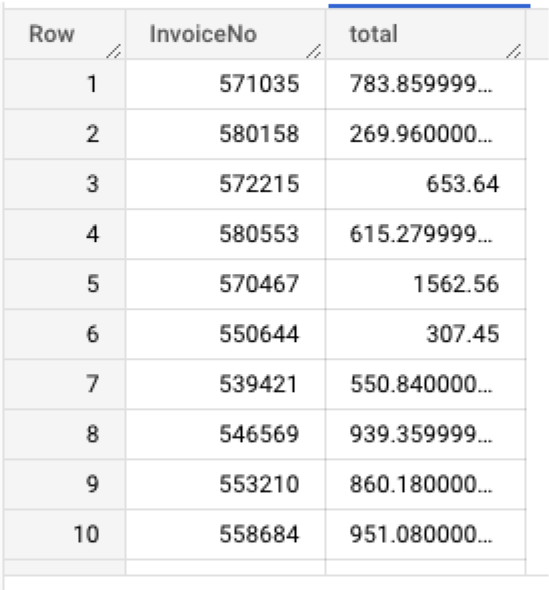
Row	InvoiceNo	StockCode	Quantity	UnitPrice	amount
1	571035	21238	8	0.85	6.8
2	571035	21243	8	1.69	13.52
3	571035	23240	6	4.15	24.9000000...
4	571035	21936	5	2.95	14.75
5	571035	23348	6	2.08	12.48
6	571035	23389	4	4.15	16.6
7	571035	47590B	3	5.45	16.35
8	571035	23528	6	3.75	22.5
9	571035	21580	6	2.25	13.5

Now that we have got the total cost for each product we need to find out the amount spent on each visit.

For each invoice id there may be different products, and till now we have calculated the total for each product, but we do not have the total bill amount for individual invoice ids.

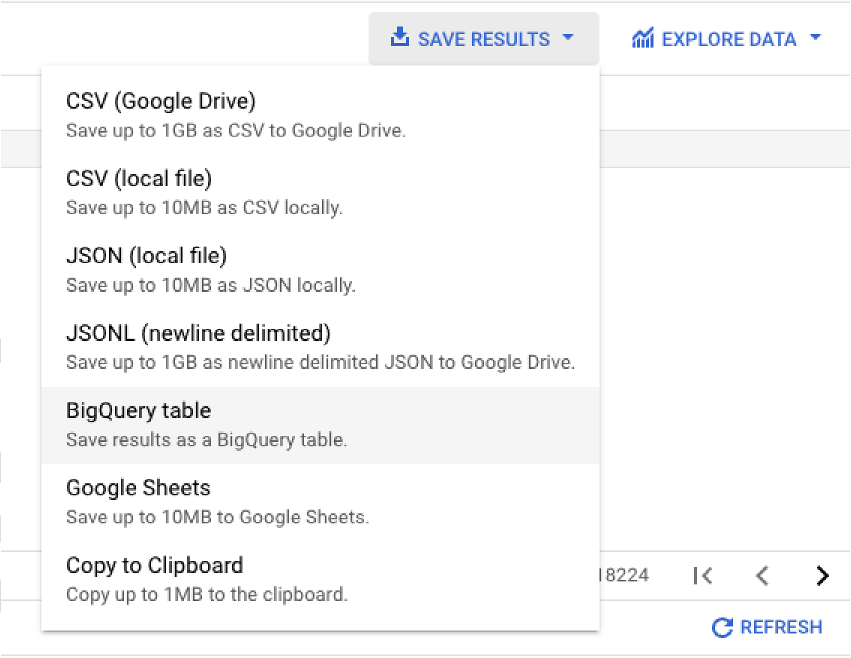
For this we use the above query and create a CTE. Then group it by invoice id and sum the total cost, getting the actual bill amount.

```
WITH
 bills AS (
 SELECT
   InvoiceNo,
   (Quantity*UnitPrice) AS amount
 FROM
   `customer-segmentation-373712.retail.sales` )
SELECT
 InvoiceNo,
 SUM(amount) AS total
FROM
 bills
GROUP BY
 InvoiceNo
```



Row	InvoiceNo	total
1	571035	783.859999...
2	580158	269.960000...
3	572215	653.64
4	580553	615.279999...
5	570467	1562.56
6	550644	307.45
7	539421	550.840000...
8	546569	939.359999...
9	553210	860.180000...
10	558684	951.080000...

Save this data as a **`bill`** table in the same dataset by using the save button below the query editor.
Note: we can do this without saving this a table but that will make the query pretty long


SAVE RESULTS
EXPLORE DATA
CSV (Google Drive)
Save up to 1GB as CSV to Google Drive.
CSV (local file)
Save up to 10MB as CSV locally.
JSON (local file)
Save up to 10MB as JSON locally.
JSONL (newline delimited)
Save up to 1GB as newline delimited JSON to Google Drive.
BigQuery table
Save results as a BigQuery table.
Google Sheets
Save up to 10MB to Google Sheets.
Copy to Clipboard
Copy up to 1MB to the clipboard.
18224
REFRESH

Compute for recency, frequency and monetary values per customer :

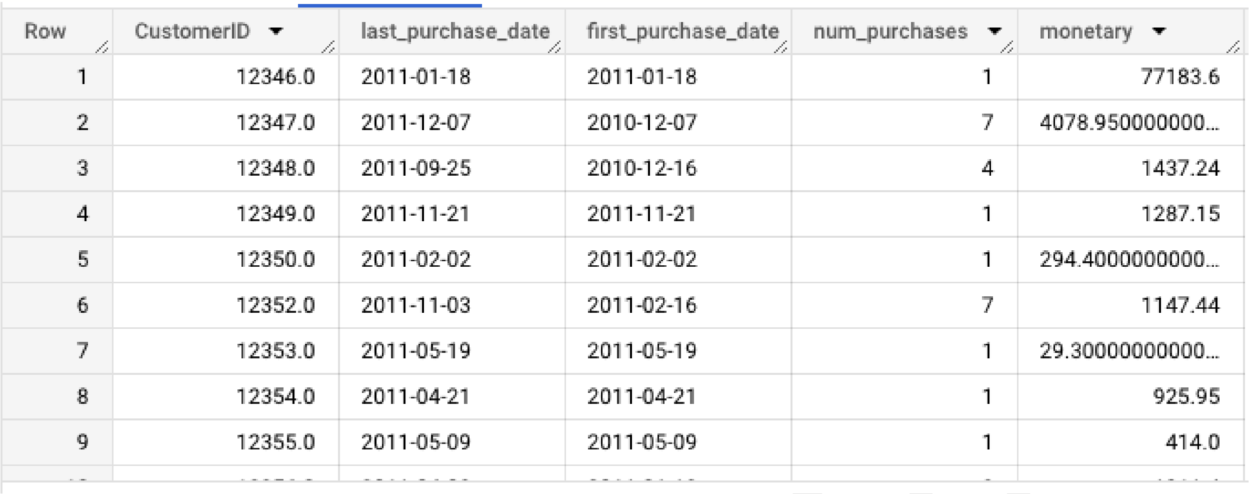
Because we will be joining the bill and sales table we will get multiple rows on the key that we are going to join ie: InvoiceNo, so we'll just take one row per InvoiceNo.
For that we will use the row number and get the data from the sales table that we need.

Now for calculating RFM

- For monetary, this is just a simple *sum of sales*,
- while for frequency, this is a *count of distinct invoice numbers per customer for the time they have been a customer ie: the number of separate purchases/ num of months they have been a customer*. So we will get the first and last purchase for all customers and also the number of purchases
- For calculating recency we will first get the last purchase for each customer

We will join the `bill` table that we saved with the `sales` table and add the total cost on the customer level for monetary value.

```
SELECT
CustomerID,
DATE(MAX(InvoiceDate)) AS last_purchase_date,
DATE(MIN(InvoiceDate)) AS first_purchase_date,
COUNT(DISTINCT InvoiceNo) AS num_purchases,
SUM(total) AS monetary,
FROM(
       Select s.CustomerID, s.InvoiceDate, s.InvoiceNo, b.total
           ,ROW_NUMBER() OVER(PARTITION BY s.InvoiceNo ORDER BY s.InvoiceNo) AS RN
       From
       `customer-segmentation-373712.retail.sales` s
LEFT JOIN
`customer-segmentation-373712.retail.bill` b
ON
s.InvoiceNo=b.InvoiceNo
       ) A
WHERE A.RN = 1
GROUP BY CustomerID
```



Row	CustomerID	last_purchase_date	first_purchase_date	num_purchases	monetary
1	12346.0	2011-01-18	2011-01-18	1	77183.6
2	12347.0	2011-12-07	2010-12-07	7	4078.950000000...
3	12348.0	2011-09-25	2010-12-16	4	1437.24
4	12349.0	2011-11-21	2011-11-21	1	1287.15
5	12350.0	2011-02-02	2011-02-02	1	294.4000000000...
6	12352.0	2011-11-03	2011-02-16	7	1147.44
7	12353.0	2011-05-19	2011-05-19	1	29.30000000000...
8	12354.0	2011-04-21	2011-04-21	1	925.95
9	12355.0	2011-05-09	2011-05-09	1	414.0

We can save this table as monetary

Recency

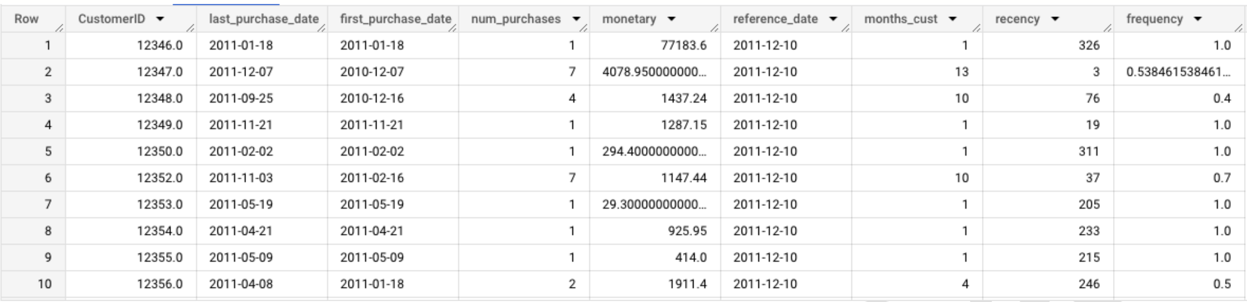
For recency, we chose a reference date, which is the *most recent purchase* in the dataset. In other situations, one may select the date when the data was analyzed instead.

After choosing the reference date, we get the date *difference between the reference date and the last purchase date of each customer*. This is the recency value for that particular customer.

For frequency we calculate the months the person has been a customer by difference in first and last purchase +1 (for when first and last month are same and the customer should be considered a customer for at least 1 month)

```
SELECT
*,
DATE_DIFF(reference_date, last_purchase_date, DAY) AS recency,
num_purchases/ (months_cust) AS frequency,
FROM
(
SELECT *,
MAX(last_purchase_date) OVER () + 1 AS reference_date,
DATE_DIFF(last_purchase_date, first_purchase_date, month)+1 AS months_cust
FROM `customer-segmentation-373712.retail.monetary` )
```

```
ORDER BY CustomerID ;
```



Row	CustomerID	last_purchase_date	first_purchase_date	num_purchases	monetary	reference_date	months_cust	recency	frequency
1	12346.0	2011-01-18	2011-01-18	1	77183.6	2011-12-10	1	326	1.0
2	12347.0	2011-12-07	2010-12-07	7	4078.950000000...	2011-12-10	13	3	0.538461538461...
3	12348.0	2011-09-25	2010-12-16	4	1437.24	2011-12-10	10	76	0.4
4	12349.0	2011-11-21	2011-11-21	1	1287.15	2011-12-10	1	19	1.0
5	12350.0	2011-02-02	2011-02-02	1	294.4000000000...	2011-12-10	1	311	1.0
6	12352.0	2011-11-03	2011-02-16	7	1147.44	2011-12-10	10	37	0.7
7	12353.0	2011-05-19	2011-05-19	1	29.30000000000...	2011-12-10	1	205	1.0
8	12354.0	2011-04-21	2011-04-21	1	925.95	2011-12-10	1	233	1.0
9	12355.0	2011-05-09	2011-05-09	1	414.0	2011-12-10	1	215	1.0
10	12356.0	2011-04-08	2011-01-18	2	1911.4	2011-12-10	4	246	0.5

Now that we have the RFM data we can save it as another table named `RFM`.

Determine quintiles for each RFM metric

The next step would be to group the customers into quintiles in terms of their RFM values — we divide the customers into 5 equal groups, according to how high and low they scored in the RFM metrics.

The main advantage of using percentile is we do not have to change or set the values. It will be automatically calculated.

What is a Quintile?

- A quintile is a 1/5th (20 percent) portion of the whole. In statistics, it’s a population or sample divided into five equal groups, according to values of a particular variable. Quintiles are like percentiles, but instead of dividing the data into 100 parts, you divide it in 5 equal parts. Quintiles work with any industry since the data itself defines the ranges; they distribute customers evenly.

We do this for each of recency, frequency and monetary values per customer.
I used BigQuery’s **APPROX_QUANTILES()** to achieve this.

How does APPROX_QUANTILES() work?

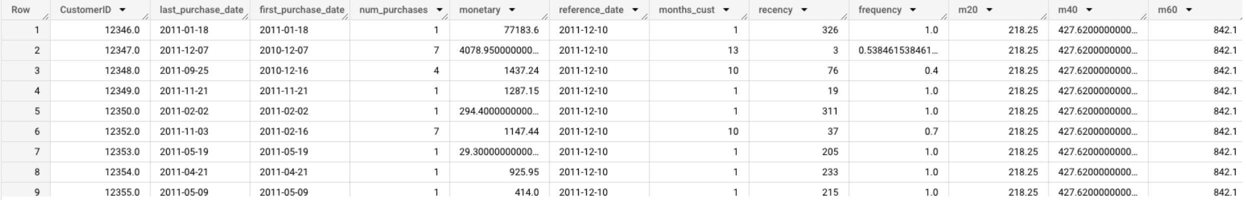
- Returns the approximate boundaries for a group of expression values, where number represents the number of quantiles to create.
- This function returns an array of number+1 elements, where the first element is the approximate minimum and the last element is the approximate maximum.

NOTE : Approximate aggregate functions are scalable in terms of memory usage and time, but produce approximate results instead of exact results.

- OFFSET() accesses an ARRAY element by position and returns the element. The approximate_quantiles will return an array for each percentile and for creating quintiles out of it we will need values at 20, 40 and so on. We save those values as m20, m40 for monetary and f, r for frequency and recency respectively.

```
SELECT
   a.*,
   --All percentiles for MONETARY
   b.percentiles[offset(20)] AS m20,
   b.percentiles[offset(40)] AS m40,
   b.percentiles[offset(60)] AS m60,
   b.percentiles[offset(80)] AS m80,
   b.percentiles[offset(100)] AS m100,
   --All percentiles for FREQUENCY
   c.percentiles[offset(20)] AS f20,
   c.percentiles[offset(40)] AS f40,
   c.percentiles[offset(60)] AS f60,
   c.percentiles[offset(80)] AS f80,
   c.percentiles[offset(100)] AS f100,
   --All percentiles for RECENCY
   d.percentiles[offset(20)] AS r20,
   d.percentiles[offset(40)] AS r40,
   d.percentiles[offset(60)] AS r60,
   d.percentiles[offset(80)] AS r80,
   d.percentiles[offset(100)] AS r100
FROM
```

```
   `customer-segmentation-373712.retail.RFM` a,
   (SELECT APPROX_QUANTILES(monetary, 100) percentiles FROM
   `customer-segmentation-373712.retail.RFM`) b,
   (SELECT APPROX_QUANTILES(frequency, 100) percentiles FROM
   `customer-segmentation-373712.retail.RFM`) c,
   (SELECT APPROX_QUANTILES(recency, 100) percentiles FROM
   `customer-segmentation-373712.retail.RFM`) d
ORDER BY CustomerID
```



Row	CustomerID	last_purchase_date	first_purchase_date	num_purchases	monetary	reference_date	months_cust	recency	frequency	m20	m40	m60
1	12346.0	2011-01-18	2011-01-18	1	77183.6	2011-12-10	1	326	1.0	218.25	427.6200000000...	842.1
2	12347.0	2011-12-07	2010-12-07	7	4078.950000000...	2011-12-10	13	3	0.538461538461...	218.25	427.6200000000...	842.1
3	12348.0	2011-09-25	2010-12-16	4	1437.24	2011-12-10	10	76	0.4	218.25	427.6200000000...	842.1
4	12349.0	2011-11-21	2011-11-21	1	1287.15	2011-12-10	1	19	1.0	218.25	427.6200000000...	842.1
5	12350.0	2011-02-02	2011-02-02	1	294.4000000000...	2011-12-10	1	311	1.0	218.25	427.6200000000...	842.1
6	12352.0	2011-11-03	2011-02-16	7	1147.44	2011-12-10	10	37	0.7	218.25	427.6200000000...	842.1
7	12353.0	2011-05-19	2011-05-19	1	29.30000000000...	2011-12-10	1	205	1.0	218.25	427.6200000000...	842.1
8	12354.0	2011-04-21	2011-04-21	1	925.95	2011-12-10	1	233	1.0	218.25	427.6200000000...	842.1
9	12355.0	2011-05-09	2011-05-09	1	414.0	2011-12-10	1	215	1.0	218.25	427.6200000000...	842.1

Again, we save these as a new table named `quantile`.

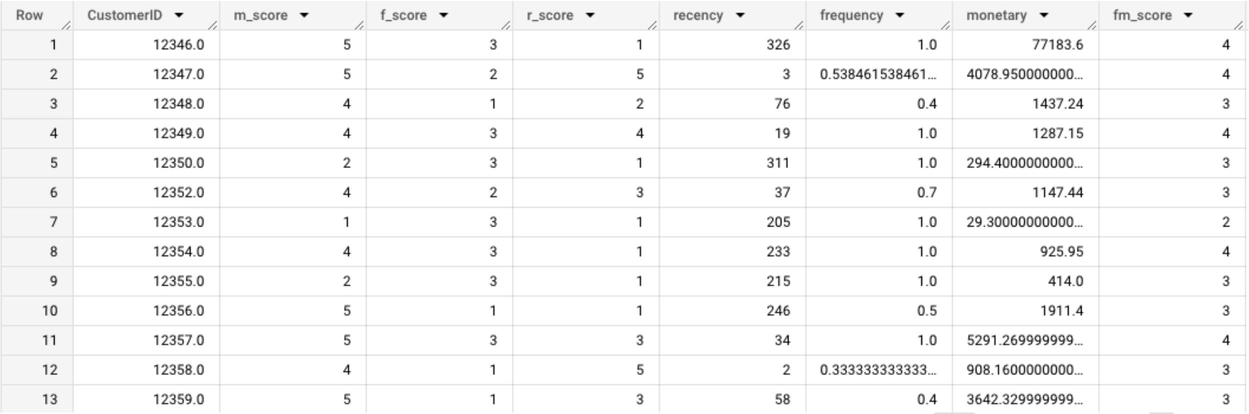
Assign scores for each RFM metric :

Now that we know how each customer fares relative to other customers in terms of RFM values, we can now assign scores from 1 to 5.

Just keep in mind that while with F and M, we give higher scores for higher quintiles, R should be reversed as more recent customers should be scored higher in this metric. Frequency and Monetary value are combined (as both of them are indicative to purchase volume anyway) to reduce the possible options from 125 to 50.

We will use CASE to get values and assign scores accordingly, so we just get the data from the `quintiles` table that we stored assign scores.

```
SELECT CustomerID,
m_score,f_score,r_score,
recency, frequency,monetary,
 CAST(ROUND((f_score + m_score) / 2, 0) AS INT64) AS fm_score
 FROM (
     SELECT *,
     CASE WHEN monetary <= m20 THEN 1
         WHEN monetary <= m40 AND monetary > m20 THEN 2
         WHEN monetary <= m60 AND monetary > m40 THEN 3
         WHEN monetary <= m80 AND monetary > m60 THEN 4
         WHEN monetary <= m100 AND monetary > m80 THEN 5
     END AS m_score,
     CASE WHEN frequency <= f20 THEN 1
         WHEN frequency <= f40 AND frequency > f20 THEN 2
         WHEN frequency <= f60 AND frequency > f40 THEN 3
         WHEN frequency <= f80 AND frequency > f60 THEN 4
         WHEN frequency <= f100 AND frequency > f80 THEN 5
     END AS f_score,
     --Recency scoring is reversed
     CASE WHEN recency <= r20 THEN 5
         WHEN recency <= r40 AND recency > r20 THEN 4
         WHEN recency <= r60 AND recency > r40 THEN 3
         WHEN recency <= r80 AND recency > r60 THEN 2
         WHEN recency <= r100 AND recency > r80 THEN 1
     END AS r_score,
     FROM `customer-segmentation-373712.retail.Quintiles`
     )
```



Row	CustomerID	m_score	f_score	r_score	recency	frequency	monetary	fm_score
1	12346.0	5	3	1	326	1.0	77183.6	4
2	12347.0	5	2	5	3	0.538461538461...	4078.950000000...	4
3	12348.0	4	1	2	76	0.4	1437.24	3
4	12349.0	4	3	4	19	1.0	1287.15	4
5	12350.0	2	3	1	311	1.0	294.4000000000...	3
6	12352.0	4	2	3	37	0.7	1147.44	3
7	12353.0	1	3	1	205	1.0	29.30000000000...	2
8	12354.0	4	3	1	233	1.0	925.95	4
9	12355.0	2	3	1	215	1.0	414.0	3
10	12356.0	5	1	1	246	0.5	1911.4	3
11	12357.0	5	3	3	34	1.0	5291.269999999...	4
12	12358.0	4	1	5	2	0.333333333333...	908.1600000000...	3
13	12359.0	5	1	3	58	0.4	3642.329999999...	3

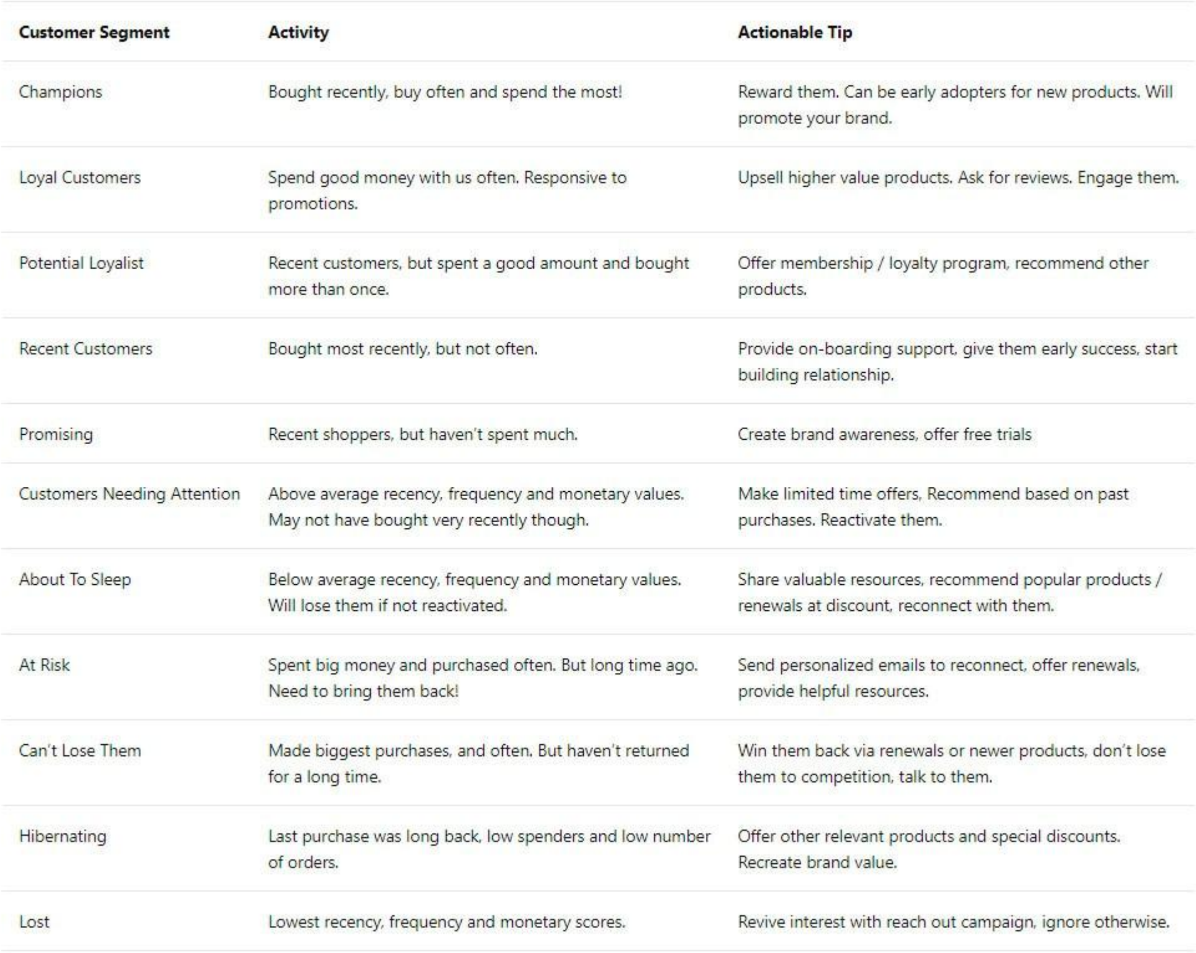
Now you can save this as another table or create a CTE named score for this and use it for further calculations.

Define the RFM segments using these scores :

The next step is to combine the scores we obtained to define the RFM segment each customer will belong to.

As there are 5 groups for each of the R, F, and M metrics, there are 125 potential permutations.

We will be using the 11 personas in the DMA as a guide and define the R vs. FM scores accordingly.



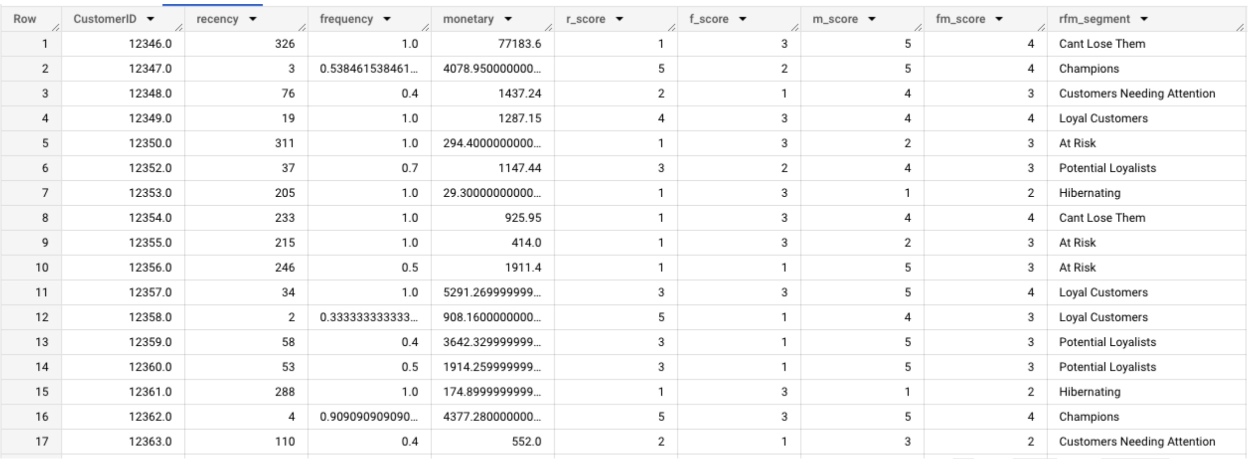
Customer Segment	Activity	Actionable Tip
Champions	Bought recently, buy often and spend the most!	Reward them. Can be early adopters for new products. Will promote your brand.
Loyal Customers	Spend good money with us often. Responsive to promotions.	Upsell higher value products. Ask for reviews. Engage 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 haven't 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 haven't returned for a long time.	Win them back via renewals or newer products, don't 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.

- For example, in the **Champions** segment, customers should have bought recently, bought often, and spent the most. Therefore, their R score should be 5 and their combined FM score should be 4 or 5.

- On the other hand, **Can't Lose Them** customers made the biggest purchases, and often, but haven't returned for a long time. Hence their R score should be 1, and FM score should be 4 or 5.

```
SELECT
      CustomerID,
      recency,frequency,monetary,
      r_score, f_score, m_score,
      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 = 5)
    THEN 'Champions'
    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 = 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 = 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 = 1 THEN 'About to Sleep'
    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 = 2 THEN 'Hibernating'
    WHEN r_score = 1 AND fm_score = 1 THEN 'Lost'
    END AS rfm_segment
FROM `customer-segmentation-373712.retail.score`
ORDER BY CustomerID
```



Row	CustomerID	recency	frequency	monetary	r_score	f_score	m_score	fm_score	rfm_segment
1	12346.0	326	1.0	77183.6	1	3	5	4	Cant Lose Them
2	12347.0	3	0.538461538461...	4078.950000000...	5	2	5	4	Champions
3	12348.0	76	0.4	1437.24	2	1	4	3	Customers Needing Attention
4	12349.0	19	1.0	1287.15	4	3	4	4	Loyal Customers
5	12350.0	311	1.0	294.4000000000...	1	3	2	3	At Risk
6	12352.0	37	0.7	1147.44	3	2	4	3	Potential Loyalists
7	12353.0	205	1.0	29.30000000000...	1	3	1	2	Hibernating
8	12354.0	233	1.0	925.95	1	3	4	4	Cant Lose Them
9	12355.0	215	1.0	414.0	1	3	2	3	At Risk
10	12356.0	246	0.5	1911.4	1	1	5	3	At Risk
11	12357.0	34	1.0	5291.269999999...	3	3	5	4	Loyal Customers
12	12358.0	2	0.333333333333...	908.1600000000...	5	1	4	3	Loyal Customers
13	12359.0	58	0.4	3642.329999999...	3	1	5	3	Potential Loyalists
14	12360.0	53	0.5	1914.259999999...	3	1	5	3	Potential Loyalists
15	12361.0	288	1.0	174.8999999999...	1	3	1	2	Hibernating
16	12362.0	4	0.909090909090...	4377.280000000...	5	3	5	4	Champions
17	12363.0	110	0.4	552.0	2	1	3	2	Customers Needing Attention

After this step, each customer should have an RFM segment assignment like this.

This type of segmentation focuses on the actual buying behavior and ignores the differences in motivations, intentions, and lifestyles of consumers.
RFM is nonetheless a useful start-off point, and because of its simplicity can be executed fast and in an automated way, giving companies the power to act and decide on business strategies swiftly.