2️⃣

RFM customer segmentation

Using the logical tree to find the dimensions for customer classification:

Customer classification	Demographic	Gender
		Age
		...
	Geographical location	
	Marital status	Single
		Married
		...
	Income	High
		Low
	Behavior	Recency
		Frequency
		Monetary

There are three factors to consider while assessing RFM, the technique for categorizing clients based on their behavior.

R is the recency, based on whether we can find out when their last purchase was, they still bought it recently (active).

R *(days) = now - last_purchase_date*

F is Frequency, this is an indicator that measures the frequency of customers in a certain period of time. This can know which customers come to your place often but which customers don't.

F = total number of orders

M is Monetary, based on the amount of money they have spent at the point of sale, it is possible to classify which customers spend more and which customers spend less.

M = total money spent

Context

We have the following table:

Recency	**F**requency	**M**onetary
Lvl 1 (leaving)	Lvl 1 (rare)	Lvl 1 (small amount)
Lvl 2	Lvl 2 (normal)	Lvl 2
Lvl 3 (active)	Lvl 3 (often)	Lvl 3 (big amount)

⇒At the simplest level we will divide into 3 groups below:

- Low value customer group: includes customers who have not come back to buy for a long time, the number of orders is small and the total order value is low. equivalent to R, F and M being at level 1.
- The group of customers with average value (Mid Value): is a group of customers with 3 average RFM indicators, neither high nor low R, F and M are at level 2 or may also have 1 of 3 criteria in level 3.
- High Value customer group: is a group of customers with recent purchases, large number of orders and high total value. all 3 RFM criteria are at level 3.

For more detail we can segment by the following table:

Customer Segment	R	F	M	Description
VIP customer	3	3	3	Purchasing regularly, most recently, and spend the most
Loyal customer	3	3	2	Customers a lot and frequently purchase.
	3	2	2	
	2	3	2	
	2	2	2	
	3	3	1	
	2	3	1	
Potential Prospective customer	3	2	3	Current purchasing, having a few times, and spending a lot
Risky customer	1	3	3	Spent big and often but haven't been back in a while
	1	2	3	
	1	3	2	
	1	3	1	
	2	2	3	
	2	3	3	
High-spending customers	1	1	3	Customers spend a lot but not frequently
	2	1	3	
	2	1	2	
	3	1	3	
New customer	3	2	1	customers buy recently but not often before
	3	1	2	
	3	1	1	
	2	1	1	
Low value customer	1	1	2	Low in three-dimension RFM
	1	2	2	
	1	2	1	
	2	2	1	
	1	1	1	

So how are the levels classified?

In a straightforward manner that is divided into three equal pieces, we may classify. Here, we categorize using **Pareto** in a more useful method.

> *With a to-do list implies using the **Pareto** principle. Select the two pieces of material that will add the most value. The majority of people, nevertheless, really opt to complete the simpler jobs first. Thus, the remaining 20% of the things that require attention are*

> *frequently postponed in around 80% of the less essential items. It requires extra time and effort because of this.*

Level 3 is the highest level in the dimensions; it will occupy the first 20% of space, while levels 2 and 3 make up the remaining 80%. The other will take up the remaining 50% to 100%, whereas the latter will range from 20% to 50%.

SQL coding

Calculate the value of 3 dimensions RFM

R - Recency:

According to the definition of R, we need to calculate the last active date of each customer and calculate the number of days from the last active customer to the current time.

Using the `MAX()` function on adjusted_created_at to get the last date the customer purchased. and use the calculation `CURRENT_DATE - MAX(adjusted_created_at)` to calculate the number of days the customer has been most active until now.

```
SELECT Customer_ID
        , max([Date]) recent_date
        , DATEDIFF(DAY, max([Date]), DATEFROMPARTS(2016, 12, 31)) recency
        , COUNT(distinct Transaction_ID) frequency
        , SUM(Sales_Amount) monetary
FROM dbo.scanner_data
GROUP BY Customer_ID
```

F - Frequency

Using the `COUNT()` function to count the number of incoming customers. Here, each time a customer comes can buy many items, so we use `DISTINCT` to count the number of sale_id instead of counting the number of lines.

```
COUNT(distinct Transaction_ID) frequency
```

M - Monetary

Using the SUM() function with the net_sale field of each customer, we will get the total spending of the customer in the past 1 year.

```
SUM(Sales_Amount) monetary
```

⇒

```
SELECT Customer_ID
        , max([Date]) recent_date
        , DATEDIFF(DAY, max([Date]), DATEFROMPARTS(2016, 12, 31)) recency
        ,COUNT(distinct Transaction_ID) frequency
        , SUM(Sales_Amount) monetary
FROM dbo.scanner_data
GROUP BY Customer_ID
```

RFM . taxonomy

We can use `NTILE()` with base 100 to line up the dimensions. Since recency returns the number of days of customer inactivity so far, we need to sort in descending order with `DESC`.

```
WITH rfm_calcu as (SELECT Customer_ID
        , max([Date]) recent_date
        , DATEDIFF(DAY, max([Date]), DATEFROMPARTS(2016, 12, 31)) recency
        ,COUNT(distinct Transaction_ID) frequency
        , SUM(Sales_Amount) monetary
FROM dbo.scanner_data
GROUP BY Customer_ID),

        rfm_rank as (SELECT Customer_ID, recent_date
        , recency
        , NTILE(100) OVER(ORDER by recency DESC) recency_rank
        , frequency
        , NTILE(100) OVER(ORDER by frequency) frequency_rank
        , monetary
        , NTILE(100) OVER(ORDER by monetary) monetary_rank
FROM rfm_calcu )

SELECT * FROM rfm_rank
```

As defined above, level 1 is the lowest level that will occupy the `recency_rank`, `frequency_rank` and `monetary_rank` fields from 1 to 50; level 2 will be from 51 to 80 and level will be from 81 to 100. We use `CASE` to assign them a value when the condition with `WHEN` is satisfied.

```
WITH rfm_calcu as (SELECT Customer_ID
        , max([Date]) recent_date
        , DATEDIFF(DAY, max([Date]), DATEFROMPARTS(2016, 12, 31)) recency
        , COUNT(distinct [Date]) frequency
        , SUM(Sales_Amount) monetary
FROM dbo.scanner_data
GROUP BY Customer_ID),

        rfm_rank as (SELECT Customer_ID, recent_date
        , recency
        , NTILE(100) OVER(ORDER by recency DESC) recency_rank
        , frequency
        , NTILE(100) OVER(ORDER by frequency) frequency_rank
        , monetary
        , NTILE(100) OVER(ORDER by monetary) monetary_rank
FROM rfm_calcu )

SELECT Customer_ID, recent_date
        , case when recency_rank <= 50 then '1'
            when recency_rank <= 80 then '2'
            when recency_rank <= 100 then '3'
            end as r
        ,case when frequency_rank <= 50 then '1'
            when frequency_rank <= 80 then '2'
            when frequency_rank <= 100 then '3'
            end as f
        ,case when monetary_rank <= 50 then '1'
            when monetary_rank <= 80 then '2'
            when monetary_rank <= 100 then '3'
            end as m
FROM rfm_rank
```

Next, based on the RFM value just found, we apply it to the table above to classify customers.

For faster sorting, we need to combine 3 RFM values into one column. with `CONCAT()` or `||`

```
WITH rfm_calcu as (SELECT Customer_ID
        , max([Date]) recent_date
        , DATEDIFF(DAY, max([Date]), DATEFROMPARTS(2016, 12, 31)) recency
```

```
        , COUNT(distinct [Date]) frequency
        , SUM(Sales_Amount) monetary
FROM dbo.scanner_data
GROUP BY Customer_ID),

        rfm_rank as (SELECT Customer_ID, recent_date
        , recency
        , NTILE(100) OVER(ORDER by recency DESC) recency_rank
        , frequency
        , NTILE(100) OVER(ORDER by frequency) frequency_rank
        , monetary
        , NTILE(100) OVER(ORDER by monetary) monetary_rank
FROM rfm_calcu ),

    rfm_lvl as (SELECT Customer_ID, recent_date
        , case when recency_rank <= 50 then '1'
            when recency_rank <= 80 then '2'
            when recency_rank <= 100 then '3'
            end as r
        ,case when frequency_rank <= 50 then '1'
            when frequency_rank <= 80 then '2'
            when frequency_rank <= 100 then '3'
            end as f
        ,case when monetary_rank <= 50 then '1'
            when monetary_rank <= 80 then '2'
            when monetary_rank <= 100 then '3'
            end as m
FROM rfm_rank),

    rfm_tbl as (SELECT *
                , CONCAT(r,f,m) rfm
        FROM rfm_lvl)

SELECT *
        , case when rfm = '333' then 'VIP customer'
            when rfm in ('332','322','232','222','331','231') then 'Loyal customer'
            when rfm in ('323') then 'Potential prospective customers'
            when rfm in ('133','123','223','233','132','131') then 'Risky customer'
            when rfm in ('113','213','212','313') then 'Customer spend a lot'
            when rfm in ('321','312','311','211') then 'New customer'
            when rfm in ('111','112','121','122','221') then 'Low value customer'
        end customer_segmentation
FROM rfm_tbl
order by customer_segmentation
```

We have the output: