1. Dữ liệu sử dụng

Paytm là một công ty công nghệ tài chính đa quốc gia của Ấn Độ chuyên về hệ thống thanh toán kỹ thuật số, thương mại điện tử và dịch vụ tài chính. Ví Paytm là ví kỹ thuật số/di động an toàn và được RBI phê duyệt mà bạn có thể sử dụng cho nhiều mục đích. Nó giống như tiền kỹ thuật số mà bạn có thể sử dụng cho bất kỳ hình thức thanh toán nào của người tiêu dùng. Bạn có thể thêm tiền vào ví Paytm thông qua UPI, ngân hàng trực tuyến hoặc thẻ tín dụng/thẻ ghi nợ. Ngoài ra, bạn có thể gửi tiền từ ví Paytm đến tài khoản ngân hàng hoặc ví Paytm của người khác.

Đây là một cơ sở dữ liệu nhỏ về các giao dịch thanh toán từ năm 2019 đến 2020 của PayTM. Cơ sở dữ liệu bao gồm 6 bảng:

- fact_transaction: Lưu trữ thông tin của tất cả các loại giao dịch: Thanh toán, Nạp tiền, Chuyển khoản, Rút tiền
- dim_scenario: Mô tả chi tiết các loại giao dịch
- dim_payment_channel: Mô tả chi tiết phương thức thanh toán
- dim_platform: Mô tả chi tiết về thiết bị thanh toán
- dim_status: Mô tả chi tiết kết quả của giao dịch

1.1.Dữ liệu bảng "dim_scenario"

Thực hiện truy vấn bằng ngôn ngữ T-SQL:

```
SELECT * FROM dim_scenario
```

Kết quả như sau:

	scenario_id	transaction_type	sub_category	category
1	S1_10	Payment	TV	Billing
2	S1_100	Payment	Supermarket	Shopping
3	S1_101	Payment	Shopping Stores	Shopping
4	S1_102	Payment	Digital Service	Other Services
5	S1_103	Payment	Electricity	Billing
6	S1_104	Payment	Electricity	Billing
7	S1_105	Payment	Electricity	Billing
8	S1_106	Payment	Others	Other Services
9	S1_107	Payment	Restaurant Chain	FnB
10	S1_108	Payment	Restaurant Chain	FnB

1.2.Dữ liệu bảng "dim_payment_channel"

Thực hiện truy vấn bằng ngôn ngữ T-SQL:

```
SELECT * FROM dim_payment_channel
```

Kết quả như sau:

	payment_channel_id	payment_method
1	11	Credit
2	12	Bank account
3	13	Balance
4	14	Local card
5	15	Debit

1.3.Dữ liệu bảng “dim_platform”

Thực hiện truy vấn bằng ngôn ngữ T-SQL:

```
SELECT * FROM dim_platform
```

Kết quả như sau:

	platform_id	payment_platform
1	P1	android
2	P2	ios
3	P3	web

1.4.Dữ liệu bảng “dim_status”

Thực hiện truy vấn bằng ngôn ngữ T-SQL:

```
SELECT * FROM dim_status
```

Kết quả như sau:

	status_id	status_description
1	-15	Your account is temporarily …
2	-14	Transactions suspected to be…
3	-13	Unknown error from the bank
4	-12	The bank transaction process…
5	-11	Payment failed
6	-10	Transactions suspected to be…
7	-9	Payment failed
8	-8	Wrong OTP
9	-7	Wrong password to pay more t…
10	-6	Exceeded the allowed amount …
11	-5	Payment password is incorrect
12	-4	Transaction failed due to du…
13	-3	The account does not have en…
14	-2	Payment expired transaction
15	1	Success

1.5.Dữ liệu bảng “fact_transaction”

Thực hiện truy vấn bằng ngôn ngữ T-SQL:

```
SELECT * FROM fact_transaction_2019
```

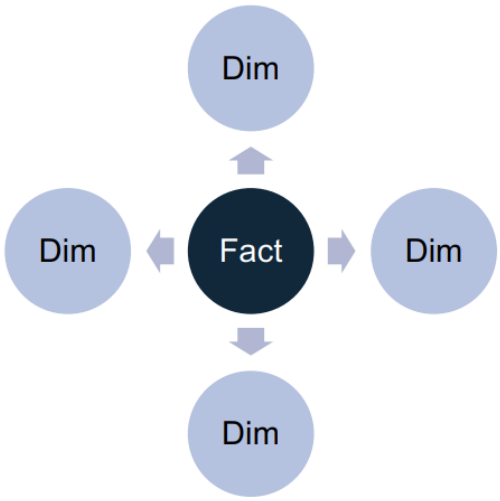
Kết quả như sau:

	transaction_id	customer_id	scenario_id	payment_channel_id	promotion_id	platform_id	status_id
1	101	3203	S2_1	12	0	P2	1
2	102	2324	S2_1	12	0	P2	1
3	103	18641	S2_1	12	0	P1	1
4	104	18785	S5_2	13	0	P1	1
5	105	5224	S1_3	13	0	P2	1
6	106	18828	S1_10	13	P_15	P1	1
7	107	18828	S1_40	13	0	P1	1
8	108	677	S2_1	12	0	P2	1
9	109	4003	S2_1	14	0	P1	1
10	110	4918	S2_1	12	0	P1	1

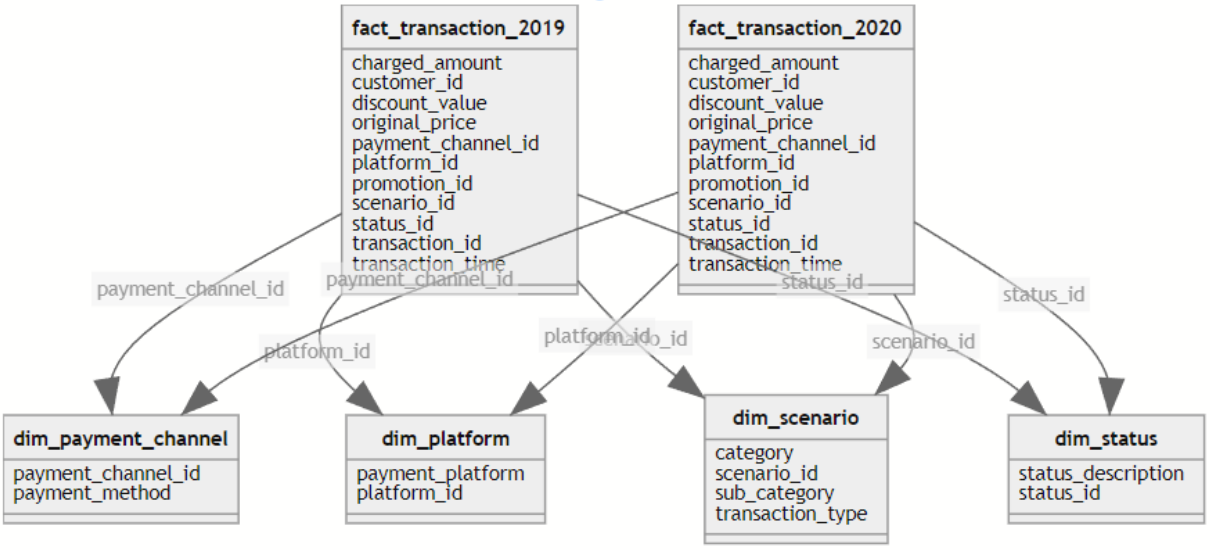
original_price	discount_value	charged_amount	transaction_time
100000	0	100000	2019-01-01 00:01:43.3000000
300000	0	300000	2019-01-01 00:25:41.0100000
100000	0	100000	2019-01-01 00:22:45.9420000
300000	0	300000	2019-01-01 00:31:43.0800000
97000	0	97000	2019-01-01 00:09:57.0160000
80000	40000	40000	2019-01-01 00:40:29.0010000
110000	0	110000	2019-01-01 00:46:12.8640000
200000	0	200000	2019-01-01 01:13:32.0110000
2000000	0	2000000	2019-01-01 01:17:16.1300000
100000	0	100000	2019-01-01 01:20:34.2370000

1.6.Mối quan hệ giữa các bảng

Mối quan hệ giữa các bảng được thiết kế theo mô hình Star-schema. Dưới đây là bản thiết kế mô hình với dữ liệu bảng fact nằm ở trung tâm, nối đến các bảng dim ở xung quanh.

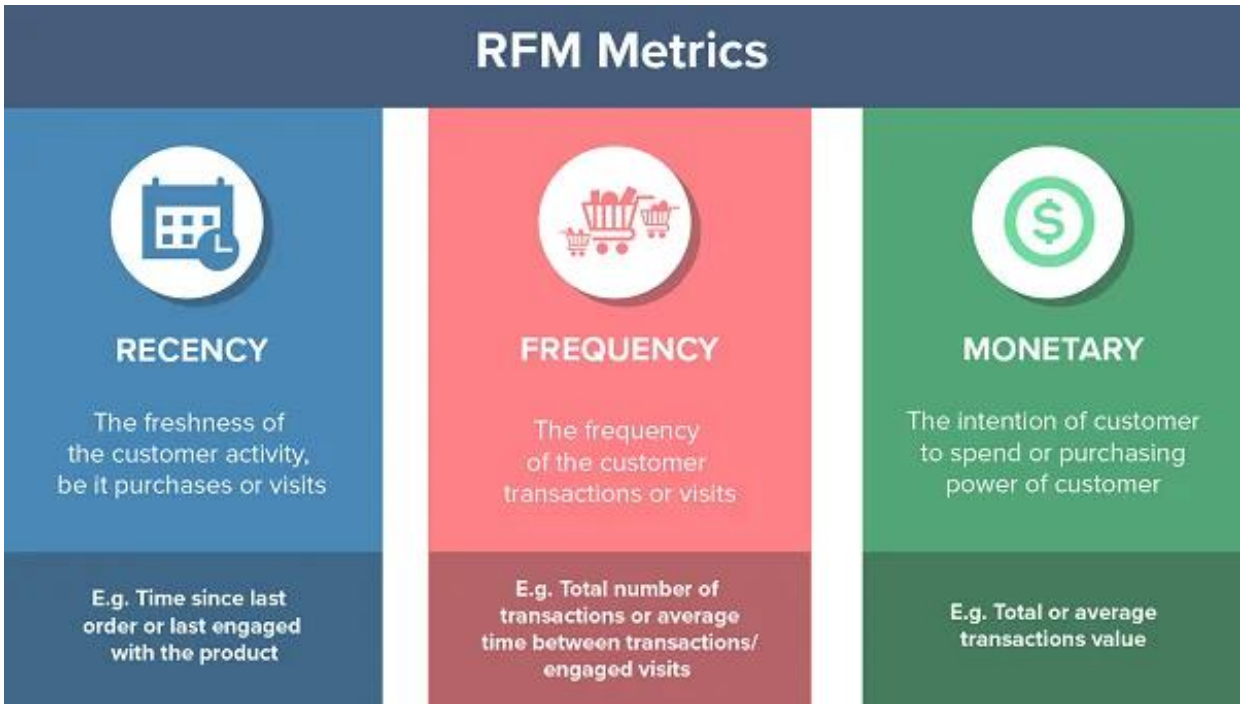


Schema visualization ta được kết quả như sau:



2. Phân nhóm khách hàng

Phân tích RFM (RECENCY – FREQUENCY – MONETERY) là một kỹ thuật được sử dụng trong marketing để xếp hạng và phân nhóm khách hàng dựa trên số lần truy cập gần đây, tần suất và tổng số tiền giao dịch gần đây để có thể tìm ra những khách hàng tiềm năng và thực hiện các chiến dịch marketing.



(Theo https://datapot.vn/)

Dựa trên phương pháp phân tích RFM đã đề cập, ta tiến hành phân nhóm người dùng đối với Billing category (từ năm 2019 đến năm 2020, chỉ chọn các giao dịch thành công) thành 9 nhóm với các tiêu chí như sau:

Segment	R tier	F tier	M tier
Best customers	1	1	1
Lost Bad customers	3,4	3,4	1,2,3,4
Lost customer	3,4	2	1,2,3,4
Almost lost	2	1	1,2,3,4
Loyal customers	1	1	2,3,4
Big Spender	1,2	1,2,3	1
New customers	1,2	4	1,2,3,4
Hibernating	3,4	1	1,2,3,4
Potential Loyalist	1,2	2,3	2,3,4

2.1. Giá trị recency, frequency, monetary

Đầu tiên ta tính các giá trị recency, frequency, monetary của từng khách hàng.

```
WITH fact_table AS (
   SELECT customer_id, transaction_id, transaction_time, charged_amount
       , CONVERT (varchar, transaction_time, 112) AS day_formated
   FROM fact_transaction_2019 AS fact_19
   JOIN dim_scenario AS scena ON fact_19.scenario_id = scena.scenario_id
   WHERE category = 'Billing' AND status_id = 1
   UNION
   SELECT customer_id, transaction_id, transaction_time, charged_amount
       , CONVERT (varchar, transaction_time, 112) AS day_formated
   FROM fact_transaction_2020 AS fact_20
   JOIN dim_scenario AS scena ON fact_20.scenario_id = scena.scenario_id
   WHERE category = 'Billing' AND status_id = 1
)
SELECT customer_id
   , DATEDIFF (day, MAX ( transaction_time ), '2020-12-31')  AS recency
   , COUNT ( DISTINCT day_formated) AS frequency
   , SUM (charged_amount *1.0) AS monetary
FROM fact_table
GROUP BY customer_id
```

	customer_id	recency	frequency	monetary
1	7162	269	3	353100.0
2	69509	86	2	422000.0
3	28387	147	1	55200.0
4	17003	149	1	203190.0
5	22768	28	4	356300.0
6	5788	716	1	103615.0
7	17172	725	1	216000.0
8	1397	237	5	2020456.0
9	12758	718	1	22900.0
10	1520	212	4	820000.0

2.2.R-tier, f-tier, m-tier

Phân loại r-tier, f-tier, m-tier dựa trên cách tính như sau:

- Khách hàng có xếp hạng từ 0% đến 25%: được gán là tier 1.
- Khách hàng có xếp hạng từ 25% đến 50%: được gán là tier 2.
- Khách hàng có xếp hạng từ 50% đến 75%: được gán là tier 1.
- Khách hàng có xếp hạng từ 75% đến 100%: được gán là tier 1.

Từ đó ta xếp được điểm tier như sau:

```
WITH fact_table AS (
   SELECT customer_id, transaction_id, transaction_time, charged_amount
       , CONVERT (varchar, transaction_time, 112) AS day_formated
   FROM fact_transaction_2019 AS fact_19
   JOIN dim_scenario AS scena ON fact_19.scenario_id = scena.scenario_id
   WHERE category = 'Billing' AND status_id = 1
   UNION
   SELECT customer_id, transaction_id, transaction_time, charged_amount
       , CONVERT (varchar, transaction_time, 112) AS day_formated
   FROM fact_transaction_2020 AS fact_20
   JOIN dim_scenario AS scena ON fact_20.scenario_id = scena.scenario_id
   WHERE category = 'Billing' AND status_id = 1
)
, rfm_table AS (
SELECT customer_id
   , DATEDIFF (day, MAX ( transaction_time ), '2020-12-31')  AS recency
   , COUNT ( DISTINCT day_formated) AS frequency
   , SUM (charged_amount * 1.0) AS monetary
FROM fact_table
GROUP BY customer_id
)
, rfm_rank_table AS (
    SELECT *
        , PERCENT_RANK () OVER (ORDER BY recency ASC) AS recency_rank
        , PERCENT_RANK () OVER (ORDER BY frequency DESC) AS frequency_rank
        , PERCENT_RANK () OVER (ORDER BY monetary DESC) AS monetary_rank
    FROM rfm_table
```

```
)
SELECT customer_id
    , CASE WHEN recency_rank BETWEEN 0 AND 0.25 THEN 1
        WHEN recency_rank BETWEEN 0.25 AND 0.5 THEN 2
        WHEN recency_rank BETWEEN 0.5 AND 0.75 THEN 3
        ELSE 4 END AS r_tier
    , CASE WHEN frequency_rank BETWEEN 0 AND 0.25 THEN 1
        WHEN frequency_rank BETWEEN 0.25 AND 0.5 THEN 2
        WHEN frequency_rank BETWEEN 0.5 AND 0.75 THEN 3
        ELSE 4 END AS f_tier
    , CASE WHEN monetary_rank BETWEEN 0 AND 0.25 THEN 1
        WHEN monetary_rank BETWEEN 0.25 AND 0.5 THEN 2
        WHEN monetary_rank BETWEEN 0.5 AND 0.75 THEN 3
        ELSE 4 END AS m_tier
FROM rfm_rank_table
```

	customer_id	r_tier	f_tier	m_tier
1	5145	1	1	1
2	69319	1	1	1
3	55755	1	1	1
4	2879	1	1	1
5	10409	1	1	1
6	38426	1	1	1
7	696	1	2	1
8	2209	1	1	1
9	15504	1	1	1
10	5291	4	1	1

2.3.Phân loại tệp khách hàng

Thực hiên truy vấn bằng ngôn ngữ T-SQL:

```
WITH fact_table AS (
   SELECT customer_id, transaction_id, transaction_time, charged_amount
       , CONVERT (varchar, transaction_time, 112) AS day_formated
   FROM fact_transaction_2019 AS fact_19
   JOIN dim_scenario AS scena ON fact_19.scenario_id = scena.scenario_id
   WHERE category = 'Billing' AND status_id = 1
   UNION
   SELECT customer_id, transaction_id, transaction_time, charged_amount
       , CONVERT (varchar, transaction_time, 112) AS day_formated
   FROM fact_transaction_2020 AS fact_20
   JOIN dim_scenario AS scena ON fact_20.scenario_id = scena.scenario_id
   WHERE category = 'Billing' AND status_id = 1
)
```

```
, rfm_table AS (
SELECT customer_id
   , DATEDIFF (day, MAX ( transaction_time ), '2020-12-31')  AS recency
   , COUNT ( DISTINCT day_formated) AS frequency
   , SUM (charged_amount * 1.0) AS monetary
FROM fact_table
GROUP BY customer_id
)
, rfm_rank_table AS (
    SELECT *
        , PERCENT_RANK () OVER (ORDER BY recency ASC) AS recency_rank
        , PERCENT_RANK () OVER (ORDER BY frequency DESC) AS frequency_rank
        , PERCENT_RANK () OVER (ORDER BY monetary DESC) AS monetary_rank
    FROM rfm_table
)
, rfm_tier AS (
    SELECT customer_id
        , CASE WHEN recency_rank BETWEEN 0 AND 0.25 THEN 1
            WHEN recency_rank BETWEEN 0.25 AND 0.5 THEN 2
            WHEN recency_rank BETWEEN 0.5 AND 0.75 THEN 3
            ELSE 4 END AS r_tier
        , CASE WHEN frequency_rank BETWEEN 0 AND 0.25 THEN 1
            WHEN frequency_rank BETWEEN 0.25 AND 0.5 THEN 2
            WHEN frequency_rank BETWEEN 0.5 AND 0.75 THEN 3
            ELSE 4 END AS f_tier
        , CASE WHEN monetary_rank BETWEEN 0 AND 0.25 THEN 1
            WHEN monetary_rank BETWEEN 0.25 AND 0.5 THEN 2
            WHEN monetary_rank BETWEEN 0.5 AND 0.75 THEN 3
            ELSE 4 END AS m_tier
    FROM rfm_rank_table
)
SELECT *
    , CASE WHEN r_tier = 1 AND f_tier = 1 AND m_tier = 1 THEN 'Best customers'
        WHEN r_tier IN ('3', '4') AND f_tier IN ('3', '4') THEN 'Lost Bad customers'
        WHEN r_tier IN ('3', '4') AND f_tier = 2 THEN 'Lost customer'
        WHEN r_tier = 2 AND f_tier = 1 THEN 'Alomost lost'
        WHEN r_tier = 1 AND f_tier = 1 AND m_tier IN ('2', '3', '4') THEN 'Loyal customers'
       WHEN r_tier IN ('1', '2') AND f_tier IN ('1', '2', '3') AND m_tier = 1 THEN 'Big Spender'
        WHEN r_tier IN ('1', '2') AND f_tier = 4 THEN 'New customers'
        WHEN r_tier IN ('3', '4') AND f_tier = 1 THEN 'Hibernating'
         WHEN r_tier IN ('1', '2') AND f_tier IN ('2', '3') AND m_tier IN ('2', '3', '4') THEN
'Big Spender'
        END AS segment
FROM rfm_tier
```

Sau khi truy vấn ta phân chia khách hàng thành 9 nhóm khác nhau. Với từng nhóm khách hàng cụ thể được lưu trữ trên hệ thống, doanh nghiệp sẽ dễ dàng hơn trong việc đưa ra các chiến dịch marketing phù hợp. Việc tiếp cận đúng phân khúc khách hàng dù là trước hay sau khi mua hàng đều vô cùng quan trọng. Bởi điều này sẽ giúp tăng tỷ lệ chuyển đổi đơn hàng, mang về doanh thu đáng kể cho doanh nghiệp. Đặc biệt, chăm sóc đúng đối tượng khách hàng sẽ giúp doanh nghiệp có số lượng khách hàng thường xuyên đáng mơ ước.

	customer_id	r_tier	f_tier	m_tier	segment
1	5145	1	1	1	Best customers
2	69319	1	1	1	Best customers
3	55755	1	1	1	Best customers
4	2879	1	1	1	Best customers
5	10409	1	1	1	Best customers
6	38426	1	1	1	Best customers
7	696	1	2	1	Big Spender
8	2209	1	1	1	Best customers
9	15504	1	1	1	Best customers
10	5291	4	1	1	Hibernating
11	12568	2	1	1	Alomost lost
12	2450	1	1	1	Best customers
13	7798	2	1	1	Alomost lost
14	42315	1	1	1	Best customers
15	11184	1	1	1	Best customers
16	22849	1	1	1	Best customers
17	10820	1	1	1	Best customers
18	300	2	1	1	Alomost lost
19	12034	1	1	1	Best customers
20	20726	1	1	1	Best customers

3. Cohort analysis

Phân tích theo nhóm là một cách hữu ích để so sánh các nhóm thực thể theo thời gian. Nhiều hành vi quan trọng phải mất hàng tuần, hàng tháng hoặc hàng năm mới xảy ra hoặc phát triển và phân tích theo nhóm là một cách để hiểu những thay đổi này. Phân tích theo nhóm phát hiện mối tương quan giữa các đặc điểm của nhóm và các xu hướng dài hạn này, điều này có thể dẫn đến các giả thuyết về nguyên nhân thúc đẩy. Ví dụ: những khách hàng có được thông qua một chiến dịch tiếp thị có thể có các kiểu mua hàng dài hạn khác với những khách hàng bị bạn bè thuyết phục dùng thử sản phẩm của công ty. Phân tích theo nhóm có thể được sử dụng để theo dõi các nhóm người dùng hoặc khách hàng mới và đánh giá cách họ so sánh với các nhóm trước đó.

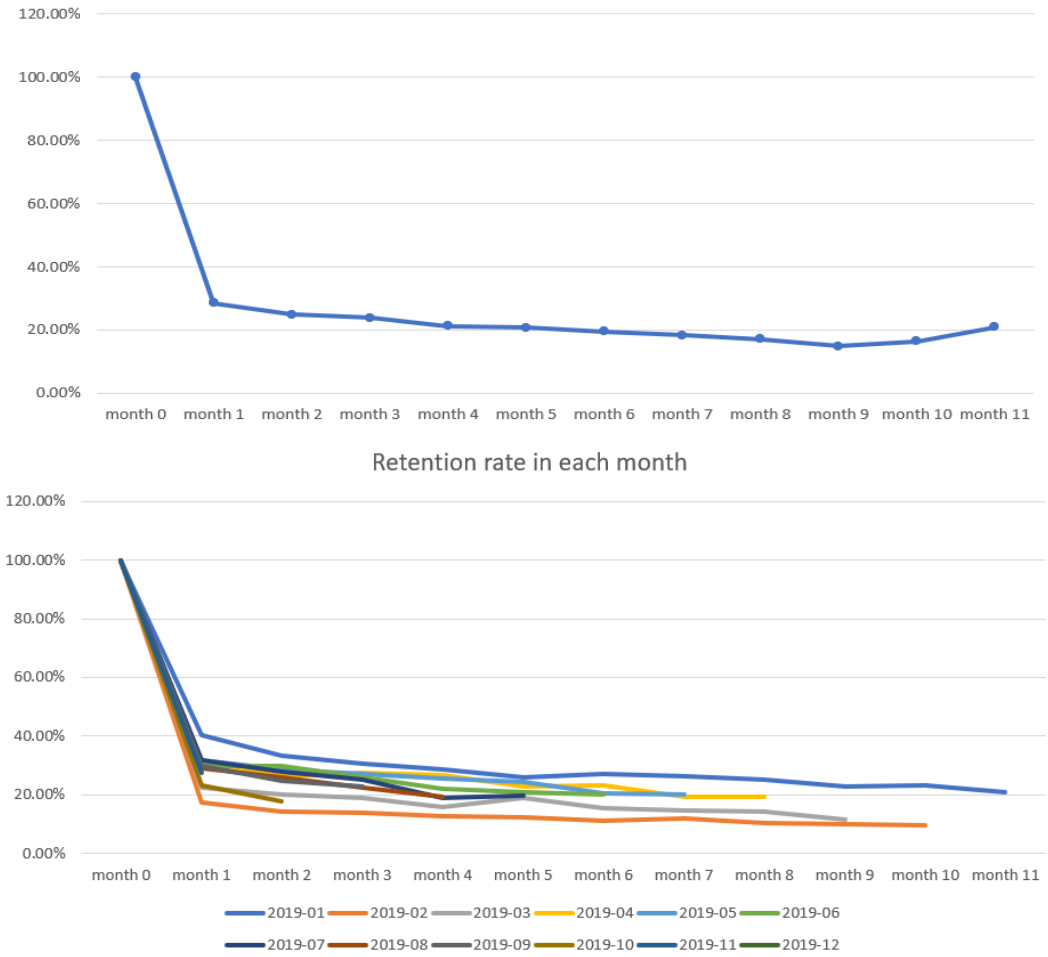
Một trong những loại phân tích nhóm phổ biến nhất là phân tích tỷ lệ giữ chân. Các doanh nghiệp thường muốn khách hàng của họ tiếp tục mua sản phẩm hoặc sử dụng dịch vụ của họ, vì việc giữ chân khách hàng mang lại nhiều lợi nhuận hơn là thu hút khách hàng mới. Kết quả của phân tích retention được thể hiện dưới dạng phần trăm và tỷ lệ giữ chân trong thời gian bắt đầu luôn là 100%. Theo thời gian, tỷ lệ giữ chân dựa trên số lượng thường giảm và không bao giờ có thể vượt quá 100%.

Ta truy vấn với câu lệnh như sau:

```
WITH table_first_month AS (
   SELECT customer_id, transaction_id, transaction_time
       ,  MIN ( MONTH (transaction_time)) OVER ( PARTITION BY customer_id ) AS first_month
       ,  MONTH (transaction_time) - MIN ( MONTH (transaction_time)) OVER ( PARTITION BY
customer_id ) AS subsequent_month
   FROM fact_transaction_2019 fact_19
   JOIN dim_scenario sce ON fact_19.scenario_id = sce.scenario_id
   WHERE sub_category = 'Telco Card' AND status_id = 1
)
, table_sub_month AS (
   SELECT first_month AS acquisition_month
       , subsequent_month
       , COUNT (DISTINCT customer_id) AS number_retained_customers
   FROM table_first_month
   GROUP BY first_month, subsequent_month
   -- ORDER BY first_month, subsequent_month
)
, table_retention AS (
   SELECT *
       , FIRST_VALUE ( number_retained_customers ) OVER ( PARTITION BY acquisition_month ORDER
BY subsequent_month ASC ) AS original_customers
       , number_retained_customers * 1.0 /
           FIRST_VALUE ( number_retained_customers ) OVER ( PARTITION BY acquisition_month
ORDER BY subsequent_month ASC ) AS pct
   FROM table_sub_month
)
SELECT acquisition_month
   , original_customers
   , "0", "1", "2", "3", "4", "5", "6","7", "8", "9", "10", "11"
FROM ( SELECT acquisition_month, subsequent_month, original_customers, pct
       FROM table_retention) AS source_table
PIVOT (
   MAX (pct)
   FOR subsequent_month IN ( "0", "1", "2", "3", "4", "5", "6",  "7", "8", "9", "10", "11" )
) AS pivot_logic
ORDER BY acquisition_month
```

Sau khi có được kết quả truy vấn ta Color scales bằng excel được như sau.

acquisition_month	original_customers	month 0	month 1	month 2	month 3	month 4	month 5	month 6	month 7	month 8	month 9	month 10	month 11
1	2111	100.00%	40.45%	33.16%	30.70%	28.61%	26.01%	26.95%	26.24%	24.96%	22.97%	23.35%	20.99%
2	2074	100.00%	17.26%	14.18%	13.84%	12.58%	12.39%	11.14%	11.72%	10.41%	10.08%	9.55%	
3	1051	100.00%	22.45%	19.98%	18.84%	15.98%	18.93%	15.51%	14.46%	14.08%	11.70%		
4	699	100.00%	30.62%	26.47%	27.47%	26.75%	22.89%	23.18%	19.46%	19.17%			
5	754	100.00%	31.56%	28.51%	27.06%	25.46%	24.40%	20.29%	20.03%				
6	647	100.00%	29.98%	29.83%	25.81%	22.10%	20.87%	20.09%					
7	693	100.00%	31.89%	27.99%	25.11%	19.05%	19.62%						
8	834	100.00%	28.90%	25.78%	22.54%	19.42%							
9	760	100.00%	29.87%	24.87%	22.89%								
10	981	100.00%	23.34%	17.84%									
11	685	100.00%	27.30%										
12	915	100.00%											
All uesrs	12204	100.00%	28.51%	24.86%	23.81%	21.25%	20.73%	19.53%	18.38%	17.16%	14.92%	16.45%	20.99%



Từ bảng tỉ lệ giữ chân ở trên, ta có thể suy ra:

- Có 2111 người dùng khởi chạy ứng dụng PayTM vào tháng 1 năm 2019.
- Tỉ lệ giữ chân sau đó 1 tháng (tức là vào tháng 2 năm 2019) là 40,45%, sau 7 tháng (tức là tháng 8 năm 2019) là 26,24%.
- Trong số tất cả những người dùng mới trong năm 2019 (12204 người dùng). có 28,51% người dùng tiếp tục sử dụng sau 1 tháng, 18,38% tiếp tục sử dụng sau 7 tháng.