**RECENCY - FREQUENCY – MONETARY ANALYSIS WITH SQL**

# Tổng quan

Phân tích RFM 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, tấn 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. Bài viết trình bày cách giải quyết bài toán với database: AdventureWorkDW2019

1. **Các vấn đề và giải pháp**
   * + 1. ***Các vấn đề gặp phải:***

* NoPurchasePerYear (Số lần mua hàng trung bình năm kỳ vọng tính ra sẽ là một số thập phân nhưng thực tế tính ra chỉ có giá trị nguyên: COUNT(DISTINCT(fi.SalesOrderNumber))/DATEDIFF(YEAR,MIN(CONVERT(CHAR(10), fi.OrderDate, 120)), '2015-01-01')
* Tìm khách hàng trong nhóm 20% khách có AmountPerYear và TotalProfit cao nhất.
* Sau khi tính điểm khách hàng theo quy tắc đề bài, kết quả trà về là những cột TopActive, TopYear, TopProfit và TopPur riêng biệt. Kết quả này khó có thể tính được tổng điểm từng khách hàng vì không thể cộng tổng cột.
  + - 1. ***Cách giải quyết:***
* Với NoPurchasePerYear: chuyển một trong 2 giá trị trong công thức thành số thập phân: COUNT(DISTINCT(fi.SalesOrderNumber))/CAST(DATEDIFF(YEAR, MIN(CONVERT(CHAR(10), fi.OrderDate, 120)), '2015-01-01') AS float)
* Với top 20% AmountPerYear và TotalProfit, dùng WindowFuntion: PERCENT\_RANK (), sắp xếp theo AmountPerYear và TotalProfit giảm dần, đánh dấu 1 điểm cho khách hàng trong khoảng từ 0% đến 20%.
* Với tổng điểm khách hàng: Dùng UNPIVOT, kết quả sẽ trả về một cột với CustomerKey mỗi khách hàng được tăng thêm nhiều lần, ứng với điểm tương ứng. Kết quả này có thể dùng để tính tổng điểm của mỗi khách hàng.

# Các bước thực hiện

## *3.1) Mô tả các trường và cách tính (Các trường được tính đến ngày 2015-01-01)*

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên trường** | **Mô tả** | **Cách tính** |
| 1 | CustomerKey | Mã khách hàng |  |
| 2 | CustomerName | Tên khách hàng( gồm FirstName, MiddleName (nếu có) và LastName) | CONCAT\_WS(' ', c.FirstName, c.MiddleName, c.LastName) |
| 3 | MonthsFrom1stPurchchase | Số tháng từ ngày mua hàng đầu tiên | DATEDIFF(MONTH, MIN(CONVERT(CHAR(10), fi.OrderDate, 120)), '2015-01-01') |
| 4 | NoPurchasePerYear | Số lần mua hàng trung bình năm(Tổng số lần mua hàng/số năm từ ngày mua hàng đầu tiên) | Tổng số lần mua hàng: đơn hàng được tính theo ngày, giả định khách không mua hàng hơn 1 lần một ngày  COUNT(DISTINCT(fi.SalesOrderNumber)) /CAST(DATEDIFF(YEAR, MIN(CONVERT(CHAR(10), fi.OrderDate, 120)), '2015-01-01') AS float) |
| 5 | AmountPerYear | Giá trị mua hàng trung bình năm(Tổng giá trị mua hàng/số năm từ ngày mua hàng đầu tiên) | CAST(SUM(SalesAmount)  / DATEDIFF(YEAR, MIN(CONVERT(CHAR(10), fi.OrderDate, 120)), '2015-01-01') AS float) |
| 6 | TotalProfit | Tổng SalesAmount- Tổng ProductStandardCost | SUM(SalesAmount) - SUM(fi.ProductStandardCost) |
| 7 | Customer Segment | Phân nhóm theo quy tắc |  |

Output cuối cùng là bảng sau:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **CustomerKey** | **CustomerName** | **MonthsFrom1stPurchase** | **NoPurchasePerYear** | **AmountPerYear** | **TotalProfit** | **CustomerSegment** |
| 11000 | Jon V Yang | 48 | 0.75 | 2062.2475 | 3513.6905 | Gold |
| 11001 | Eugene L Huang | 48 | 0.75 | 1595.97 | 2795.8839 | Gold |
| 11002 | Ruben Torres | 48 | 0.75 | 2028.51 | 3454.8801 | Gold |
| 11003 | Christy Zhu | 49 | 0.6 | 1627.858 | 3467.1264 | Gold |
| 11004 | Elizabeth Johnson | 48 | 0.75 | 2049.0025 | 3501.9051 | Gold |
| 11005 | Julio Ruiz | 49 | 0.6 | 1624.266 | 3459.8654 | Gold |

***Cách tính điểm khách hàng:***

* Khách hàng Active: Mua hàng trong vòng 1 năm gần nhất: 1 điểm
* Khách hàng top 20% có AmountPerYear cao nhất: 2 điểm
* Khách hàng top 20% có TotalProfit cao nhất: 2 điểm
* Khách hàng có NoPurchasePerYear >1 : 1 điểm

***Phân loại khách hàng:***

* Lớn hơn hoặc bằng 5 điểm: Diamond
* 4 điểm: Gold
* 3 điểm: Silver
* Dưới 3 điểm: Normal

***3.2) Các bước thực hiện***

**3.2.1) Tính các trường sau, group by khách hàng:**

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên trường** | **Cách tính** |
| 1 | CustomerKey |  |
| 2 | CustomerName | CONCAT\_WS(' ', c.FirstName, c.MiddleName, c.LastName) |
| 3 | MonthsFrom1stPurchchase | DATEDIFF(MONTH, MIN(CONVERT(CHAR(10), fi.OrderDate, 120)), '2015-01-01') |
| 4 | NoPurchasePerYear | COUNT(DISTINCT(fi.SalesOrderNumber)) /CAST(DATEDIFF(YEAR, MIN(CONVERT(CHAR(10), fi.OrderDate, 120)), '2015-01-01') AS float) |
| 5 | AmountPerYear | CAST(SUM(SalesAmount)  / DATEDIFF(YEAR, MIN(CONVERT(CHAR(10), fi.OrderDate, 120)), '2015-01-01') AS float) |
| 6 | TotalProfit | SUM(SalesAmount) - SUM(fi.ProductStandardCost) |

1. Câu lệnh:

SELECT fi.CustomerKey

, CONCAT\_WS(' ', c.FirstName, c.MiddleName, c.LastName) AS CustomerName

, MAX(OrderDate) AS RecentOrder

, DATEDIFF(MONTH, MIN(CONVERT(CHAR(10), fi.OrderDate, 120)), '2015-01-01') AS Monthsfrom1stpur

, COUNT(DISTINCT(fi.SalesOrderNumber)) AS TotalOrder

, DATEDIFF(YEAR, MIN(CONVERT(CHAR(10), fi.OrderDate, 120)), '2015-01-01')

AS Yearsfrom1spur

, COUNT(DISTINCT(fi.SalesOrderNumber))

/CAST(DATEDIFF(YEAR, MIN(CONVERT(CHAR(10), fi.OrderDate, 120)), '2015-01-01') AS float) AS NoPurchasePerYear

, CAST(SUM(SalesAmount)

/ DATEDIFF(YEAR, MIN(CONVERT(CHAR(10), fi.OrderDate, 120)),

'2015-01-01') AS float) AS AmountPerYear

, SUM(SalesAmount) - SUM(fi.ProductStandardCost) AS TotalProfit

FROM FactInternetSales AS fi

LEFT JOIN DimCustomer AS c

ON fi.CustomerKey = c.CustomerKey

GROUP BY fi.CustomerKey

, CONCAT\_WS(' ', FirstName, MiddleName, LastName)

### b) Kết quả:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Customer**  **Key** | **Customer**  **Name** | **Recent**  **Order** | **Month**  **From**  **1stPurchase** | **Total**  **Order** | **Years**  **From**  **1stOrder** | **No**  **Purchase**  **PerYear** | **Amount**  **PerYear** | **Total**  **Profit** |
| 20075 | Aaron A Allen | 12/2/11 0:00 | 37 | 1 | 4 | 0.25 | 849.9975 | 1487.836 |
| 15568 | Aaron A Hayes | 9/28/13 0:00 | 28 | 2 | 3 | 0.666667 | 1037.657 | 1356.908 |
| 28866 | Aaron B Adams | 4/28/13 0:00 | 21 | 1 | 2 | 0.5 | 58.98 | 52.4628 |
| 21414 | Aaron Butler | 12/26/13 0:00 | 13 | 1 | 2 | 0.5 | 7.49 | 9.3774 |
| 18695 | Aaron C Diaz | 3/25/13 0:00 | 43 | 2 | 4 | 0.5 | 1507.393 | 2300.355 |

## 3.2.2) Phân loại khách hàng:

### *a) Tìm phần trăm khách hàng có AmountPeryear và TotalProfit cao nhất:*

- Dùng window function Percent\_Rank ()

- Câu lệnh: SELECT \*

, PERCENT\_RANK () OVER (ORDER BY AmountPerYear DESC) AS AmountPerYear\_rank

, PERCENT\_RANK () OVER (ORDER BY TotalProfit DESC) AS TotalProfit\_rank

FROM rfm

### *b) Gắn điểm khách hàng theo yêu cầu đề bài:*

SELECT CustomerKey

, CASE

WHEN YEAR(RecentOrder) = 2014 THEN 1

ELSE 0

END AS TopActive

, CASE

WHEN AmountPerYear\_rank BETWEEN 0 AND 0.2 THEN 2

ELSE 0

END AS TopYear

, CASE

WHEN TotalProfit\_rank BETWEEN 0 AND 0.2 THEN 2

ELSE 0

END AS TopProfit

, CASE

WHEN NoPurchasePerYear > 1 THEN 1

ELSE 0

END AS TopPur

FROM percentrank

* Kết quả:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CustomerKey** | **TopActive** | **TopYear** | **TopProfit** | **TopPur** |
| 12132 | 0 | 2 | 2 | 1 |
| 12308 | 0 | 2 | 2 | 1 |
| 12301 | 0 | 2 | 2 | 1 |
| 12300 | 0 | 2 | 2 | 1 |
| 12131 | 0 | 2 | 2 | 1 |
| 12321 | 0 | 2 | 2 | 1 |

### *c) Thu các cột TopActive, TopYear, TopProfit, TopPur thành một cột*

- Dùng Unpivot:

- Câu lệnh: SELECT CustomerKey

, Score

, SegmentedCustomer

FROM segment

UNPIVOT(

Score FOR SegmentedCustomer IN(TopActive, TopYear, TopProfit, TopPur)

) AS u

- Kết quả:

|  |  |  |
| --- | --- | --- |
| **CustomerKey** | **Score** | **SegmentedCustomer** |
| 12308 | 0 | TopActive |
| 12308 | 2 | TopYear |
| 12308 | 2 | TopProfit |
| 12308 | 1 | TopPur |
| 12300 | 0 | TopActive |
| 12300 | 2 | TopYear |
| 12300 | 2 | TopProfit |
| 12300 | 1 | TopPur |
| 12321 | 0 | TopActive |
| 12321 | 2 | TopYear |
| 12321 | 2 | TopProfit |
| 12321 | 1 | TopPur |
| 12307 | 0 | TopActive |
| 12307 | 2 | TopYear |
| 12307 | 2 | TopProfit |

### *d) Tính tổng điểm cuối cùng theo CustomerKey*

- Câu lệnh:

SELECT CustomerKey

, SUM(Score) AS TotalScore

FROM ScoreCustomer

GROUP BY CustomerKey

- Kết quả:

|  |  |  |
| --- | --- | --- |
| **CustomerKey** | **Score** | **SegmentedCustomer** |
| 12308 | 0 | TopActive |
| 12308 | 2 | TopYear |
| 12308 | 2 | TopProfit |
| 12308 | 1 | TopPur |
| 12300 | 0 | TopActive |
| 12300 | 2 | TopYear |
| 12300 | 2 | TopProfit |
| 12300 | 1 | TopPur |
| 12321 | 0 | TopActive |
| 12321 | 2 | TopYear |
| 12321 | 2 | TopProfit |
| 12321 | 1 | TopPur |
| 12307 | 0 | TopActive |
| 12307 | 2 | TopYear |
| 12307 | 2 | TopProfit |

### *e) Phân loại khách hàng:*

- Dùng CASE WHEN

- Câu lệnh:

SELECT CustomerKey

, CASE

WHEN TotalScore >=5 THEN 'Diamond'

WHEN TotalScore = 4 THEN 'Gold'

WHEN TotalScore = 3 THEN 'Silver'

WHEN TotalScore < 3 THEN 'Normal'

END AS CustomerSegement

FROM CustomerFinalScore

- Kết quả:

|  |  |
| --- | --- |
| **CustomerKey** | **CustomerSegmented** |
| 14324 | Gold |
| 15652 | Gold |
| 19897 | Normal |
| 28387 | Normal |
| 27059 | Normal |
| 11407 | Normal |
| 24142 | Normal |
| 18569 | Normal |
| 22814 | Normal |

### *f) Mỗi bước ở trên là một CTE, JOIN CTE đầu là các trường dữ liệu đã được tính và CTE cuối có cột phân loại khách hàng cuối cùng.*

- Câu lệnh:

SELECT rfm.CustomerKey

, rfm.CustomerName

, rfm.Monthsfrom1stpur AS MonthFrom1stPurchase

, rfm.NoPurchasePerYear

, rfm.AmountPerYear

, rfm.TotalProfit

, fcs.CustomerSegement

FROM rfm

LEFT JOIN FinalCustomerSegment AS fcs

ON rfm.CustomerKey = fcs.CustomerKey

ORDER BY CustomerKey

-Kết quả:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **CustomerKey** | **CustomerName** | **MonthsFrom1stPurchase** | **NoPurchasePerYear** | **AmountPerYear** | **TotalProfit** | **CustomerSegment** |
| 11000 | Jon V Yang | 48 | 0.75 | 2062.2475 | 3513.6905 | Gold |
| 11001 | Eugene L Huang | 48 | 0.75 | 1595.97 | 2795.8839 | Gold |
| 11002 | Ruben Torres | 48 | 0.75 | 2028.51 | 3454.8801 | Gold |
| 11003 | Christy Zhu | 49 | 0.6 | 1627.858 | 3467.1264 | Gold |
| 11004 | Elizabeth Johnson | 48 | 0.75 | 2049.0025 | 3501.9051 | Gold |
| 11005 | Julio Ruiz | 49 | 0.6 | 1624.266 | 3459.8654 | Gold |

4. Kết luận:

Phân tích RFM giúp phân loại khách hàng và trả lời cho những câu hỏi:

* Khách hàng nào thuộc nhóm trung thành với lượt mua trung bình năm nhiều nhất?
* Khách hàng công ty đang có nguy cơ mất?
* Cần tập trung chiến lược marketing cho nhóm khách hàng nào?

Kết quả cuối cùng cho thấy nhóm khách hàng ‘Diamond’ và ‘Gold' tuy có số lượt mua trung bình năm ít hơn các nhóm khách hàng khác nhưng là những nhóm mang lại lợi nhuận nhiều nhất cho công ty do có sức mua lớn. Kết quả này cũng có mối liên hệ với nguyên lý Pareto: “80% doanh thu công ty đến từ 20% khách hàng.”

Dựa vào kết quả phân tích, công ty có thể cân nhắc chiến lược kinh marketing phù hợp cho từng nhóm đổi tượng khách hàng.

**Reference**

#### *How to Apply the 80-20 Rule* (2022). Available at: https://www.investopedia.com/terms/1/80-20-rule.asp (Accessed: 16 January 2022).

#### *Recency, Frequency, Monetary Value (RFM) Definition* (2022). Available at: https://www.investopedia.com/terms/r/rfm-recency-frequency-monetary-value.asp (Accessed: 16 January 2022).

#### *PERCENT\_RANK (Transact-SQL) - SQL Server* (2021). Available at: https://docs.microsoft.com/en-us/sql/t-sql/functions/percent-rank-transact-sql?view=sql-server-ver15#:~:text=The%20PERCENT\_RANK%20function%20computes%20the,the%20rows%20in%20each%20partition. (Accessed: 16 January 2022).

#### *Using PIVOT and UNPIVOT - SQL Server* (2021). Available at: https://docs.microsoft.com/en-us/sql/t-sql/queries/from-using-pivot-and-unpivot?view=sql-server-ver15 (Accessed: 16 January 2022).