
KROTOS AUDIO



Krotos recruitment - Business Intelligence Analyst

Michael Papamikroulis

Athens, May 2025

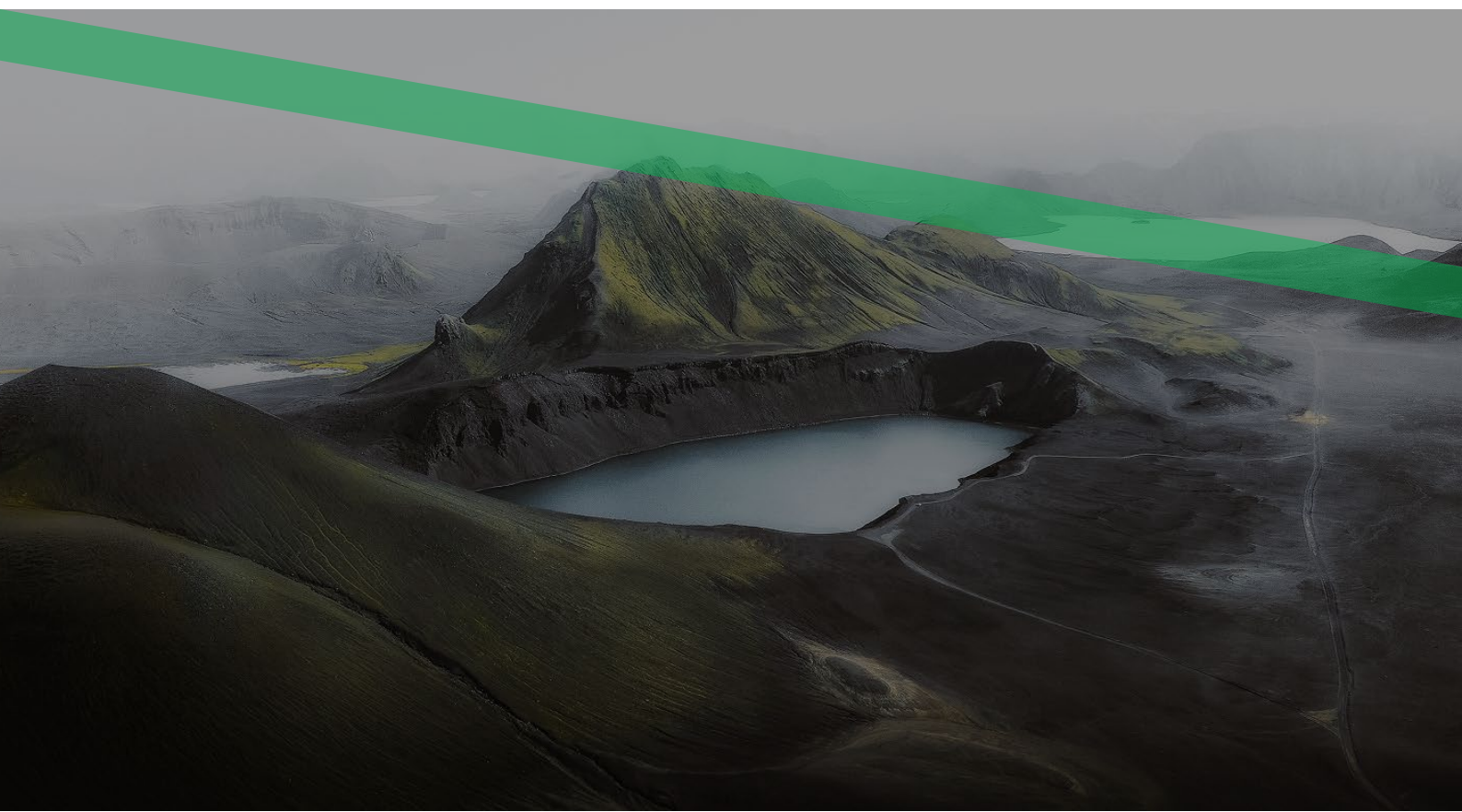


TABLE OF CONTENTS

1. INTRODUCTION.....	1
2. DATA BASE SETUP	2
3. ANALYSIS.....	4
3.1. Introduction.....	4
3.2. Customer metrics calculation (RFM).....	4
3.2.1. Recency	4
3.2.2. Frequency	5
3.2.3. Monetary	6
3.3. Segmentation Analysis.....	7
3.3.1. Recency segmentation.....	8
3.3.2. Frequency segmentation	9
3.3.3. Monetary segmentation	10
3.3.4. Final segmentation.....	11
4. POWER BI VISUALIZATION	14
4.1. Introduction.....	14
4.2. Pre-Work performed in Power BI	14
4.3. Customer Segments Dashboard.....	16
4.4. Acquisition Channel Effectiveness Dashboard.....	17
5. KEY FINDINGS - RECOMMENDATIONS	18

TABLE OF TABLES

Table 3-1: RFM Tier Classification (1–5 Scoring)	8
Table 3-2: Recency buckets	8
Table 3-3: Frequency buckets.....	9
Table 3-4: Monetary buckets.....	10
Table 3-5: RFM Segmentation Summary	11

TABLE OF FIGURES

Figure 2-1: PostgreSQL schema setup and verification queries for users and transactions tables.....	3
Figure 3-1: SQL code and output for Recency calculation — grouped by user_id, returning one row per customer.....	5
Figure 3-2: SQL code and output for Frequency calculation — grouped by user_id, returning one row per customer.....	6
Figure 3-3: SQL code and output for Monetary calculation — grouped by user_id, returning one row per customer.....	7

Figure 3-4: Final SQL Query for RFM Scoring and Customer Segmentation.....	13
Figure 4-1: Customer Segments Dashboard (RFM Analysis).....	16
Figure 4-2: Acquisition Channel Effectiveness Dashboard.....	17

1. INTRODUCTION

The present document includes the relevant documentation, steps and insights regarding the Data Analysis Case Study (Technical Assessment) for KROTOS recruitment process for the Business Intelligence Analyst role.

The case study overview is as follows:

"You are a data analyst for a retail chain with stores in various locations. The marketing team wants to implement more targeted campaigns based on customer behavior and value. Your task is to perform an RFM (Recency, Frequency, Monetary) analysis to segment customers and create an interactive dashboard to visualize the findings."

The data for this assessment include a document (.doc) file describing the tasks, and two tables containing the relevant data. In particular, there are two tables, Transactions and Users, which I downloaded in two respective csv documents. Those two files will be used later to import the values to each table, created using SQL.

Transactions table contains the most important data for the transactions, such as orders id, dates, users' information, the amount spent and region data. On the other hand, Users table contain apart from the users id, the marketing or acquisition channel through which the user first discovered or interacted with the product before making a purchase.

The main tasks of the case study include:

- 1) Database Setup
- 2) Analysis
- 3) Visualization

Finally, the deliverables of the assessment will include:

- Database
 - Screenshots or text
- SQL
 - Customer metrics calculation
 - Segmentation queries
- Dashboard
 - Interactive visualizations
 - Filtering capabilities
 - Segment performance metrics
- Presentation
 - Brief explanation of approach
 - Key findings
 - Business recommendations

2. DATA BASE SETUP

For the data base setup, I decided to work using Local PostgreSQL since I don't have direct data base setup experience, especially with Docker Desktop. Please note that I only have hands-on experience with MySQL however I know that it has similarities with other SQL types.

Furthermore, to setup the data base I installed PostgreSQL from the official website (<https://www.postgresql.org/download/>) and I set a local PostgreSQL server.

Then I created the data base using pgAdmin 4 by selecting **Create → Database**.

Then, I created the appropriate Schema by creating two tables named users and transactions with correct data types and relationships. To create the tables, I wrote the following schema code in the query tool.

```
CREATE TABLE users (  
    user_id BIGINT PRIMARY KEY,  
    user_first_transaction_channel TEXT  
);  
  
CREATE TABLE transactions (  
    order_id BIGINT PRIMARY KEY,  
    transaction_created_date DATE,  
    user_id BIGINT REFERENCES users(user_id),  
    net_amount NUMERIC(12,2),  
    email TEXT,  
    store_name TEXT,  
    city TEXT,  
    country TEXT  
);
```

Once creating the tables' structure, I imported the two (2) csv files containing each table values to the corresponding tables (users and transactions). From Krotos database that I had made, in Schemas, public, tables, I selected each table then I right clicked to choose Import/Export Data. There, I chose the relevant csv file, selected UTF8 for the encoding and I ensured from the options menu that Header is on and the Delimiter of the file is comma (,).

Finally, I run a few SQL codes to ensure that the values of the tables have been imported properly. Based on that, Schema aligns with both the data structure and column descriptions from the case study.

The following figure provides a screenshot from pgAdmin 4 and shows the Schema setup along with the verification queries for users and transactions tables.

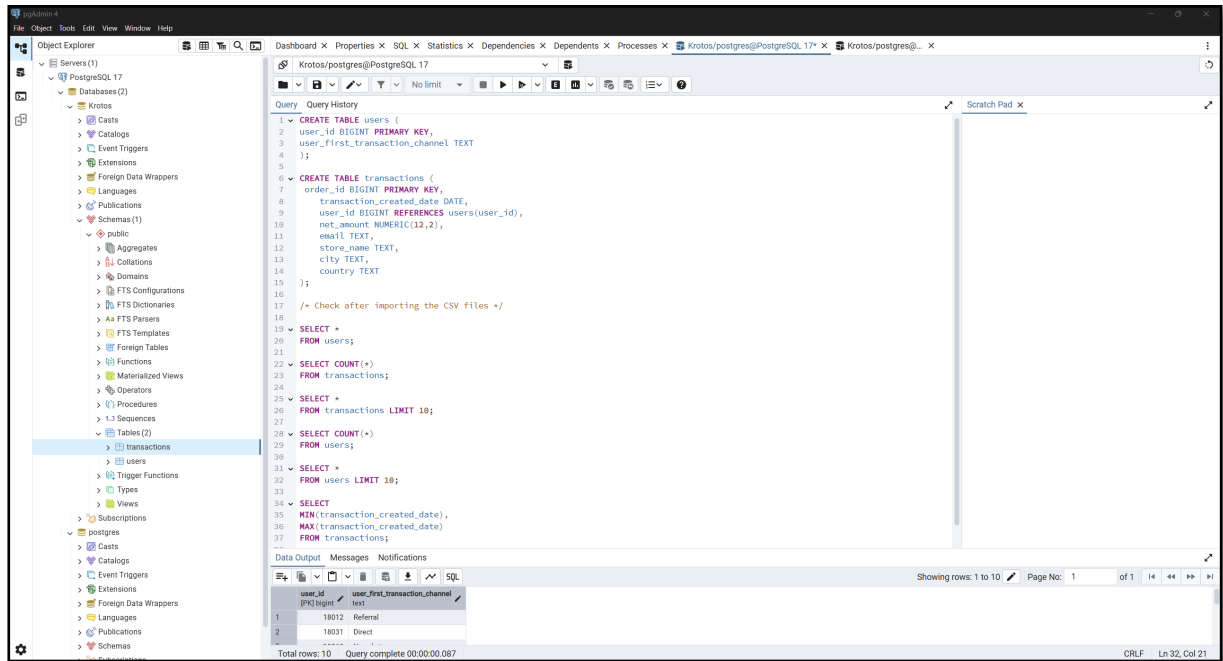


Figure 2-1: PostgreSQL schema setup and verification queries for users and transactions tables

3. ANALYSIS

3.1. INTRODUCTION

The objective of this analysis is to extract meaningful customer behavior insights from the transaction history using the RFM (Recency, Frequency, Monetary) methodology. RFM analysis is a widely used marketing technique to segment customers based on how recently they made a purchase (Recency), how often they purchase (Frequency), and how much they spend (Monetary). This approach allows businesses to identify high-value customers, re-engage lapsed buyers, and tailor marketing strategies to specific segments.

The following steps outline the calculation of RFM metrics, the scoring of customers, and the creation of customer segments based on these scores. The analysis is performed using SQL within the PostgreSQL environment, and the results will later be visualized using Power BI.

All SQL queries used throughout this project, including table creation, data import, metric calculation, RFM scoring, and customer segmentation, are included in the file `krotos_rfm_analysis.sql` provided in the deliverables.

3.2. CUSTOMER METRICS CALCULATION (RFM)

3.2.1. Recency

Recency is calculated as the number of days since each customer's most recent transaction. Rather than using the current system date, the maximum transaction date in the dataset is used as a consistent reference point to ensure reproducibility of results.

I used a Common Table Expression (CTE) to fetch the last purchase date per customer (MAX date) and then using `CURRENT_DATE` (which is equivalent to `CURDATE()` in MySQL) I calculated the difference in days between current date and the last purchase date.

The code is as follows:

```
WITH last_purchase_per_customer AS (  
    SELECT user_id, MAX(transaction_created_date) AS last_purchase_date  
    FROM transactions  
    GROUP BY user_id  
)  
SELECT  
user_id,
```

```
(CURRENT_DATE - last_purchase_date) AS recency_days  
FROM last_purchase_per_customer;
```

The following figure provides a screenshot from pgAdmin 4 showing the above query written and the data output.

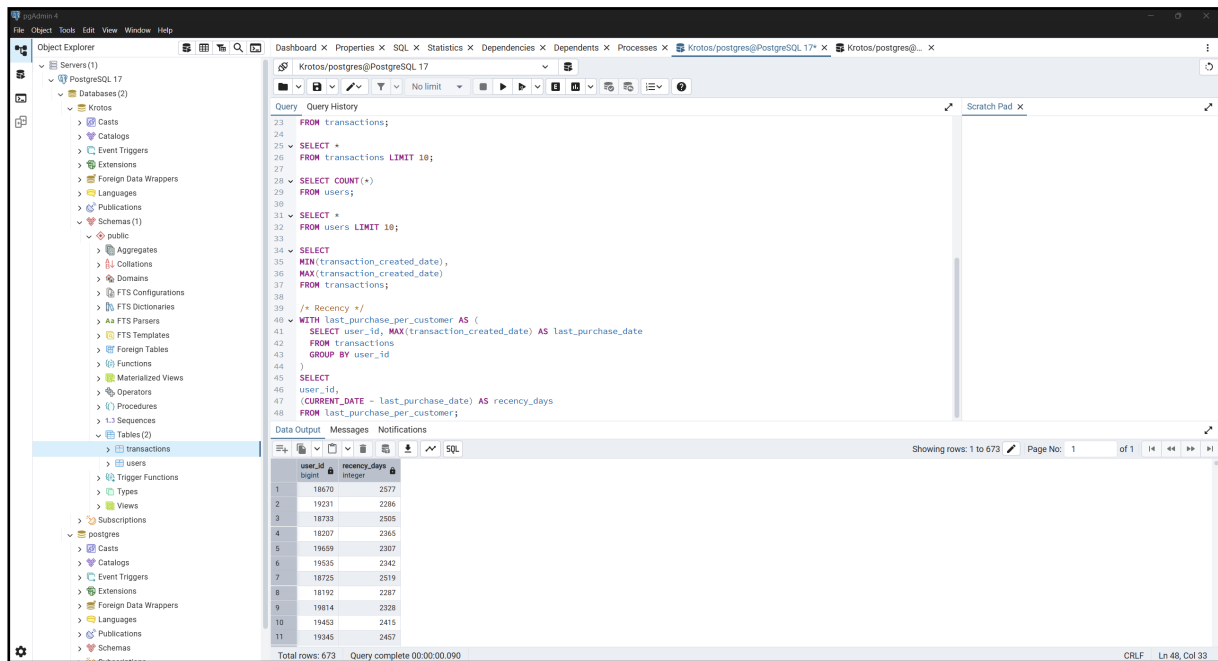


Figure 3-1: SQL code and output for Recency calculation — grouped by user_id, returning one row per customer

The query output returned 673 rows, which matches the number of distinct users in the users table. This confirms that each user has at least one recorded transaction, and validates that the grouping by user_id is correctly aligned with the dataset's structure.

Finally, since the dataset ends in 2018, most recency values exceed 2000 days, indicating that this is historical data. The logic remains valid and would work identically for up-to-date datasets.

3.2.2. Frequency

Frequency stands for how often a customer makes a purchase. Therefore, was calculated by grouping transactions by user_id and counting the number of unique order_ids per user. This gives a measure of how many times each customer placed an order, which reflects their engagement level with the brand.

The code I wrote is as follows:

```
SELECT user_id, COUNT(order_id) AS frequency
```



```
FROM transactions
GROUP BY user_id
ORDER BY frequency DESC;
```

Since order_id is already a unique purchase, I used the simple COUNT instead of COUNT(DISTINCT).

The following figure provides a screenshot from pgAdmin 4 showing the above query written and the data output for frequency calculation.

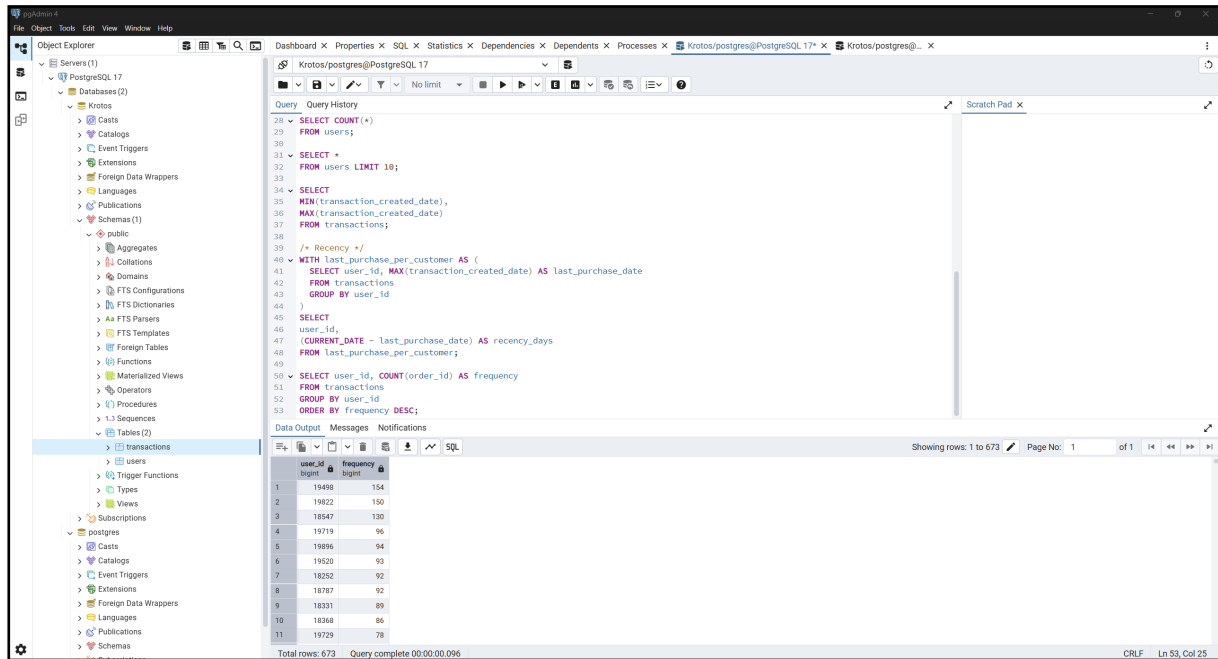


Figure 3-2: SQL code and output for Frequency calculation — grouped by user_id, returning one row per customer

3.2.3. Monetary

Monetary is a value showing how much money a customer spends on purchases. To calculate this value I sum up the net_amount using SUM() from the transactions table, and I grouped by user_id to find the amount spent by each customer.

The code I wrote is as follows:

```
SELECT user_id, SUM(net_amount) AS monetary
FROM transactions
GROUP BY user_id
ORDER BY monetary DESC;
```

The following figure provides a screenshot from pgAdmin 4 showing the above query written and the data output for monetary calculation.

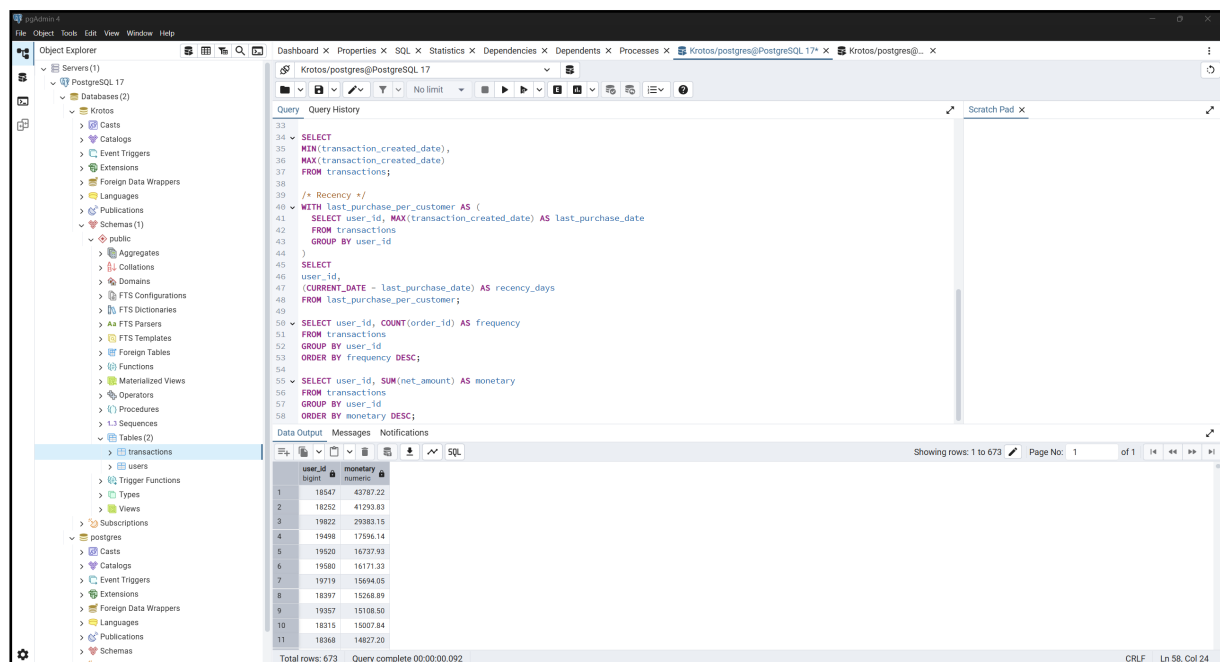


Figure 3-3: SQL code and output for Monetary calculation — grouped by user_id, returning one row per customer

3.3. SEGMENTATION ANALYSIS

In this section I conduct segmentation analysis based on RFM metrics calculated in the previous section. RMF segmentation is a customer analysis methodology that breaks down the customer base into tiered groups based on the three key behavioral dimensions:

- Recency (R): How recently a customer made a purchase (fewer days = better)
- Frequency (F): How often a customer makes purchases (more orders = better)
- Monetary (M): How much the customer spends (higher amount = better)

For my methodology I decided to assign each customer a score from 1 to 5 for each of the three RFM dimensions. The score reflects the customer's standing relative to the rest of the customer base, using quintile-based ranking. For example, a customer in R-Tier-5 made a very recent purchase, while one in F-Tier-1 made only a single purchase in the entire period. This enables targeted marketing actions and personalized engagement strategies.

Table 3-1: RFM Tier Classification (1–5 Scoring)

Recency (R)	Frequency (F)	Monetary (M)
R-Tier-5 (most recent)	F-Tier-5 (very frequent buyers)	M-Tier-5 (top spenders)
R-Tier-4	F-Tier-4	M-Tier-4
R-Tier-3	F-Tier-3	M-Tier-3
R-Tier-2	F-Tier-2	M-Tier-2
R-Tier-1 (least recent buyers)	F-Tier-1 (only 1–2 purchases)	M-Tier-1 (lowest spenders)

Please note that Recency is reversed since a lower recency value is better because it means that a customer bought recently (most recent). On the other hand, Frequency and Monetary are ascending because a higher value is better.

3.3.1. Recency segmentation

Based on the query output in **Section 3.2.1**, the range of recency_days is as follows:

- Minimum (most recent): 2277 days ago
- Maximum (least recent): 2641 days ago

Therefore, this gives a total spread of 364 days. In order to break that into 5 equal-sized tiers for scoring, I divided the 364-day range into 5 quantile buckets manually.

$$\text{Tier width} \approx \frac{364}{5} \approx 72.8 \approx 73$$

That means that each Recency score level covers 73 days of recency. Therefore, the buckets are shown in the following table.

Table 3-2: Recency buckets

R score	Range	Explanation
5	2277 – 2350	2277 + 73 = 2350
4	2351 – 2423	2350 + 1 = 2351 (then add 73)
3	2424 – 2496	2423 + 1 = 2424 (then add 73)
2	2497 – 2569	2496 + 1 = 2497 (then add 73)
1	2570 – 2641	2569 + 1 = 2570 (max is 261 which ends the range)

In order to interpret the above into SQL code, I used CASE WHEN syntax as follows:

CASE

WHEN recency_days <= 2350 THEN 5
WHEN recency_days <= 2423 THEN 4

```
    WHEN recency_days <= 2496 THEN 3
    WHEN recency_days <= 2569 THEN 2
    ELSE 1
END AS r_score
```

3.3.2. Frequency segmentation

In order to build the buckets for frequency, I wrote again the query from **Section 3.2.2** as follows:

```
SELECT MIN(frequency), MAX(frequency)
FROM (
    SELECT user_id, COUNT(order_id) AS frequency
    FROM transactions
    GROUP BY user_id
);
```

This showed me that the minimum frequency is 1 and the maximum 154. Therefore, the range is 153 and divided by 5 gives the frequency points.

$$\text{Tier width} \approx \frac{153}{5} \approx 30.6 \approx 31$$

Based on the tier classification provided on **Table 3-1**, the tier ranges for frequency are shown in the following table.

Table 3-3: Frequency buckets

F score	Range	Description
1	1 – 32	Low frequency
2	33 – 64	Below average
3	65 – 96	Moderate frequency
4	97 – 128	High frequency
5	129 – 154	Very high frequency

In order to interpret the above into SQL code, I used CASE WHEN syntax as follows:

```
CASE
    WHEN frequency <= 32 THEN 1
    WHEN frequency <= 64 THEN 2
    WHEN frequency <= 96 THEN 3
    WHEN frequency <= 128 THEN 4
    ELSE 5
END AS f_score
```

3.3.3. Monetary segmentation

In order to build the buckets for frequency, I wrote again the query from **Section 3.2.3** as follows:

```
SELECT MIN(monetary), MAX(monetary)
FROM (
    SELECT user_id, SUM(net_amount) AS monetary
    FROM transactions
    GROUP BY user_id
);
```

This showed me that the minimum monetary is 13.47 and the maximum 43787.22. Therefore, the range is approximately 43773.75 and divided by 5 gives the frequency points.

$$\text{Tier width} \approx \frac{43773.75}{5} \approx 8754.75$$

That means that each tier spans about 8755 in total spend.

Based on the tier classification provided on **Table 3-1**, the tier ranges for frequency are shown in the following table.

Table 3-4: Monetary buckets

M score	Range	Description
1	13.47 - 8768.22	Low frequency
2	8768.23 - 17522.97	Below average
3	17522.98 - 26277.72	Moderate frequency
4	26277.73 - 35032.47	High frequency
5	35032.47 - 43787.22	Very high frequency

In order to interpret the above into SQL code, I used CASE WHEN syntax as follows:

```
CASE
    WHEN monetary <= 8768.22 THEN 1
    WHEN monetary <= 17522.97 THEN 2
    WHEN monetary <= 26277.72 THEN 3
    WHEN monetary <= 35032.47 THEN 4
    ELSE 5
END AS m_score
```

3.3.4. Final segmentation

After scoring each customer from 1 to 5 on Recency, Frequency, and Monetary dimensions, I combined these into an RFM score. Based on these scores, I assigned each customer to a behavior-based segment.

For my methodology I decided to label the customers based on their RFM score as shown in the following table.

Table 3-5: RFM Segmentation Summary

Label	RFM Score Pattern	Description
Top	R=5, F=5, M≥4	Very recent, frequent, and high-spending customers. Most valuable.
Loyal	R≥4, F≥4	Consistently active and engaged. Strong repeat buyers.
Fading	R≤2, F≤2	Low recent activity and frequency. At risk of churn.
New Customer	R=5, F=1	Just made their first purchase. Opportunity for nurturing.
Other	All other combinations	Mixed behavior. Moderate or average customers.

To build the final segmentation, I first created a Common Table Expression (CTE) named `rfm_base` that brings together the core customer metrics calculated in the previous sections:

- Recency, calculated as the number of days since the customer's most recent transaction (**Section 3.2.1**)
- Frequency, measured as the total number of transactions made by each customer (**Section 3.2.2**)
- Monetary, calculated as the total amount spent by each customer (**Section 3.2.3**)

These metrics were then used to assign individual scores on a scale of 1 to 5 for each RFM dimension based on manually defined value ranges. The RFM scores were combined into a 3-digit code (e.g., 545), and each customer was labeled with a behavioral segment such as Top, Loyal, or Fading based on their score profile.

The CTE named `rfm_base` was built with the following SQL code:

```
WITH rfm_base AS (  
    SELECT user_id,  
           CURRENT_DATE - MAX(transaction_created_date) AS recency_days,  
           COUNT(order_id) AS frequency,  
           SUM(net_amount) AS monetary  
    FROM transactions
```

```
GROUP BY user_id  
)
```

Then, I made a second CTE named rfm_score to apply the scoring rules and assign segment labels based on predefined thresholds.

The CTE named rfm_score was built with the following SQL code:

```
rfm_scored AS (  
  SELECT *,  
  CASE  
    WHEN recency_days <= 2350 THEN 5  
    WHEN recency_days <= 2423 THEN 4  
    WHEN recency_days <= 2496 THEN 3  
    WHEN recency_days <= 2569 THEN 2  
    ELSE 1  
  END AS r_score,  
  CASE  
    WHEN frequency <= 32 THEN 1  
    WHEN frequency <= 64 THEN 2  
    WHEN frequency <= 96 THEN 3  
    WHEN frequency <= 128 THEN 4  
    ELSE 5  
  END AS f_score,  
  CASE  
    WHEN monetary <= 8768.22 THEN 1  
    WHEN monetary <= 17522.97 THEN 2  
    WHEN monetary <= 26277.72 THEN 3  
    WHEN monetary <= 35032.47 THEN 4  
    ELSE 5  
  END AS m_score  
  FROM rfm_base  
)
```

Finally, I made a query that combines the RFM scores into a 3-digit code and each customer was labeled respectively.

The code is as follows:

```
SELECT *,  
CONCAT(r_score, f_score, m_score) AS rfm_score,  
CASE  
  WHEN r_score = 5 AND f_score = 5 AND m_score >= 4 THEN 'Top'  
  WHEN r_score >= 4 AND f_score >= 4 THEN 'Loyal'
```

```

WHEN r_score <= 2 AND f_score <= 2 THEN 'Fading'
WHEN r_score = 5 AND f_score = 1 THEN 'New Customer'
ELSE 'Other'

```

```
END AS segment
```

```
FROM rfm_scored;
```

The final SQL query, which combines the logic from the previous steps, is shown in the figure below along with a preview of the resulting customer segmentation output.

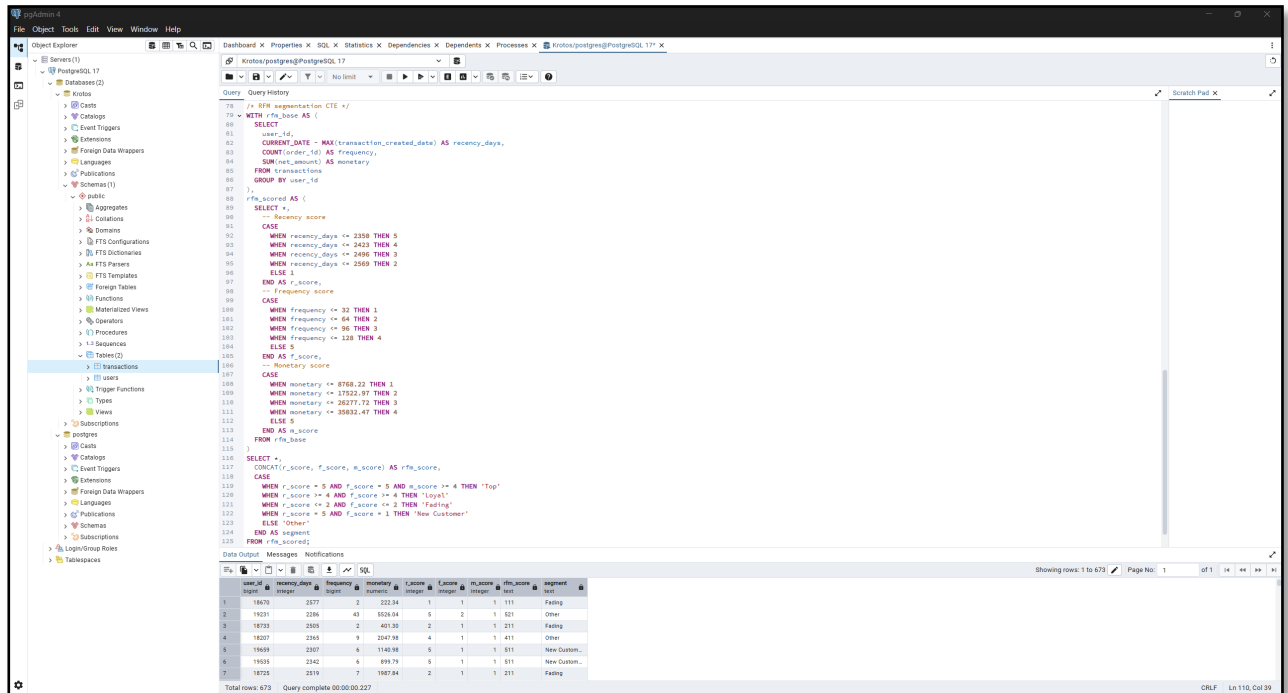


Figure 3-4: Final SQL Query for RFM Scoring and Customer Segmentation

The final RFM segmentation table was exported as `rfm_segmentation_output.csv` and used as the main dataset in Power BI (Section 4). This file contains all customer scoring and segmentation logic from Task 2, enabling clean visual analysis without duplicating business logic inside Power BI.

4. POWER BI VISUALIZATION

4.1. INTRODUCTION

The purpose of this section is to present interactive dashboards developed in Microsoft Power BI to fulfill the requirements outlined in Task 3 of the case study. These dashboards are designed to provide insights into customer behavior and acquisition effectiveness using the RFM segmentation results generated in the previous task.

Based on the provided case study the visualization will include Customer segments dashboard (RFM) and Acquisition channel effectiveness. Furthermore, the deliverable will cover a) Interactive visualizations, b) Filtering capabilities and c) Segment performance metrics.

The data files that will be loaded into Power BI concern `rfm_segmentation_output.csv`, which contains pre-processed customer metrics, RFM scores, and assigned segment labels, and was created using SQL (see **Section 3.3.4**), along with the file `Users.csv` provided from Krotos for this assessment and includes user-level metadata, such as acquisition channels, that enables analysis of marketing performance.

4.2. PRE-WORK PERFORMED IN POWER BI

Before creating the charts, some pre-work was performed in Power BI to ensure consistency, accurate data relationships and a smooth visualization experience. Pre-work includes the following:

- Loaded `rfm_segmentation_output.csv` and `Users.csv` into Power BI.
- Verified and adjusted data types and confirmed that numeric fields interpreted decimal points correctly based on locale using Power Query.
- Renamed tables and columns via Power Query for clarity and easier reference during visualization.
- Created relationship between both datasets in the Data Model.
- Built calculated measures for KPIs/Cards using Data Analysis Expressions (DAX).
- Added slicers for dynamic filtering by segment and acquisition channel.

Furthermore, due to the use of dot (.) as the decimal separator in the monetary field, I configured Power BI to use the US locale to ensure correct interpretation of numeric values. Since the dataset does not specify a currency, and for consistency with the system locale, all monetary values are displayed using dollar (\$) formatting by default.

To fulfill the requirements outlined in Task 3 of the case study, I decided to create two (2) Dashboards named “Customer Segments Dashboard” focusing on visualizing RFM segments from `rfm_segmentation_output.csv`, and “Acquisition Channel Effectiveness” focusing on how well each channel from `Users.csv` performs.

I first loaded the Users.csv file into Power BI. For clarity and ease of use during visualization, I renamed the table to Users and the columns as follows:

- user_id → User
- user_first_transaction_channel → Transaction Channel

Columns' data types were correct so no other changes made there.

Then I loaded the rfm_segmentation_output.csv file and renamed the table to RFM for clarity. Column names were updated to improve readability within the Power BI interface as follows:

- user_id → User
- recency_days → Recency(days)
- frequency → Frequency
- monetary → Monetary
- r_score → Recency score
- f_score → Frequency score
- m_score → Monetary score
- rfm_score → RFM score
- segment → Segment

Additionally, the Monetary column was formatted as a fixed decimal number with dollar (\$) currency symbol to support visual clarity in spend-related charts.

Then in the Model view a one-to-one relationship was created between the Users and RFM tables using the User column (originally user_id). Cross-filter direction was set to Both to allow flexible filtering and interaction between visuals based on either table. Also, I selected User column from User table as the Primary Key.

In addition to data cleaning and relationship setup, several DAX measures were created to support key performance indicators (KPIs) and visual summaries in the dashboards. These measures were used to dynamically calculate totals, averages, and customer counts based on selected filters. The list of measures I created are as follows:

- ✓ Total Spend
Total Spend = SUM(RFM[Monetary])
- ✓ Average Spend per Customer
Average Spend = AVERAGE(RFM[Monetary])
- ✓ Average Frequency per Customer
Average Frequency = AVERAGE(RFM[Frequency])
- ✓ Customer Count
Total Customers = DISTINCTCOUNT(RFM[User])
- ✓ Distinct Acquisition Channels
Total Channels = DISTINCTCOUNT(Users[Transaction Channel])

Finally, I created a separate table named Measure Table to store all the calculated DAX measures listed above. This approach helps me always keep the data model organized and makes it easier for me to manage KPIs across multiple report pages.

4.3. CUSTOMER SEGMENTS DASHBOARD

The first dashboard provides an interactive overview of customer behavior based on RFM segmentation. It is designed to help identify how customers are distributed across behavioral segments and how each segment contributes to overall performance.

This dashboard includes the following:

- ✓ Four KPI cards for Total Customers, Total Spend, Average Spend and Average Frequency
- ✓ A column chart displaying the number of customers in each segment
- ✓ A matrix table showing segment-wise breakdown of total customers, Average Spend and Average Frequency
- ✓ One segment slicer allowing users to filter all visuals by selected customer segment

The interactive version of this dashboard is included in the deliverables as a Power BI (.pbix) file under the name “Michael Papamikroulis Krotos Assessment” and provided in the figure below.

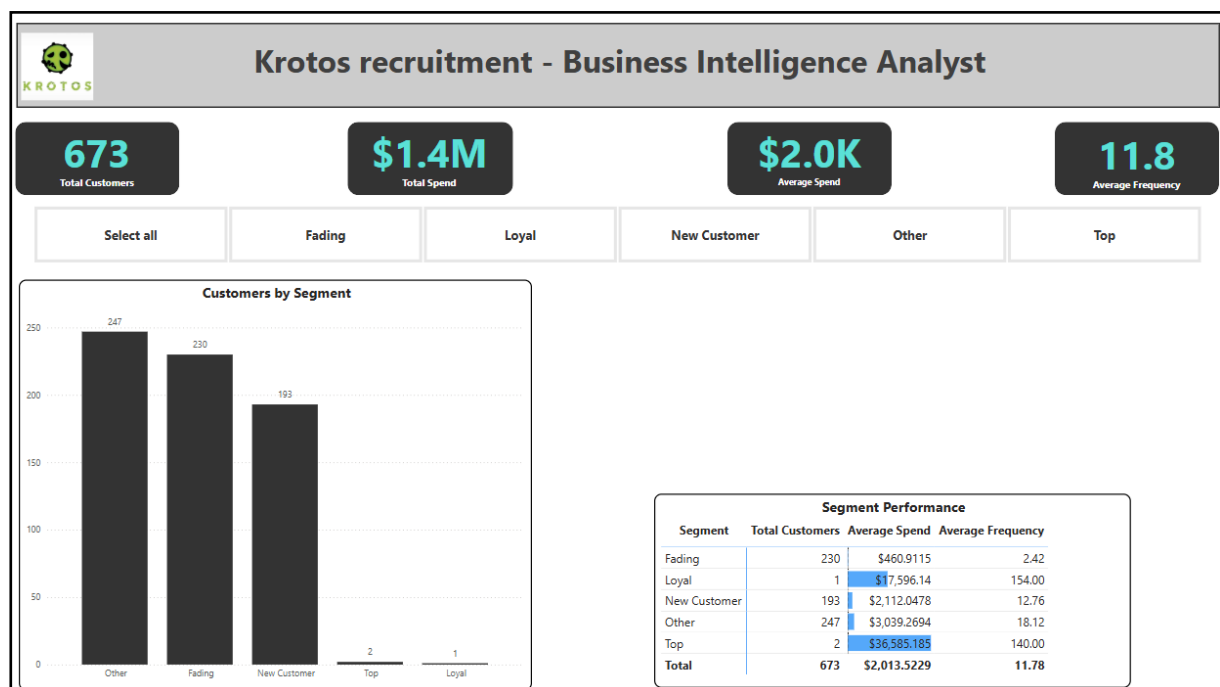


Figure 4-1: Customer Segments Dashboard (RFM Analysis)

Based on this dashboard the main insights are:

- The majority of customers fall into the "Other" and "Fading" segments, showing mixed or declining behavior.
- Only three customers qualify as "Top" or "Loyal" in total, though they show high spend and frequency.

4.4. ACQUISITION CHANNEL EFFECTIVENESS DASHBOARD

This dashboard evaluates the performance of acquisition channels by visualizing customer volume and segment distribution. The dashboard includes:

- ✓ One KPI card for Total Channels
- ✓ One Top-N card showing the channel with the highest average spend
- ✓ A column chart showing customers per acquisition channel
- ✓ A clustered column chart showing segment distribution per channel

The interactive version of this dashboard is included in the deliverables as a Power BI (.pbix) file under the name “Michael Papamikroulis Krotos Assessment” and provided in the figure below.

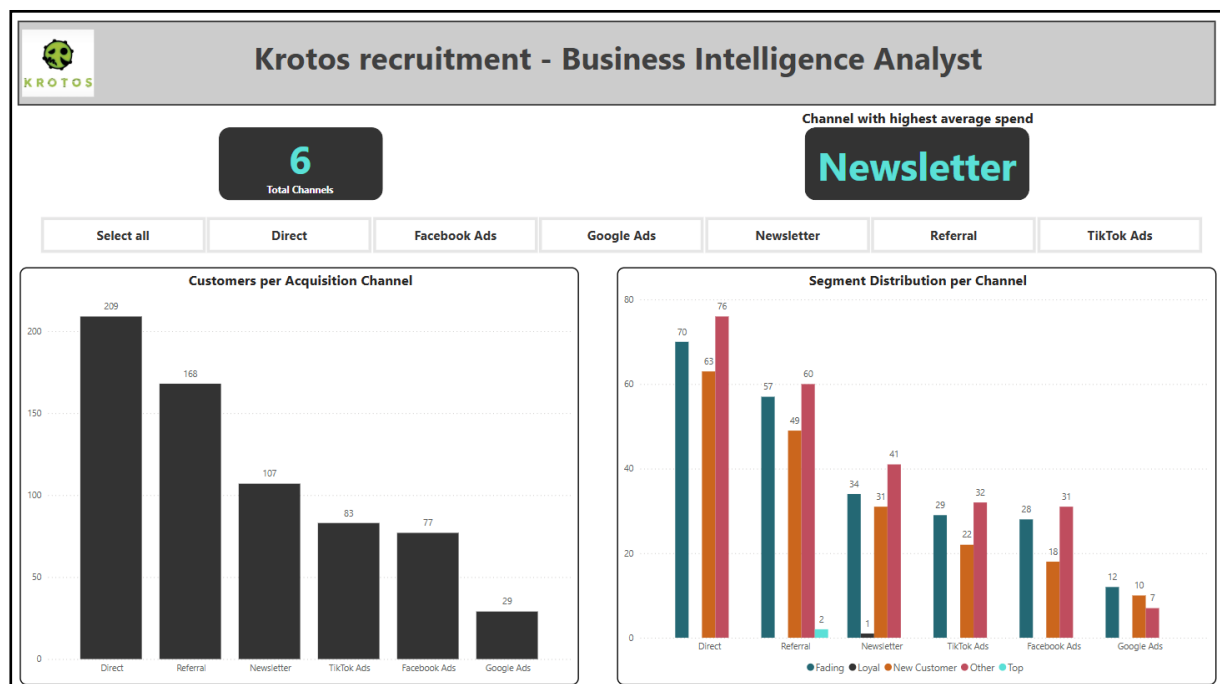


Figure 4-2: Acquisition Channel Effectiveness Dashboard

Based on this dashboard the main insights are:

- Out of six (6) channels Newsletter is the one with the highest average spend (Monetary)
- Customers label as “Other” are the majority in every channel
- The majority of customers fall into the Direct channel

5. KEY FINDINGS - RECOMMENDATIONS

This case study, as part of Krotos recruitment process, aimed to analyze customer behavior and acquisition channel effectiveness using the Recency, Frequency, Monetary (RFM) segmentation methodology and visual exploration through Power BI.

The analysis was conducted by importing customer and transaction data into a PostgreSQL database, where customer level RFM metrics were calculated using SQL. These metrics were then exported and visualized in Power BI, where interactive dashboards were developed to uncover key insights about customer segments and marketing channel performance.

The process included structured data modeling, creation of reusable DAX measures, relationship setup between datasets, and the development of filtering and interactivity features to support exploratory analysis.

Based on Power BI visualization the key findings are as follows:

- The majority of customers fall into the "Other" and "Fading" segments, indicating that most users do not show strong engagement or high value.
- Only three customers qualify as "Top" or "Loyal" in total, though they show high spend and frequency.
- The "Other" segment dominates across all channels, suggesting a large base of mixed behavior users.
- Among all acquisition channels, Newsletter brings in customers with the highest average spend, despite being 3rd in volume.
- Direct traffic accounts for the highest number of customers, though many of these fall into the "Other" or "Fading" segments.
- Referral and TikTok Ads also show balanced performance across segments, while Google Ads shows low volume and low engagement.

Finally, the following recommendations are provided:

- **Consider re-segmenting the "Other" group.** This is the largest segment, but also the least defined. A deeper breakdown could help uncover subgroups worth targeting.
- **Focus on the Newsletter channel.** Despite not being the largest source of customers, it brings in users with the highest average spend. Consider whether this channel can be scaled further or used as a model for messaging strategies in other channels.
- **Prioritize engagement of Direct channel users.** The Direct channel contributes the most users but many fall into lower-value segments. These users could be targeted with loyalty offers, emails, or promotions to boost retention or activity.
- **Use the RFM segments in future marketing campaigns.** For example, rewarding Loyal customers or reactivating Fading ones.