PROJECT EXAM - Emotive

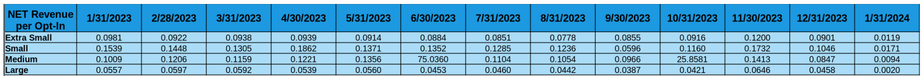
Name: Derrick Adrian Payas

Email: derykpayas@gmail.com

Date: March 16, 2024

Questions

1. Write an SQL script (or multiple scripts) to help answer the following: What is the Net Revenue per Opted In Count broken down by brand_size from Jan 2023 to Jan 2024? Populate the table below.



NET Revenue per Opt-In	1/31/2023	2/28/2023	3/31/2023	4/30/2023	5/31/2023	6/30/2023	7/31/2023	8/31/2023	9/30/2023	10/31/2023	11/30/2023	12/31/2023	1/31/2024
Extra Small	0.0981	0.0922	0.0938	0.0939	0.0914	0.0884	0.0851	0.0778	0.0855	0.0916	0.1200	0.0901	0.0119
Small	0.1539	0.1448	0.1305	0.1862	0.1371	0.1352	0.1285	0.1236	0.0596	0.1160	0.1732	0.1046	0.0171
Medium	0.1009	0.1206	0.1159	0.1221	0.1356	75.0360	0.1104	0.1054	0.0966	25.8581	0.1413	0.0847	0.0094
Large	0.0557	0.0597	0.0592	0.0539	0.0560	0.0453	0.0460	0.0442	0.0387	0.0421	0.0646	0.0458	0.0020

Figure 1: img2

2. Please briefly outline the steps you took to answer the question (Include tools if any)

Steps for the extraction of Net Revenue per Opted In
a. Familiarize and explore datasets using Google Sheets. This includes identification of potential errors and data cleaning.

b. Initial data cleaning using Neovim (text editor) and Nvim-R (R-Studio extension for Neovim). Used tidyverse package to clean identified errors.

c. Automated migration of data to PostgreSQL using a Python script

d. Coducted Exploratory Data Analysis (EDA) in SQL to identify potential errors from the migration process.

e. Corrected errors that may potentially skew the results and updated existing tables in the database using SQL scripts.

f. Proceeded to data analysis using SQL script to extract Net Revenue per Opted In Count broken down by brand size from each month.

g. Data Validation through another SQL script just to check if values are matching and correct.

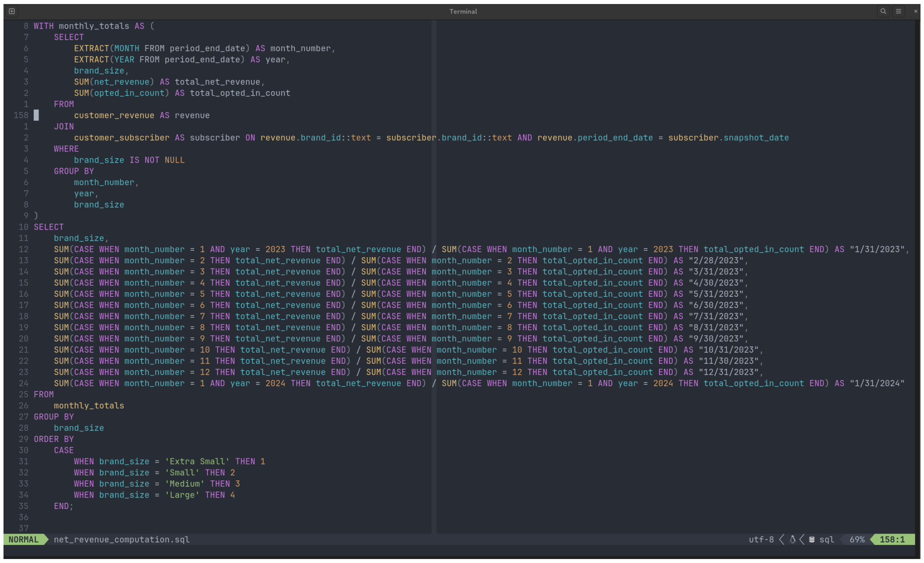
3. What data cleaning & correcting did you do, if any?

```
a. Dataset 1 (Customer Revenue)
    * brand_size
        replace 'Mediumm' to 'Medium'
        replace 'Smal' to 'Small'
    * brand_id
        replace 'National Foundation...' to '28671'
    * period_end_date
        converted data type to date

b. Dataset 2 (Customer Subscriber)
    * snapshot_date
        replace '2/29/2023' to '2/28/2023'
        converted data type to date
    * opted_in_count
        updated 0 values to (total_brand_subscriber - opted_out_count)
    * generalized naming convention of title headers for better coding
```

Check this link: https://github.com/paadde/revenue_analysis for the full data cleaning documentation. Documentations are good practices for data transparency.

4. Please paste or attach the SQL you wrote



```sql
WITH monthly_totals AS (
    SELECT
        EXTRACT(MONTH FROM period_end_date) AS month_number,
        EXTRACT(YEAR FROM period_end_date) AS year,
        brand_size,
        SUM(net_revenue) AS total_net_revenue,
        SUM(opted_in_count) AS total_opted_in_count
    FROM
        customer_revenue AS revenue
    JOIN
        customer_subscriber AS subscriber ON revenue.brand_id::text = subscriber.brand_id::text AND revenue.period_end_date = subscriber.snapshot_date
    WHERE
        brand_size IS NOT NULL
    GROUP BY
        month_number,
        year,
        brand_size
)
SELECT
    brand_size,
    SUM(CASE WHEN month_number = 1 AND year = 2023 THEN total_net_revenue END) / SUM(CASE WHEN month_number = 1 AND year = 2023 THEN total_opted_in_count END) AS "1/31/2023",
    SUM(CASE WHEN month_number = 2 THEN total_net_revenue END) / SUM(CASE WHEN month_number = 2 THEN total_opted_in_count END) AS "2/28/2023",
    SUM(CASE WHEN month_number = 3 THEN total_net_revenue END) / SUM(CASE WHEN month_number = 3 THEN total_opted_in_count END) AS "3/31/2023",
    SUM(CASE WHEN month_number = 4 THEN total_net_revenue END) / SUM(CASE WHEN month_number = 4 THEN total_opted_in_count END) AS "4/30/2023",
    SUM(CASE WHEN month_number = 5 THEN total_net_revenue END) / SUM(CASE WHEN month_number = 5 THEN total_opted_in_count END) AS "5/31/2023",
    SUM(CASE WHEN month_number = 6 THEN total_net_revenue END) / SUM(CASE WHEN month_number = 6 THEN total_opted_in_count END) AS "6/30/2023",
    SUM(CASE WHEN month_number = 7 THEN total_net_revenue END) / SUM(CASE WHEN month_number = 7 THEN total_opted_in_count END) AS "7/31/2023",
    SUM(CASE WHEN month_number = 8 THEN total_net_revenue END) / SUM(CASE WHEN month_number = 8 THEN total_opted_in_count END) AS "8/31/2023",
    SUM(CASE WHEN month_number = 9 THEN total_net_revenue END) / SUM(CASE WHEN month_number = 9 THEN total_opted_in_count END) AS "9/30/2023",
    SUM(CASE WHEN month_number = 10 THEN total_net_revenue END) / SUM(CASE WHEN month_number = 10 THEN total_opted_in_count END) AS "10/31/2023",
    SUM(CASE WHEN month_number = 11 THEN total_net_revenue END) / SUM(CASE WHEN month_number = 11 THEN total_opted_in_count END) AS "11/30/2023",
    SUM(CASE WHEN month_number = 12 THEN total_net_revenue END) / SUM(CASE WHEN month_number = 12 THEN total_opted_in_count END) AS "12/31/2023",
    SUM(CASE WHEN month_number = 1 AND year = 2024 THEN total_net_revenue END) / SUM(CASE WHEN month_number = 1 AND year = 2024 THEN total_opted_in_count END) AS "1/31/2024"
FROM
    monthly_totals
GROUP BY
    brand_size
ORDER BY
    CASE
        WHEN brand_size = 'Extra Small' THEN 1
        WHEN brand_size = 'Small' THEN 2
        WHEN brand_size = 'Medium' THEN 3
        WHEN brand_size = 'Large' THEN 4
    END;
```

Figure 2: img1

5. Given the SQL below, suggest improvements and highlight any issues you can identify:

a. Using Common Table Expression (CTE) might help in readability of the script

b. Use of capitalizations in SQL keywords, functions and clauses may also improve readability of this SQL script.

c. Make the ALIASES more readable. Using meaningful and descriptive aliases also improve readability and maintainability. It may be difficult for someone to understand the query if Aliases are set to 'x', 'y', and 'b'. Suggested ALIAS would be 'lines', 'invoices', and 'brands'.

d. Nested subquery in the LEFT JOIN clause can be improved for query optimization.

```
LEFT JOIN emotive_brands AS brands ON invoices.customer_id = brands.stripe_id
```

e. While grouping columns in positional notation (GROUP BY 1, 2, 3, 4) may be concise, it may pose readability issues and may be error-prone if columns were moved or added.