Documentation

DATABASE TABLES

Company: List of companies with their respective associate partner (if any) and the date they signed a contract with Treble.

Conversations: Record of conversations grouped by day, account and whether they were successful or not.

Stripe Invoice: Invoices sent to companies. Each company can have multiple accounts in Stripe and the company_id column represents the internal ID that Stripe creates for each account.

Company Identifiers: Provides the association between the Company, Conversations and Stripe Invoice tables. It indicates for each company what the account IDs are in Treble and Stripe.

TASK #1:

The following are the logical and manageable steps that I will use to solve the problem efficiently:

a. Filtering Companies: I will filter the companies that were brought by the alliance team. I will use the "company" table for this step.

b. Activation Detection: For each company it is necessary to review the "conversations" table to identify the activation date, that is, when they have 350 or more conversations in any period of three consecutive days.

c. Count of Successful Conversations: With the activation date I proceed to count the successful conversations from that date until the end of a two-month period.

d. Determine Success: This step consists of determining which companies are successful, i.e. have 500 successful conversations within two months of activation.

e. Visualization: Finally with the data obtained, plot in a linear fashion how many companies became successful, week by week in the year 2023.

TASK #2

a. Definition of "recently closed company":

A company that closed within the last 2 months. For example, taking October 2023 as a reference, recently closed companies are those that closed in September and October 2023.

b. Definition of "successful company":

A company is successful in a given month if it had more than 1500 successful conversations in that month.

c. Procedure:

a. Identify all companies that closed in the last 2 months.

b. Of those companies, determine how many had more than 1500 successful conversations in the reference month.

c. Calculate the percentage of successful companies: (Number of successful companies / Total number of recently closed companies) * 100.

TASK #3

Obtain the necessary data

Perform a JOIN operation between tables to get the information you need. The important connections are:

company table with close_date

stripe_invoice with sent_date and amount

company_identifiers to establish the connections between company and stripe_invoice

Step 3: Data processing

For each company, it is necessary to

Determine the cohort (based on close_date).

Track the revenue collected each month after the close date.

Step 4: Plot the data

Using matplotlib, create a heat map similar to the given graph.

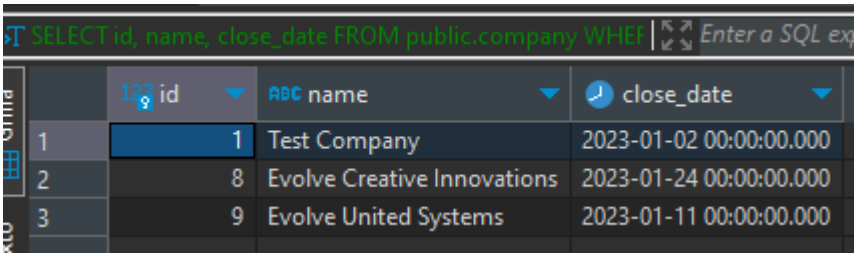
Observations of this task:

Following the pattern and logic of the proposed task we obtain data different from those shown in the reference graph, this can be corroborated as follows:

Let's verify how many companies closed in January 2023.

```
SELECT id, name, close_date
FROM public.company
WHERE EXTRACT(YEAR FROM close_date) = 2023 AND EXTRACT(MONTH FROM
close_date) = 1;
```

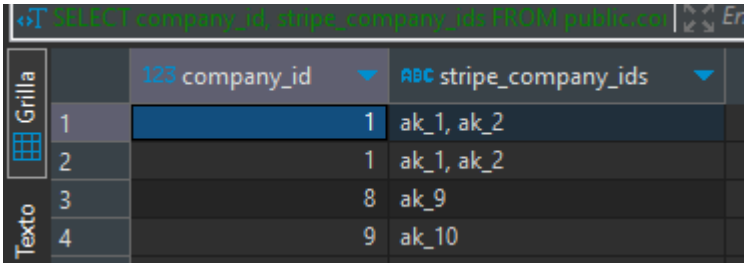
Results:



Once we know the companies that closed in January 2023, let's check how many of these companies have Stripe identifiers in public.company_identifiers.

```
SELECT company_id, stripe_company_ids
FROM public.company_identifiers
WHERE company_id IN (
    SELECT id
    FROM public.company
    WHERE EXTRACT(YEAR FROM close_date) = 2023 AND EXTRACT(MONTH FROM
close_date) = 1
);
```

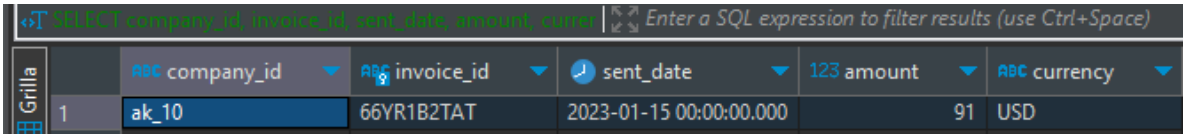
Results:



For companies with Stripe identifiers, we will check for invoices in public.stripe_invoice by January 2023.

```
SELECT company_id, invoice_id, sent_date, amount, currency
FROM public.stripe_invoice
WHERE company_id IN (
    SELECT stripe_company_ids
    FROM public.company_identifiers
    WHERE company_id IN (
        SELECT id
        FROM public.company
        WHERE EXTRACT(YEAR FROM close_date) = 2023 AND EXTRACT(MONTH FROM
close_date) = 1
    )
) AND EXTRACT(YEAR FROM sent_date) = 2023 AND EXTRACT(MONTH FROM
sent_date) = 1;
```

Results:



I limited the query to only one record and we can see that the invoice that we identified in the month of January closing is related to another invoice, inside the script 3 this function is called "fetch_monthly_data" and can be executed in the main function, for the development of the task to be the most similar to the reference graph I decided not to execute this function and bring the data from the reference graph from a dictionary with python (get_mock_revenue).

For any related questions feel free to contact me via email or whatsapp.