SOLVED SQL PROBLEMS [INTERVIEW QUESTIONS] OF MY 30 DAYS SQL PROBLEM SOLVING CHALLENG

#30dayofsql

Hi everyone, Hope you all fine. I took 30 days of SQL problem solving challenge. I completed it on 31st March 2023. I wish to share SQL problems and solution which I solved in this challenge.

This journey gave me a good experience and made me more familiar with windows functions, subqueries, CTE, date types and functions then as I know already aggregation function, where group order by clauses. Here, I gave solved SQL problems with solution and their respective problem links. Hope this will help you all to recall some of approaches you done with brush up your skills.

DAY -1

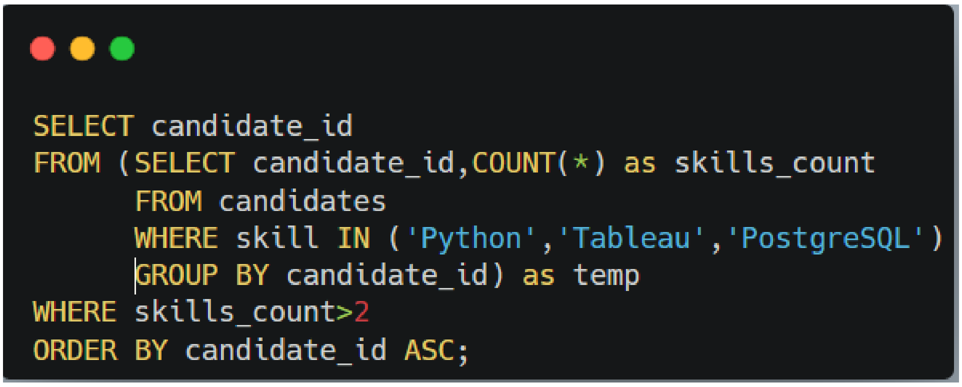
Problem 1 : Data Science Skills [LinkedIn SQL Interview Question]

Problem Statement:

Given a table of candidates and their skills, you're tasked with finding the candidates best suited for an open Data Science job. You want to find candidates who are proficient in Python, Tableau, and PostgreSQL.
Write a query to list the candidates who possess all of the required skills for the job. Sort the output by candidate ID in ascending order.
Assumption:
There are no duplicates in the candidates table.
candidates Table:
Column NameType
candidate_id int
egerskillvar char

Problem Link: https://lnkd.in/ggMwJhpJ

My Solution:



```
SELECT candidate_id
FROM (SELECT candidate_id,COUNT(*) as skills_count
      FROM candidates
      WHERE skill IN ('Python','Tableau','PostgreSQL')
      GROUP BY candidate_id) as temp
WHERE skills_count>2
ORDER BY candidate_id ASC;
```

DAY -2

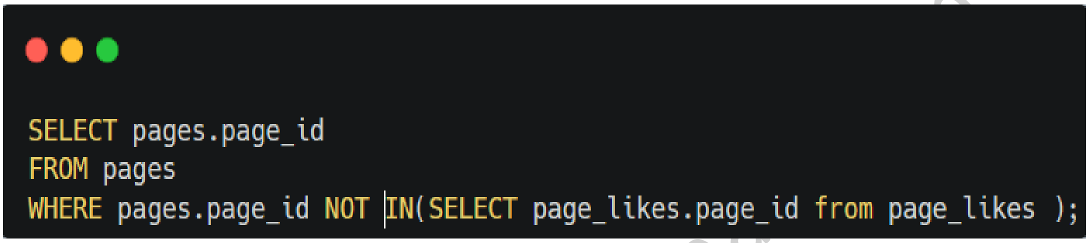
Problem 1 : Page With No Likes [Facebook SQL Interview Question]

Problem Statement:

Assume you are given the tables below about Facebook pages and page likes. Write a query to return the page IDs of all the Facebook pages that don't have any likes. The output should be in ascending order.

Problem Link: https://lnkd.in/gMYzTZHg

My Solution:



```sql
SELECT pages.page_id
FROM pages
WHERE pages.page_id NOT IN(SELECT page_likes.page_id from page_likes );
```

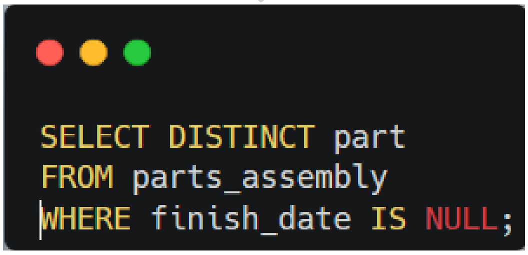
Problem 2 : Unfinished Parts [Tesla SQL Interview Question]

Problem Statement:

Tesla is investigating bottlenecks in their production, and they need your help to extract the relevant data. Write a query that determines which parts have begun the assembly process but are not yet finished.

Problem Link: https://lnkd.in/gYupaQRf

My Solution:



```sql
SELECT DISTINCT part
FROM parts_assembly
WHERE finish_date IS NULL;
```

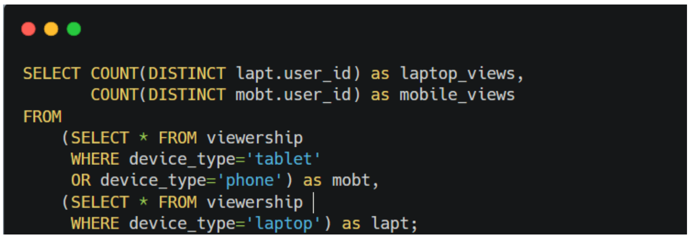
Problem 3 : Laptop vs. Mobile Viewership [New York Times SQL Interview Question]

Problem Statement:

Assume you are given the tables below about Facebook pages and page likes. Write a query to return the page IDs of all the Facebook pages that don't have any likes. The output should be in ascending order.

Problem Link: https://lnkd.in/ghGeHsgj

My Solution:



```
SELECT COUNT(DISTINCT lapt.user_id) as laptop_views,
       COUNT(DISTINCT mobt.user_id) as mobile_views
FROM
    (SELECT * FROM viewership
     WHERE device_type='tablet'
     OR device_type='phone') as mobt,
    (SELECT * FROM viewership
     WHERE device_type='laptop') as lapt;
```

DAY -3

Problem 1 : Duplicate Job Listings [Linkedin SQL Interview Question]

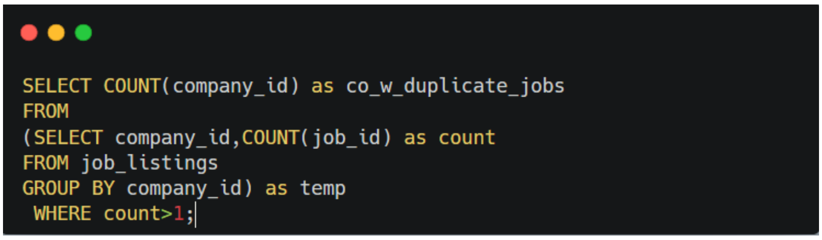
Problem Statement:

Assume you are given the table below that shows job postings for all companies on the LinkedIn platform. Write a query to get the number of companies that have posted duplicate job listings.

Clarification:
job listings refer to two jobs at the same company with the same title and description.

Problem Link: https://lnkd.in/gRs9un_a

My Solution:



```
SELECT COUNT(company_id) as co_w_duplicate_jobs
FROM
(SELECT company_id,COUNT(job_id) as count
FROM job_listings
GROUP BY company_id) as temp
 WHERE count>1;
```

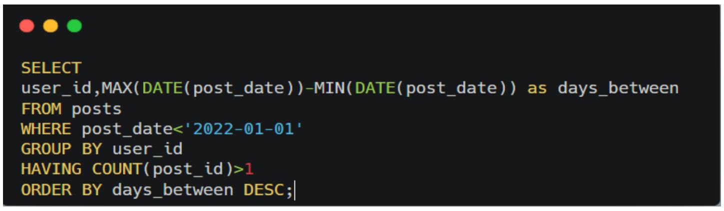
Problem 2 : Average Post Hiatus (Part 1) [Facebook SQL Interview Question]

Problem Statement:

Given a table of Facebook posts, for each user who posted at least twice in 2021, write a query to find the number of days between each user's first post of the year and last post of the year in the year 2021. Output the user and number of the days between each user's first and last post.

Problem Link: https://lnkd.in/gyHwfh4b

My Solution:



```
SELECT
user_id,MAX(DATE(post_date))-MIN(DATE(post_date)) as days_between
FROM posts
WHERE post_date<'2022-01-01'
GROUP BY user_id
HAVING COUNT(post_id)>1
ORDER BY days_between DESC;
```

Problem 3: Teams Power Users [Microsoft SQL Interview Question]

Problem Statement:

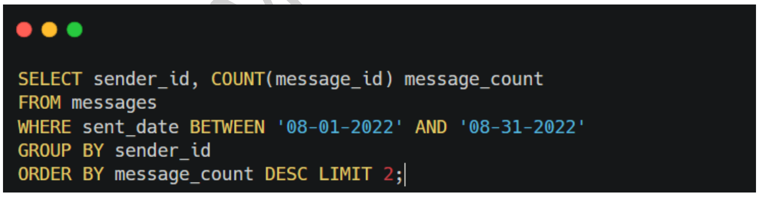
Write a query to find the top 2 power users who sent the most messages on Microsoft Teams in August 2022. Display the IDs of these 2 users along with the total number of messages they sent. Output the results in descending count of the messages.

Assumption:

No two users has sent the same number of messages in August 2022.

Problem Link: https://lnkd.in/gVbW2McP

My Solution:



```
SELECT sender_id, COUNT(message_id) message_count
FROM messages
WHERE sent_date BETWEEN '08-01-2022' AND '08-31-2022'
GROUP BY sender_id
ORDER BY message_count DESC LIMIT 2;
```

DAY -4

Problem 1 : Cities With Completed Trades [Robinhood SQL Interview Question]

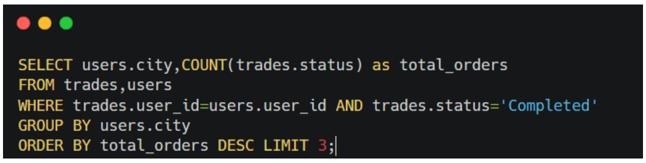
Problem Statement:

You are given the tables below containing information on Robinhood trades and users. Write a query to list the top three cities that have the most completed trade orders in descending order.

Output the city and number of orders.

Problem Link: https://lnkd.in/gCjq-Ttr

My Solution:



```
SELECT users.city,COUNT(trades.status) as total_orders
FROM trades,users
WHERE trades.user_id=users.user_id AND trades.status='Completed'
GROUP BY users.city
ORDER BY total_orders DESC LIMIT 3;
```

Problem 2 : Average Review Ratings [Amazon SQL Interview Question]

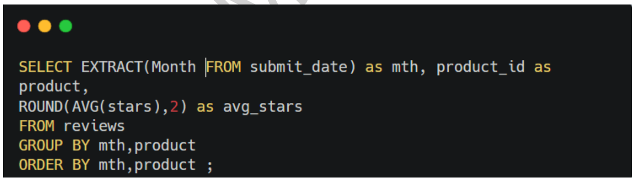
Problem Statement:

Given the reviews table, write a query to get the average stars for each product every month.

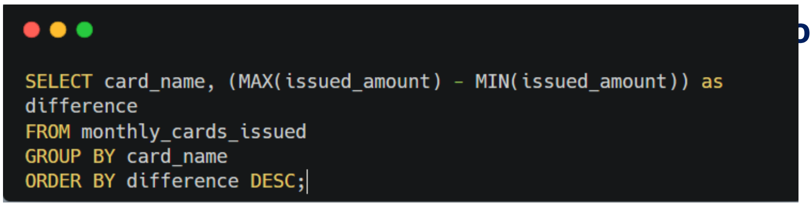
The output should include the month in numerical value, product id, and average star rating rounded to two decimal places. Sort the output based on month followed by the product id.

Problem Link: https://lnkd.in/gm6Km3HE

My Solution:



```
SELECT EXTRACT(Month FROM submit_date) as mth, product_id as
product,
ROUND(AVG(stars),2) as avg_stars
FROM reviews
GROUP BY mth,product
ORDER BY mth,product ;
```



```
SELECT card_name, (MAX(issued_amount) - MIN(issued_amount)) as
difference
FROM monthly_cards_issued
GROUP BY card_name
ORDER BY difference DESC;
```

ook SQL Interview

This is the same question as problem #1 in the SQL Chapter of Ace the Data Science Interview!

Assume you have an events table on app analytics. Write a query to get the app's click-through rate (CTR %) in 2022. Output the results in percentages rounded to 2 decimal places.

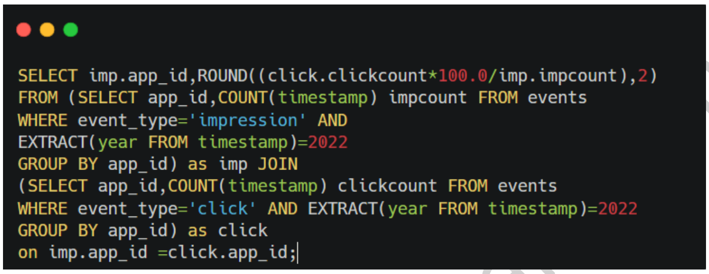
Notes:

Percentage of click-through rate = 100.0 * Number of clicks / Number of impressions

To avoid integer division, you should multiply the click-through rate by 100.0, not 100.

Problem Link: https://lnkd.in/g8VHmcqE

My Solution:



```
SELECT imp.app_id,ROUND((click.clickcount*100.0/imp.impcount),2)
FROM (SELECT app_id,COUNT(timestamp) impcount FROM events
WHERE event_type='impression' AND
EXTRACT(year FROM timestamp)=2022
GROUP BY app_id) as imp JOIN
(SELECT app_id,COUNT(timestamp) clickcount FROM events
WHERE event_type='click' AND EXTRACT(year FROM timestamp)=2022
GROUP BY app_id) as click
on imp.app_id =click.app_id;
```

DAY -5

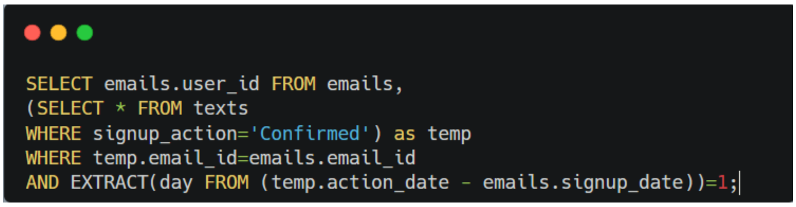
Problem 1: Second Day Confirmation [TikTok SQL Interview Question]

Problem Statement:

New TikTok users sign up with their emails and each user receives a text confirmation to activate their account. Assume you are given the below tables about emails and texts.
Write a query to display the ids of the users who did not confirm on the first day of sign-up, but confirmed on the second day.

Problem Link: https://lnkd.in/gQ542cwP

My Solution:



```
SELECT emails.user_id FROM emails,
(SELECT * FROM texts
WHERE signup_action='Confirmed') as temp
WHERE temp.email_id=emails.email_id
AND EXTRACT(day FROM (temp.action_date - emails.signup_date))=1;
```

Problem 2: Cards Issued Difference [JPMorgan Chase SQL Interview Question]

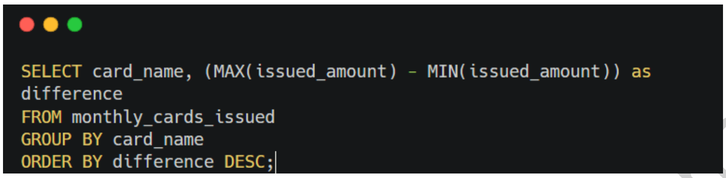
Problem Statement:

JPMorgan Chase is soon launching a new credit card, and to gain some context, you are analyzing how many credit cards were issued each month.

Write a query that outputs the name of each credit card and the difference in issued amount between the month with the most cards issued, and the least cards issued. Order the results according to the biggest difference.

Problem Link: https://lnkd.in/ga773a9T

My Solution:



```sql
SELECT card_name, (MAX(issued_amount) - MIN(issued_amount)) as
difference
FROM monthly_cards_issued
GROUP BY card_name
ORDER BY difference DESC;
```

Problem 3: Compressed Mean [Alibaba SQL Interview Question]

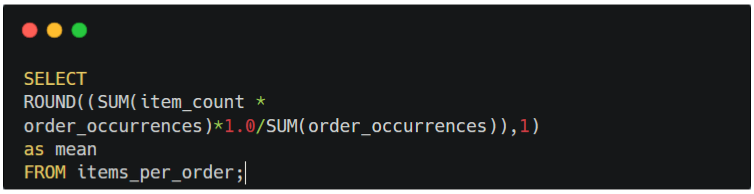
Problem Statement:

You are trying to find the mean number of items bought per order on Alibaba, rounded to 1 decimal place.

However, instead of doing analytics on all Alibaba orders, you have access to a summary table, which describes how many items were in an order (item_count), and the number of orders that had that many items (order_occurrences)

Problem Link: https://lnkd.in/gDN3quqz

My Solution:



```sql
SELECT
ROUND((SUM(item_count *
order_occurrences)*1.0/SUM(order_occurrences)),1)
as mean
FROM items_per_order;
```

DAY -6

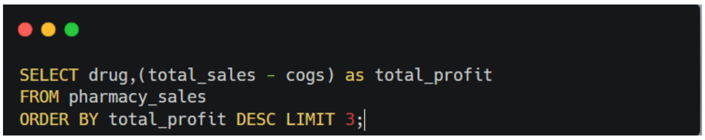
Problem 1: Pharmacy Analytics (Part 1) [CVS Health SQL Interview Question]

Problem Statement:

CVS Health
Write a query to find the top 3 most profitable drugs sold, and how much profit they made.
Assume that there are no ties in the profits. Display the result from the highest to the lowest total profit.

Problem Link: https://lnkd.in/gQQ_njDK

My Solution:



```
SELECT drug,(total_sales - cogs) as total_profit
FROM pharmacy_sales
ORDER BY total_profit DESC LIMIT 3;
```

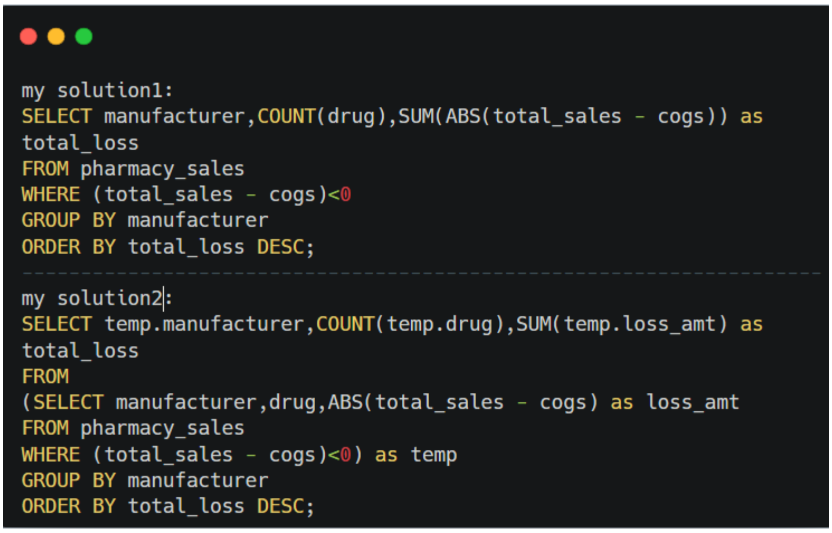
Problem 2: Pharmacy Analytics (Part 2) [CVS Health SQL Interview Question]

Problem Statement:

it is part 2 of problem 1
Write a query to find out which manufacturer is associated with the drugs that were not profitable and how much money CVS lost on these drugs.
Output the manufacturer, number of drugs and total losses. Total losses should be in absolute value. Display the results with the highest losses on top.

Problem Link: https://lnkd.in/gRQjMQ-P

My Solution:



```
my solution1:
SELECT manufacturer,COUNT(drug),SUM(ABS(total_sales - cogs)) as
total_loss
FROM pharmacy_sales
WHERE (total_sales - cogs)<0
GROUP BY manufacturer
ORDER BY total_loss DESC;
----------------------------------------------------------------------
my solution2:
SELECT temp.manufacturer,COUNT(temp.drug),SUM(temp.loss_amt) as
total_loss
FROM
(SELECT manufacturer,drug,ABS(total_sales - cogs) as loss_amt
FROM pharmacy_sales
WHERE (total_sales - cogs)<0) as temp
GROUP BY manufacturer
ORDER BY total_loss DESC;
```

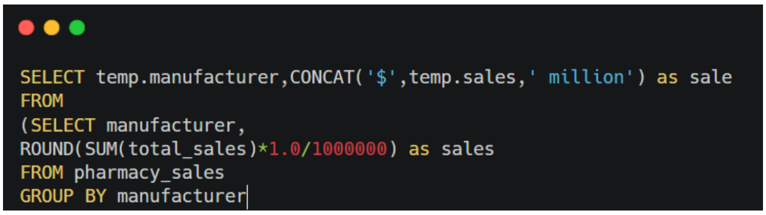
Problem 3: Pharmacy Analytics (Part 3) [CVS Health SQL Interview Question]

Problem Statement:

continue of problem 1 part-3
Write a query to find the total sales of drugs for each manufacturer. Round your answer to the closest million, and report your results in descending order of total sales.
result like this: "$36 million".

Problem Link: https://lnkd.in/gHYveBUD

My Solution:



```
SELECT temp.manufacturer,CONCAT('$',temp.sales,' million') as sale
FROM
(SELECT manufacturer,
ROUND(SUM(total_sales)*1.0/1000000) as sales
FROM pharmacy_sales
GROUP BY manufacturer
```

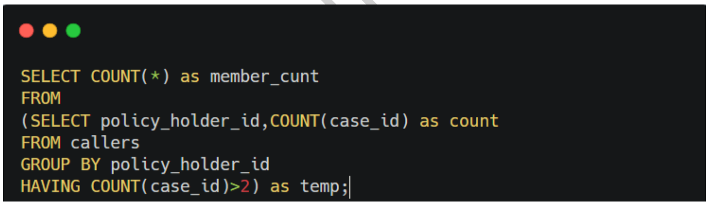
Problem 4: Patient Support Analysis (Part 1) [UnitedHealth SQL Interview Question]

Problem Statement:

Write a query to find how many UHG members made 3 or more calls. case_id column uniquely identifies each call made.

Problem Link: https://lnkd.in/grR_EW5d

My Solution:



```
SELECT COUNT(*) as member_cunt
FROM
(SELECT policy_holder_id,COUNT(case_id) as count
FROM callers
GROUP BY policy_holder_id
HAVING COUNT(case_id)>2) as temp;
```

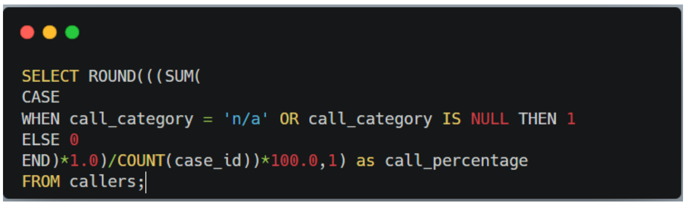
Problem 5: Patient Support Analysis (Part 2) [UnitedHealth SQL Interview Question]

Problem Statement:

continue of problem 4 part-2
alls to the Advocate4Me call centre are categorised, but sometimes they can't fit neatly into a category. These uncategorised calls are labelled “n/a”, or are just empty
Write a query to find the percentage of calls that cannot be categorised. Round your answer to 1 decimal place.

Problem Link: https://lnkd.in/g975YMbj

My Solution:



```
SELECT ROUND(((SUM(
CASE
WHEN call_category = 'n/a' OR call_category IS NULL THEN 1
ELSE 0
END)*1.0)/COUNT(case_id))*100.0,1) as call_percentage
FROM callers;
```

DAY -7

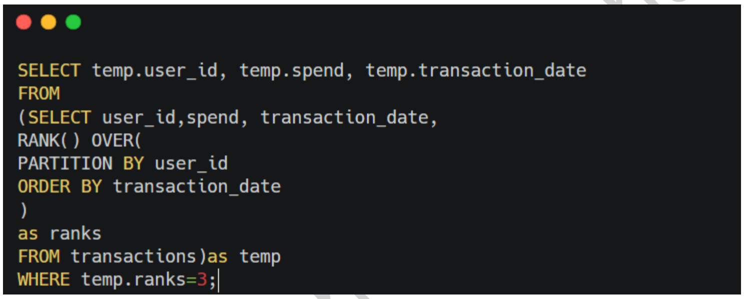
Problem 1: User's Third Transaction [Uber SQL Interview Question]

Problem Statement:

Assume you are given the table below on Uber transactions made by users. Write a query to obtain the third transaction of every user. Output the user id, spend and transaction date.

Problem Link: https://lnkd.in/gwhX5YZp

My Solution:



```
SELECT temp.user_id, temp.spend, temp.transaction_date
FROM
(SELECT user_id,spend, transaction_date,
RANK() OVER(
PARTITION BY user_id
ORDER BY transaction_date
)
as ranks
FROM transactions)as temp
WHERE temp.ranks=3;
```

Problem 2: Sending vs. Opening Snaps [Snapchat SQL Interview Question]

Problem Statement:

Assume you are given the tables below containing information on Snapchat users, their ages, and their time spent sending and opening snaps. Write a query to obtain a breakdown of the time spent sending vs. opening snaps (as a percentage of total time spent on these activities) for each age group.

Output the age bucket and percentage of sending and opening snaps. Round the percentage to 2 decimal places.

Notes:

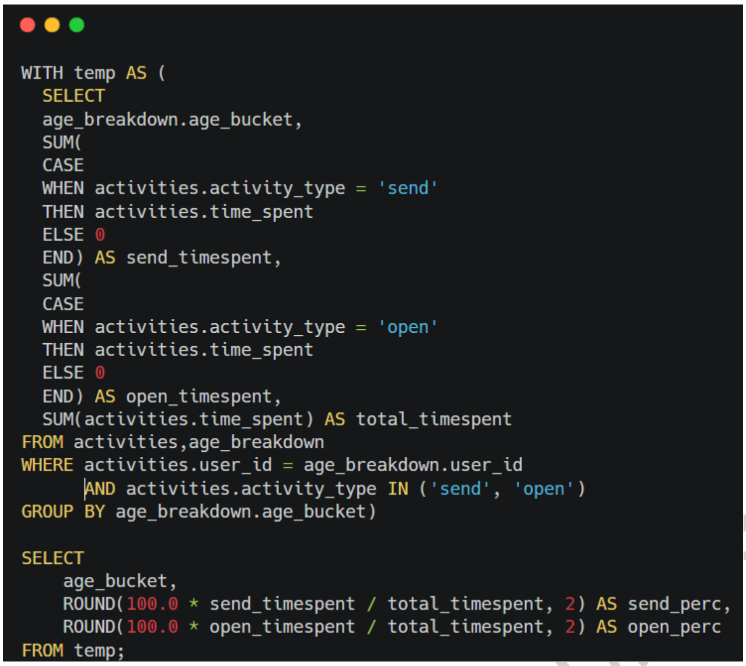
You should calculate these percentages:

time sending / (time sending + time opening) time opening / (time sending + time opening)

To avoid integer division in percentages, multiply by 100.0 and not 100.

Problem Link: https://lnkd.in/gnvwaJ5H

My Solution:



```
WITH temp AS (
  SELECT
  age_breakdown.age_bucket,
  SUM(
  CASE
  WHEN activities.activity_type = 'send'
  THEN activities.time_spent
  ELSE 0
  END) AS send_timespent,
  SUM(
  CASE
  WHEN activities.activity_type = 'open'
  THEN activities.time_spent
  ELSE 0
  END) AS open_timespent,
  SUM(activities.time_spent) AS total_timespent
FROM activities,age_breakdown
WHERE activities.user_id = age_breakdown.user_id
      AND activities.activity_type IN ('send', 'open')
GROUP BY age_breakdown.age_bucket)

SELECT
    age_bucket,
    ROUND(100.0 * send_timespent / total_timespent, 2) AS send_perc,
    ROUND(100.0 * open_timespent / total_timespent, 2) AS open_perc
FROM temp;
```

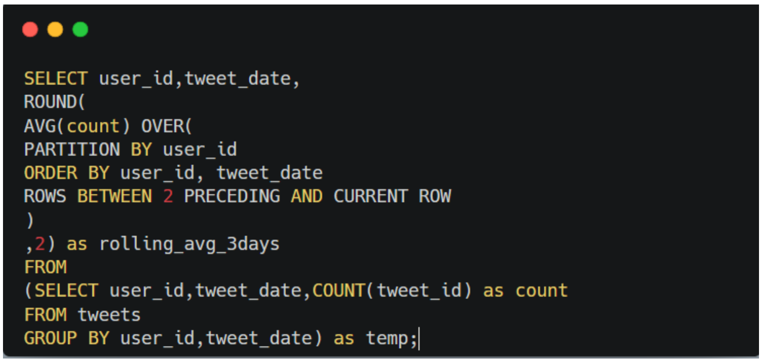
DAY -8

Problem 1: Tweets' Rolling Averages [Twitter SQL Interview Question]Problem Statement:

The table below contains information about tweets over a given period of time. Calculate the 3-day rolling average of tweets published by each user for each date that a tweet was posted. Output the user id, tweet date, and rolling averages rounded to 2 decimal places.

Problem Link: https://lnkd.in/gDhyXKsx

My Solution:



```
SELECT user_id,tweet_date,
ROUND(
AVG(count) OVER(
PARTITION BY user_id
ORDER BY user_id, tweet_date
ROWS BETWEEN 2 PRECEDING AND CURRENT ROW
)
,2) as rolling_avg_3days
FROM
(SELECT user_id,tweet_date,COUNT(tweet_id) as count
FROM tweets
GROUP BY user_id,tweet_date) as temp;
```

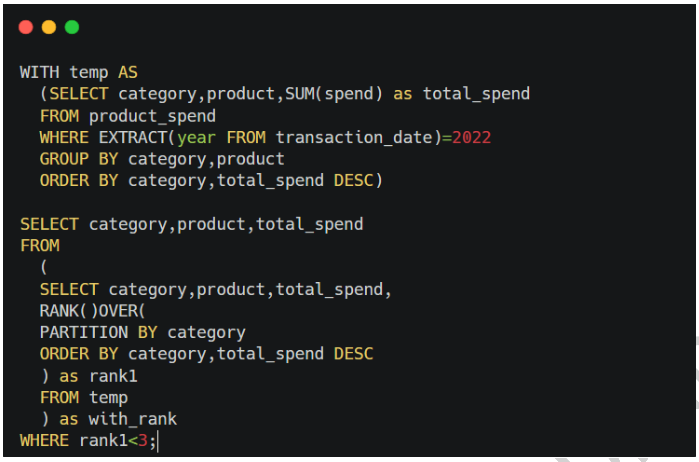
Problem 2: Highest-Grossing Items [Amazon SQL Interview Question]

Problem Statement:

Assume you are given the table containing information on Amazon customers and their spending on products in various categories.
Identify the top two highest-grossing products within each category in 2022. Output the category, product, and total spend.

Problem Link: https://lnkd.in/gBY_zcy6

My Solution:



```
WITH temp AS
  (SELECT category,product,SUM(spend) as total_spend
  FROM product_spend
  WHERE EXTRACT(year FROM transaction_date)=2022
  GROUP BY category,product
  ORDER BY category,total_spend DESC)

SELECT category,product,total_spend
FROM
  (
  SELECT category,product,total_spend,
  RANK()OVER(
  PARTITION BY category
  ORDER BY category,total_spend DESC
  ) as rank1
  FROM temp
  ) as with_rank
WHERE rank1<3;
```

DAY -9

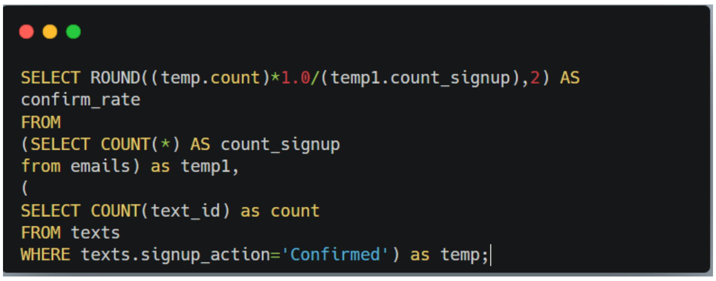
Problem 1: Signup Activation Rate [TikTok SQL Interview Question]

Problem Statement:

New TikTok users sign up with their emails. They confirmed their signup by replying to the text confirmation to activate their accounts. Users may receive multiple text messages for account confirmation until they have confirmed their new account.
Write a query to find the activation rate of the users. Round the percentage to 2 decimal places.
full question on link

Problem Link: https://lnkd.in/gDREv64Q

My Solution:



```
SELECT ROUND((temp.count)*1.0/(temp1.count_signup),2) AS
confirm_rate
FROM
(SELECT COUNT(*) AS count_signup
from emails) as temp1,
(
SELECT COUNT(text_id) as count
FROM texts
WHERE texts.signup_action='Confirmed') as temp;
```

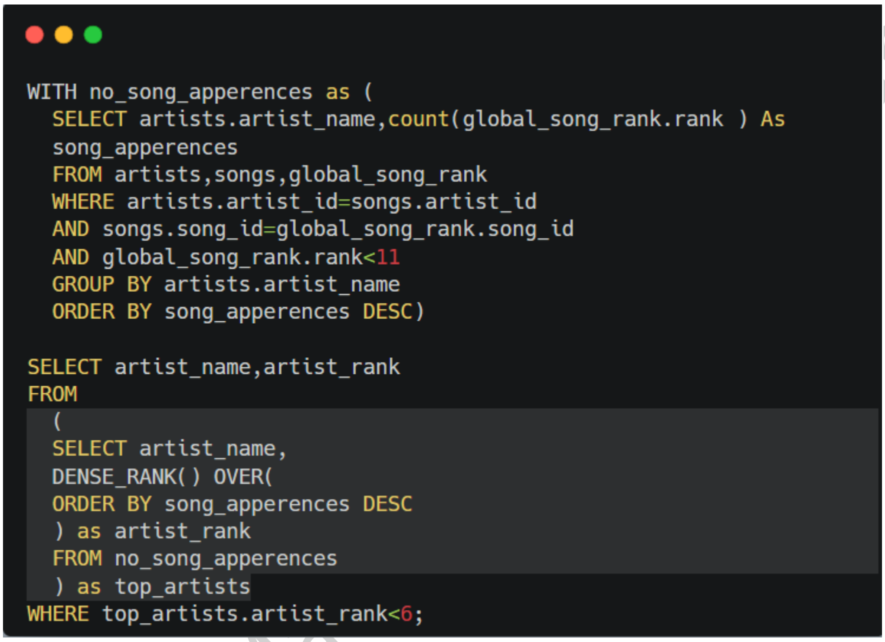
Problem 2: Top 5 Artists [Spotify SQL Interview Question]

Problem Statement:

Assume there are three Spotify tables containing information about the artists, songs, and music charts. Write a query to determine the top 5 artists whose songs appear in the Top 10 of the global_song_rank table the highest number of times. From now on, we'll refer to this ranking number as "song appearances".
Output the top 5 artist names in ascending order along with their song appearances ranking (not the number of song appearances, but the rank of who has the most appearances). The order of the rank should take precedence. full question on link

Problem Link: https://lnkd.in/gmq3SH3D

My Solution:



```
WITH no_song_apperences as (
  SELECT artists.artist_name,count(global_song_rank.rank ) As
  song_apperences
  FROM artists,songs,global_song_rank
  WHERE artists.artist_id=songs.artist_id
  AND songs.song_id=global_song_rank.song_id
  AND global_song_rank.rank<11
  GROUP BY artists.artist_name
  ORDER BY song_apperences DESC)

SELECT artist_name,artist_rank
FROM
  (
  SELECT artist_name,
  DENSE_RANK() OVER(
  ORDER BY song_apperences DESC
  ) as artist_rank
  FROM no_song_apperences
  ) as top_artists
WHERE top_artists.artist_rank<6;
```

DAY -10

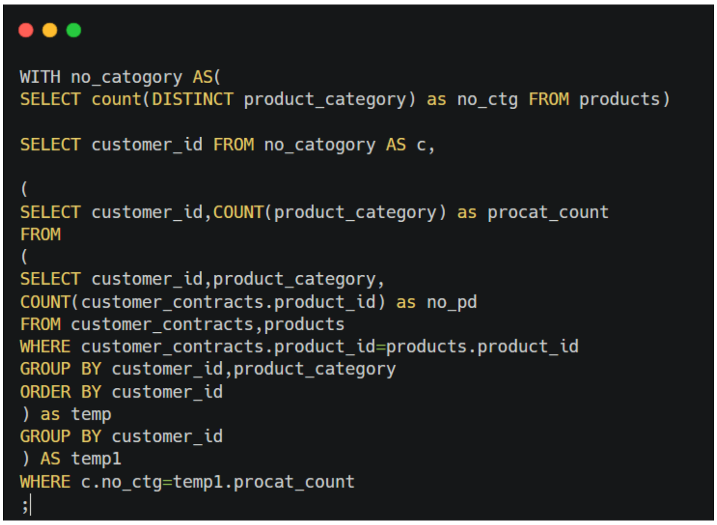
Problem 1: Supercloud Customer [Microsoft SQL Interview Question]

Problem Statement:

A Microsoft Azure Supercloud customer is a company which buys at least 1 product from each product category.
Write a query to report the company ID which is a Supercloud customer.
full question on link

Problem Link: https://lnkd.in/gzREXRvV

My Solution:



```
WITH no_catogory AS(
SELECT count(DISTINCT product_category) as no_ctg FROM products)

SELECT customer_id FROM no_catogory AS c,

(
SELECT customer_id,COUNT(product_category) as procat_count
FROM
(
SELECT customer_id,product_category,
COUNT(customer_contracts.product_id) as no_pd
FROM customer_contracts,products
WHERE customer_contracts.product_id=products.product_id
GROUP BY customer_id,product_category
ORDER BY customer_id
) as temp
GROUP BY customer_id
) AS temp1
WHERE c.no_ctg=temp1.procat_count
;
```

DAY -11

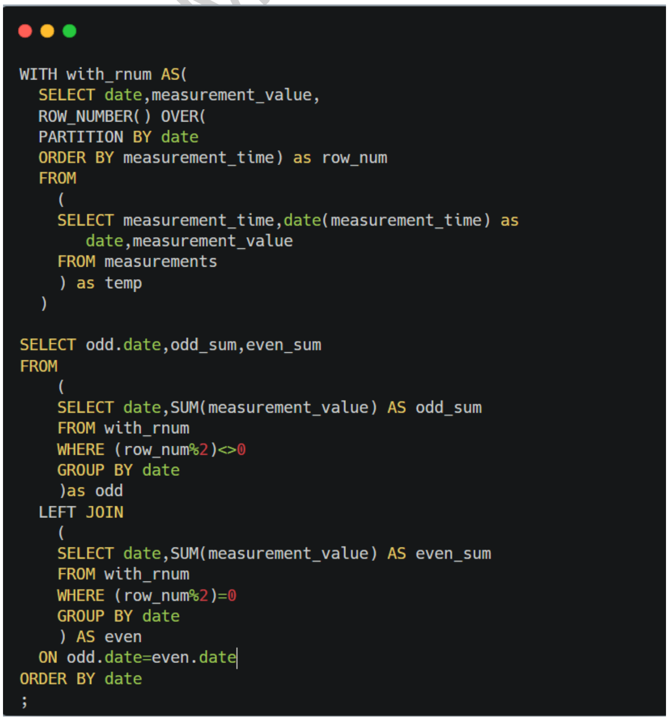
Problem 1: Odd and Even Measurements [Google SQL Interview Question]

Problem Statement:

Assume you are given the table containing measurement values obtained from a Google sensor over several days. Measurements are taken several times within a given day.
Write a query to obtain the sum of the odd-numbered and even-numbered measurements on a particular day, in two different columns. Refer to the Example Output below for the outputformat.
full question on link

Problem Link: https://lnkd.in/gYDRtExF

My Solution:



```
WITH with_rnum AS(
  SELECT date,measurement_value,
  ROW_NUMBER( ) OVER(
  PARTITION BY date
  ORDER BY measurement_time) as row_num
  FROM
    (
    SELECT measurement_time,date(measurement_time) as
       date,measurement_value
    FROM measurements
    ) as temp
  )

SELECT odd.date,odd_sum,even_sum
FROM
    (
    SELECT date,SUM(measurement_value) AS odd_sum
    FROM with_rnum
    WHERE (row_num%2)<>0
    GROUP BY date
    )as odd
  LEFT JOIN
    (
    SELECT date,SUM(measurement_value) AS even_sum
    FROM with_rnum
    WHERE (row_num%2)=0
    GROUP BY date
    ) AS even
  ON odd.date=even.date
ORDER BY date
;
```

DAY -12

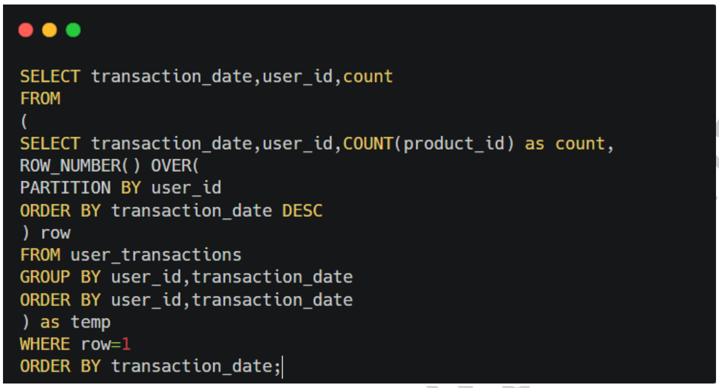
Problem 1: Histogram of Users and Purchases [Walmart SQL Interview Question]

Problem Statement:

Assume you are given the table on Walmart user transactions. Based on a user's most recent transaction date, write a query to obtain the users and the number of products bought. Output the user's most recent transaction date, user ID and the number of products sorted by the transaction date in chronological order.

Problem Link: https://lnkd.in/gbr27qyx

My Solution:



```
SELECT transaction_date,user_id,count
FROM
(
SELECT transaction_date,user_id,COUNT(product_id) as count,
ROW_NUMBER() OVER(
PARTITION BY user_id
ORDER BY transaction_date DESC
) row
FROM user_transactions
GROUP BY user_id,transaction_date
ORDER BY user_id,transaction_date
) as temp
WHERE row=1
ORDER BY transaction_date;
```

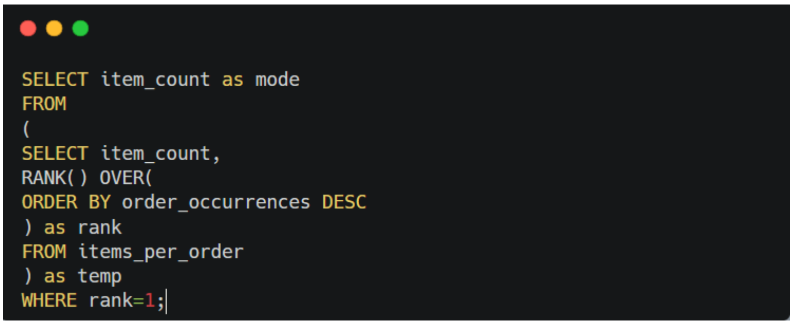
Problem 2: Compressed Mode [Alibaba SQL Interview Question]

Problem Statement:

You are trying to find the most common (aka the mode) number of items bought per order on Alibaba. However, instead of doing analytics on all Alibaba orders, you have access to a summary table, which describes how many items were in an order (item_count), and the number of orders that had that many items (order_occurrences).In case of multiple item counts, display the item_counts in ascending order.

Problem Link: https://lnkd.in/gzE97zmH

My Solution:



```
SELECT item_count as mode
FROM
(
SELECT item_count,
RANK() OVER(
ORDER BY order_occurrences DESC
) as rank
FROM items_per_order
) as temp
WHERE rank=1;
```

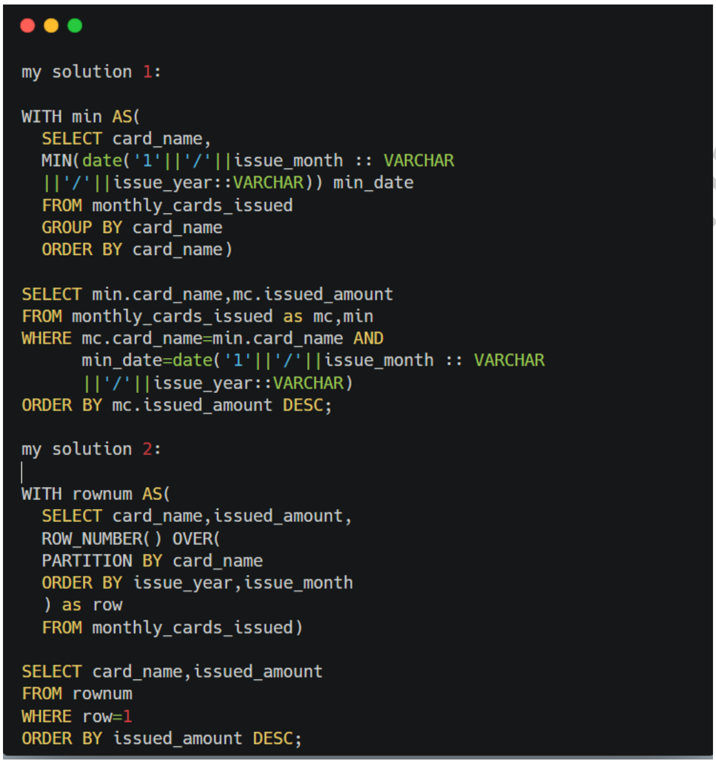
Problem 3: Card Launch Success [JPMorgan Chase SQL Interview Question]

Problem Statement:

Your team at JPMorgan Chase is soon launching a new credit card. You are asked to estimate how many cards you'll issue in the first month.
Before you can answer this question, you want to first get some perspective on how well new credit card launches typically do in their first month.
Write a query that outputs the name of the credit card, and how many cards were issued in its launch month. The launch month is the earliest record in the monthly_cards_issued table for a given card. Order the results starting from the biggest issued amount
I solved it in two ways

Problem Link: https://lnkd.in/gPZgbJHp

My Solution:



```
my solution 1:

WITH min AS(
  SELECT card_name,
  MIN(date('1'||'/'||issue_month :: VARCHAR
  ||'/'||issue_year::VARCHAR)) min_date
  FROM monthly_cards_issued
  GROUP BY card_name
  ORDER BY card_name)

SELECT min.card_name,mc.issued_amount
FROM monthly_cards_issued as mc,min
WHERE mc.card_name=min.card_name AND
      min_date=date('1'||'/'||issue_month :: VARCHAR
      ||'/'||issue_year::VARCHAR)
ORDER BY mc.issued_amount DESC;

my solution 2:

WITH rownum AS(
  SELECT card_name,issued_amount,
  ROW_NUMBER() OVER(
  PARTITION BY card_name
  ORDER BY issue_year,issue_month
  ) as row
  FROM monthly_cards_issued)

SELECT card_name,issued_amount
FROM rownum
WHERE row=1
ORDER BY issued_amount DESC;
```

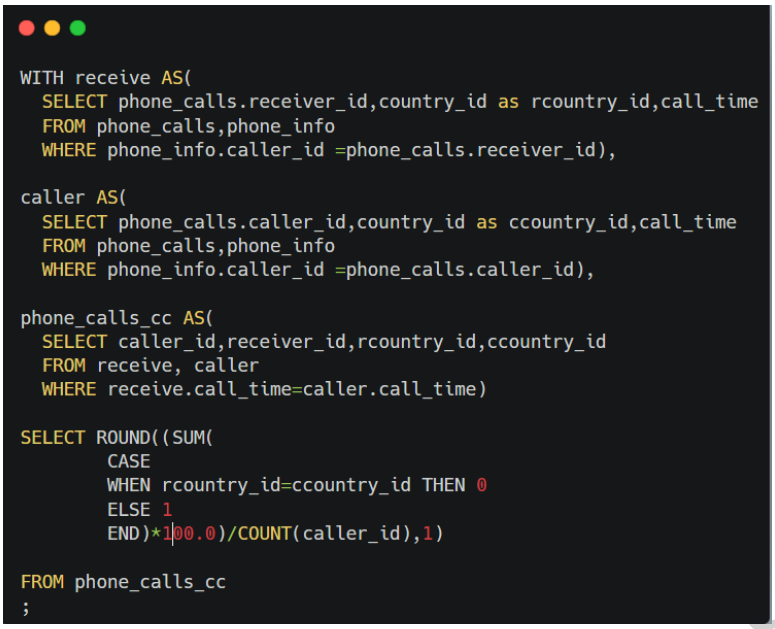
DAY -13

Problem 1: International Call Percentage [Verizon SQL Interview Question]

A phone call is considered an international call when the person calling is in a different country than the person receiving the call.
What percentage of phone calls are international? Round the result to 1 decimal.
Assumption:
The caller_id in phone_info table refers to both the caller and receiver.

Problem Link: https://lnkd.in/g-ArRBDk

My Solution:



```
WITH receive AS(
  SELECT phone_calls.receiver_id,country_id as rcountry_id,call_time
  FROM phone_calls,phone_info
  WHERE phone_info.caller_id =phone_calls.receiver_id),

caller AS(
  SELECT phone_calls.caller_id,country_id as ccountry_id,call_time
  FROM phone_calls,phone_info
  WHERE phone_info.caller_id =phone_calls.caller_id),

phone_calls_cc AS(
  SELECT caller_id,receiver_id,rcountry_id,ccountry_id
  FROM receive, caller
  WHERE receive.call_time=caller.call_time)

SELECT ROUND((SUM(
        CASE
        WHEN rcountry_id=ccountry_id THEN 0
        ELSE 1
        END)*100.0)/COUNT(caller_id),1)

FROM phone_calls_cc
;
```

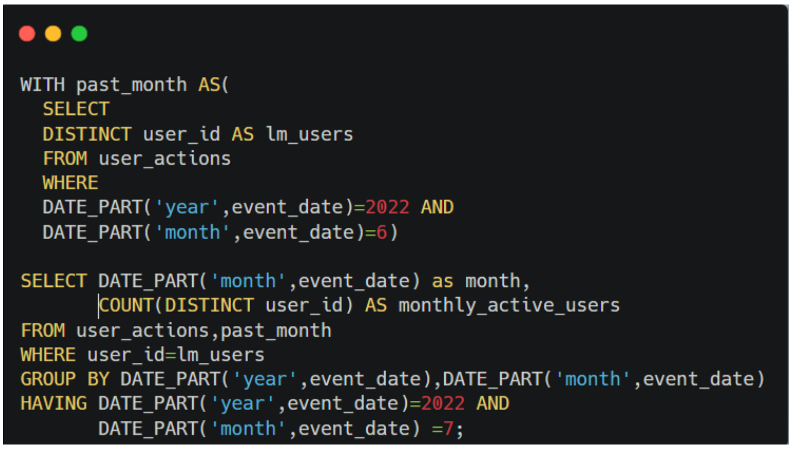
DAY -14

Problem 1: Active User Retention [Facebook SQL Interview Question]

Assume you have the table below containing information on Facebook user actions. Write a query to obtain the active user retention in July 2022. Output the month (in numerical format 1, 2, 3) and the number of monthly active users (MAUs).
Hint: An active user is a user who has user action ("sign-in", "like", or "comment") in the current month and last month

Problem Link: https://lnkd.in/g8zXV954

My Solution:



```
WITH past_month AS(
  SELECT
  DISTINCT user_id AS lm_users
  FROM user_actions
  WHERE
  DATE_PART('year',event_date)=2022 AND
  DATE_PART('month',event_date)=6)

SELECT DATE_PART('month',event_date) as month,
       COUNT(DISTINCT user_id) AS monthly_active_users
FROM user_actions,past_month
WHERE user_id=lm_users
GROUP BY DATE_PART('year',event_date),DATE_PART('month',event_date)
HAVING DATE_PART('year',event_date)=2022 AND
       DATE_PART('month',event_date) =7;
```

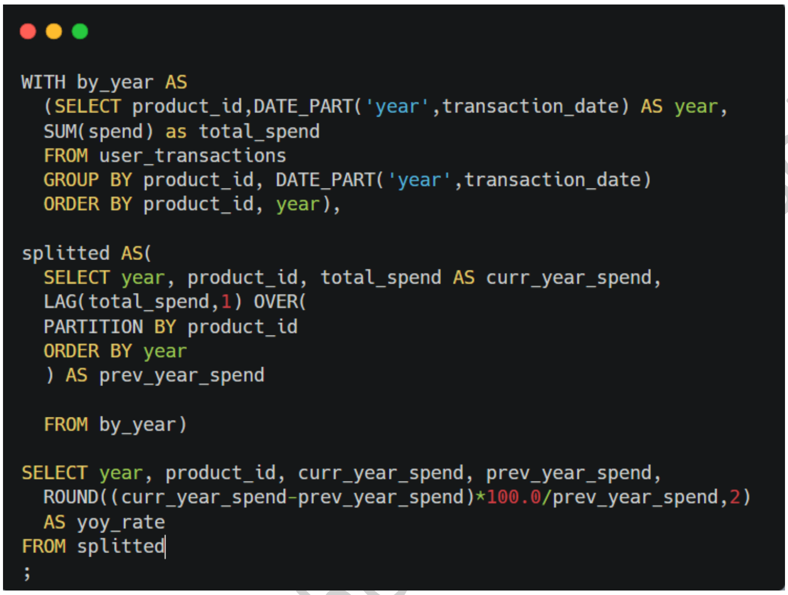
DAY -15

Problem 1: A Y-on-Y Growth Rate [Wayfair SQL Interview Question]

Assume you are given the table below containing information on user transactions for particular products. Write a query to obtain the year-on-year growth rate for the total spend of each product for each year.
Output the year (in ascending order) partitioned by product id, current year's spend, previous year's spend and year-on-year growth rate (percentage rounded to 2 decimal places.

Problem Link: https://lnkd.in/g8ZWSVmg

My Solution:



```sql
WITH by_year AS
  (SELECT product_id,DATE_PART('year',transaction_date) AS year,
  SUM(spend) as total_spend
  FROM user_transactions
  GROUP BY product_id, DATE_PART('year',transaction_date)
  ORDER BY product_id, year),

splitted AS(
  SELECT year, product_id, total_spend AS curr_year_spend,
  LAG(total_spend,1) OVER(
  PARTITION BY product_id
  ORDER BY year
  ) AS prev_year_spend

  FROM by_year)

SELECT year, product_id, curr_year_spend, prev_year_spend,
  ROUND((curr_year_spend-prev_year_spend)*100.0/prev_year_spend,2)
  AS yoy_rate
FROM splitted
;
```

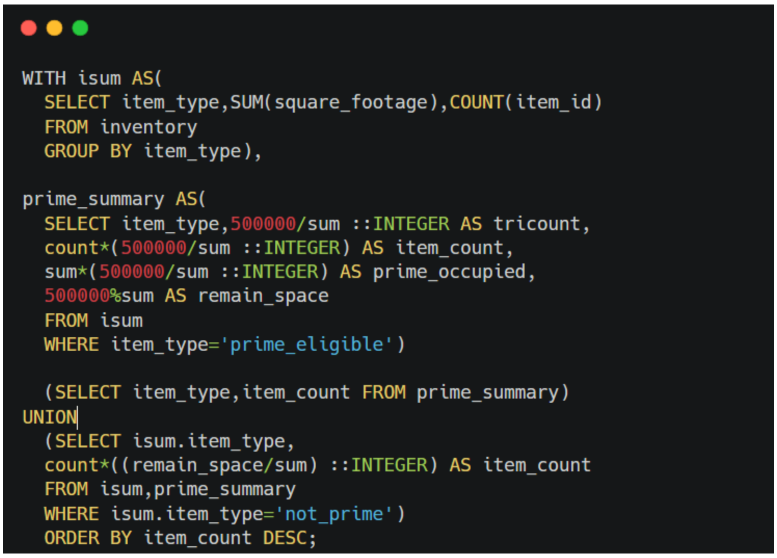
DAY -16

Problem 1: Maximize Prime Item Inventory [Amazon SQL Interview Question]

Amazon wants to maximize the number of items it can stock in a 500,000 square feet warehouse. It wants to stock as many prime items as possible, and afterwards use the remaining square footage to stock the most number of non-prime items.
Write a SQL query to find the number of prime and non-prime items that can be stored in the 500,000 square feet warehouse. Output the item type and number of items to be stocked.

Problem Link: https://lnkd.in/ggbjCVxx

My Solution:



```
WITH isum AS(
  SELECT item_type,SUM(square_footage),COUNT(item_id)
  FROM inventory
  GROUP BY item_type),

prime_summary AS(
  SELECT item_type,500000/sum ::INTEGER AS tricount,
  count*(500000/sum ::INTEGER) AS item_count,
  sum*(500000/sum ::INTEGER) AS prime_occupied,
  500000%sum AS remain_space
  FROM isum
  WHERE item_type='prime_eligible')

  (SELECT item_type,item_count FROM prime_summary)
UNION
  (SELECT isum.item_type,
  count*((remain_space/sum) ::INTEGER) AS item_count
  FROM isum,prime_summary
  WHERE isum.item_type='not_prime')
  ORDER BY item_count DESC;
```

DAY -17

Problem 1: Median Google Search Frequency [Google SQL Interview Question]

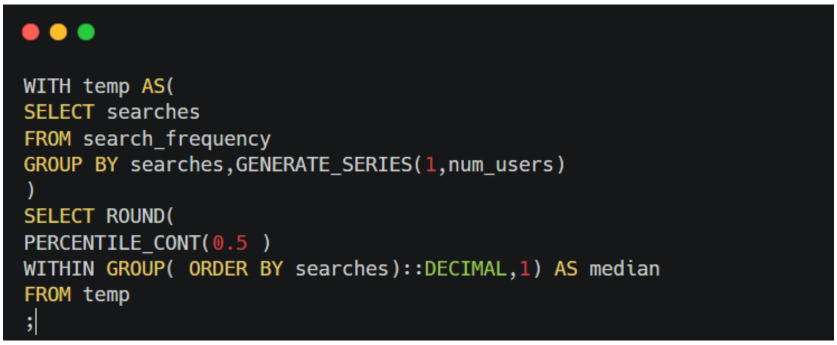
Google's marketing team is making a Superbowl commercial and needs a simple statistic to put on their TV ad: the median number of searches a person made last year.

However, at Google scale, querying the 2 trillion searches is too costly. Luckily, you have access to the summary table which tells you the number of searches made last year and how many Google users fall into that bucket.

Write a query to report the median of searches made by a user. Round the median to one decimal point

Problem Link: https://lnkd.in/gq3A3q6j

My Solution:



```
WITH temp AS(
SELECT searches
FROM search_frequency
GROUP BY searches,GENERATE_SERIES(1,num_users)
)
SELECT ROUND(
PERCENTILE_CONT(0.5 )
WITHIN GROUP( ORDER BY searches)::DECIMAL,1) AS median
FROM temp
;
```

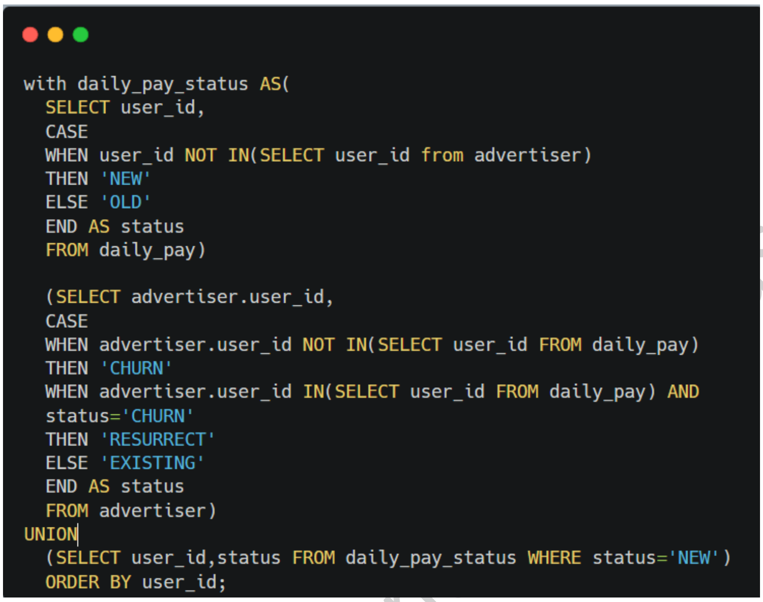
DAY -18

Problem 1: Advertiser Status [Facebook SQL Interview Question]

Write a query to update the Facebook advertiser's status using the daily_pay table. Advertiser is a two-column table containing the user id and their payment status based on the last payment and daily_pay table has current information about their payment. Only advertisers who paid will show up in this table.
Output the user id and current payment status sorted by the user id.

Problem Link: https://lnkd.in/g5WWYS7W

My Solution:



```
with daily_pay_status AS(
  SELECT user_id,
  CASE
  WHEN user_id NOT IN(SELECT user_id from advertiser)
  THEN 'NEW'
  ELSE 'OLD'
  END AS status
  FROM daily_pay)

  (SELECT advertiser.user_id,
  CASE
  WHEN advertiser.user_id NOT IN(SELECT user_id FROM daily_pay)
  THEN 'CHURN'
  WHEN advertiser.user_id IN(SELECT user_id FROM daily_pay) AND
  status='CHURN'
  THEN 'RESURRECT'
  ELSE 'EXISTING'
  END AS status
  FROM advertiser)
UNION
  (SELECT user_id,status FROM daily_pay_status WHERE status='NEW')
  ORDER BY user_id;
```

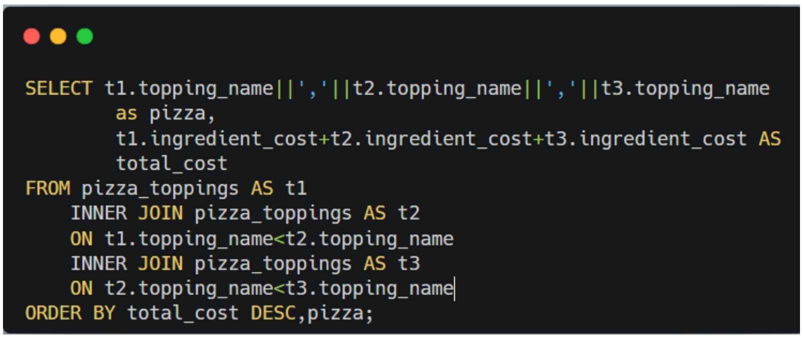
DAY -19

Problem 1: Topping Pizzas [McKinsey SQL Interview Question]

You're a consultant for a major pizza chain that will be running a promotion where all 3-topping pizzas will be sold for a fixed price, and are trying to understand the costs involved.
Given a list of pizza toppings, consider all the possible 3-topping pizzas, and print out the total cost of those 3 toppings. Sort the results with the highest total cost on the top followed by pizza toppings in ascending order.
Break ties by listing the ingredients in alphabetical order, starting from the first ingredient, followed by the second and third.

Problem Link: https://lnkd.in/gc9ehpxn

My Solution:



```
SELECT t1.topping_name||','||t2.topping_name||','||t3.topping_name
        as pizza,
        t1.ingredient_cost+t2.ingredient_cost+t3.ingredient_cost AS
        total_cost
FROM pizza_toppings AS t1
    INNER JOIN pizza_toppings AS t2
    ON t1.topping_name<t2.topping_name
    INNER JOIN pizza_toppings AS t3
    ON t2.topping_name<t3.topping_name
ORDER BY total_cost DESC,pizza;
```

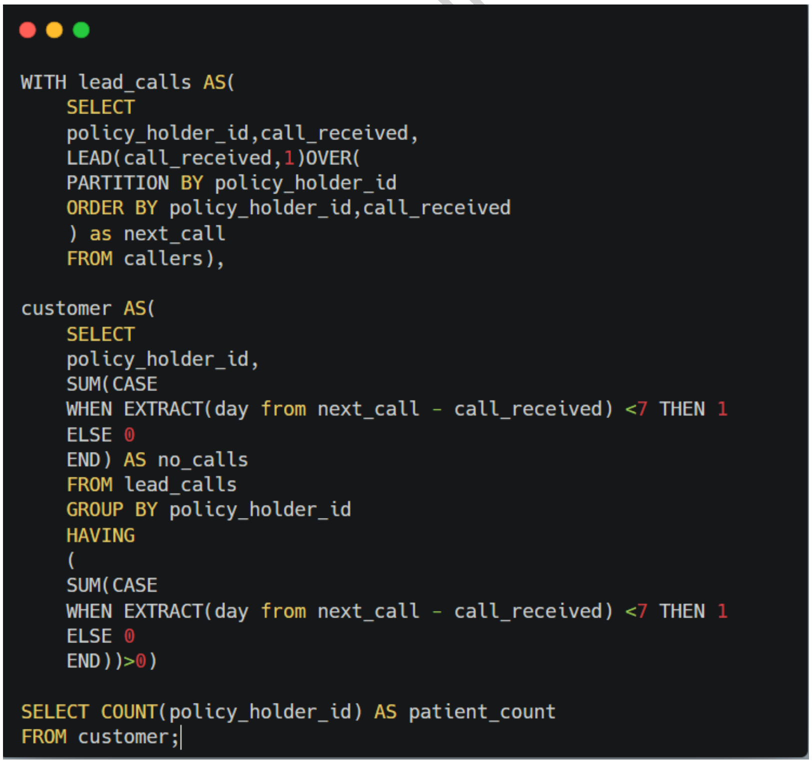
DAY -20

Problem 1: Patient Support Analysis (Part 3) [UnitedHealth SQL Interview Question]

UnitedHealth Group has a program called Advocate4Me, which allows members to call an advocate and receive support for their health care needs – whether that's behavioural, clinical, well-being, health care financing, benefits, claims or pharmacy help.
Write a query to get the patients who made a call within 7 days of their previous call. If a patient called more than twice in a span of 7 days, count them as once.

Problem Link: https://lnkd.in/gSearVV4

My Solution:



```
WITH lead_calls AS(
    SELECT
    policy_holder_id,call_received,
    LEAD(call_received,1)OVER(
    PARTITION BY policy_holder_id
    ORDER BY policy_holder_id,call_received
    ) as next_call
    FROM callers),

customer AS(
    SELECT
    policy_holder_id,
    SUM(CASE
    WHEN EXTRACT(day from next_call - call_received) <7 THEN 1
    ELSE 0
    END) AS no_calls
    FROM lead_calls
    GROUP BY policy_holder_id
    HAVING
    (
    SUM(CASE
    WHEN EXTRACT(day from next_call - call_received) <7 THEN 1
    ELSE 0
    END))>0)

SELECT COUNT(policy_holder_id) AS patient_count
FROM customer;
```

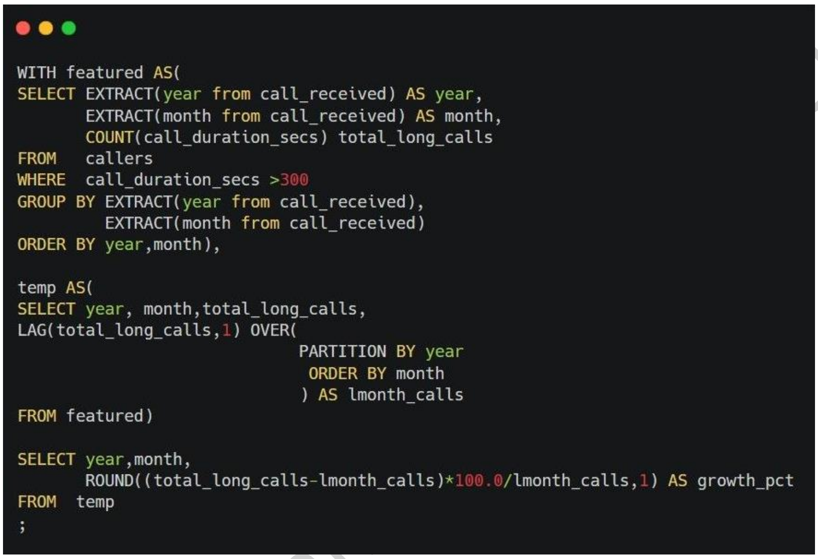
DAY -21

Problem 1: Patient Support Analysis (Part 4) [UnitedHealth SQL Interview Question]

UnitedHealth Group has a program called Advocate4Me, which allows members to call an advocate and receive support for their health care needs – whether that's behavioural, clinical, well-being, health care financing, benefits, claims or pharmacy help.
A long-call is categorised as any call that lasts more than 5 minutes (300 seconds). What's the month-over-month growth of long-calls?
Output the year, month (both in numerical and chronological order) and growth percentage rounded to 1 decimal place.

Problem Link: https://lnkd.in/gDi5hvPb

My Solution:



```
WITH featured AS(
SELECT EXTRACT(year from call_received) AS year,
       EXTRACT(month from call_received) AS month,
       COUNT(call_duration_secs) total_long_calls
FROM   callers
WHERE  call_duration_secs >300
GROUP BY EXTRACT(year from call_received),
         EXTRACT(month from call_received)
ORDER BY year,month),

temp AS(
SELECT year, month,total_long_calls,
LAG(total_long_calls,1) OVER(
                             PARTITION BY year
                              ORDER BY month
                             ) AS lmonth_calls
FROM featured)

SELECT year,month,
       ROUND((total_long_calls-lmonth_calls)*100.0/lmonth_calls,1) AS growth_pct
FROM  temp
;
```

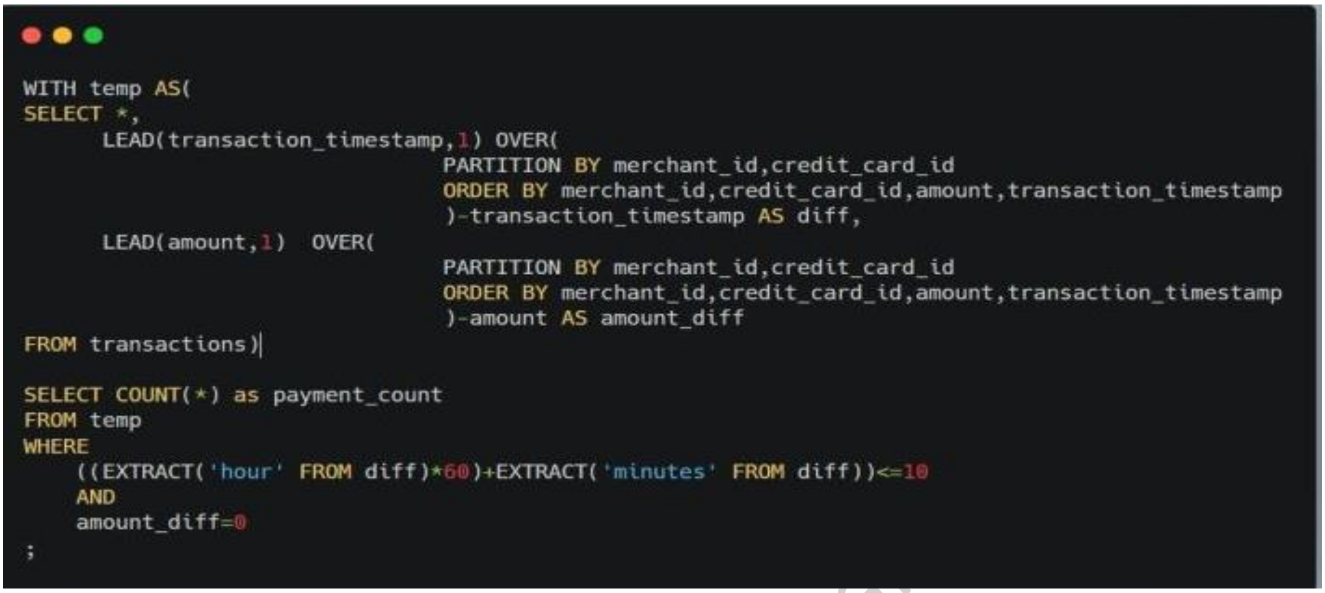
DAY -22

Problem 1: Repeated Payments [Stripe SQL Interview Question]

Sometimes, payment transactions are repeated by accident; it could be due to user error, API failure or a retry error that causes a credit card to be charged twice.
Using the transactions table, identify any payments made at the same merchant with the same credit card for the same amount within 10 minutes of each other. Count such repeated payments.
Assumptions:
The first transaction of such payments should not be counted as a repeated payment. This means, if there are two transactions performed by a merchant with the same credit card and for the same amount within 10 minutes, there will only be 1 repeated payment.

Problem Link: https://lnkd.in/gfhY6tEz

My Solution:



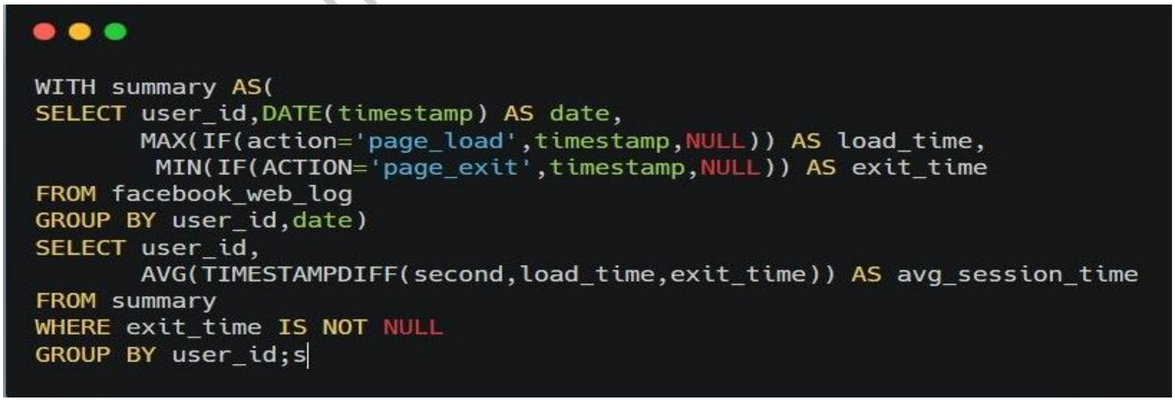
```
WITH temp AS(
SELECT *,
      LEAD(transaction_timestamp,1) OVER(
                                PARTITION BY merchant_id,credit_card_id
                                ORDER BY merchant_id,credit_card_id,amount,transaction_timestamp
                                )-transaction_timestamp AS diff,
      LEAD(amount,1)  OVER(
                                PARTITION BY merchant_id,credit_card_id
                                ORDER BY merchant_id,credit_card_id,amount,transaction_timestamp
                                )-amount AS amount_diff
FROM transactions)

SELECT COUNT(*) as payment_count
FROM temp
WHERE
    ((EXTRACT('hour' FROM diff)*60)+EXTRACT('minutes' FROM diff))<=10
    AND
    amount_diff=0
;
```

Problem 2: Users By Average Session Time [Facebook Interview Question]

Calculate each user's average session time. A session is defined as the time difference between a page_load and page_exit. For simplicity, assume a user has only 1 session per day and if there are multiple of the same events on that day, consider only the latest page_load and earliest page_exit. Output the user_id and their average session time.

Problem Link: https://lnkd.in/gw5Wu9Ct

My Solution:



```
WITH summary AS(
SELECT user_id,DATE(timestamp) AS date,
       MAX(IF(action='page_load',timestamp,NULL)) AS load_time,
        MIN(IF(ACTION='page_exit',timestamp,NULL)) AS exit_time
FROM facebook_web_log
GROUP BY user_id,date)
SELECT user_id,
       AVG(TIMESTAMPDIFF(second,load_time,exit_time)) AS avg_session_time
FROM summary
WHERE exit_time IS NOT NULL
GROUP BY user_id;s
```

DAY -23

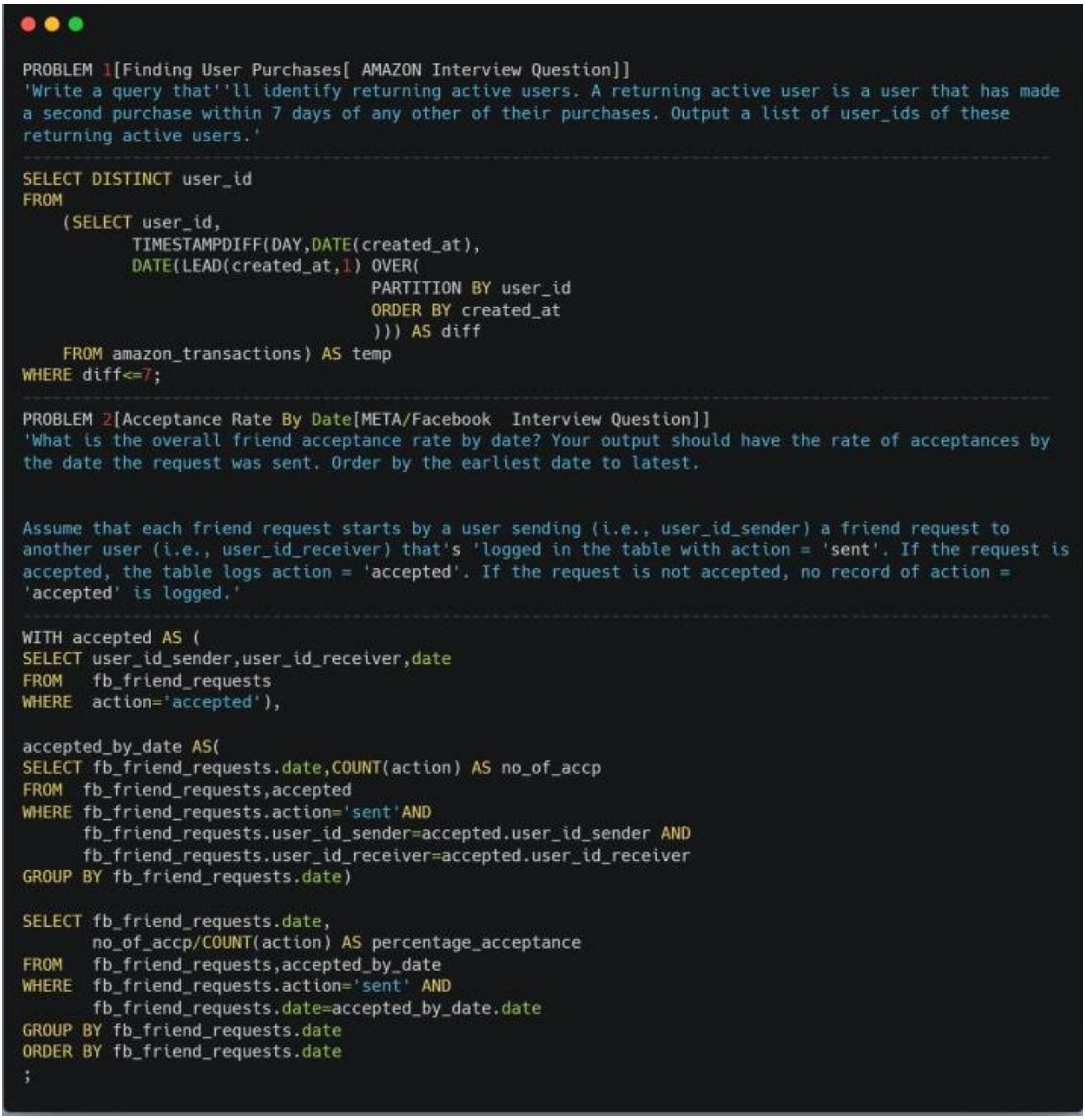
Problem 1: Finding User Purchases[AMAZON Interview Question]

Problem Link: https://lnkd.in/g-ZgtR-A

Problem 2: Acceptance Rate By Date[META/Facebook Interview Question]

Problem Link: https://lnkd.in/g7fTAQzS

My Solution:



```
PROBLEM 1[Finding User Purchases[ AMAZON Interview Question]]
'Write a query that''ll identify returning active users. A returning active user is a user that has made
a second purchase within 7 days of any other of their purchases. Output a list of user_ids of these
returning active users.'
--------------------------------------------------------------------------------------------------------
SELECT DISTINCT user_id
FROM
    (SELECT user_id,
           TIMESTAMPDIFF(DAY,DATE(created_at),
           DATE(LEAD(created_at,1) OVER(
                                   PARTITION BY user_id
                                   ORDER BY created_at
                                   ))) AS diff
    FROM amazon_transactions) AS temp
WHERE diff<=7;
--------------------------------------------------------------------------------------------------------
PROBLEM 2[Acceptance Rate By Date[META/Facebook  Interview Question]]
'What is the overall friend acceptance rate by date? Your output should have the rate of acceptances by
the date the request was sent. Order by the earliest date to latest.


Assume that each friend request starts by a user sending (i.e., user_id_sender) a friend request to
another user (i.e., user_id_receiver) that's 'logged in the table with action = 'sent'. If the request is
accepted, the table logs action = 'accepted'. If the request is not accepted, no record of action =
'accepted' is logged.'
--------------------------------------------------------------------------------------------------------
WITH accepted AS (
SELECT user_id_sender,user_id_receiver,date
FROM   fb_friend_requests
WHERE  action='accepted'),

accepted_by_date AS(
SELECT fb_friend_requests.date,COUNT(action) AS no_of_accp
FROM  fb_friend_requests,accepted
WHERE fb_friend_requests.action='sent'AND
      fb_friend_requests.user_id_sender=accepted.user_id_sender AND
      fb_friend_requests.user_id_receiver=accepted.user_id_receiver
GROUP BY fb_friend_requests.date)

SELECT fb_friend_requests.date,
       no_of_accp/COUNT(action) AS percentage_acceptance
FROM   fb_friend_requests,accepted_by_date
WHERE  fb_friend_requests.action='sent'  AND
       fb_friend_requests.date=accepted_by_date.date
GROUP BY fb_friend_requests.date
ORDER BY fb_friend_requests.date
;
```

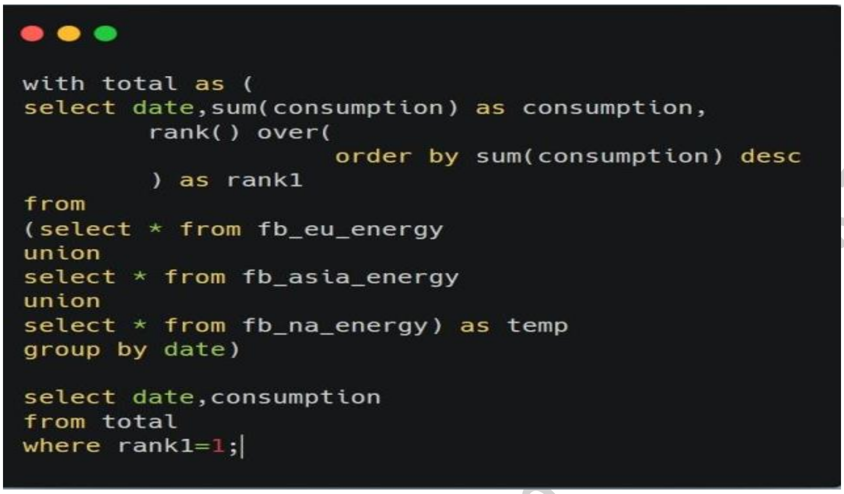
DAY -24

Problem 1: Highest Energy Consumption[Meta/Facebook interview Question]

Find the date with the highest total energy consumption from the Meta/Facebook data centers. Output the date along with the total energy consumption across all data centers.
Tables: fb_eu_energy, fb_asia_energy, fb_na_energy.

Problem Link: https://lnkd.in/gd2VaxyT

My Solution:



```
with total as (
select date,sum(consumption) as consumption,
        rank() over(
                    order by sum(consumption) desc
        ) as rank1
from
(select * from fb_eu_energy
union
select * from fb_asia_energy
union
select * from fb_na_energy) as temp
group by date)

select date,consumption
from total
where rank1=1;
```

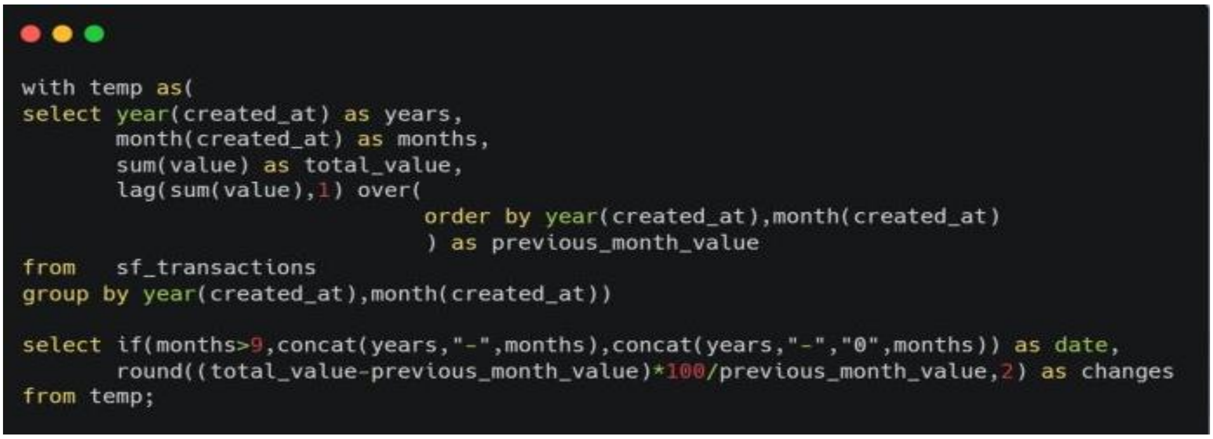
DAY -25

Problem 1: Monthly Percentage Difference[Amazon Interview Question]

Given a table of purchases by date, calculate the month-over-month percentage change in revenue. The output should include the year-month date (YYYY-MM) and percentage change, rounded to the 2nd decimal point, and sorted from the beginning of the year to the end of the year.
The percentage change column will be populated from the 2nd month forward and can be calculated as ((this month's revenue - last month's revenue) / last month's revenue)*100.

Problem Link: https://lnkd.in/gEF8WCcB

My Solution:



```
with temp as(
select year(created_at) as years,
       month(created_at) as months,
       sum(value) as total_value,
       lag(sum(value),1) over(
                              order by year(created_at),month(created_at)
                              ) as previous_month_value
from   sf_transactions
group by year(created_at),month(created_at))

select if(months>9,concat(years,"-",months),concat(years,"-","0",months)) as date,
       round((total_value-previous_month_value)*100/previous_month_value,2) as changes
from temp;
```

DAY -26

Problem 1: Highest Cost Orders[Amazon/Shopify Interview Question]

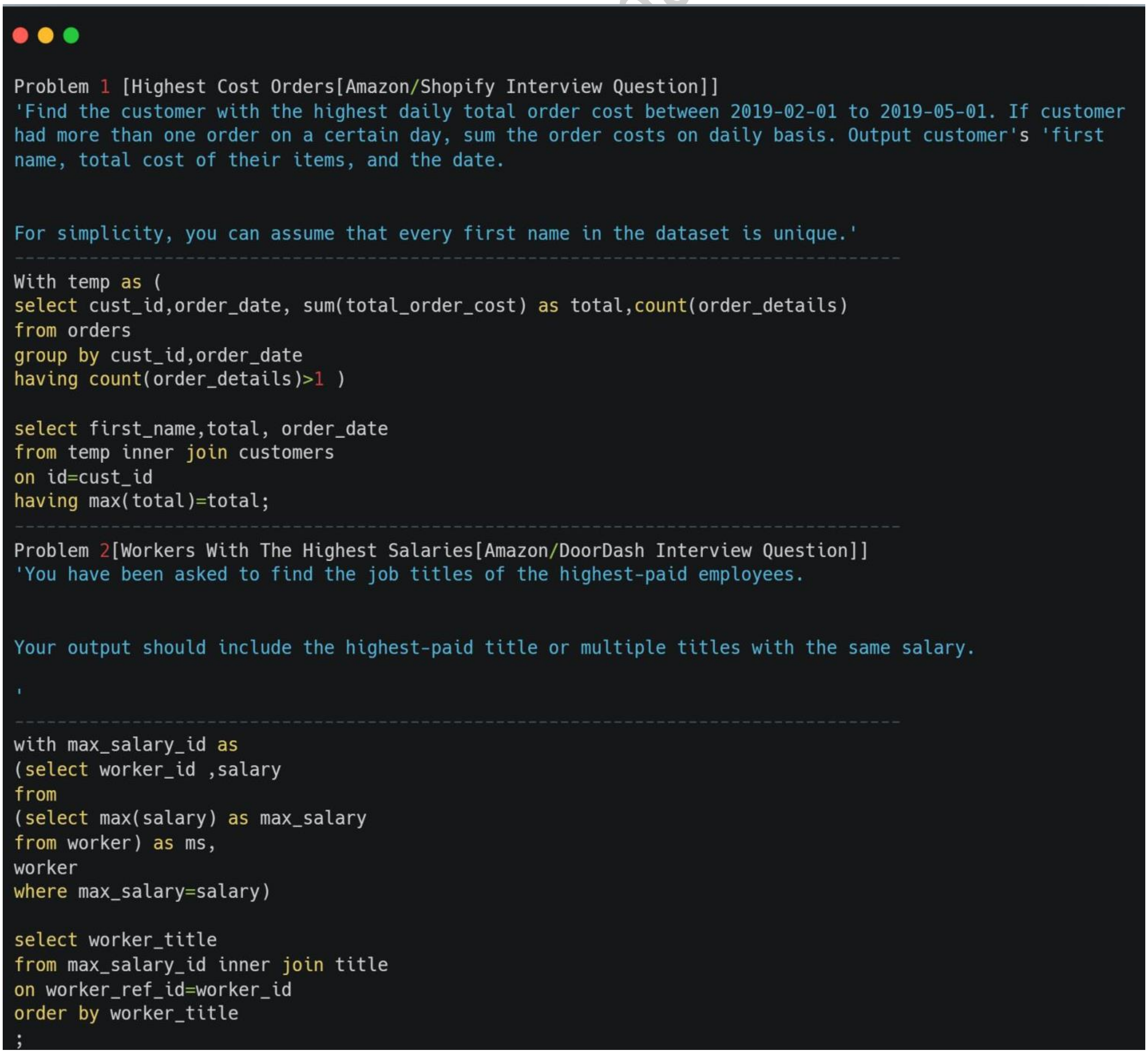
Find the customer with the highest daily total order cost between 2019-02-01 to 2019-05-01. If customer had more than one order on a certain day, sum the order costs on daily basis. Output customer's first name, total cost of their items, and the date.
For simplicity, you can assume that every first name in the dataset is unique.

Problem Link: StrataScratch - Highest Cost Orders

Problem 2: Workers With The Highest Salaries[Amazon/DoorDash Interview Question]

You have been asked to find the job titles of the highest-paid employees.
Your output should include the highest-paid title or multiple titles with the same salary.**Problem Link**: https://lnkd.in/g9zyjdn8

My Solution:



```
Problem 1 [Highest Cost Orders[Amazon/Shopify Interview Question]]
'Find the customer with the highest daily total order cost between 2019-02-01 to 2019-05-01. If customer
had more than one order on a certain day, sum the order costs on daily basis. Output customer's 'first
name, total cost of their items, and the date.


For simplicity, you can assume that every first name in the dataset is unique.'
-------------------------------------------------------------------------------
With temp as (
select cust_id,order_date, sum(total_order_cost) as total,count(order_details)
from orders
group by cust_id,order_date
having count(order_details)>1 )

select first_name,total, order_date
from temp inner join customers
on id=cust_id
having max(total)=total;
-------------------------------------------------------------------------------
Problem 2[Workers With The Highest Salaries[Amazon/DoorDash Interview Question]]
'You have been asked to find the job titles of the highest-paid employees.


Your output should include the highest-paid title or multiple titles with the same salary.

'
-------------------------------------------------------------------------------
with max_salary_id as
(select worker_id ,salary
from
(select max(salary) as max_salary
from worker) as ms,
worker
where max_salary=salary)

select worker_title
from max_salary_id inner join title
on worker_ref_id=worker_id
order by worker_title
;
```

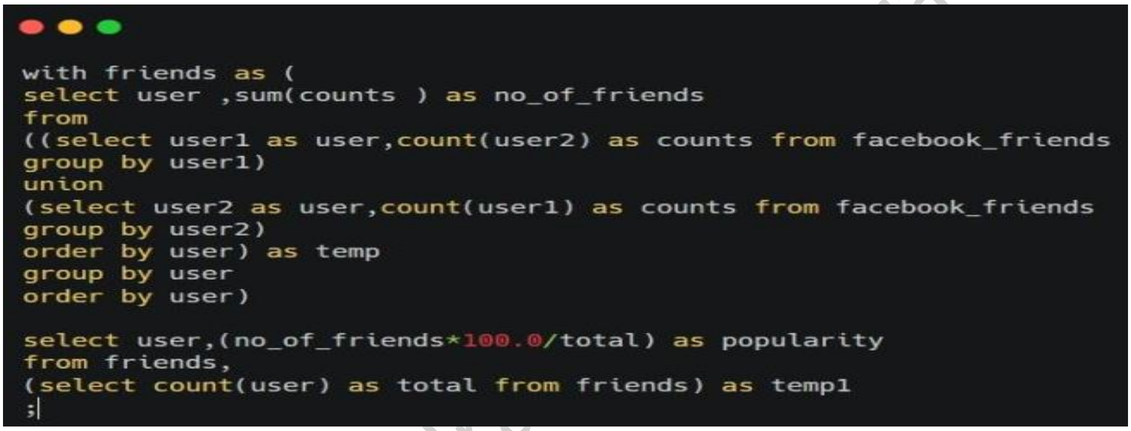
DAY -27

Problem 1: Popularity Percentage [Meta/Facebook Interview Question]

Find the popularity percentage for each user on Meta/Facebook. The popularity percentage is defined as the total number of friends the user has divided by the total number of users on the platform, then converted into a percentage by multiplying by 100.
Output each user along with their popularity percentage. Order records in ascending order by user id.
The 'user1' and 'user2' column are pairs of friends.

Problem Link: https://lnkd.in/gsAqNHQy

My Solution:



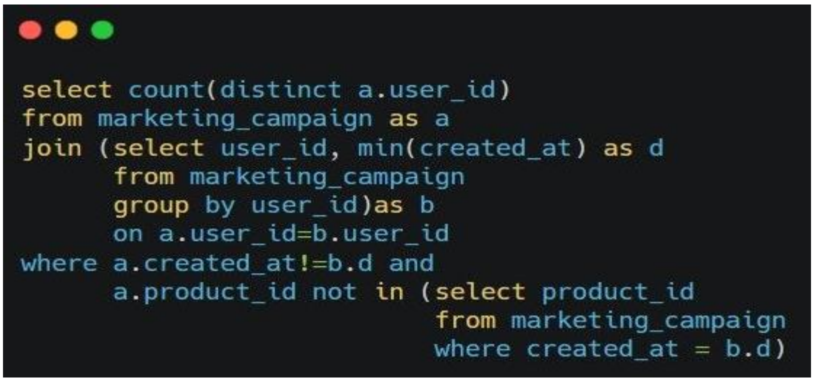
```
with friends as (
select user ,sum(counts ) as no_of_friends
from
((select user1 as user,count(user2) as counts from facebook_friends
group by user1)
union
(select user2 as user,count(user1) as counts from facebook_friends
group by user2)
order by user) as temp
group by user
order by user)

select user,(no_of_friends*100.0/total) as popularity
from friends,
(select count(user) as total from friends) as temp1
;
```

DAY -28

Problem 1: Highest Cost OrdersMarketing Campaign Success [Advanced][Amazon/Shopify Interview Question]

You have a table of in-app purchases by user. Users that make their first in-app purchase are placed in a marketing campaign where they see call-to-actions for more in-app purchases. Find the number of users that made additional in-app purchases due to the success of the marketing campaign.
The marketing campaign doesn't start until one day after the initial in-app purchase so users that only made one or multiple purchases on the first day do not count, nor do we count users that over time purchase only the products they purchased on the first day.

Problem Link: https://lnkd.in/gxjHjM4u



```
select count(distinct a.user_id)
from marketing_campaign as a
join (select user_id, min(created_at) as d
      from marketing_campaign
      group by user_id)as b
      on a.user_id=b.user_id
where a.created_at!=b.d and
      a.product_id not in (select product_id
                           from marketing_campaign
                           where created_at = b.d)
```

DAY -29

Problem 1: [Customer Details[Apple Interview Question]]

Find the details of each customer regardless of whether the customer made an order. Output the customer's 'first name, last name, and the city along with the order details.
You may have duplicate rows in your results due to a customer ordering several of the same items. Sort records based on the customer's 'first name and the order details in ascending order.

Problem Link https://lnkd.in/gaeCg5QC

Problem 2: [Number Of Bathrooms And Bedrooms[Airbnb Interview Question]]

Find the average number of bathrooms and bedrooms for each city's property types. Output the result along with the city name and the property type..

Problem Link: https://lnkd.in/gfCyWAts

Problem 3: [Host Popularity Rental Prices[Airbnb Interview Question]]

You're given a table of rental property searches by users. The table consists of search results and outputs host information for searchers. Find the minimum, average, maximum rental prices for each host's popularity rating. The host's popularity rating is defined as below:
0 reviews: New
1 to 5 reviews: Rising
6 to 15 reviews: Trending Up
16 to 40 reviews: Popular
more than 40 reviews: Hot

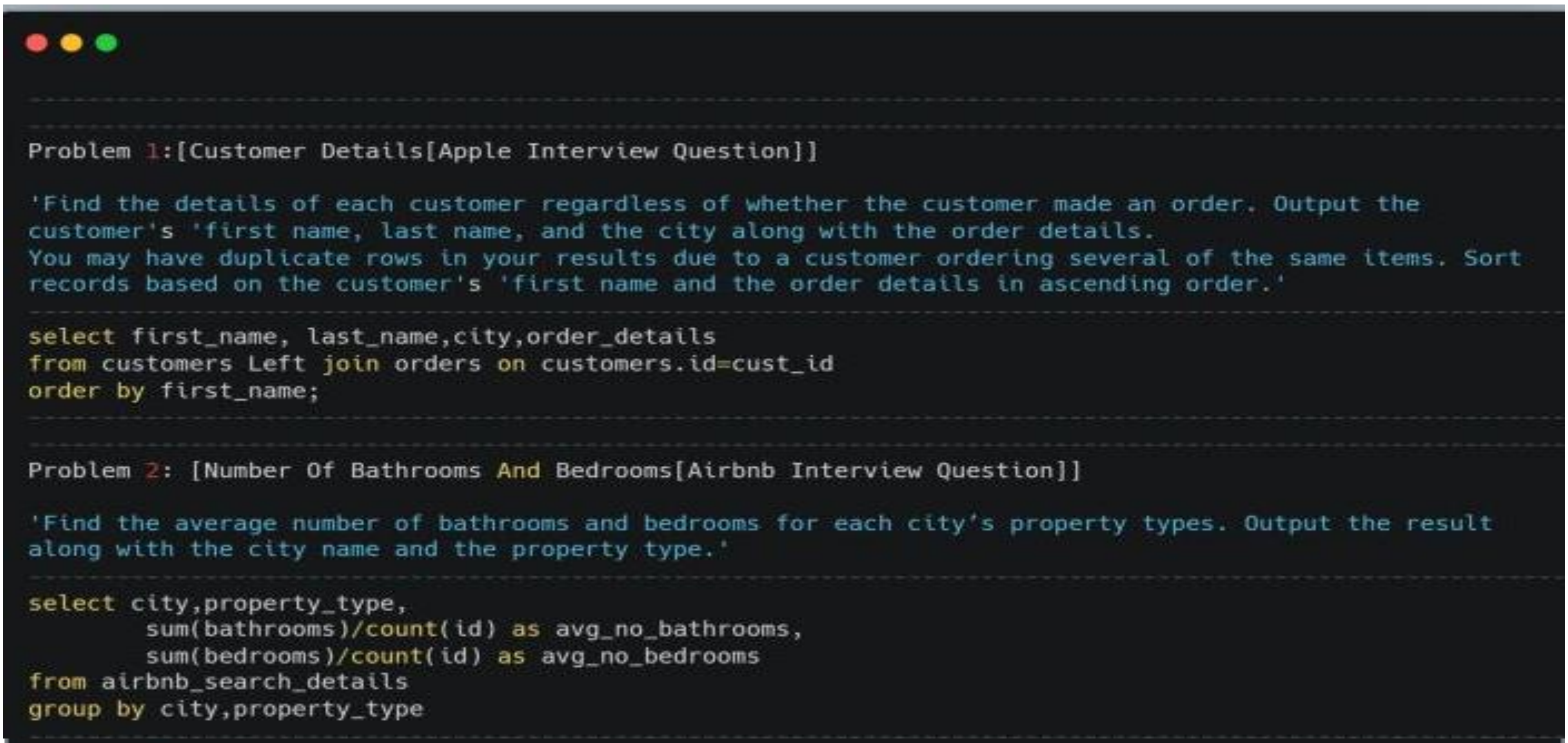
Problem Link: https://lnkd.in/g9mqJNi7

Problem 4: [Customer Revenue In March[Meta/Facebook Interview Question]]

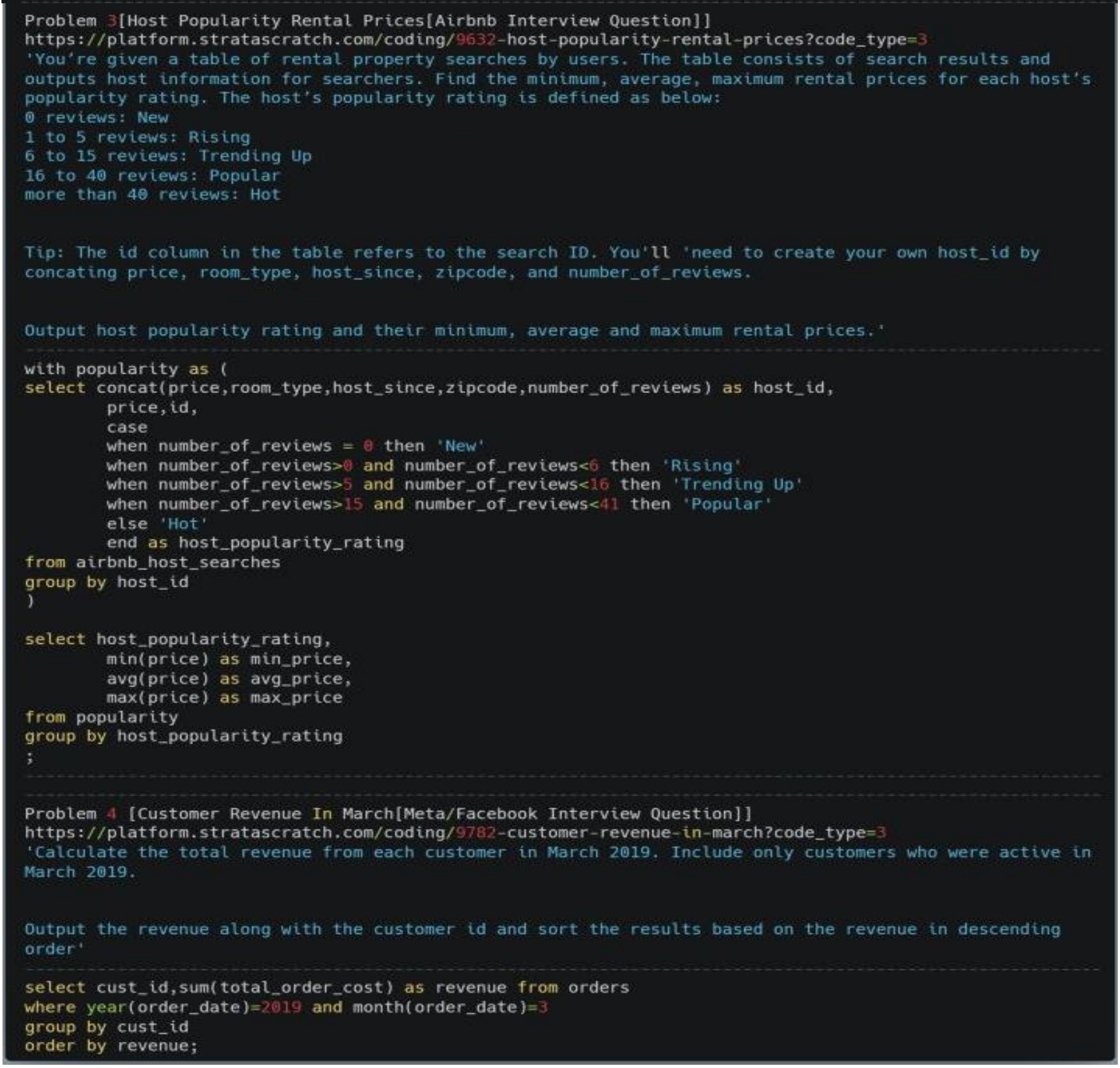
Calculate the total revenue from each customer in March 2019. Include only customers who were active in March 2019.
Output the revenue along with the customer id and sort the results based on the revenue in descending order

Problem Link: https://lnkd.in/gjYsYhQT

My Solution:



```
Problem 1:[Customer Details[Apple Interview Question]]

'Find the details of each customer regardless of whether the customer made an order. Output the
customer's 'first name, last name, and the city along with the order details.
You may have duplicate rows in your results due to a customer ordering several of the same items. Sort
records based on the customer's 'first name and the order details in ascending order.'

select first_name, last_name,city,order_details
from customers Left join orders on customers.id=cust_id
order by first_name;


Problem 2: [Number Of Bathrooms And Bedrooms[Airbnb Interview Question]]

'Find the average number of bathrooms and bedrooms for each city's property types. Output the result
along with the city name and the property type.'

select city,property_type,
        sum(bathrooms)/count(id) as avg_no_bathrooms,
        sum(bedrooms)/count(id) as avg_no_bedrooms
from airbnb_search_details
group by city,property_type
```



```
Problem 3[Host Popularity Rental Prices[Airbnb Interview Question]]
https://platform.stratascratch.com/coding/9632-host-popularity-rental-prices?code_type=3
'You're given a table of rental property searches by users. The table consists of search results and
outputs host information for searchers. Find the minimum, average, maximum rental prices for each host's
popularity rating. The host's popularity rating is defined as below:
0 reviews: New
1 to 5 reviews: Rising
6 to 15 reviews: Trending Up
16 to 40 reviews: Popular
more than 40 reviews: Hot


Tip: The id column in the table refers to the search ID. You'll 'need to create your own host_id by
concating price, room_type, host_since, zipcode, and number_of_reviews.


Output host popularity rating and their minimum, average and maximum rental prices.'

with popularity as (
select concat(price,room_type,host_since,zipcode,number_of_reviews) as host_id,
        price,id,
        case
        when number_of_reviews = 0 then 'New'
        when number_of_reviews>0 and number_of_reviews<6 then 'Rising'
        when number_of_reviews>5 and number_of_reviews<16 then 'Trending Up'
        when number_of_reviews>15 and number_of_reviews<41 then 'Popular'
        else 'Hot'
        end as host_popularity_rating
from airbnb_host_searches
group by host_id
)

select host_popularity_rating,
        min(price) as min_price,
        avg(price) as avg_price,
        max(price) as max_price
from popularity
group by host_popularity_rating
;


Problem 4 [Customer Revenue In March[Meta/Facebook Interview Question]]
https://platform.stratascratch.com/coding/9782-customer-revenue-in-march?code_type=3
'Calculate the total revenue from each customer in March 2019. Include only customers who were active in
March 2019.


Output the revenue along with the customer id and sort the results based on the revenue in descending
order'

select cust_id,sum(total_order_cost) as revenue from orders
where year(order_date)=2019 and month(order_date)=3
group by cust_id
order by revenue;
```

DAY -30

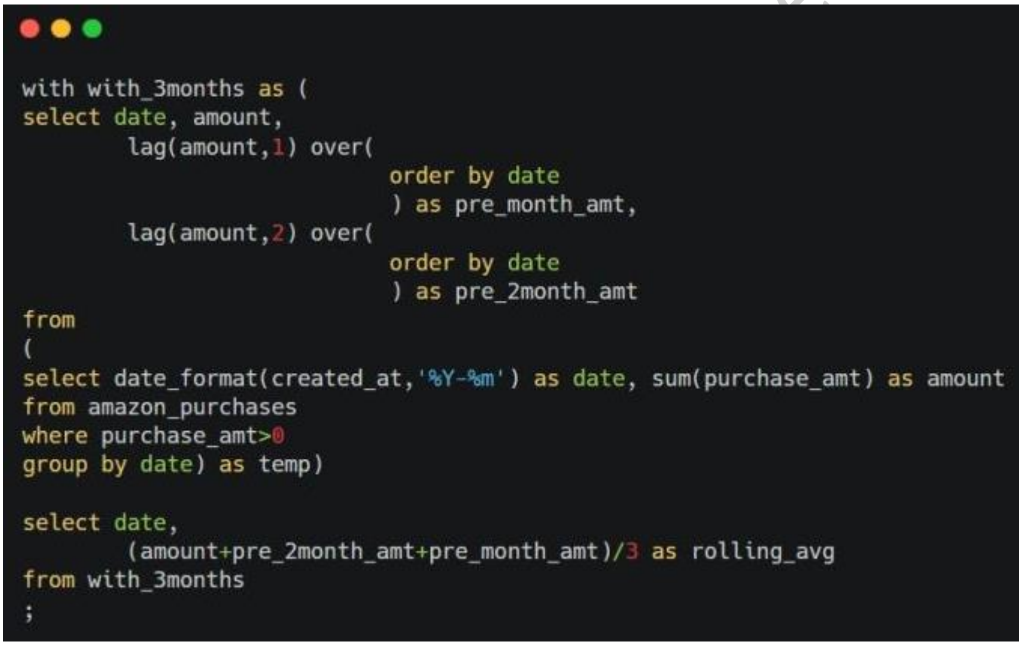
Problem 1: [Revenue Over Time[Amazon interview Question]]

Find the 3-month rolling average of total revenue from purchases given a table with users, their purchase amount, and date purchased. Do not include returns which are represented by negative purchase values. Output the year-month (YYYY-MM) and 3-month rolling average of revenue, sorted from earliest month to latest month.

A 3-month rolling average is defined by calculating the average total revenue from all user purchases for the current month and previous two months. The first two months will not be a true 3-month rolling average since we are not given data from last year. Assume each month has at least one purchase.

Problem Link: https://lnkd.in/gVyx_bcA

My Solution:



```
with with_3months as (
select date, amount,
        lag(amount,1) over(
                            order by date
                            ) as pre_month_amt,
        lag(amount,2) over(
                            order by date
                            ) as pre_2month_amt
from
(
select date_format(created_at,'%Y-%m') as date, sum(purchase_amt) as amount
from amazon_purchases
where purchase_amt>0
group by date) as temp)

select date,
        (amount+pre_2month_amt+pre_month_amt)/3 as rolling_avg
from with_3months
;
```

THANKS FOR READING

My Profile Link: Mahendran Dhanasekaran | LinkedIn