

Day50 - March 26th 2024

1. Started my day as usual
2. Solved one leetcode medium problem
3. Practicing data modeling from manish kumar

Find more about the doc : [Data Modeling 3/26/2024](#)

The screenshot shows a Google Docs interface with a document titled "Data Modeling 3/26/2024". The document content includes:

14. Now lets answer this questions

- ① What is the total no. of available product on 25-11-23 ?
- ① What is the total no. of available product in each store on 26-11-23 ?
- ① Total no. of product available in the in-

-inventory

15. Here we can get 1st question and 2nd question

16. But in the 3rd question...as the quantity in store's varies on given date..we cannot calculate the no.of products in the inventory. ..

17. So this is semi additive

The bottom of the screenshot shows a Windows taskbar with the date and time 8:31 PM 3/26/2024.

- 4.
5. Spent most of the time applying for my STEM OPT

6. ENded my day by solving two complex SQL questions from data lemur

The screenshot shows the DataLemur website interface. At the top, there's a navigation bar with the DataLemur logo and links for 'Practice Interview Questions', 'Learn SQL', and 'Get 1:1 Coaching'. The user's name 'kaushik varma' is displayed in the top right corner. Below the navigation bar, there's a breadcrumb link 'Back to questions' and the title of the question: 'Laptop vs. Mobile Viewership [New York Times SQL Interview Question]'. The question is categorized as 'Easy' and is tagged with 'New York Times'. There are social sharing buttons for Twitter and LinkedIn. The problem description states: 'This is the same question as problem #3 in the SQL Chapter of [Ace the Data Science Interview!](#). Assume you're given the table on user viewership categorised by device type where the three types are laptop, tablet, and phone. Write a query that calculates the total viewership for laptops and mobile devices where mobile is defined as the sum of tablet and phone viewership. Output the total viewership for laptops as `laptop_views` and the total viewership for mobile devices as `mobile_views`. Effective 15 April 2023, the solution has been updated with a more concise and easy-to-understand approach.'

The 'viewership' table is defined as follows:

| Column Name | Type |
|-------------|--------------------------------------|
| user_id | integer |
| device_type | string ('laptop', 'tablet', 'phone') |
| view_time | timestamp |

Example Input:

| user_id | device_type | view_time |
|---------|-------------|---------------------|
| 123 | tablet | 01/02/2022 00:00:00 |
| 125 | laptop | 01/07/2022 00:00:00 |

The SQL query provided in the editor is:

```
1 SELECT sum(case when device_type = 'laptop' then 1 end) as laptop_views,
2 SUM(case when device_type = 'tablet' or device_type = 'phone' then 1 end) as mobile_views
3 FROM viewership;
4
5
```

The output of the query is shown below:

| laptop_views | mobile_views |
|--------------|--------------|
| 2 | 3 |

The bottom of the screenshot shows a Windows taskbar with various application icons and a system tray displaying the date and time as 11:06 PM on 3/26/2024.

Shirdi Sai BabaGeminiJump Game - LJump Game - CFirestre - LeetCNetflixAverage P(1) YouTubeSpace SonNotifications

datalemur.com/questions/sql-average-post-hiatus-1

DataLemur

Practice Interview QuestionsLearn SQLGet 1:1 Coaching

kaushik varmaKV

Back to questions

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

DescriptionSolutionDiscussionSubmissions

EasyFacebookShare on TwitterShare on LinkedIn

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.

p.s. If you've read the [Ace the Data Science Interview](#) and liked it, consider writing us a review?

posts Table:

| Column Name | Type |
|--------------|-----------|
| user_id | integer |
| post_id | integer |
| post_date | timestamp |
| post_content | text |

posts Example Input:

| user_id | post_id | post_date | post_content |
|---------|---------|---------------------|--|
| 151652 | 599415 | 07/10/2021 12:00:00 | Need a hug |
| 661093 | 624356 | 07/29/2021 13:00:00 | Bed. Class 8-12. Work 12-3. Gym 3-5 or 6. Then class 6-10. Another day that's gonna fly by. I miss my girlfriend |
| 661093 | 704354 | 07/04/2021 | Happy 4th of July! |

```
1 SELECT user_id, EXTRACT(day from max(post_date)-min(post_date)) as days_between FROM posts
2 WHERE EXTRACT(year from post_date) = 2021
3 GROUP BY user_id
4 HAVING count(user_id) > 1;
```

PostgreSQL 14Run CodeSubmit

Output

| user_id | days_between |
|---------|--------------|
| 151652 | 307 |
| 661093 | 206 |

Shirdi Sai BabaGeminiJump Game - LJump Game - CFirestre - LeetCNetflixMicrosoft(1) YouTubeAnd To Notifications

datalemur.com/questions/teams-power-users

DataLemur

Practice Interview QuestionsLearn SQLGet 1:1 Coaching

kaushik varmaKV

Back to questions

Teams Power Users [Microsoft SQL Interview Question]

DescriptionSolutionDiscussionSubmissions

EasyMicrosoftShare on TwitterShare on LinkedIn

Write a query to identify the top 2 Power Users who sent the highest number of 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 order based on the count of the messages.

Assumption:

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

messages Table:

| Column Name | Type |
|-------------|----------|
| message_id | integer |
| sender_id | integer |
| receiver_id | integer |
| content | varchar |
| sent_date | datetime |

messages Example Input:

| message_id | sender_id | receiver_id | content | sent_date |
|------------|-----------|-------------|---------------------------|---------------------|
| 901 | 3601 | 4500 | You up? | 08/03/2022 00:00:00 |
| 902 | 4500 | 3601 | Only if you're buying | 08/03/2022 00:00:00 |
| 743 | 3601 | 8752 | I let's take this offline | 06/14/2022 00:00:00 |

```
1 SELECT sender_id, count(sender_id) as message_count FROM messages
2 WHERE EXTRACT(month from sent_date) = 8 and EXTRACT(year from sent_date) = 2022
3 GROUP BY sender_id
4 ORDER BY message_count desc
5 LIMIT 2;
```

PostgreSQL 14Run CodeSubmit

Output

| sender_id | message_count |
|-----------|---------------|
| 3601 | 4 |
| 2520 | 3 |