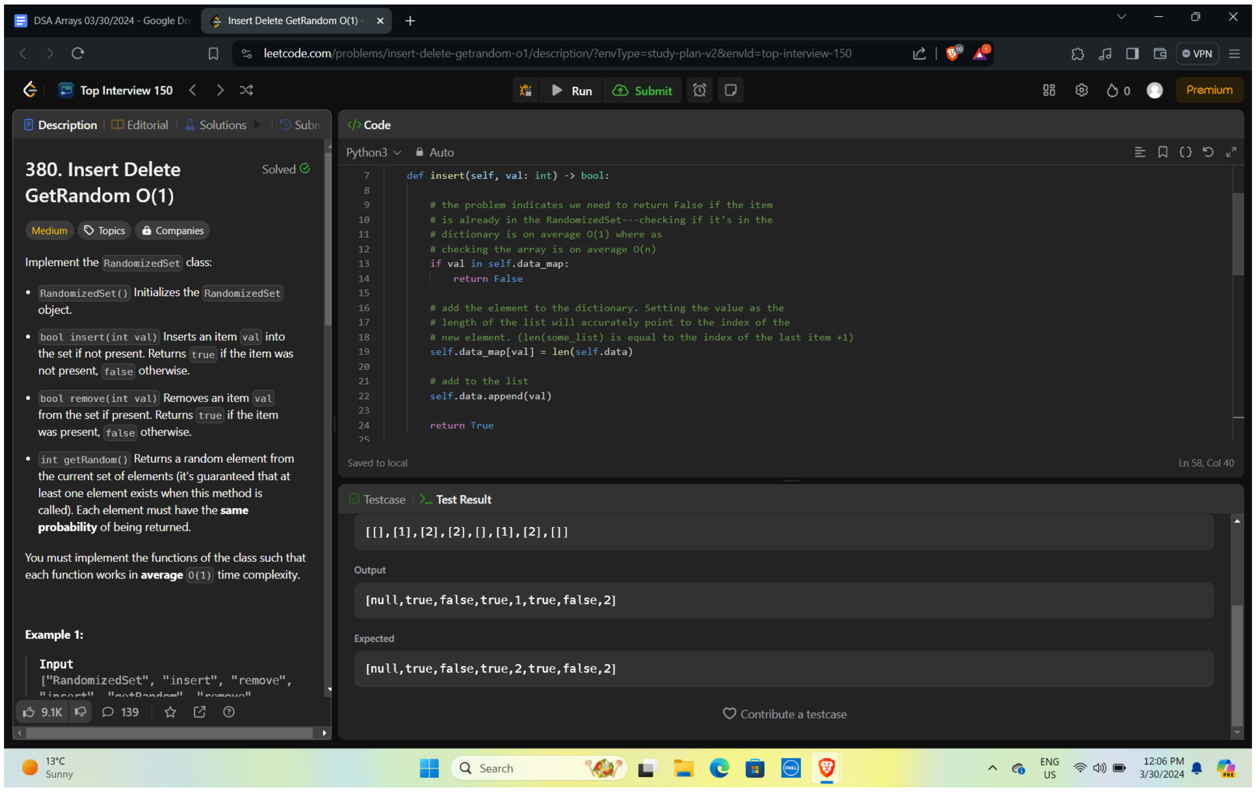
Day54 - March 30th 2024

1. Started my day at 6am
2. Solved 380. Insert Delete GetRandom and added some explanations int the doc : DSA Arrays 03/30/2024
3. 

   380. Insert Delete GetRandom O(1) Solved

   Medium · Topics · Companies

   Implement the `RandomizedSet` class:

   - `RandomizedSet()` Initializes the `RandomizedSet` object.
   - `bool insert(int val)` Inserts an item `val` into the set if not present. Returns `true` if the item was not present, `false` otherwise.
   - `bool remove(int val)` Removes an item `val` from the set if present. Returns `true` if the item was present, `false` otherwise.
   - `int getRandom()` Returns a random element from the current set of elements (it's guaranteed that at least one element exists when this method is called). Each element must have the **same probability** of being returned.

   You must implement the functions of the class such that each function works in **average** `O(1)` time complexity.

   Example 1:

   ```
   Input
   ["RandomizedSet", "insert", "remove",
   ```

   ```python
       def insert(self, val: int) -> bool:

           # the problem indicates we need to return False if the item
           # is already in the RandomizedSet---checking if it's in the
           # dictionary is on average O(1) where as
           # checking the array is on average O(n)
           if val in self.data_map:
               return False

           # add the element to the dictionary. Setting the value as the
           # length of the list will accurately point to the index of the
           # new element. (len(some_list) is equal to the index of the last item +1)
           self.data_map[val] = len(self.data)

           # add to the list
           self.data.append(val)

           return True
   ```

   Testcase

   ```
   [[],[1],[2],[2],[],[1],[2],[]]
   ```

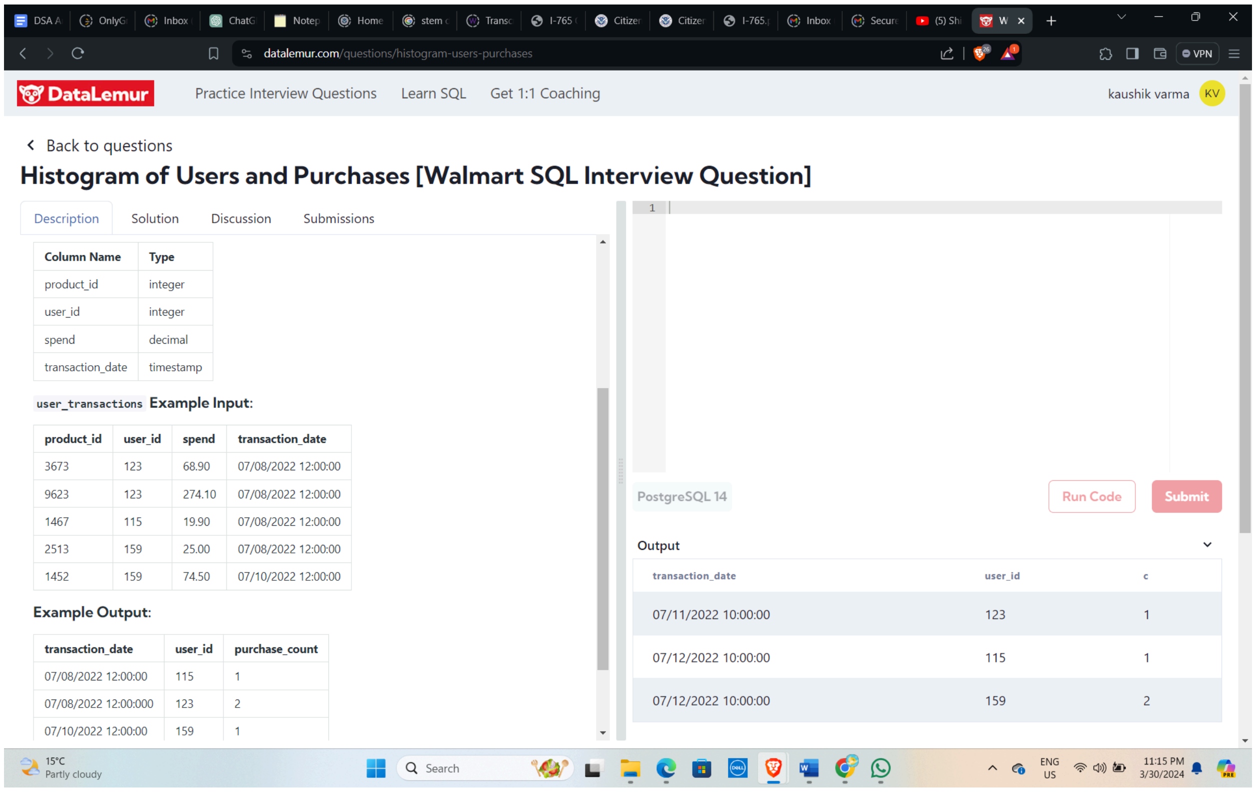
   Output

   ```
   [null,true,false,true,1,true,false,2]
   ```

   Expected

   ```
   [null,true,false,true,2,true,false,2]
   ```
4. Spent my rest of the time collecting documents required for my stem-OPT

5. Ended my day by solving complex SQL questions



Histogram of Users and Purchases [Walmart SQL Interview Question]

Column Name	Type
product_id	integer
user_id	integer
spend	decimal
transaction_date	timestamp

`user_transactions` **Example Input:**

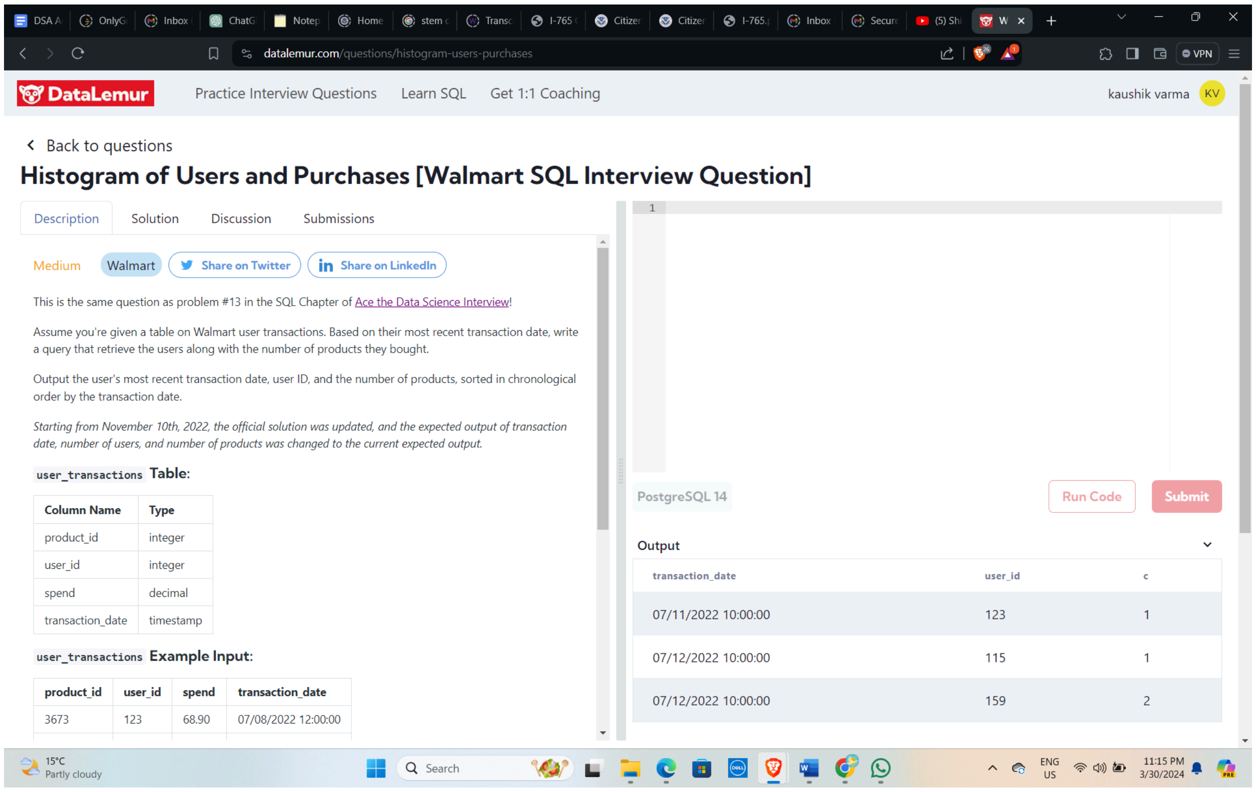
product_id	user_id	spend	transaction_date
3673	123	68.90	07/08/2022 12:00:00
9623	123	274.10	07/08/2022 12:00:00
1467	115	19.90	07/08/2022 12:00:00
2513	159	25.00	07/08/2022 12:00:00
1452	159	74.50	07/10/2022 12:00:00

Example Output:

transaction_date	user_id	purchase_count
07/08/2022 12:00:00	115	1
07/08/2022 12:00:000	123	2
07/10/2022 12:00:00	159	1

Output

```
transaction_date       user_id   c
07/11/2022 10:00:00    123       1
07/12/2022 10:00:00    115       1
07/12/2022 10:00:00    159       2
```



Histogram of Users and Purchases [Walmart SQL Interview Question]

This is the same question as problem #13 in the SQL Chapter of Ace the Data Science Interview!

Assume you're given a table on Walmart user transactions. Based on their most recent transaction date, write a query that retrieve the users along with the number of products they bought.

Output the user's most recent transaction date, user ID, and the number of products, sorted in chronological order by the transaction date.

Starting from November 10th, 2022, the official solution was updated, and the expected output of transaction date, number of users, and number of products was changed to the current expected output.

`user_transactions` **Table:**

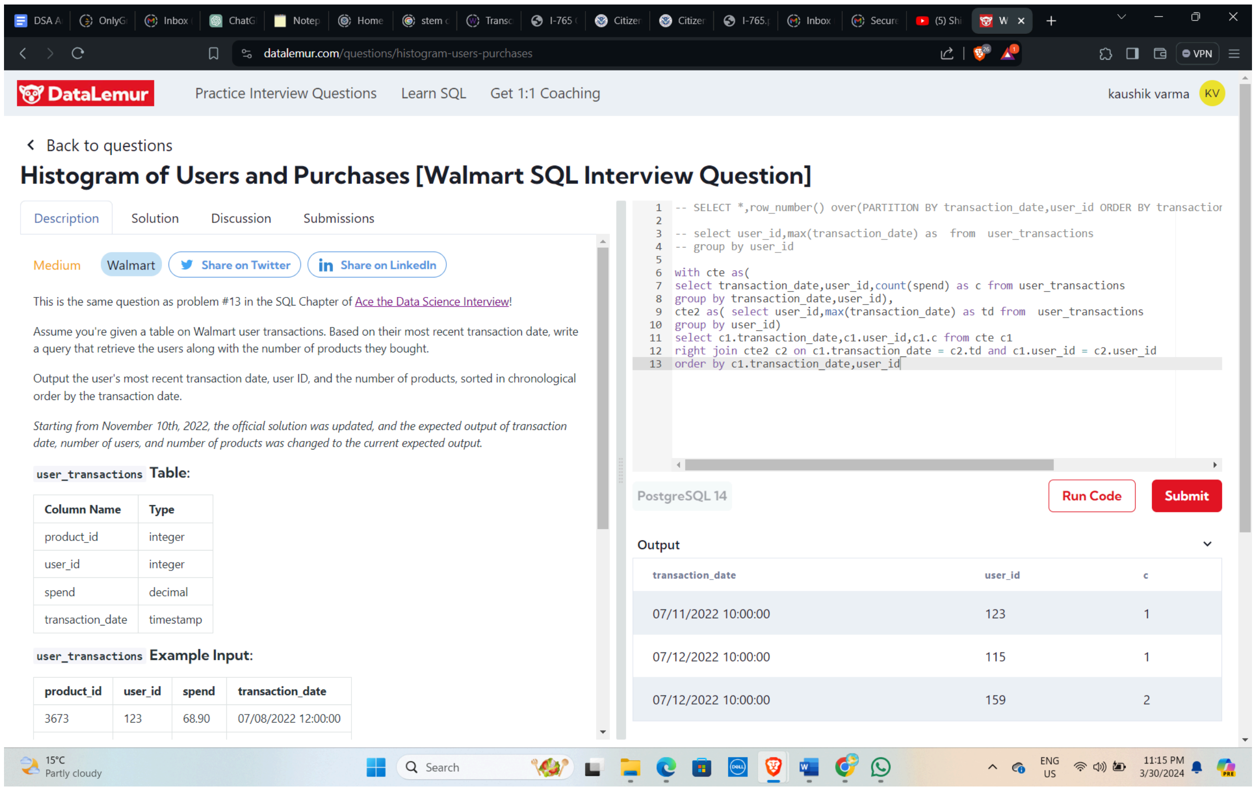
Column Name	Type
product_id	integer
user_id	integer
spend	decimal
transaction_date	timestamp

`user_transactions` **Example Input:**

product_id	user_id	spend	transaction_date
3673	123	68.90	07/08/2022 12:00:00

Output

```
transaction_date       user_id   c
07/11/2022 10:00:00    123       1
07/12/2022 10:00:00    115       1
07/12/2022 10:00:00    159       2
```



datalemur.com/questions/histogram-users-purchases

DataLemur Practice Interview Questions Learn SQL Get 1:1 Coaching kaushik varma

Back to questions

Histogram of Users and Purchases [Walmart SQL Interview Question]

Description Solution Discussion Submissions

Medium Walmart Share on Twitter Share on LinkedIn

This is the same question as problem #13 in the SQL Chapter of Ace the Data Science Interview!

Assume you're given a table on Walmart user transactions. Based on their most recent transaction date, write a query that retrieve the users along with the number of products they bought.

Output the user's most recent transaction date, user ID, and the number of products, sorted in chronological order by the transaction date.

Starting from November 10th, 2022, the official solution was updated, and the expected output of transaction date, number of users, and number of products was changed to the current expected output.

`user_transactions` Table:

Column Name	Type
product_id	integer
user_id	integer
spend	decimal
transaction_date	timestamp

`user_transactions` Example Input:

product_id	user_id	spend	transaction_date
3673	123	68.90	07/08/2022 12:00:00

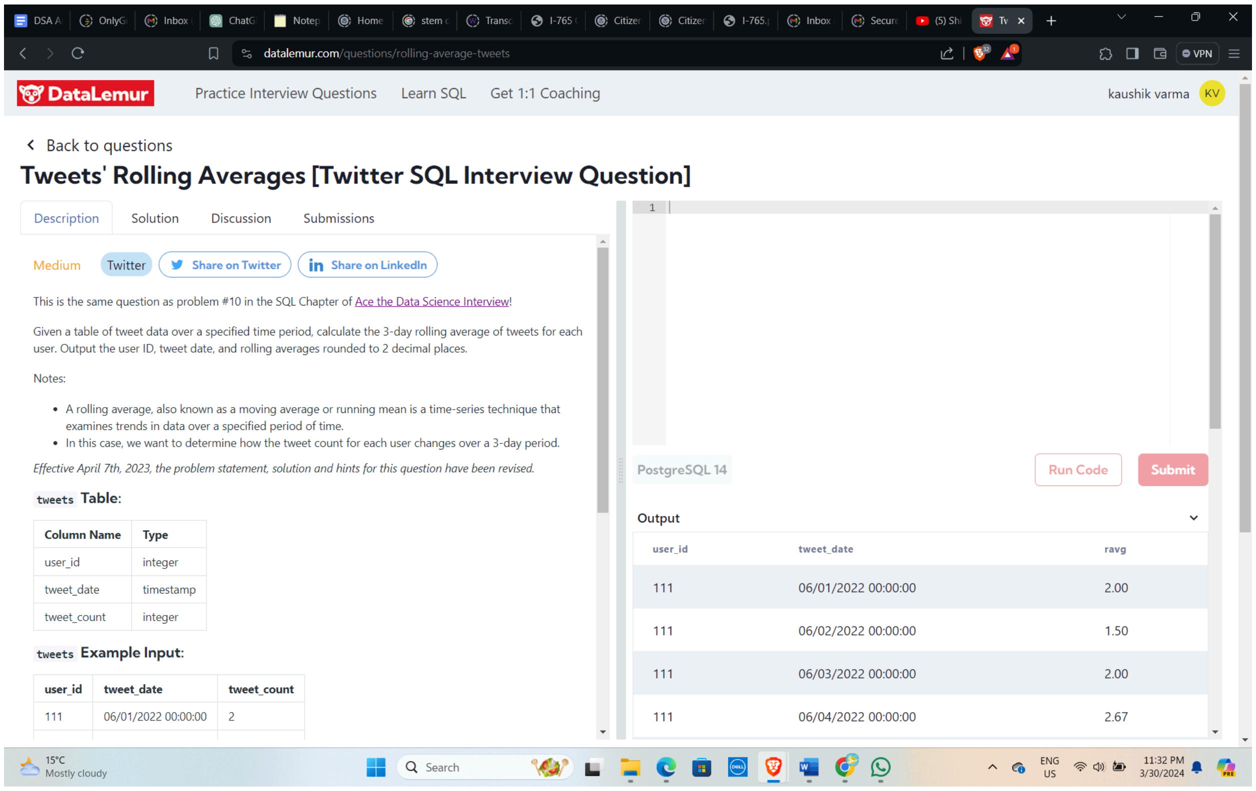
```sql
-- SELECT *,row_number() over(PARTITION BY transaction_date,user_id ORDER BY transactior

-- select user_id,max(transaction_date) as  from  user_transactions
-- group by user_id

with cte as(
select transaction_date,user_id,count(spend) as c from user_transactions
group by transaction_date,user_id),
cte2 as( select user_id,max(transaction_date) as td from  user_transactions
group by user_id)
select c1.transaction_date,c1.user_id,c1.c from cte c1
right join cte2 c2 on c1.transaction_date = c2.td and c1.user_id = c2.user_id
order by c1.transaction_date,user_id
```

PostgreSQL 14 Run Code Submit

Output

```
transaction_date       user_id   c
07/11/2022 10:00:00    123       1
07/12/2022 10:00:00    115       1
07/12/2022 10:00:00    159       2
```

15°C Partly cloudy Search ENG US 11:15 PM 3/30/2024


datalemur.com/questions/rolling-average-tweets
DataLemur
Practice Interview Questions
Learn SQL
Get 1:1 Coaching
kaushik varma
Back to questions

Tweets' Rolling Averages [Twitter SQL Interview Question]

This is the same question as problem #10 in the SQL Chapter of Ace the Data Science Interview!

Given a table of tweet data over a specified time period, calculate the 3-day rolling average of tweets for each user. Output the user ID, tweet date, and rolling averages rounded to 2 decimal places.

Notes:

- A rolling average, also known as a moving average or running mean is a time-series technique that examines trends in data over a specified period of time.
- In this case, we want to determine how the tweet count for each user changes over a 3-day period.

Effective April 7th, 2023, the problem statement, solution and hints for this question have been revised.

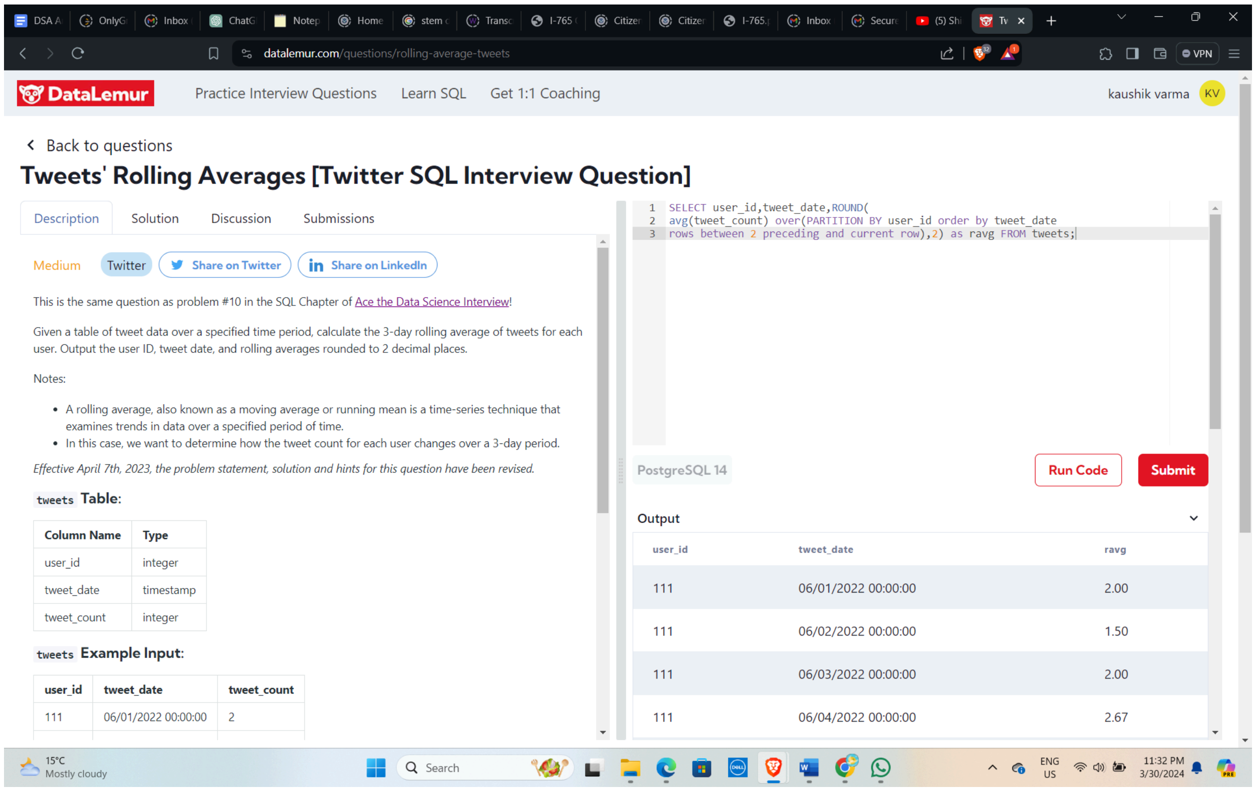
`tweets` **Table:**

Column Name	Type
user_id	integer
tweet_date	timestamp
tweet_count	integer

`tweets` **Example Input:**

user_id	tweet_date	tweet_count
111	06/01/2022 00:00:00	2

user_id	tweet_date	ravg
111	06/01/2022 00:00:00	2.00
111	06/02/2022 00:00:00	1.50
111	06/03/2022 00:00:00	2.00
111	06/04/2022 00:00:00	2.67


Tweets' Rolling Averages [Twitter SQL Interview Question]

This is the same question as problem #10 in the SQL Chapter of Ace the Data Science Interview!

Given a table of tweet data over a specified time period, calculate the 3-day rolling average of tweets for each user. Output the user ID, tweet date, and rolling averages rounded to 2 decimal places.

Notes:

- A rolling average, also known as a moving average or running mean is a time-series technique that examines trends in data over a specified period of time.
- In this case, we want to determine how the tweet count for each user changes over a 3-day period.

Effective April 7th, 2023, the problem statement, solution and hints for this question have been revised.

```sql
SELECT user_id,tweet_date,ROUND(
avg(tweet_count) over(PARTITION BY user_id order by tweet_date
rows between 2 preceding and current row),2) as ravg FROM tweets;
```

user_id	tweet_date	ravg
111	06/01/2022 00:00:00	2.00
111	06/02/2022 00:00:00	1.50
111	06/03/2022 00:00:00	2.00
111	06/04/2022 00:00:00	2.67