

Day48 - March 24th 2024

1. Started my day as usual
2. Solved one leetcode problem

The screenshot shows a web browser window displaying a LeetCode submission for the 'Rotate Array' problem. The submission is accepted and was made by 'kaushik varma N' on March 24, 2024, at 11:59. The runtime is 159 ms, which beats 1.63% of users with Python3. The memory usage is 28.05 MB, which beats 36.82% of users with Python3. A performance graph is visible, showing the submission's performance relative to other users. The code is written in Python3 and uses a recursive helper function 'HelpR' to rotate the array in-place. The test result shows that the code passed all test cases, with a runtime of 74 ms for the first case. The input for the first case is [1, 2, 3, 4, 5, 6, 7].

Rotate Array - LeetCode

leetcode.com/problems/rotate-array/submissions/1212754984/?envType=study-plan-v2&envId=top-interview-150

Top Interview 150

Description Editorial Solutions Submissions

All Submissions

Accepted

kaushik varma N submitted at Mar 24, 2024 11:59

Runtime: 159 ms, Beats 1.63% of users with Python3

Memory: 28.05 MB, Beats 36.82% of users with Python3

Code

```
class Solution:
    def rotate(self, nums: List[int], k: int) -> None:
        """
        Do not return anything, modify nums in-place instead.
        """
        k = k % len(nums)
        self.HelpR(0, len(nums)-1, nums)
        self.HelpR(0, k-1, nums)
        self.HelpR(k, len(nums)-1, nums)

    def HelpR(self, l, r, nums):
        while(l < r):
            nums[l], nums[r] = nums[r], nums[l]
            l, r = l+1, r-1
        return nums
```

Testcase Test Result

Accepted Runtime: 74 ms

Case 1 Case 2

Input

nums = [1, 2, 3, 4, 5, 6, 7]

3. Started revising pyspark from my notes..which helps in my interview

4. Ended my day by solving complex SQL questions from DataLemur

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

Back to questions

Odd and Even Measurements [Google SQL Interview Question]

Description Solution Discussion Submissions

measurement_day	odd_sum	even_sum
07/10/2022 00:00:00	2355.75	1662.74
07/11/2022 00:00:00	2377.12	2480.70
07/12/2022 00:00:00	2903.40	1244.30

Expected

measurement_day	odd_sum	even_sum
07/10/2022 00:00:00	2355.75	1662.74
07/11/2022 00:00:00	2377.12	2480.70
07/12/2022 00:00:00	2903.40	1244.30

TIME **STATUS** **YOUR SUBMISSION**

03/24/2024 22:59	Solved	Copy To Clipboard
03/24/2024 22:58	Error	Copy To Clipboard

```
1 with cte as(
2 SELECT *,row_number() over(partition by EXTRACT(day from measurement_time) order by measurement_time) as
3 --select * from cte
4 select date(measurement_time),sum(case when rn%2 != 0 then (measurement_value) end) as odd_sum
5 ,sum(case when rn%2 = 0 then measurement_value end) as even_sum
6 from cte
7 GROUP BY date(measurement_time)
8 order by date(measurement_time)
9
10 --select * from measurements
11
12
13
```

PostgreSQL 14 Run Code Submit

Output

date	odd_sum	even_sum
07/11/2022 00:00:00	2377.12	2480.70
07/12/2022 00:00:00	2903.40	1244.30
07/10/2022 00:00:00	2355.75	1662.74

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

Back to questions

Odd and Even Measurements [Google SQL Interview Question]

Description Solution Discussion Submissions

Medium Google [Share on Twitter](#) [Share on LinkedIn](#)

This is the same question as problem #28 in the SQL Chapter of [Ace the Data Science Interview!](#)

Assume you're given a table with measurement values obtained from a Google sensor over multiple days with measurements taken multiple times within each day.

Write a query to calculate the sum of odd-numbered and even-numbered measurements separately for a particular day and display the results in two different columns. Refer to the Example Output below for the desired format.

Definition:

- Within a day, measurements taken at 1st, 3rd, and 5th times are considered odd-numbered measurements, and measurements taken at 2nd, 4th, and 6th times are considered even-numbered measurements.

Effective April 15th, 2023, the question and solution for this question have been revised.

measurements Table:

Column Name	Type
measurement_id	integer
measurement_value	decimal
measurement_time	datetime

measurements Example Input:

measurement_id	measurement_value	measurement_time
----------------	-------------------	------------------

```
1
```

PostgreSQL 14 Run Code Submit

Output

date	odd_sum	even_sum
07/11/2022 00:00:00	2377.12	2480.70
07/12/2022 00:00:00	2903.40	1244.30
07/10/2022 00:00:00	2355.75	1662.74

Data Modeling 3/24/24 - Google DGD8 online Debugger | Compiler - CGoogle Interview Question | DataAce the Data Science Interview: 201How to Get the Date from a Datetime

datalemur.com/questions/odd-even-measurements

VPN

DataLemur

Practice Interview QuestionsLearn SQLGet 1:1 Coaching

kaushik varmaKV

Back to questions

Odd and Even Measurements [Google SQL Interview Question]

DescriptionSolutionDiscussionSubmissions

07/10/2022 00:00:00	2355.75	1662.74
07/11/2022 00:00:00	2377.12	2480.70
07/12/2022 00:00:00	2903.40	1244.30

Expected

measurement_day	odd_sum	even_sum
07/10/2022 00:00:00	2355.75	1662.74
07/11/2022 00:00:00	2377.12	2480.70
07/12/2022 00:00:00	2903.40	1244.30

TIME	STATUS	YOUR SUBMISSION
03/24/2024 22:59	Solved	Copy To Clipboard
03/24/2024 22:58	Error	Copy To Clipboard

```
1 with cte as(
2   SELECT *,row_number() over(partition by EXTRACT(day from measurement_time) order by measurement_time) as
3   --select * from cte
4   select date(measurement_time),sum(case when rn%2 != 0 then (measurement_value) end) as odd_sum
5   ,sum(case when rn%2 = 0 then measurement_value end) as even_sum
6   from cte
7   GROUP BY date(measurement_time)
8   order by date(measurement_time)
9
10
11 --select * from measurements
12
13
```

PostgreSQL 14

Run CodeSubmit

Output

date	odd_sum	even_sum
07/11/2022 00:00:00	2377.12	2480.70
07/12/2022 00:00:00	2903.40	1244.30
07/10/2022 00:00:00	2355.75	1662.74

11°C Cloudy

Search

ENG US

11:00 PM 3/24/2024

Data Modeling 3/24/24GD8 online Debugger | Amazon SQL InterviewAce the Data Science InterviewHow to Get the Date from SQL Window FunctionsPostgreSQL DATEDIFF - 5

datalemur.com/questions/total-utilization-time

kaushik varmaKV

Back to questions

Server Utilization Time [Amazon SQL Interview Question]

DescriptionSolutionDiscussionSubmissions

HardAmazonShare on TwitterShare on LinkedIn

Amazon Web Services (AWS) is powered by fleets of servers. Senior management has requested data-driven solutions to optimize server usage.
Write a query that calculates the total time that the fleet of servers was running. The output should be in units of **full days**.
Assumptions:

- Each server might start and stop several times.
- The total time in which the server fleet is running can be calculated as the sum of each server's uptime.

server_utilization Table:

Column Name	Type
server_id	integer
status_time	timestamp
session_status	string

server_utilization Example Input:

server_id	status_time	session_status
1	08/02/2022 10:00:00	start
1	08/04/2022 10:00:00	stop
2	08/17/2022 10:00:00	start
2	08/24/2022 10:00:00	stop

PostgreSQL 14

Run CodeSubmit

Output

total_uptime_days

26

10°C Cloudy

Search

ENG US

11:35 PM 3/24/2024

Data Modeling 3/24/24GD8 online Debugger | Amazon SQL InterviewAce the Data Science InterviewHow to Get the Date from SQL Window FunctionsPostgreSQL DATEDIFF - 5

datalemur.com/questions/total-utilization-time

kaushik varmaKV

Back to questions

Server Utilization Time [Amazon SQL Interview Question]

DescriptionSolutionDiscussionSubmissions

Accepted

Congrats 🎉 - Share this problem, and your solution, on LinkedIn or Twitter!
Share on TwitterShare on LinkedIn

In your post, don't forget to tag Nick Singh, so that he can comment on and share your post with his audience of 150k+ followers on LinkedIn and 25k+ followers on Twitter (which will give your post and profile more visibility!)

Output

total_uptime_days

26

Expected

total_uptime_days

26

TIMESTATUSYOUR SUBMISSION

03/24/2024 23:29SolvedCopy To Clipboard

1with cte as
2SELECT server_id,session_status,lead(session_status,1)
3OVER(PARTITION BY server_id order by server_id,status_time) as next_session_status,
4status_time,lead(status_time,1)
5DIVER(PARTITION BY server_id order by server_id,status_time) as le FROM server_utilization
6select sum(extract(day from le)-extract(day from status_time)) as total_uptime_days
7from cte
8where session_status=next_session_status
9
10
11
12

PostgreSQL 14

Run CodeSubmit

Output

total_uptime_days

26

10°C Cloudy

Search

ENG US

11:35 PM 3/24/2024