

QUESTION 1

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:

1. $\text{time sending} / (\text{time sending} + \text{time opening})$
2. $\text{time opening} / (\text{time sending} + \text{time opening})$
3. To avoid integer division in percentages, multiply by 100.0 and not 100.

Sending vs. Opening Snaps [Snapchat SQL Interview Question]

Description Solution Discussion Submissions

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

[Share on Twitter](#) [Share 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 90k+ followers on LinkedIn and 6k+ followers on Twitter (which will give your post and profile more visibility!)

Output

age_bucket	send_perc	open_perc
21-25	54.31	45.69
26-30	82.26	17.74
31-35	37.84	62.16

Expected

age_bucket	send_perc	open_perc
21-25	54.31	45.69
26-30	82.26	17.74
31-35	37.84	62.16

SQL Query:

```
1 WITH snap_cte AS (
2   SELECT
3     age_breakdown.age_bucket,
4     SUM(
5       CASE WHEN activities.activity_type = 'send'
6         THEN activities.time_spent
7       ELSE 0
8     ) AS time_send,
9     SUM(
10      CASE WHEN activities.activity_type = 'open'
11        THEN activities.time_spent
12      ELSE 0
13    ) AS time_open,
14    SUM(activities.time_spent) AS total_time_spent
15  FROM activities
16  INNER JOIN age_breakdown
17    ON activities.user_id = age_breakdown.user_id
18  WHERE activities.activity_type IN ('send', 'open')
19  GROUP BY age_breakdown.age_bucket
20 )
21
22 SELECT
23   age_bucket,
24   ROUND(100.0 * (time_send / total_time_spent), 2) AS send_perc,
25   ROUND(100.0 * (time_open / total_time_spent), 2) AS open_perc
26 FROM snap_cte;
```

PostgreSQL 14

[Run Code](#) [Submit](#)

QUESTION 2

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.

The screenshot shows a web browser with multiple tabs open. The active tab is 'datalemur.com/questions/sql-page-with-no-likes'. The page title is 'Page With No Likes [Facebook SQL Interview Question]'. The user 'Jermaine Sangiwa' is logged in. The page has tabs for 'Description', 'Solution', 'Discussion', and 'Submissions'. The 'Submissions' tab is active, showing a list of submissions. The first submission is 'Accepted' and shows the following SQL query:

```
1 SELECT
2   pages.page_id
3
4 FROM pages
5 LEFT JOIN page_likes
6 ON pages.page_id = page_likes.page_id
7
8 WHERE
9   page_likes.liked_date IS NULL
10
11 ORDER BY
12   pages.page_id ASC;
```

The output of the query is shown in a table with the following data:

page_id
20701
32728

The 'Expected' output is also shown, matching the actual output. The submission is marked as 'Solved' and the time taken is '02/13/2023 20:18'. The status is 'Solved' and the submission is 'Copy To Clipboard'.