DESCRIPTION

Table: Activity

```
+--------------+---------+
| Column Name  | Type    |
+--------------+---------+
| player_id    | int     |
| device_id    | int     |
| event_date   | date    |
| games_played | int     |
+--------------+---------+
```

(player_id, event_date) is the primary key (combination of columns with unique values) of this table.

This table shows the activity of players of some games.

Each row is a record of a player who logged in and played a number of games (possibly 0) before logging out on someday using some device.

Write a solution to report the **fraction** of players that logged in again on the day after the day they first logged in, **rounded to 2 decimal places**. In other words, you need to count the number of players that logged in for at least two consecutive days starting from their first login date, then divide that number by the total number of players.

The result format is in the following example.

Example 1:

Input:

Activity table:

```
+-----------+-----------+------------+--------------+
| player_id | device_id | event_date | games_played |
+-----------+-----------+------------+--------------+
| 1         | 2         | 2016-03-01 | 5            |
| 1         | 2         | 2016-03-02 | 6            |
| 2         | 3         | 2017-06-25 | 1            |
| 3         | 1         | 2016-03-02 | 0            |

| 3         | 4         | 2018-07-03 | 5            |

+-----------+-----------+------------+--------------+
```

Output:

```
+-----------+

| fraction  |

+-----------+

| 0.33      |

+-----------+
```

Explanation:

Only the player with id 1 logged back in after the first day he had logged in so the answer is 1/3 = 0.33

SOLUTION

MySQL:

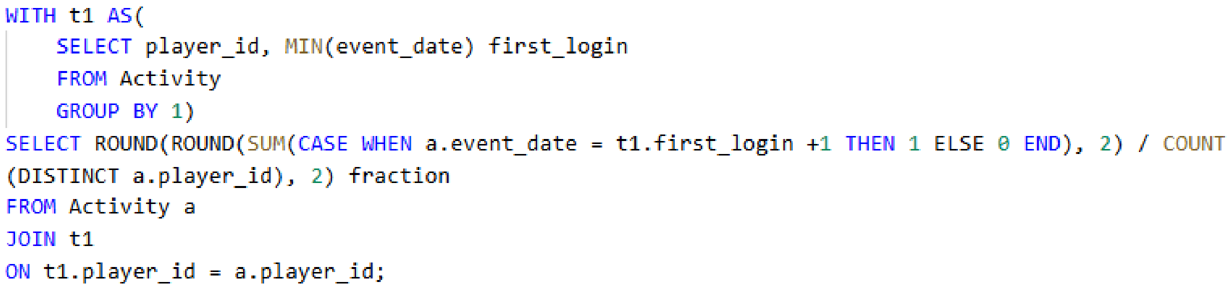
- Using CTE, select player_id and first_login using MIN() and grouped by player_id
- Calculate total players who made two consecutive days logins using DATEDIFF(), IF() and SUM()
- Calculate fraction by dividing the above sum with total number of distinct players, and round the result to 2 decimals using ROUND()
- Join t and Activity

```
WITH t AS
(SELECT player_id, MIN(event_date) first_login
FROM Activity
GROUP BY player_id)
SELECT ROUND(SUM(IF(DATEDIFF(a.event_date,t.first_login) = 1, 1, 0)) / COUNT(DISTINCT
a.player_id), 2) fraction
FROM t
JOIN Activity a
ON t.player_id = a.player_id;
```

PostgreSQL:

- Using CTE, select player_id and first_login using MIN() and grouped by player_id
- Calculate total number of players who made two consecutive days logins using CASE and SUM()
- Calculate fraction by dividing the above sum with total number of distinct players, and round the result to 2 decimals using ROUND()

- Join t1 and Activity



```
WITH t1 AS(
    SELECT player_id, MIN(event_date) first_login
    FROM Activity
    GROUP BY 1)
SELECT ROUND(ROUND(SUM(CASE WHEN a.event_date = t1.first_login +1 THEN 1 ELSE 0 END), 2) / COUNT
(DISTINCT a.player_id), 2) fraction
FROM Activity a
JOIN t1
ON t1.player_id = a.player_id;
```