Codility_

CodeCheck Report: training5DWR3C-YJA

Test Name:

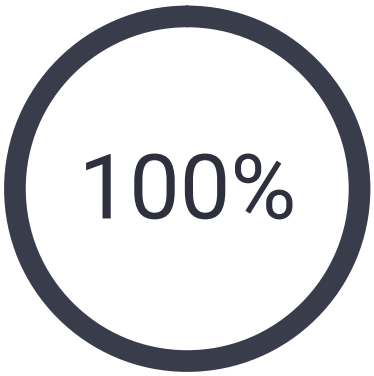
Check out Codility training tasks

Summary Timeline

Tasks summary

Task	Time spent	Score
SqlEventsDelta SQL (PostgreSQL)	6 min	100%

Total score



Tasks Details

1. SqlEventsDelta

Easy

Compute the difference between the latest and the second latest value for each event type.

Task Score	Correctness	Performance
100%	100%	Not assessed

Task description

Given a table `events` with the following structure:

```
create table events (
    event_type integer not null,
    value integer not null,
    time timestamp not null,
    unique(event_type, time)
);
```

write an SQL query that, for each `event_type` that has been registered more than once, returns the difference between the latest (i.e. the most recent in terms of `time`) and the second latest `value`. The table should be ordered by `event_type` (in ascending order).

Solution

Programming language used:	SQL (PostgreSQL)
Total time used:	6 minutes
Effective time used:	6 minutes
Notes:	*not defined yet*

Task timeline

For example, given the following data:

```
   event_type | value       | time

------------+-------------+-----------------
---
   2          | 5           | 2015-05-09
12:42:00
   4          | -42         | 2015-05-09
13:19:57
   2          | 2           | 2015-05-09
14:48:30
   2          | 7           | 2015-05-09
12:54:39
   3          | 16          | 2015-05-09
13:19:57
   3          | 20          | 2015-05-09
15:01:09
```

your query should return the following rowset:

```
   event_type | value
  ------------+------------
   2          | -5
   3          | 4
```

For the `event_type` 2, the latest `value` is 2 and the second latest `value` is 7, so the difference between them is -5.

The names of the columns in the rowset don't matter, but their order does.

Copyright 2009–2023 by Codility Limited. All Rights Reserved.
Unauthorized copying, publication or disclosure prohibited.

07:57:32 08:03:29

Code: 08:03:29 UTC, sql-postgres, final, score: **100** show code in pop-up

```
1  -- Implement your solution here
2  SELECT
3      event_type,
4      latest_value - second_latest_value AS
5  FROM (
6      -- Subquery to get the latest and sec
7      SELECT
8          event_type,
9          MAX(CASE WHEN rn = 1 THEN value E
10         MAX(CASE WHEN rn = 2 THEN value E
11     FROM (
12         -- Subquery to rank the records b
13         SELECT
14             event_type,
15             value,
16             ROW_NUMBER() OVER(PARTITION B
17         FROM events
18     ) ranked_events
19     GROUP BY event_type
20 ) event_values
21 WHERE
22     -- Filter out event_types that only h
23     latest_value IS NOT NULL
24     AND second_latest_value IS NOT NULL
25 ORDER BY event_type;
26
27
```

Analysis summary

The solution obtained perfect score.

Analysis

expand all	Example tests	
▶	example example test	✔ OK
expand all	**Correctness tests**	
▶	simple_one_type One type of events	✔ OK
▶	extreme_unique_types Unique types of events	✔ OK
▶	extreme_empty_data Empty data set	✔ OK
▶	simple Event types repeating various number of times	✔ OK
▶	cyclic_polling N=16, four event types, four series of events of different types	✔ OK

▶	bracketed_polling N=12, two rounds of polling with reversed order	✔ OK
▶	single_event_type N=12, one type of events	✔ OK
▶	double_events N=12; two event types mixed	✔ OK
▶	random1 random sequence; N=100, 100 event types	✔ OK
▶	random2 random sequence; N=100, 12 event types	✔ OK
▶	random3 random sequence; N=100, 4 event types	✔ OK