Codility_

CodeCheck Report: trainingJNR254-G9B

Test Name:

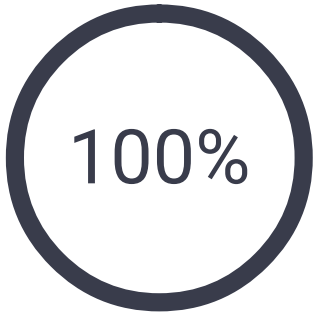
Check out Codility training tasks

Summary Timeline

Tasks summary

Task	Time spent	Score
SqlEventsDelta SQL (PostgreSQL) ⚠	42 min	100%

Total score



Tasks Details

Easy

1. SqlEventsDelta

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).

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.

Solution

Programming language used:	SQL (PostgreSQL)
Total time used:	42 minutes
Effective time used:	42 minutes
Notes:	*not defined yet*

Task timeline

17:48:47 18:30:25

Code: 18:30:25 UTC, sql-postgres, final, score: **100** show code in pop-up

```
1  WITH events_numbered AS (
2      SELECT
3          event_type
4          , value
5          , row_number() OVER (PARTITION BY event_type
6      FROM events
7  )
8  SELECT
9      e1.event_type
10     , e1.value - e2.value
11 FROM events_numbered e2
12     INNER JOIN events_numbered e1
13         ON e2.event_type = e1.event_type
```

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.

```
14          AND e2.row_num = 2
15          AND e1.row_num = 1
16  ORDER BY event_type
```

Analysis summary

The solution obtained perfect score.

Analysis

collapse all — Example tests

▼ example — ✔ OK
example test

1. 0.044 s OK
function result:
```
+---+----+
| 2 | -5 |
| 3 |  4 |
+---+----+
```

collapse all — Correctness tests

▼ simple_one_type — ✔ OK
One type of events

1. 0.044 s OK

▼ extreme_unique_types — ✔ OK
Unique types of events

1. 0.044 s OK
2. 0.044 s OK

▼ extreme_empty_data — ✔ OK
Empty data set

1. 0.044 s OK
2. 0.040 s OK

▼ simple — ✔ OK
Event types repeating various number of times

1. 0.044 s OK

▼ cyclic_polling — ✔ OK
N=16, four event types, four series of events of different types

1. 0.040 s OK

▼ bracketed_polling — ✔ OK
N=12, two rounds of polling with reversed order

1. 0.044 s OK
2. 0.044 s OK

▼ single_event_type — ✔ OK
N=12, one type of events

1. 0.044 s OK

▼ double_events — ✔ OK
N=12; two event types mixed

1. 0.044 s OK

▼ random1 — ✔ OK
random sequence; N=100, 100 event types

1. 0.044 s OK

▼ random2 random sequence; N=100, 12 event types	✔ OK
1. 0.040 s OK	
▼ random3 random sequence; N=100, 4 event types	✔ OK
1. 0.044 s OK	