Codility_

CodeCheck Report: training5C4G5M-E37

Test Name:

Check out Codility training tasks

Summary Timeline AI Assistant Transcript

Tasks summary

Task	Time spent	Score
SqlEventsDelta ⚠ SQL (PostgreSQL)	17 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).

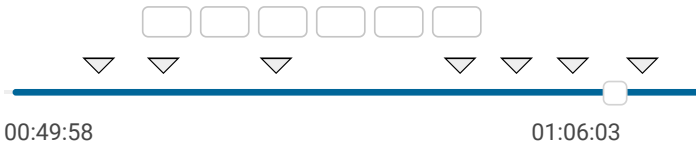
For example, given the following data:

```
 event_type | value      | time
------------+------------+--------------------
```

Solution

Programming language used:	SQL (PostgreSQL)
Total time used:	17 minutes
Effective time used:	17 minutes
Notes:	*not defined yet*

Task timeline



```
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.

Code: 01:06:03 UTC, sql-postgres, final, score: **100** show code in pop-up

```sql
-- Implement your solution here
;WITH events_CTE AS
(

    SELECT *,
            LEAD(value) OVER (PARTITION BY event_ty
            ROW_NUMBER() OVER (PARTITION BY event_t
    FROM events
    WHERE event_type IN (
                            SELECT event_type
                            FROM events
                            GROUP BY event_type
                            HAVING(COUNT(event_type
                        )
)

SELECT event_type, (value - SecondLatest) AS value
FROM events_CTE
WHERE RowNumber = 1


```

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	