Codility_

CodeCheck Report: trainingWC2ZQF-G9V

Test Name:

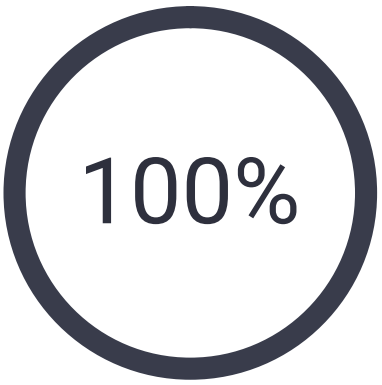
Check out Codility training tasks

Summary Timeline

Tasks summary

Task	Time spent	Score
SqlWorldCup SQL (PostgreSQL)	8 min	100%

Total score



Tasks Details

1. SqlWorldCup

Medium

Given a list of matches in a group stage of the soccer World Cup, compute the number of points each team currently has.

Task Score	Correctness	Performance
100%	100%	Not assessed

Task description

You are given two tables, `teams` and `matches`, with the following structures:

```
create table teams (
    team_id integer not null,
    team_name varchar(30) not null,
    unique(team_id)
);
```

Solution

Programming language used:	SQL (PostgreSQL)
Total time used:	8 minutes
Effective time used:	8 minutes
Notes:	*not defined yet*

```
create table matches (
    match_id integer not null,
    host_team integer not null,
    guest_team integer not null,
    host_goals integer not null,
    guest_goals integer not null,
    unique(match_id)
);
```

Each record in the table `teams` represents a single soccer team. Each record in the table `matches` represents a finished match between two teams. Teams (`host_team`, `guest_team`) are represented by their IDs in the `teams` table (`team_id`). No team plays a match against itself. You know the result of each match (that is, the number of goals scored by each team).

You would like to compute the total number of points each team has scored after all the matches described in the table. The scoring rules are as follows:

- If a team wins a match (scores strictly more goals than the other team), it receives three points.
- If a team draws a match (scores exactly the same number of goals as the opponent), it receives one point.
- If a team loses a match (scores fewer goals than the opponent), it receives no points.

Write an SQL query that returns a ranking of all teams (`team_id`) described in the table `teams`. For each team you should provide its name and the number of points it received after all described matches (`num_points`). The table should be ordered by `num_points` (in decreasing order). In case of a tie, order the rows by `team_id` (in increasing order).

For example, for:

```
teams:

 team_id | team_name
---------+---------------
 10      | Give
 20      | Never
 30      | You
 40      | Up
 50      | Gonna


matches:

 match_id | host_team | guest_team |
host_goals | guest_goals

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

Task timeline

08:06:55 08:14:20

Code: 08:14:20 UTC, sql-postgres, final, score: **100** show code in pop-up

```
1  -- Implement your solution here
2  WITH host_matches AS (
3      -- calculate points from matches whe
4      SELECT host_team AS team_id,
5             CASE
6                 WHEN host_goals > guest_(
7                 WHEN host_goals = guest_(
8                 ELSE 0
9             END AS points
10     FROM matches
11 ),
12
13 guest_matches AS (
14     -- calculate points from matches whe
15     SELECT guest_team AS team_id,
16            CASE
17                WHEN guest_goals > host_(
18                WHEN guest_goals = host_(
19                ELSE 0
20            END AS points
21     FROM matches
22 ),
23
24 total_points AS (
25     -- combine points from host and gues
26     SELECT * FROM host_matches
27     UNION ALL
28     SELECT * FROM guest_matches
29 )
30
31 -- sum total points and join with teams
32 SELECT t.team_id, t.team_name, COALESCE(
33 FROM teams t
34 LEFT JOIN total_points tp ON t.team_id =
35 GROUP BY t.team_id, t.team_name
36 ORDER BY num_points DESC, t.team_id ASC
37
```

Analysis summary

The solution obtained perfect score.

Analysis

expand all	Example tests	
▶	example example test	✔ OK
expand all	Correctness tests	

```
------+-------------
  1         | 30         | 20          | 1
| 0
  2         | 10         | 20          | 1
| 2
  3         | 20         | 50          | 2
| 2
  4         | 10         | 30          | 1
| 0
  5         | 30         | 50          | 0
| 1
```

your query should return:

```
  team_id | team_name | num_points
 ---------+-----------+------------
  20      | Never     | 4
  50      | Gonna     | 4
  10      | Give      | 3
  30      | You       | 3
  40      | Up        | 0
```


▶	both_tables_empty no teams nor matches	✔ OK
▶	no_matches some teams, but no matches	✔ OK
▶	one_match many teams, only one match	✔ OK
▶	simple_only_draws all teams drawed their matches	✔ OK
▶	simple_no_draws no draw in any match	✔ OK
▶	simple_distinct_scores all teams have different score at the end, also no need to reorder anything	✔ OK
▶	many_draws test where teams with many draws are better than a single win	✔ OK
▶	same_scores test where many matches have been conducted and all teams tie	✔ OK
▶	only_two_teams_playing many teams, only two of them playing 50 matches	✔ OK
▶	random totally random test, 8 teams, 100 matches	✔ OK
▶	random_some_teams totally random test, 20 teams, 100 matches between 8 first teams	✔ OK