Codility_

CodeCheck Report: trainingS6GNJZ-B2X
Test Name:

Summary   Timeline   🤖 AI Assistant Transcript

Tasks summary

Task		Time spent	Score
SqlWorldCup ⚠️ SQL (PostgreSQL)		30 min	100%

Total score

100%

Tasks Details

Medium

1. SqlWorldCup
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)
);
```

```
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,
```

Solution

Programming language used:	SQL (PostgreSQL)
Total time used:	30 minutes ❓
Effective time used:	30 minutes ❓
Notes:	*not defined yet*

Task timeline ❓

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

Code: 01:42:31 UTC, sql-postgres, final, score: **100**   show code in pop-up

```
1    -- Implement your solution here
2    ; WITH host_scores_CTE AS
3    (
4        SELECT host_team AS team_id,
5            CASE WHEN (host_goals > guest_goals) THEN 3
6                WHEN (host_goals = guest_goals) THEN 1
7                ELSE 0 END AS scores
8        FROM matches
9    )
10
11   , guest_scores_CTE AS
12   (
13       SELECT guest_team AS team_id,
14           CASE WHEN (host_goals < guest_goals) THEN 3
15               WHEN (host_goals = guest_goals) THEN 1
16               ELSE 0 END AS scores
17       FROM matches
18   )
19
20   , all_scores_CTE AS
21   (
22       SELECT * FROM host_scores_CTE
23       UNION ALL
24       SELECT * FROM guest_scores_CTE
25   )
26
27   SELECT t.team_id, t.team_name, COALESCE(SUM(scores)
28   FROM teams t
29       LEFT JOIN all_scores_CTE s ON s.team_id = t.tea
30   GROUP BY t.team_id, t.team_name
31   ORDER BY num_points DESC, team_id
```

Analysis summary

The solution obtained perfect score.

Analysis

expand all	Example tests	
▶ example	✓ OK	
example test		
expand all	Correctness tests	
▶ both_tables_empty	✓ OK	
no teams nor matches		
▶ no_matches	✓ OK	
some teams, but no matches		
▶ one_match	✓ OK	
many teams, only one match		
▶ simple_only_draws	✓ OK	
all teams drawed their matches		
▶ simple_no_draws	✓ OK	
no draw in any match		
▶ simple_distinct_scores	✓ OK	
all teams have different score at the		

```
30       | You       | 3
40       | Up        | 0
```

end, also no need to reorder anything

▶	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