Problem 3278: Find Candidates for Data Scientist Position II

Problem Information

Difficulty: Medium
Acceptance Rate: 42.45%
Paid Only: Yes
Tags: Database

Problem Description

Table: `Candidates`

+--------------+---------+ | Column Name | Type | +--------------+---------+ | candidate_id | int | | skill | varchar | | proficiency | int | +--------------+---------+ (candidate_id, skill) is the unique key for this table. Each row includes candidate_id, skill, and proficiency level (1-5).

Table: `Projects`

+--------------+---------+ | Column Name | Type | +--------------+---------+ | project_id | int | | skill | varchar | | importance | int | +--------------+---------+ (project_id, skill) is the primary key for this table. Each row includes project_id, required skill, and its importance (1-5) for the project.

Leetcode is staffing for multiple data science projects. Write a solution to find the **best candidate** for**each project** based on the following criteria:

1. Candidates must have **all** the skills required for a project. 2. Calculate a **score** for each candidate-project pair as follows: * **Start** with `100` points * **Add** `10` points for each skill where **proficiency > importance** * **Subtract** `5` points for each skill where **proficiency < importance** * If the candidate's skill proficiency **equal** to the project's skill importance, the score remains unchanged

Include only the top candidate (highest score) for each project. If there's a **tie** , choose the candidate with the **lower** `candidate_id`. If there is **no suitable candidate** for a project, **do not return** that project.

Return a result table ordered by `project_id` in ascending order.

The result format is in the following example.

Example:

Input:

`Candidates` table:

+--------------+-----------+-------------+ | candidate_id | skill | proficiency | +--------------+-----------+-------------+ | 101 | Python | 5 | | 101 | Tableau | 3 | | 101 | PostgreSQL| 4 | | 101 | TensorFlow| 2 | | 102 | Python | 4 | | 102 | Tableau | 5 | | 102 | PostgreSQL| 4 | | 102 | R | 4 | | 103 | Python | 3 | | 103 | Tableau | 5 | | 103 | PostgreSQL| 5 | | 103 | Spark | 4 | +--------------+-----------+-------------+

`Projects` table:

+-------------+-----------+------------+ | project_id | skill | importance | +-------------+-----------+------------+ | 501 | Python | 4 | | 501 | Tableau | 3 | | 501 | PostgreSQL| 5 | | 502 | Python | 3 | | 502 | Tableau | 4 | | 502 | R | 2 | +-------------+-----------+------------+

Output:

+-------------+--------------+-------+ | project_id | candidate_id | score | +-------------+--------------+-------+ | 501 | 101 | 105 | | 502 | 102 | 130 | +-------------+--------------+-------+

Explanation:

* For Project 501, Candidate 101 has the highest score of 105. All other candidates have the same score but Candidate 101 has the lowest candidate_id among them. * For Project 502, Candidate 102 has the highest score of 130.

The output table is ordered by project_id in ascending order.

Code Snippets

MySQL:

```
# Write your MySQL query statement below
```

MS SQL Server:

```
/* Write your T-SQL query statement below */
```

PostgreSQL:

```
-- Write your PostgreSQL query statement below
```