Problem 3057: Employees Project Allocation

Problem Information

Difficulty: Hard
Acceptance Rate: 0.00%
Paid Only: No

Problem Description

Table:

Project

+-------------+---------+ | Column Name | Type | +-------------+---------+ | project_id | int | | employee_id | int | | workload | int | +-------------+---------+ employee_id is the primary key (column with unique values) of this table. employee_id is a foreign key (reference column) to

Employee

table. Each row of this table indicates that the employee with employee_id is working on the project with project_id and the workload of the project.

Table:

Employees

+------------------+---------+ | Column Name | Type | +------------------+---------+ | employee_id | int | | name | varchar | | team | varchar | +------------------+---------+ employee_id is the primary key (column with unique values) of this table. Each row of this table contains information about one employee.

Write a solution to find the

employees

who are allocated to projects with a

workload that exceeds the average

workload of all employees for

their respective teams

Return t

he result table ordered by

employee_id

,

project_id

in

ascending

order.

The result format is in the following example.

Example 1:

Input:

Project table: +-------------+-------------+----------+ | project_id | employee_id | workload | +-------------+-------------+----------+ | 1 | 1 | 45 | | 1 | 2 | 90 | | 2 | 3 | 12 | | 2 | 4 | 68 | +-------------+-------------+----------+ Employees table: +-------------+--------+------+ | employee_id | name | team | +-------------+--------+------+ | 1 | Khaled | A | | 2 | Ali | B | | 3 | John | B | | 4 | Doe | A | +-------------+--------+------+

Output:

+-------------+------------+---------------+------------------+ | employee_id | project_id | employee_name | project_workload | +-------------+------------+---------------+------------------+ | 2 | 1 | Ali | 90 | | 4 | 2 | Doe | 68 | +-------------+------------+---------------+------------------+

Explanation:

- Employee with ID 1 has a project workload of 45 and belongs to Team A, where the average workload is 56.50. Since his project workload does not exceed the team's average workload, he will be excluded. - Employee with ID 2 has a project workload of 90 and belongs to Team B, where the average workload is 51.00. Since his project workload does exceed the team's average workload, he will be included. - Employee with ID 3 has a project workload of 12 and belongs to Team B, where the average workload is 51.00. Since his project workload does not exceed the team's average workload, he will be excluded. - Employee with ID 4 has a project workload of 68 and belongs to Team A, where the average workload is 56.50. Since his project workload does exceed the team's average workload, he will be included. Result table orderd by employee_id, 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
```

Oracle:

```
/* Write your PL/SQL query statement below */
```

Pandas:

```
import pandas as pd

def employees_with_above_avg_workload(project: pd.DataFrame, employees:
pd.DataFrame) -> pd.DataFrame:
```

Solutions

MySQL Solution:

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

MS SQL Server Solution:

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

PostgreSQL Solution:

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

Oracle Solution:

```
/* Write your PL/SQL query statement below */
```

Pandas Solution:

```
import pandas as pd

def employees_with_above_avg_workload(project: pd.DataFrame, employees:
pd.DataFrame) -> pd.DataFrame:
```