Problem 2010: The Number of Seniors and Juniors to Join the Company II

Problem Information

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

Problem Description

Table:

Candidates

+-------------+------+ | Column Name | Type | +-------------+------+ | employee_id | int | | experience | enum | | salary | int | +-------------+------+ employee_id is the column with unique values for this table. experience is an ENUM (category) of types ('Senior', 'Junior'). Each row of this table indicates the id of a candidate, their monthly salary, and their experience. The salary of each candidate is guaranteed to be

unique

.

A company wants to hire new employees. The budget of the company for the salaries is

$70000

. The company's criteria for hiring are:

Keep hiring the senior with the smallest salary until you cannot hire any more seniors.

Use the remaining budget to hire the junior with the smallest salary.

Keep hiring the junior with the smallest salary until you cannot hire any more juniors.

Write a solution to find the ids of seniors and juniors hired under the mentioned criteria.

Return the result table in

any order

.

The result format is in the following example.

Example 1:

Input:

Candidates table: +-------------+------------+--------+ | employee_id | experience | salary | +-------------+------------+--------+ | 1 | Junior | 10000 | | 9 | Junior | 15000 | | 2 | Senior | 20000 | | 11 | Senior | 16000 | | 13 | Senior | 50000 | | 4 | Junior | 40000 | +-------------+------------+--------+

Output:

+-------------+ | employee_id | +-------------+ | 11 | | 2 | | 1 | | 9 | +-------------+

Explanation:

We can hire 2 seniors with IDs (11, 2). Since the budget is $70000 and the sum of their salaries is $36000, we still have $34000 but they are not enough to hire the senior candidate with ID 13. We can hire 2 juniors with IDs (1, 9). Since the remaining budget is $34000 and the sum of their salaries is $25000, we still have $9000 but they are not enough to hire the junior candidate with ID 4.

Example 2:

Input:

Candidates table: +-------------+------------+--------+ | employee_id | experience | salary | +-------------+------------+--------+ | 1 | Junior | 25000 | | 9 | Junior | 10000 | | 2 | Senior | 85000 | | 11 | Senior | 80000 | | 13 | Senior | 90000 | | 4 | Junior | 30000 | +-------------+------------+--------+

Output:

+-------------+ | employee_id | +-------------+ | 9 | | 1 | | 4 | +-------------+

Explanation:

We cannot hire any seniors with the current budget as we need at least $80000 to hire one senior. We can hire all three juniors with the remaining budget.

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 number_of_joiners(candidates: 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 number_of_joiners(candidates: pd.DataFrame) -> pd.DataFrame:
```