

Documentation for the "Top Three Salaries" Solution

Problem Overview:

The problem requires us to find the employees with the top three unique salaries in each department of a company. The data is provided in two tables:

- Employee table, which contains details of each employee's ID, name, salary, and department affiliation.
- Department table, which contains the department ID and department name.

The objective is to:

1. Identify the top three unique salaries within each department.
2. Display the employees earning those top three salaries along with their salary and department information.

Input Structure:

1. Employee Table:

- ***id***: An integer representing the unique ID of each employee.
- ***name***: A string representing the name of the employee.
- ***salary***: An integer representing the salary of the employee.
- ***departmentId***: An integer that references the ID of the department to which the employee belongs.

2. Department Table:

- ***id***: An integer representing the unique ID of each department.
- ***name***: A string representing the name of the department.

Output Structure:

The solution should return a table with the following columns:

- **Department:** The name of the department.
- **Employee:** The name of the employee who earns one of the top three unique salaries in their department.
- **Salary:** The salary of the employee.

Key Points:

- The result should include all employees who have one of the top three unique salaries in their department.
- If a department has fewer than three distinct salaries, include all available salaries.
- Employees with the same salary should all be included in the output if their salary is in the top three.

Solution Steps:

1. Merge the Employee and Department Tables:

- To get the department name alongside employee details, merge the Employee and Department tables on the departmentId and id fields, respectively.
- This combined table will allow us to associate each employee with their corresponding department's name.

2. Sort the Merged Data:

- After merging the two tables, sort the data by department name and salary. Sorting the salary in descending order helps to easily identify the top earners in each department.

3. Rank the Salaries:

- Group the data by the department name and apply a ranking function to the salaries. The ranking will help to determine the top three unique salaries for each department.
- Use a dense ranking method, which ensures that the same salary is assigned the same rank. For example, if two employees earn the same salary, they will both get the same rank, and the next highest salary will receive the following rank.

4. Filter for Top Three Salaries:

- After ranking, filter out employees whose salary rank exceeds 3. This step ensures that only the top three unique salary earners are retained for each department.
- If a department has fewer than three unique salaries, return all the available ranks (1st, 2nd, etc.).

5. Prepare the Final Output:

- After filtering, select only the relevant columns (Department name, Employee name, and Salary) from the merged table.
- Rename the columns to match the required output format: "Department", "Employee", and "Salary".
- The final result should contain the top three unique salary earners for each department, listed in any order.

Example:

Input:

Employee Table:

id	name	salary	departmentId
1	Joe	85000	1
2	Henry	80000	2
3	Sam	60000	2
4	Max	90000	1
5	Janet	69000	1
6	Randy	85000	1
7	Will	70000	1

Department Table:

id	name
1	IT
2	Sales

Output:

Department	Employee		Salary
IT	Max		90000
IT	Joe		85000
IT	Randy		85000
IT	Will		70000
Sales	Henry		80000
Sales	Sam		60000

Explanation:

In the IT department, the top three unique salaries are:

1. Max earns 90000.
2. Joe and Randy both earn 85000.
3. Will earns 70000.

In the Sales department, the top two salaries are:

1. Henry earns 80000.
2. Sam earns 60000.

There is no third-highest salary in this department because there are only two employees.

Edge Cases:

1. **Fewer than 3 Employees in a Department:** If a department has fewer than three employees, simply return all employees' salaries.

2. **Duplicate Salaries:** Employees with the same salary are considered for the same rank. For example, if two employees earn the second-highest salary, both should appear in the results.
3. **Departments with Only 1 Unique Salary:** In this case, only employees with that salary are returned.

Efficiency Considerations:

- The solution involves sorting the data and ranking salaries within each department, which are standard operations in pandas and can be efficiently handled even for large datasets.
- By using dense ranking, we ensure that the logic remains simple and effective in dealing with duplicate salaries.

This approach ensures that the top three unique salary earners are correctly identified for each department, meeting the problem's requirements efficiently.