Analytical Functions in Oracle

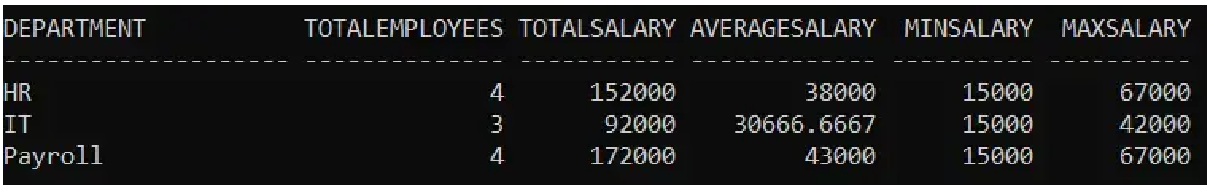
Analytical Functions in Oracle

Oracle supports the following three types of analytical functions,

1. **Row_Number()**
2. **Rank()**
3. **Dense_Rank()**

These analytical functions are automatically generated ranking numbers for each row-wise (or) group of rows-wise. These are also called “Ranking Functions” in Oracle.

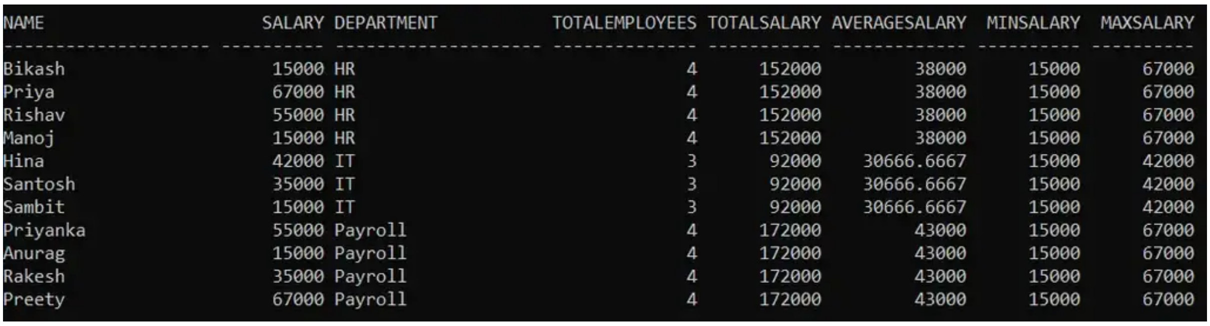
Our business requirement is to generate a report that should display the total number of employees in each department. Along with the total number of employees in each department, we also need to display the Total Salary, Average Salary, Minimum Salary, and Maximum Salary of each department.



```
DEPARTMENT           TOTALEMPLOYEES TOTALSALARY AVERAGESALARY  MINSALARY  MAXSALARY
-------------------- -------------- ----------- ------------- ---------- ----------
HR                                4      152000         38000      15000      67000
IT                                3       92000    30666.6667      15000      42000
Payroll                           4      172000         43000      15000      67000
```

```
SELECT  Department,
 COUNT(*) AS TotalEmployees,
 SUM(Salary) AS TotalSalary,
 AVG(Salary) AS AverageSalary,
 MIN(Salary) AS MinSalary,
 MAX(Salary) AS MaxSalary
FROM Employees
GROUP BY Department;
```

Now our business requirement changes. Now we also want to show the non-aggregated values (Name and Salary) in the report along with the aggregated values as shown in the below image.



```
NAME                     SALARY DEPARTMENT           TOTALEMPLOYEES TOTALSALARY AVERAGESALARY  MINSALARY  MAXSALARY
-------------------- ---------- -------------------- -------------- ----------- ------------- ---------- ----------
Bikash                    15000 HR                                4      152000         38000      15000      67000
Priya                     67000 HR                                4      152000         38000      15000      67000
Rishav                    55000 HR                                4      152000         38000      15000      67000
Manoj                     15000 HR                                4      152000         38000      15000      67000
Hina                      42000 IT                                3       92000    30666.6667      15000      42000
Santosh                   35000 IT                                3       92000    30666.6667      15000      42000
Sambit                    15000 IT                                3       92000    30666.6667      15000      42000
Priyanka                  55000 Payroll                           4      172000         43000      15000      67000
Anurag                    15000 Payroll                           4      172000         43000      15000      67000
Rakesh                    35000 Payroll                           4      172000         43000      15000      67000
Preety                    67000 Payroll                           4      172000         43000      15000      67000
```

```
SELECT Name, Salary, Department,
 COUNT(*) AS TotalEmployees,
 SUM(Salary) AS TotalSalary,
 AVG(Salary) AS AverageSalary,
 MIN(Salary) AS MinSalary,
 MAX(Salary) AS MaxSalary
FROM Employees
GROUP BY Department;
```

```
SELECT Name, Salary, Department,
 COUNT(Department) OVER(PARTITION BY Department) AS TotalEmployees,
 SUM(Salary) OVER(PARTITION BY Department) AS TotalSalary,
 AVG(Salary) OVER(PARTITION BY Department) AS AverageSalary,
 MIN(Salary) OVER(PARTITION BY Department) AS MinSalary,
 MAX(Salary) OVER(PARTITION BY Department) AS MaxSalary
FROM Employees;
```

Analytical Functions Without Partition by Clause in Oracle:

```
SELECT Name, Salary,
ROW_NUMBER() OVER(ORDER BY Salary DESC) ROW_NUMBER,
RANK() OVER(ORDER BY Salary DESC) RANK,
DENSE_RANK() OVER(ORDER BY Salary DESC) DENSE_RANK
FROM Employees;
```

Analytical Functions With Partition by Clause in Oracle:

```
SELECT NAME, SALARY,
ROW_NUMBER() OVER(PARTITION BY DEPARTMENT ORDER BY SALARY DESC) ROW_NUMBER,
RANK() OVER(PARTITION BY DEPARTMENT  ORDER BY SALARY DESC) RANK,
DENSE_RANK() OVER(PARTITION BY DEPARTMENT ORDER BY SALARY DESC) DENSE_RANK
FROM EMPLOYEES;
```

WAQ to display 3rd highest salary employee details from Employees table in each department wise by using DENSE_RANK() function along with inline view?

Solution:

```
SELECT * FROM (SELECT NAME, SALARY, DEPARTMENT,DENSE_RANK() OVER (PARTITION BY DEPARTMENT ORDER BY SALARY DESC) R FROM EMPLOYEES) WHERE R=3;
```

ROW_NUMBER Function in Oracle

RANK Function in Oracle

DENSE_RANK Function in Oracle