



Objective: Create aggregate queries using GROUP BY and using analytic functions

Problem 1:

Create a query displaying the department_name, minimum salary, maximum salary, average salary for employees by department_name. Round the average salary to two decimal places. Order the query by department name. Label the columns Min Salary, Max Salary, Average Salary.

Problem 2:

Write a query to show job title, average salary, difference between maximum and minimum of salary of employees for sales people only (those having a commission or job_id starts with SA). Order the query by job title. Use the column names "Average Salary" and "Diff Max – Min".

Problem 3:

Create a query showing all employees (employee_id, Last_name, Salary) that make more money than the maximum salary of all sales people. (those having a commission or job_id starts with SA). Sort it by last_name.

Problem 4:

Create a query counting the number of locations by region name and country name. Alias the count column with # Locs. Implement the following custom sort order for region name: Europe, Americas, Asia. Within each region name, sort it by country name.

Problem 5:

Create a query counting the number of employees by administration, sales, or other. Any job id starting with IT, AD, AC, or PU is considered administration, any job id starting with ST, SA, SH is considered sales. For any other job id use other. Use the column headings "Job Category" and "Count of Emp". Sort it by job category expression.

Problem 6:

Show the count of employees by department_id for employees that are not sales people. Display only those records where the count is greater than 5. Order it by department_id.

Problem 7:

Display the hire year and the count of employees hired in each year. Additionally, show the total number of employees (use analytic function sum) and alias it with sum_total. Extend this query to show the percentage of employees hired in each year (use the count, multiply it by 100, then divide by sum_total expression). Round this value to two decimal places, format the number so that always two decimal digits are displayed, add the percent sign at the end (2.8 → 2.80%) and finally, alias it with sum_percent. Sort the output by hire year.