

# 

**LAB REPORT 13**

# Chapter 5 exercise

Submitted to Ms.Humaira Batool

Submitted By Hamza Mehmood

Roll No SP-21-110

Subject DBMS-Lab

Department SE

Class BSSE-EVE

1. Group functions work across many rows to produce one result.

**True**

1. Group functions include nulls in calculations.

**False. Group functions ignore null values. If you want to include null values, use the NVL function**

1. The WHERE clause restricts rows prior to inclusion in a group calculation.

**True**

1. Display the highest, lowest, sum, and average salary of all employees. Label the columns Maximum, Minimum, Sum, and Average, respectively. Round your results to the nearest whole number. Place your SQL statement in a text file named lab5\_6.sql.

**SELECT ROUND(MAX(salary),0) "Maximum", ROUND(MIN(salary),0) "Minimum", ROUND(SUM(salary),0) "Sum", ROUND(AVG(salary),0) "Average" FROM employees;**

1. Modify the query in lab5\_4.sql to display the minimum, maximum, sum, and average salary for each job type. Resave lab5\_4.sql to lab5\_5.sql. Run the statement in lab5\_5.sql.

**SELECT job\_id, ROUND(MAX(salary),0) "Maximum", ROUND(MIN(salary),0) "Minimum", ROUND(SUM(salary),0) "Sum", ROUND(AVG(salary),0) "Average" FROM employees GROUP BY job\_id;**

1. Write a query to display the number of people with the same job.

**SELECT job\_id, COUNT(\*) FROM employees GROUP BY job\_id;**

1. Determine the number of managers without listing them. Label the column Number of Managers. Hint: Use the MANAGER\_ID column to determine the number of managers.

**SELECT COUNT(DISTINCT manager\_id) "Number of Managers" FROM employees;**

1. Write a query that displays the difference between the highest and lowest salaries. Label the column DIFFERENCE.

**SELECT MAX(salary) - MIN(salary) DIFFERENCE FROM employees;**

1. Display the manager number and the salary of the lowest paid employee for that manager. Exclude anyone whose manager is not known. Exclude any groups where the minimum salary is less than $6,000. Sort the output in descending order of salary.

**SELECT manager\_id, MIN(salary) FROM employees WHERE manager\_id IS NOT NULL GROUP BY manager\_id HAVING MIN(salary) > 6000 ORDER BY MIN(salary) DESC;**

1. Write a query to display each department’s name, location, number of employees, and the average salary for all employees in that department. Label the columns Name, Location, Number of People, and Salary, respectively. Round the average salary to two decimal places.

**SELECT d.department\_name "Name", d.location\_id "Location ", COUNT(\*) "Number of People", ROUND(AVG(salary),2) "Salary" FROM employees e, departments d WHERE e.department\_id = d.department\_id GROUP BY d.department\_name, d.location\_id;**

1. Create a query that will display the total number of employees and, of that total, the number of employees hired in 1995, 1996, 1997, and 1998. Create appropriate column headings.

**SELECT COUNT(\*) total, SUM(DECODE(TO\_CHAR(hire\_date, 'YYYY'),1995,1,0))"1 995", SUM(DECODE(TO\_CHAR(hire\_date, 'YYYY'),1996,1,0))"1 996", SUM(DECODE(TO\_CHAR(hire\_date, 'YYYY'),1997,1,0))"1 997", SUM(DECODE(TO\_CHAR(hire\_date, 'YYYY'),1998,1,0))"1 998" FROM employees;**

1. Create a matrix query to display the job, the salary for that job based on department number, and the total salary for that job, for departments 20, 50, 80, and 90, giving each column an appropriate heading.

**SELECT job\_id "Job", SUM(DECODE(department\_id , 20, salary)) "Dept 20" , SUM(DECODE(department\_id , 50, salary)) "Dept 50" , SUM(DECODE(department\_id , 80, salary)) "Dept 80" , SUM(DECODE(department\_id , 90, salary)) "Dept 90" , SUM(salary) "Total" FROM employees GROUP BY job\_id;**