Department of Computer Technology and Information Systems
2025-2026 Fall Semester

**CTIS259 Database Management Systems and Applications**

# Lab Guide 11

| | | | |
|---|---|---|---|
| **Instructor** | : Nimet Ceren SERİM | **Week:** | 8 |
| **Assistant** | : Engin Zafer KIRAÇBEDEL, Hatice Zehra YILMAZ | **Date:** | 06-07.11.2025 |

**Aim of this lab session:** **1.** Reporting Aggregated Data Using the Group Functions
**2.** Practice 7-1: Retrieving Data by Using Subqueries

**ORACLE Server Configurations:**
**IP Address: 139.179.33.231**
**Port number: 1522**
**SID: orclctis**

**Please USE oraxx accounts!**

## Practices for Lesson 6

### Lesson Overview

At the end of this practice, you should be familiar with using group functions and selecting groups of data.

## Practice 6-1: Reporting Aggregated Data Using the Group Functions

Determine the validity of the following three statements. Circle either True or False.

1. Group functions work across many rows to produce one result per group.
   True/False

2. Group functions include nulls in calculations.
   True/False

3. The `WHERE` clause restricts rows before inclusion in a group calculation.
   True/False

The HR department needs the following reports:

4. Find 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. Save your SQL statement as `lab_06_04.sql`. Run the query.

| | Maximum | Minimum | Sum | Average |
|---|---|---|---|---|
| 1 | 24000 | 2500 | 175500 | 8775 |

5. Modify the query in `lab_06_04.sql` to display the minimum, maximum, sum, and average salary for each job type. Save `lab_06_04.sql` as `lab_06_05.sql` again. Run the statement in `lab_06_05.sql`.

| | JOB_ID | Maximum | Minimum | Sum | Average |
|---|---|---|---|---|---|
| 1 | AC_MGR | 12000 | 12000 | 12000 | 12000 |
| 2 | AC_ACCOUNT | 8300 | 8300 | 8300 | 8300 |
| 3 | IT_PROG | 9000 | 4200 | 19200 | 6400 |
| 4 | ST_MAN | 5800 | 5800 | 5800 | 5800 |
| 5 | AD_ASST | 4400 | 4400 | 4400 | 4400 |
| 6 | AD_VP | 17000 | 17000 | 34000 | 17000 |
| 7 | MK_MAN | 13000 | 13000 | 13000 | 13000 |
| 8 | SA_MAN | 10500 | 10500 | 10500 | 10500 |
| 9 | MK_REP | 6000 | 6000 | 6000 | 6000 |
| 10 | AD_PRES | 24000 | 24000 | 24000 | 24000 |
| 11 | SA_REP | 11000 | 7000 | 26600 | 8867 |
| 12 | ST_CLERK | 3500 | 2500 | 11700 | 2925 |

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

| | JOB_ID | COUNT(*) |
|---|---|---|
| 1 | AC_ACCOUNT | 1 |
| 2 | AC_MGR | 1 |
| 3 | AD_ASST | 1 |
| 4 | AD_PRES | 1 |
| 5 | AD_VP | 2 |
| 6 | IT_PROG | 3 |
| 7 | MK_MAN | 1 |
| 8 | MK_REP | 1 |
| 9 | SA_MAN | 1 |
| 10 | SA_REP | 3 |
| 11 | ST_CLERK | 4 |
| 12 | ST_MAN | 1 |

Generalize the query so that the user in the HR department is prompted for a job title. Save the script to a file named `lab_06_06.sql`. Run the query. Enter `IT_PROG` when prompted.

| | JOB_ID | COUNT(*) |
|---|---|---|
| 1 | IT_PROG | 3 |

7. 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.

| | Number of Managers |
|---|---|
| 1 | 8 |

8. Find the difference between the highest and lowest salaries. Label the column `DIFFERENCE`.

| | DIFFERENCE |
|---|---|
| 1 | 21500 |

9. Create a report to 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 $6,000 or less. Sort the output in descending order of salary.

| | MANAGER_ID | MIN(SALARY) |
|---|---|---|
| 1 | 102 | 9000 |
| 2 | 205 | 8300 |
| 3 | 149 | 7000 |

10. Create a query to 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.

| | TOTAL | 1995 | 1996 | 1997 | 1998 |
|---|---|---|---|---|---|
| 1 | 20 | 1 | 2 | 2 | 3 |

11. 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.

| | Job | Dept 20 | Dept 50 | Dept 80 | Dept 90 | Total |
|---|---|---|---|---|---|---|
| 1 | AC_MGR | (null) | (null) | (null) | (null) | 12000 |
| 2 | AC_ACCOUNT | (null) | (null) | (null) | (null) | 8300 |
| 3 | IT_PROG | (null) | (null) | (null) | (null) | 19200 |
| 4 | ST_MAN | (null) | 5800 | (null) | (null) | 5800 |
| 5 | AD_ASST | (null) | (null) | (null) | (null) | 4400 |
| 6 | AD_VP | (null) | (null) | (null) | 34000 | 34000 |
| 7 | MK_MAN | 13000 | (null) | (null) | (null) | 13000 |
| 8 | SA_MAN | (null) | (null) | 10500 | (null) | 10500 |
| 9 | MK_REP | 6000 | (null) | (null) | (null) | 6000 |
| 10 | AD_PRES | (null) | (null) | (null) | 24000 | 24000 |
| 11 | SA_REP | (null) | (null) | 19600 | (null) | 26600 |
| 12 | ST_CLERK | (null) | 11700 | (null) | (null) | 11700 |

# Practice 7-1: Retrieving Data by Using Subqueries

## Overview

In this practice, you write multiple-column subqueries, and correlated and scalar subqueries.

## Task

**1.** Write a query to display the last name, department number, and salary of any employee whose department number and salary both match the department number and salary of any employee who earns a commission.

| | LAST_NAME | DEPARTMENT_ID | SALARY |
|---|---|---|---|
| 1 | Taylor | 80 | 8600 |
| 2 | Zlotkey | 80 | 10500 |
| 3 | Abel | 80 | 11000 |

**2.** Display the last name, department name, and salary of any employee whose salary and commission match the salary and commission of any employee located in location ID 1700.

| | LAST_NAME | DEPARTMENT_NAME | SALARY |
|---|---|---|---|
| 1 | King | Executive | 24000 |
| 2 | Kochhar | Executive | 17000 |
| 3 | De Haan | Executive | 17000 |
| 4 | Whalen | Administration | 4400 |
| 5 | Higgins | Accounting | 12000 |
| 6 | Gietz | Accounting | 8300 |

**3.** Create a query to display the last name, hire date, and salary for all employees who have the same salary and commission as Kochhar.

**Note:** Do not display Kochhar in the result set.

| | LAST_NAME | HIRE_DATE | SALARY |
|---|---|---|---|
| 1 | De Haan | 13-JAN-1993 | 17000 |

**4.** Create a query to display the employees who earn a salary that is higher than the salary of all the sales managers (`JOB_ID = 'SA_MAN'`). Sort the results from the highest to the lowest.

| | LAST_NAME | JOB_ID | SALARY |
|---|---|---|---|
| 1 | King | AD_PRES | 24000 |
| 2 | De Haan | AD_VP | 17000 |
| 3 | Kochhar | AD_VP | 17000 |
| 4 | Hartstein | MK_MAN | 13000 |
| 5 | Higgins | AC_MGR | 12000 |
| 6 | Abel | SA_REP | 11000 |

**5.** Display details such as the employee ID, last name, and department ID of those employees who live in cities the names of which begin with *T*.

| | EMPLOYEE_ID | LAST_NAME | DEPARTMENT_ID |
|---|---|---|---|
| 1 | 202 | Fay | 20 |
| 2 | 201 | Hartstein | 20 |

**6.**

Write a query to find all employees who earn more than the average salary in their departments. Display last name, salary, department ID, and the average salary for the department. Sort by average salary and round to two decimals. Use aliases for the columns retrieved by the query as shown in the sample output.

| | ENAME | SALARY | DEPTNO | DEPT_AVG |
|---|---|---|---|---|
| 1 | Mourgos | 5800 | 50 | 3500 |
| 2 | Hunold | 9000 | 60 | 6400 |
| 3 | Hartstein | 13000 | 20 | 9500 |
| 4 | Zlotkey | 10500 | 80 | 10033.333333333333333333333333333333333 |
| 5 | Abel | 11000 | 80 | 10033.333333333333333333333333333333333 |
| 6 | Higgins | 12000 | 110 | 10150 |
| 7 | King | 24000 | 90 | 19333.333333333333333333333333333333333 |

**7.** Find all employees who are not supervisors.
**a.** First, do this using the NOT EXISTS operator.

| | LAST_NAME |
|---|---|
| 1 | Abel |
| 2 | Davies |
| 3 | Ernst |
| 4 | Fay |
| 5 | Gietz |
| 6 | Grant |
| 7 | Lorentz |
| 8 | Matos |
| 9 | Rajs |
| 10 | Taylor |
| 11 | Vargas |
| 12 | Whalen |

**b.** Can this be done by using the `NOT IN` operator? How, or why not?

**8.**
Write a query to display the last names of the employees who earn less than the average salary in their departments.

| | LAST_NAME |
|---|---|
| 1 | Kochhar |
| 2 | De Haan |
| 3 | Fay |
| 4 | Gietz |
| 5 | Davies |
| 6 | Matos |
| 7 | Vargas |
| 8 | Taylor |
| 9 | Ernst |
| 10 | Lorentz |

**9.** Write a query to display the last names of the employees who have one or more coworkers in their departments with later hire dates but higher salaries.

| | LAST_NAME |
|---|---|
| 1 | Vargas |
| 2 | Matos |
| 3 | Davies |
| 4 | Rajs |
| 5 | Taylor |

**10.** Write a query to display the employee ID, last names, and department names of all the employees.
**Note:** Use a scalar subquery to retrieve the department name in the `SELECT` statement.

| | EMPLOYEE_ID | LAST_NAME | DEPARTMENT |
|---|---|---|---|
| 1 | 205 | Higgins | Accounting |
| 2 | 206 | Gietz | Accounting |
| 3 | 200 | Whalen | Administration |
| 4 | 102 | De Haan | Executive |
| 5 | 100 | King | Executive |
| 6 | 101 | Kochhar | Executive |
| 7 | 103 | Hunold | IT |
| 8 | 104 | Ernst | IT |
| 9 | 107 | Lorentz | IT |
| 10 | 202 | Fay | Marketing |
| 11 | 201 | Hartstein | Marketing |
| 12 | 149 | Zlotkey | Sales |
| 13 | 174 | Abel | Sales |
| 14 | 176 | Taylor | Sales |
| 15 | 143 | Matos | Shipping |
| 16 | 142 | Davies | Shipping |
| 17 | 141 | Rajs | Shipping |
| 18 | 124 | Mourgos | Shipping |
| 19 | 144 | Vargas | Shipping |
| 20 | 178 | Grant | (null) |