# Practices for Lesson 16: Retrieving Data by Using Subqueries

Practices for Lesson 16: Overview

Practice Overview

This practice covers the following topics:

Creating multiple-column subqueries

Writing correlated subqueries

Using the EXISTS operator

Using scalar subqueries

Using the WITH clause

Practice 16-1: Retrieving Data by Using Subqueries

Overview

In this practice, you write multiple-column, correlated, and scalar subqueries. You also solve problems by writing the WITH clause.

Tasks

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.

**…**

Display the last name, department name, and salary of any employee whose salary and

job\_ID match the salary and job\_ID of any employee located in location ID 1700.

**…**

Create a query to display the last name, hire date, and salary for all employees who have the same salary and manager\_ID as Kochhar.

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

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 on salary from the highest to the lowest.

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*.

Write a query to find all employees who earn more than the average salary in their departments. Display the 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.

**…**

Find all employees who are not supervisors.

First, do this by using the NOT EXISTS operator.

**…**

Can this be done by using the NOT IN operator? How, or why not? If not, try out using another solution.

**…**

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

**…**

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.

**…**

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.

**…**

Write a query to display the department names of those departments whose total salary cost is above one-eighth (1/8) of the total salary cost of the whole company. Use the WITH clause to write this query. Name the query SUMMARY.

Solution 16-1: Retrieving Data by Using Subqueries

Solution

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

Display the last name, department name, and salary of any employee whose salary and

job\_ID match the salary and job\_ID of any employee located in location ID 1700.

Create a query to display the last name, hire date, and salary for all employees who have the same salary and manager\_ID as Kochhar.

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

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 on salary from the highest to the lowest.

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*.

Write a query to find all employees who earn more than the average salary in their departments. Display the 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.

Find all employees who are not supervisors.

First, do this by using the NOT EXISTS operator.

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

This alternative solution is not a good one. The subquery picks up a NULL value, so the entire query returns no rows. The reason is that all conditions that compare a NULL value result in NULL. Whenever NULL values are likely to be part of the value set, *do not* use NOT IN as a substitute for NOT EXISTS. A much better solution would be a subquery like the following:

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

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

Write a query to display the employee ID, last names, and department names of all employees.

**Note:** Use a scalar subquery to retrieve the department name in the SELECT statement.

Write a query to display the department names of those departments whose total salary cost is above one-eighth (1/8) of the total salary cost of the whole company. Use the WITH clause to write this query. Name the query SUMMARY.