Database Programming with SQL
10-4: Correlated Subqueries
Practice Activities

Objectives

- Identify when correlated subqueries are needed
- Construct correlated subqueries
- Construct named subqueries using the WITH clause

Try It / Solve It

1. Explain the main difference between correlated and non-correlated subqueries?

Correlated subqueries executes completely differently to non-correlated subqueries, in as much as they are driven by the outer query. So the outer query is executed, the first row returned and for THAT row the inner query is executed.

2. Write a query that lists the highest earners for each department. Include the last_name, department_id, and the salary for each employee.

```
SELECT oe.last_name, oe.department_id, oe.salary
FROM employees oe
WHERE oe.salary = (SELECT MAX(ie.salary) FROM employees ie
WHERE NVL(ie.department_id,-1) = NVL(oe.department_id,-1));
```

3. Examine the following select statement and finish it so that it will return the last_name, department_id, and salary of employees who have at least one person reporting to them. So we are effectively looking for managers only. In the partially written SELECT statement, the WHERE clause will work as it is. It is simply testing for the existence of a row in the subquery.

```
SELECT (enter columns here)
FROM   (enter table name here) outer
WHERE  'x' IN (SELECT 'x'
               FROM   (enter table name here) inner
               WHERE  inner(enter column name here)  = inner(enter column name here)
```

```
SELECT outer.last_name, outer.department_id, outer.salary
FROM employees outer
WHERE outer.employee_id IN (SELECT DISTINCT inner.manager_id
FROM employees  inner
WHERE inner.manager_id IS NOT NULL)
ORDER BY outer.department_id;
```

Finish off the statement by sorting the rows on the department_id column.

4. Using a WITH clause, write a SELECT statement to list the job_title of those jobs whose maximum salary is more than half the maximum salary of the entire company. Name your subquery MAX_CALC_SAL. Name the columns in the result JOB_TITLE and JOB_TOTAL, and sort the result on JOB_TOTAL in descending order.

Hint: Examine the jobs table. You will need to join JOBS and EMPLOYEES to display the job_title.

```
WITH max_calc_sal as (SELECT MAX(max_salary)/2 FROM jobs)
SELECT job_title
FROM jobs
WHERE jobs.max_salary > (SELECT * FROM max_calc_sal );
```