****

**UPA**

**Universidad Politécnica de Aguascalientes.**

**ISC06B**

**DATA BASE ADMINISTRATION**

# DP 10-4 PRACTICE

**TEACHER: Juan Carlos Herrera Hernández**

**STUDENTS:**

**Uriel Isaac Vazquez Martínez UP210934**

**Juan Pablo López González UP200053**

**José Miguel Escalera Rubalcava UP200667**

**Derek Gilberto Ramírez López UP200424**

**Aguascalientes, Ags, October 23th, 2022**

**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?**

In a correlated subquery, the inner query is evaluated once for each row processed by the outer query, while, in a non-correlated subquery, the inner query is evaluated first and only once, then the result is used by the outer query.

**2. Write a query that lists the highest earners for each department. Include the last\_name, department\_id, and the salary for each employee.**

|  |  |
| --- | --- |
| Problem No: 2 | No. Rows in Result: |
| **2. Write a query that lists the highest earners for each department. Include the last\_name, department\_id, and the salary for each employee.** | 7 |
| Text Code (No image) : | |
| --DP10-4  --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 last\_name, department\_id, salary  from employees e  where e.salary = (select max(salary)  from employees m  where e.department\_id = m.department\_id)  order by department\_id; | |
| Image Result: | |
|  | |
|  | |

**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)**

**Finish off the statement by sorting the rows on the department\_id column.**

|  |  |
| --- | --- |
| Problem No: 3 | No. Rows in Result: |
| **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.** | 8 |
| Text Code (No image) : | |
| --DP10-4  --3. return the last\_name, department\_id, and salary of employees who have at least one person reporting to them  SELECT outer.employee\_id, 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 = outer.employee\_id )  order by department\_id; | |
| Image Result: | |
|  | |
|  | |

**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.**

|  |  |
| --- | --- |
| Problem No: 4 | No. Rows in Result: |
| **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** | 5 |
| Text Code (No image) : | |
| --DP10-4  --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  with MAX\_CALC\_SAL as (select max(salary) as compar  from employees)  select j.job\_title as "JOB\_TITLE", j.max\_salary as "JOB\_TOTAL"  from jobs j, MAX\_CALC\_SAL m  where j.max\_salary > (m.compar/2)  order by JOB\_TOTAL desc; | |
| Image Result: | |
|  | |
|  | |