****

**UPA**

**Universidad Politécnica de Aguascalientes.**

**ISC06B**

**DATA BASE ADMINISTRATION**

# DP 10-3 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 7th, 2022**

**Database Programming with SQL**

**10-3: Multiple-Row Subqueries**

**Practice Activities**

**Objectives**

• Correctly use the comparison operators IN, ANY, and ALL in multiple-row subqueries

• Describe what happens if a multiple-row subquery returns a null value

• Construct and execute a multiple-row subquery in the WHERE clause or HAVING clause

• Understand when multiple-row subqueries should be used, and when it is safe to use a single-row subquery

• Distinguish between pair-wise and non-pair-wise subqueries

• Create a query using the EXISTS and NOT EXISTS operators to test for returned rows from the subquery

**Try It / Solve It**

1. **What will be returned by a query if it has a subquery that returns a null?**

No results

1. **Write a query that returns jazz and pop songs. Write a multi-row subquery and use the d\_songs and d\_types tables. Include the id, title, duration, and the artist name.**

|  |  |
| --- | --- |
| Problem No: 2 | No. Rows in Result: |
| 2. Write a query that returns jazz and pop songs. Write a multi-row subquery and use the d\_songs and d\_types tables. Include the id, title, duration, and the artist name. | 3 |
| Text Code (No image) : | |
| --DP 10-3  --2. Write a query that returns jazz and pop songs.  --Write a multi-row subquery and use the d\_songs and d\_types tables. Include the id, title, duration, and the artist name.  select id, title, duration, artist  from d\_songs  where type\_code in (select code  from d\_types  where description like 'Jazz'  or description like 'Pop'  ); | |
| Image Result: | |
|  | |
|  | |

1. **Find the last names of all employees whose salaries are the same as the minimum salary for any department.**

|  |  |
| --- | --- |
| Problem No: 3 | No. Rows in Result: |
| --3. 3. Find the last names of all employees whose salaries are the same as the minimum salary for any department.  select last\_name,department\_id, salary | 8 |
| Text Code (No image) : | |
| --DP10-3  --3. 3. Find the last names of all employees whose salaries are the same as the minimum salary for any department.  select last\_name,department\_id, salary  from employees  where (department\_id, salary) in (select department\_id,min(salary)  from employees  group by department\_id)  order by department\_id; | |
| Image Result: | |
|  | |
|  | |

1. **Which Global Fast Foods employee earns the lowest salary? Hint: You can use either a single-row or a multiple-row subquery.**

|  |  |
| --- | --- |
| Problem No: 4 | No. Rows in Result: |
| 4 Which Global Fast Foods employee earns the lowest salary? Hint: You can use either a single-row or a multiple-row subquery. | 1 |
| Text Code (No image) : | |
| --DP10-3  --4.Which Global Fast Foods employee earns the lowest salary?  --Hint: You can use either a single-row or a multiple-row subquery.  SELECT first\_name, salary  from f\_staffs  where salary = (select min(salary)  from f\_staffs); | |
| Image Result: | |
|  | |
|  | |

**5. Place the correct multiple-row comparison operators in the outer query WHERE clause of each of the following:**

**a. Which CDs in our d\_cds collection were produced before “Carpe Diem” was produced?**

**WHERE year \_\_\_\_<\_\_\_\_\_ (SELECT year ...**

**b. Which employees have salaries lower than any one of the programmers in the IT department?**

**WHERE salary \_\_\_\_<ANY\_\_\_\_\_\_(SELECT salary ...**

**c. What CD titles were produced in the same year as “Party Music for All Occasions” or “Carpe Diem”?**

**WHERE year \_\_\_\_IN\_\_\_\_\_(SELECT year ...**

**d. What song title has a duration longer than every type code 77 title?**

**WHERE duration \_\_\_\_>ALL\_\_\_\_(SELECT duration ...**

**6. If each WHERE clause is from the outer query, which of the following are true?**

**\_\_TRUE\_\_a. WHERE size > ANY -- If the inner query returns sizes ranging from 8 to 12, the value 9 could be returned in the outer query.**

**\_\_FALSE\_\_b. WHERE book\_number IN -- If the inner query returns books numbered 102, 105, 437, and 225 then 325 could be returned in the outer query.**

**\_FALSE\_\_\_c. WHERE score <= ALL -- If the inner query returns the scores 89, 98, 65, and 72, then 82 could be returned in the outer query.**

**\_\_TRUE\_\_d. WHERE color NOT IN -- If the inner query returns red, green, blue, black, and then the outer query could return white.**

**\_\_FALSE\_\_e. WHERE game\_date = ANY -- If the inner query returns 05-Jun-1997, 10-Dec-2002, and 2-Jan-2004, then the outer query could return 10-Sep-2002.**

**7. The goal of the following query is to display the minimum salary for each department whose minimum salary is less than the lowest salary of the employees in department 50. However, the subquery does not execute because it has five errors. Find them, correct them, and run the query.**

**SELECT department\_id**

**FROM employees**

**WHERE MIN(salary)**

**HAVING MIN(salary) >**

**GROUP BY department\_id**

**SELECT MIN(salary)**

**WHERE department\_id < 50;**

|  |  |
| --- | --- |
| Problem No: 7 | No. Rows in Result: |
| 7. **minimum salary for each department whose minimum salary is less than the lowest salary of the employees in department 50** | 0 |
| Text Code (No image) : | |
| --DP10-3  --7. minimum salary for each department whose minimum salary is less than the lowest salary of the employees in department 50  --DP10-3  SELECT department\_id, MIN(salary)  FROM employees  GROUP BY department\_id  HAVING MIN(salary) <(  SELECT MIN(salary)  from employees  WHERE department\_id = 50); | |
| Image Result: | |
|  | |
|  | |

**8. Which statements are true about the subquery below?**

**SELECT employee\_id, last\_name**

**FROM employees**

**WHERE salary =**

**(SELECT MIN(salary)**

**FROM employees**

**GROUP BY department\_id);**

**\_FALSE\_\_\_\_\_ a. The inner query could be eliminated simply by changing the WHERE clause to WHERE MIN(salary).**

**\_\_TRUE\_\_\_\_ b. The query wants the names of employees who make the same salary as the smallest salary in any department.**

**\_\_\_FALSE\_\_\_ c. The query first selects the employee ID and last name, and then compares that to the salaries in every department.**

**\_\_TRUE\_\_\_\_ d. This query will not execute.**

**9. Write a pair-wise subquery listing the last\_name, first\_name, department\_id, and manager\_id for all employees that have the same department\_ id and manager\_id as employee 141. Exclude employee 141 from the result set.**

|  |  |
| --- | --- |
| Problem No: 9 | No. Rows in Result: |
| **9. Write a pair-wise subquery listing the last\_name, first\_name, department\_id, and manager\_id for all employees that have the same department\_ id and manager\_id as employee 141. Exclude employee 141 from the result set.** | 3 |
| Text Code (No image) : | |
| --DP10-3  --9. Write a pair-wise subquery listing the last\_name, first\_name, department\_id, and manager\_id for all employees  --that have the same department\_ id and manager\_id as employee 141. Exclude employee 141 from the result set.  SELECT last\_name, first\_name, department\_id, manager\_id  FROM employees  Where (department\_id,manager\_id)in (select department\_id, manager\_id  from employees  where employee\_id=141)  and employee\_id<>141; | |
| Image Result: | |
|  | |
|  | |

**10.Write a non-pair-wise subquery listing the last\_name, first\_name, department\_id, and manager\_id for all employees that have the same department\_ id and manager\_id as employee 141.**

|  |  |
| --- | --- |
| Problem No: 9 | No. Rows in Result: |
| **10.Write a non-pair-wise subquery listing the last\_name, first\_name, department\_id, and manager\_id for all employees that have the same department\_ id and manager\_id as employee 141.** | 4 |
| Text Code (No image) : | |
| --DP10-3  --10.Write a non-pair-wise subquery listing the last\_name, first\_name, department\_id, and manager\_id  --for all employees that have the same department\_ id and manager\_id as employee 141.  SELECT last\_name, first\_name, department\_id, manager\_id  FROM employees  Where (department\_id)= (select department\_id  from employees  where employee\_id=141)  and manager\_id=(select manager\_id  from employees  where employee\_id=141) ; | |
| Image Result: | |
|  | |
|  | |