Alicia Bonavita

Database Programming with SQL 10-1: Fundamentals of Subqueries Practice Activities

- *Vocabulary:* Identify the vocabulary word for each definition below
  - It accepts a value from the inner query to complete its SELECT statement
    - **outer query**
  - An inner query that returns one or more rows to the outer query
    - **multiple-row subquery**
  - An inner query that is nested within an outer query
    - **subquery**
  - An inner query that compares multiple columns at the same time
    - **multi-column subquery**
  - An inner query that returns only one row to the outer query
    - **single-row subquery**
  - An inner query that compares the multiple columns one at a time in different subqueries
    - **correlated subquery**
  - Another name for a subquery
    - **inner query**
- *Try It / Solve It*
  - 1. What is the purpose of using a subquery?

    To provide information to an outer query, which does not have all the information it needs to form a complete SELECT statement
  - 2. What is a subquery?

    It is the inner query nested within the outer query
  - 3. What DJs on Demand d_play_list_items song_id's have the same event_id as song_id 45?

```
SELECT song_id
FROM d_play_list_items
WHERE event_id = (
   SELECT event_id
   FROM d_play_list_items
   WHERE song_id = 45
   LIMIT 1
);
```

  - 4. Which events in the DJs on Demand database cost more than event_id = 100?

```
SELECT event_id, event_name, event_cost
```

```
FROM events
WHERE event_cost > (
   SELECT event_cost
   FROM events
   WHERE event_id = 100
   LIMIT 1
);
```

- 5. Find the track number of the song that has the same CD number as “Party Music for All Occasions”

```
SELECT track_number
FROM songs
WHERE cd_number = (
   SELECT cd_number
   FROM songs
   WHERE song_name = 'Party Music for All Occasions'
   LIMIT 1
);
```

- 6. List the DJs on Demand events whose theme code is the same as the code for “Tropical”

```
SELECT event_id, event_name, theme_code
FROM events
WHERE theme_code = (
   SELECT theme_code
   FROM events
   WHERE event_name = 'Tropical'
   LIMIT 1
);
```

- 7. What are the names of the Global Fast Foods staff members whose salaries are greater than the staff member whose ID is 12?

```
SELECT first_name, last_name
FROM staff
WHERE salary > (
   SELECT salary
   FROM staff
   WHERE staff_id = 12
   LIMIT 1
)
AND company = 'Global Fast Foods';
```

- 8. What are the names of the Global Fast Foods staff members whose staff types are not the same as Bob Miller’s?

```
SELECT first_name, last_name
FROM staff
WHERE staff_type != (
   SELECT staff_type
   FROM staff
   WHERE first_name = 'Bob' AND last_name = 'Miller'
   LIMIT 1
)
AND company = 'Global Fast Foods';
```

- 9. Which Oracle employees have the same department ID as the IT department?

  ```
  SELECT employee_id, first_name, last_name
  FROM employees
  WHERE department_id = (
     SELECT department_id
     FROM departments
     WHERE department_name = 'IT'
     LIMIT 1
  )
  AND company = 'Oracle';
  ```

- 10. What are the department names of the Oracle departments that have the same location ID as Seattle?

  ```
  SELECT department_name
  FROM departments
  WHERE location_id = (
     SELECT location_id
     FROM locations
     WHERE city = 'Seattle'
     LIMIT 1
  )
  AND company = 'Oracle';
  ```

- 11. Indicate whether the statement regarding subqueries is True or False
  - a. It is good programming practice to place a subquery on the right side of the comparison operator

    True
  - b. A subquery can reference a table that is not included in the outer query's FROM clause

    True
  - c. Single-row subqueries can return multiple values to the outer query

    False

Database Programming with SQL 10-2: Single-Row Subqueries Practice Activities

- *Try It / Solve It*
  - 1. Write a query to return all those employees who have a salary greater than that of Lorentz and are in the same department as Abel

    ```
    SELECT employee_id, first_name, last_name, salary, department_id
    FROM employees
    WHERE salary > (
       SELECT salary
       FROM employees
       WHERE first_name = 'Lorentz'
       LIMIT 1
    )
    AND department_id = (
       SELECT department_id
       FROM employees
       WHERE first_name = 'Abel'
       LIMIT 1
    );
    ```
  - 2. Write a query to return all those employees who have the same job id as Rajs and were hired after Davies

    ```
    SELECT employee_id, first_name, last_name, job_id, hire_date
    FROM employees
    WHERE job_id = (
       SELECT job_id
       FROM employees
       WHERE first_name = 'Rajs'
       LIMIT 1
    )
    AND hire_date > (
       SELECT hire_date
       FROM employees
       WHERE first_name = 'Davies'
       LIMIT 1
    );
    ```
  - 3. What DJs on Demand events have the same theme code as event ID = 100?

    ```
    SELECT event_id, theme_code
    FROM events
    WHERE theme_code = (
       SELECT theme_code
       FROM events
    ```

```
    WHERE event_id = 100
);
```

- 4. What is the staff type for those Global Fast Foods jobs that have a salary less than those of any Cook staff-type jobs?

```
SELECT staff_type
FROM jobs
WHERE company = 'Global Fast Foods'
  AND salary < (
    SELECT MIN(salary)
    FROM jobs
    WHERE staff_type = 'Cook'
);
```

- 5. Write a query to return a list of department id’s and average salaries where the department’s average salary is greater than Ernst’s salary

```
SELECT department_id, AVG(salary) AS avg_salary
FROM employees
GROUP BY department_id
HAVING AVG(salary) > (
    SELECT salary
    FROM employees
    WHERE first_name = 'Ernst'
    LIMIT 1
);
```

- 6. Return the department ID and minimum salary of all employees, grouped by department ID, having a minimum salary greater than the minimum salary of those employees whose department ID is not equal to 50

```
SELECT department_id, MIN(salary) AS min_salary
FROM employees
GROUP BY department_id
HAVING MIN(salary) > (
    SELECT MIN(salary)
    FROM employees
    WHERE department_id != 50
);
```

Database Programming with SQL 10-3: Multiple-Row Subqueries Practice Activities

- *Try It / Solve It*
  - 1. What will be returned by a query if it has a subquery that returns a null?

If IN or ANY are used, then the query will return rows of the outer query that match the non-null values. However, if ALL is used, the query will return no rows because the ALL function is meant to compare every outer query row with every value from the subquery, which include the null values

- 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 s.id, s.title, s.duration, s.artist_name
FROM d_songs s
WHERE s.type_id IN (
   SELECT t.id
   FROM d_types t
   WHERE t.genre IN ('Jazz', 'Pop')
);
```

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

```
SELECT e.last_name
FROM employees e
WHERE e.salary = (
   SELECT MIN(salary)
   FROM employees
   GROUP BY department_id
);
```

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

```
SELECT e.*
FROM employees e
WHERE e.salary = (
   SELECT MIN(salary)
   FROM employees
   WHERE company = 'Global Fast Foods'
);
```

- 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 FROM d_cds WHERE title = 'Carpe
Diem');
```

b. *Which employees have salaries lower than any one of the programmers in the IT department?*
WHERE salary < ANY (SELECT salary FROM employees WHERE department = 'IT' AND job_title = 'Programmer');
c. *What CD titles were produced in the same year as "Party Music for All Occasions" or "Carpe Diem"?*
WHERE year IN (SELECT year FROM d_cds WHERE title = 'Party Music for All Occasions' OR title = 'Carpe Diem');
d. *What song title has a duration longer than every type code 77 title?*
WHERE duration > ALL (SELECT duration FROM songs WHERE type_code = 77);

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

  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
  - **True**

  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
  - **True**

- 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;
```

```
SELECT department_id, MIN(salary) AS min_salary
FROM employees
GROUP BY department_id
HAVING MIN(salary) < (SELECT MIN(salary) FROM
employees WHERE department_id = 50);
```

- 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);
```

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

  b. *The query wants the names of employees who make the same salary as the smallest salary in any department*
  - **True**

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

  d. *This query will not execute*
  - **False**

- 9. Write a pairwise 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 e
WHERE (e.department_id, e.manager_id) =
   (SELECT department_id, manager_id
    FROM employees
    WHERE employee_id = 141)
AND e.employee_id != 141;
```

- 10. Write a non-pairwise 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)
AND employee_id != 141;
```

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

- *Try It / Solve It*
  - 1. Explain the main difference between correlated and non-correlated subqueries

    The difference is between the dependency of the subquery. Correlated subqueries depend on the outer query. Each subquery is executed once for every row of the outer query. While for non-correlated queries, they are independent of the outer query and is executed only once for the result 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

    ```
    SELECT last_name, department_id, salary
    FROM employees e
    WHERE salary = (
       SELECT MAX(salary)
       FROM employees
       WHERE department_id = e.department_id
    )
    ORDER BY department_id;
    ```
  - 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. Finish off the statement by sorting the rows on the department_id column

    ```
    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 last_name, department_id, salary
FROM employees outer
WHERE 'x' IN (
   SELECT 'x'
   FROM employees inner
   WHERE inner.manager_id = outer.employee_id
)
ORDER BY department_id;
```

- 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 job_id, MAX(salary) AS job_total
   FROM employees
   GROUP BY job_id
)
SELECT j.job_title, m.job_total
FROM jobs j
JOIN MAX_CALC_SAL m ON j.job_id = m.job_id
WHERE m.job_total > (SELECT MAX(salary) / 2 FROM employees)
ORDER BY m.job_total DESC;
```