

### **10-1: Fundamentals of Subqueries**

Correlated Subquery	It accepts a value from the inner query to complete its SELECT statement.
Multiple-Row Subquery	An inner query that returns one or more rows to the outer query
Nested Subquery	An inner query that is nested within an outer query
Column Comparison Subquery	An inner query that compares multiple columns at the same time
Single-Row Subquery	An inner query that returns only one row to the outer query
Multiple-Column Subquery	An inner query that compares the multiple columns one at a time in different subqueries
Inner Query	Another name for a subquery

1. What is the purpose of using a subquery?
  - A subquery helps filter data within a larger query
2. What is a subquery?
  - A subquery is a nested query within another 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);
```

4. Which events in the DJs on Demand database cost more than event\_id = 100?

```
SELECT event_id, cost
FROM events
WHERE cost > (SELECT cost FROM events WHERE event_id = 100);
```

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 title = 'Party
Music for All Occasions');
```

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

```
SELECT event_id, theme_code
FROM events
WHERE theme_code = (SELECT theme_code FROM themes WHERE
theme_name = 'Tropical');
```

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 staff_name
FROM staff
WHERE salary > (SELECT salary FROM staff WHERE staff_id = 12);
```

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

```
SELECT staff_name
FROM staff
WHERE staff_type <> (SELECT staff_type FROM staff WHERE staff_name =
'Bob Miller');
```

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

```
SELECT employee_name
FROM employees
WHERE department_id = (SELECT department_id FROM departments WHERE
department_name = 'IT');
```

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');
```

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**

## **10-2 Single-Row Subqueries**

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_name
FROM employees
WHERE salary > (SELECT salary FROM employees WHERE employee_name =
'Lorentz')
AND department_id = (SELECT department_id FROM employees WHERE
employee_name = 'Abel');
```

2. Write a query to return all those employees who have the same job id as Rajs and were hired after Davies.

```
SELECT employee_name
FROM employees
WHERE job_id = (SELECT job_id FROM employees WHERE employee_name =
'Rajs')
AND hire_date > (SELECT hire_date FROM employees WHERE
employee_name = 'Davies');
```

3. What DJs on Demand events have the same theme code as event ID = 100?

```
SELECT event_id, event_name
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 DISTINCT staff_type
FROM jobs
WHERE 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, MIN(salary) AS min_salary
FROM employees
GROUP BY department_id
HAVING MIN(salary) > (SELECT MIN(salary) FROM employees WHERE
department_id <> 50);
```

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);
```

### **10-3: Multiple-Row Subqueries Practice Activities**

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  
);
```

```
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

**Answer: 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

**Answer: 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

**Answer: True**

d. WHERE color NOT IN -- If the inner query returns red, green, blue, black, and then the outer query could return white

**Answer: 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

**Answer: 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;
```

Corrected:

```
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)

**Answer: False**

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

**Answer: True**

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

**Answer: False**

d. This query will not execute

**Answer: 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;
```

#### **10-4: Correlated Subqueries Practice Activities**

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)
Completed version:
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;
```