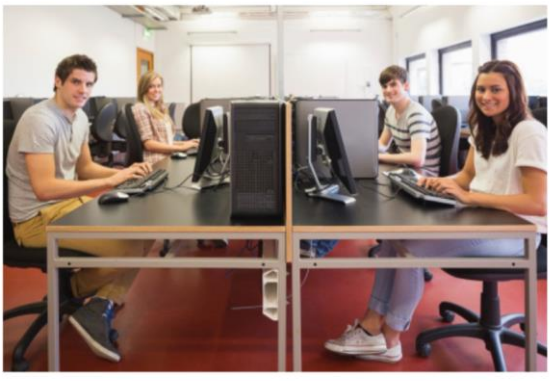

ORACLE
Academy

Database Programming with SQL

10-2

Single-Row Subqueries



ORACLE
Academy

Objectives

- This lesson covers the following objectives:
  - Construct and execute a single-row subquery in the WHERE clause or HAVING clause
  - Construct and execute a SELECT statement using more than one subquery
  - Construct and execute a SELECT statement using a group function in the subquery

Purpose

- As you have probably realized, subqueries are a lot like Internet search engines
- They are great at locating the information needed to accomplish another task
- In this lesson, you will learn how to create even more complicated tasks for subqueries to do for you
- Keep in mind that subqueries save time in that you can accomplish two tasks in one statement

Facts About Single-row Subqueries

- They:
  - Return only one row
  - Use single-row comparison operators (=, >,>=, <, <=, <>)
- Always:
  - Enclose the subquery in parentheses
  - Place the subquery on the right hand side of the comparison condition

Additional Subquery Facts

- The outer and inner queries can get data from different tables
- Only one ORDER BY clause can be used for a SELECT statement, and if specified, it must be the last clause in the main SELECT statement
- The only limit on the number of subqueries is the buffer size that the query uses

Subqueries from Different Tables

- The outer and inner queries can get data from different tables
- Who works in the Marketing department?

```
SELECT last_name, job_id, department_id
FROM employees
WHERE department_id =
    (SELECT department_id
     FROM departments
     WHERE department_name = 'Marketing')
ORDER BY job_id;
```

Result of subquery

DEPARTMENT_ID
20

LAST_NAME	JOB_ID	DEPARTMENT_ID
Hartstein	MK_MAN	20
Fay	MK_REP	20

ORACLE Academy

The subquery finds the department_id for 'Marketing', the outer query uses the returned department_id to display rows from the employees table.

Subqueries from Different Tables

- More than one subquery can return information to the outer query

```
SELECT last_name, job_id, salary, department_id
FROM employees
WHERE job_id =
   (SELECT job_id
    FROM employees
    WHERE employee_id = 141)
AND department_id =
   (SELECT department_id
    FROM departments
    WHERE location_id = 1500);
```

Result of 1st subquery

JOB_ID
ST_CLERK

Result of 2nd subquery

DEPARTMENT_ID
50

LAST_NAME	JOB_ID	SALARY	DEPARTMENT_ID
Rajs	ST_CLERK	3500	50
Davies	ST_CLERK	3100	50
Matos	ST_CLERK	2600	50
Vargas	ST_CLERK	2500	50

ORACLE Academy

The first subquery returns the job_id of employee 141 (ST_CLERK). The second subquery uses the departments table to find the department_id at location_id 1500 (50).

The outer query then returns rows from the employees table that match both these values.

Note: in this instance, as each department is situated at only one location, it is safe to use a singe-row subquery. If a department could be in multiple locations, we would need to use a multiple-row subquery.

Group Functions in Subqueries

- Group functions can be used in subqueries
- A group function without a GROUP BY clause in the subquery returns a single row
- The query on the next slide answers the question, "Which employees earn less than the average salary?"

Group Functions in Subqueries

- The subquery first finds the average salary for all employees, the outer query then returns employees with a salary of less than the average

```
SELECT last_name, salary
FROM employees
WHERE salary <
   (SELECT AVG(salary)
    FROM employees);
```

Result of subquery

AVG(SALARY)
8775

LAST_NAME	SALARY
Whalen	4400
Gietz	8300
Taylor	8600
Grant	7000
Mourgos	5800
Rajs	3500
Davies	3100
Matos	2600
Vargas	2500
Ernst	6000
Lorentz	4200
Fay	6000

ORACLE Academy

Subqueries in the HAVING Clause

- Subqueries can also be placed in the HAVING clause
- Remember that the HAVING clause is similar to the WHERE clause, except that the HAVING clause is used to restrict groups and always includes a group function such as MIN, MAX, or AVG
- Because the HAVING clause always includes a group function, the subquery will nearly always include a group function as well

Subquery Example

- Which departments have a lowest salary that is greater than the lowest salary in department 50?
- In this example, the subquery selects and returns the lowest salary in department 50

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

DEPARTMENT_ID	MIN(SALARY)
-	7000
90	17000
20	6000
110	8300
80	8600
10	4400
60	4200

Result of subquery

MIN(SALARY)
2500

ORACLE Academy

Subquery Example

- The outer query uses this value to select the department ID and lowest salaries of all the departments whose lowest salary is greater than that number
- The HAVING clause eliminated those departments whose MIN salary was less than department 50's MIN salary

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

DEPARTMENT_ID	MIN(SALARY)
-	7000
90	17000
20	6000
110	8300
80	8600
10	4400
60	4200

Result of subquery

MIN(SALARY)
2500

Summary

- In this lesson, you should have learned how to:
  - Construct and execute a single-row subquery in the WHERE clause or HAVING clause
  - Construct and execute a SELECT statement using more than one subquery
  - Construct and execute a SELECT statement using a group function in the subquery


ORACLE
Academy