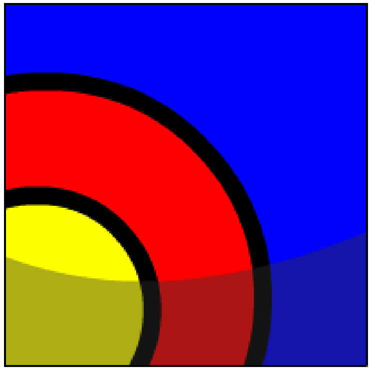
Subqueries

What Will I Learn?

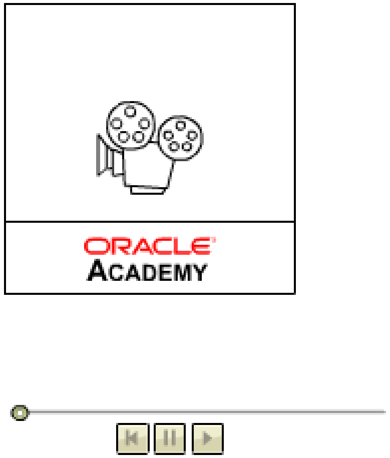


In this lesson, you will learn to:

- Define and explain the purpose of subqueries for retrieving data
- Construct and execute a single-row subquery in the WHERE clause
- Distinguish between single-row and multiple-row subqueries
- Distinguish between pair-wise and non-pair-wise subqueries

Why Learn It?

Has a friend asked you to go to a movie, but before you could answer "yes" or "no," you first had to check with your parents? Has someone asked you the answer to a math problem, but before you can give the answer, you had to do the problem yourself?



Asking parents, or doing the math problem, are examples of subqueries.

In SQL, subqueries enable us to find the information we need so we can get the information we want.

Tell Me / Show Me

Throughout this course, you have written queries to extract data from a database. What if you wanted to write a query only to find out you didn't have all the information you needed to construct it?

You can solve this problem by combining two queries, placing one query inside the other query. The inner query is called the "subquery." The subquery executes to find the information you don’t know. The outer query uses that information to find out what you need to know.

Being able to combine two queries into one can be very useful when you need to select rows from a table with a condition that depends on the data in the table itself.

Tell Me / Show Me

A subquery is a SELECT statement that is embedded in a clause of another SELECT statement. A subquery executes once before the main query The result of the subquery is used by the main or outer query.

Subqueries can be placed in a number of SQL clauses, including the WHERE clause, the HAVING clause, and the FROM clause.

The subquery syntax is:

```
SELECT select_list
FROM table
WHERE expression operator
(SELECT select_list
FROM table);
```

The SELECT statement in parentheses is the inner query or ‘subquery’.

Tell Me / Show Me

Guidelines for using subqueries are:

- The subquery is enclosed in parentheses.
- The subquery is placed on the right side of the comparison condition.
- 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 used, it must be the last clause in the outer query. A subquery cannot have its own ORDER BY clause.
- The only limit on the number of subqueries is the buffer size the query uses.

Tell Me / Show Me

There are two types of subqueries:

- **Single-row subqueries** that use single-row operators (>, =, >=, < <>, <=) and return only one row from the inner query.

- **Multiple-row subqueries** that use multiple-row operators (IN, ANY, ALL) and return more than one row from the inner query.

Tell Me / Show Me

What if you wanted to find out the names of the Global Fast Foods staff members that were born after Monique Tuttle? What is the first thing you need to know? When was Monique born? Once you know her birth date, then you can select those staff members whose birth dates are after hers.

```
SELECT staff_id, first_name, last_name, birth_date
FROM f_staffs
WHERE birth_date >=
    (SELECT birth_date
     FROM f_staffs
     WHERE last_name = ‘Tuttle’);
```

STAFF_ID	FIRST_NAME	LAST_NAME	BIRTH_DATE
1000	Roger	Morgan	17-JUN-87
1189	Nancy	Vickers	21-SEP-89
1007	Monique	Tuttle	13-JAN-87
1354	Alex	Hunter	03-JAN-90
1423	Kathryn	Bassman	08-AUG-91

Tell Me / Show Me

If a subquery returns a null value or no rows, the outer query takes the results of the subquery (null) and uses this result in its WHERE clause.

The outer query will then return no rows, because comparing a value with null always yields a null.

Who works in the same department as Grant? Grant’s department_id is null.

The outer query does not even return Grant’s row, because comparing a null with a null returns a null.

```
SELECT last_name
FROM employees
WHERE department_id =
   (SELECT department_id
    FROM employees
    WHERE last_name = ‘Grant’);
```

```
No data found.
```

Tell Me / Show Me

MULTIPLE-COLUMN SUBQUERIES

Subqueries can use one or more columns. If they use more than one column, they are called multiple-column subqueries. A multiple-column subquery can be either pair-wise comparisons or non-pair-wise comparisons.

The example on the right shows a multiple-column pair-wise subquery with the subquery highlighted in red and the result in the table below.

The query is listing the employees whose manager and departments are the same as the manager and department of employees 149 or 174.

```
SELECT  employee_id,
        manager_id,
        department_id
FROM    employees
WHERE     (manager_id,department_id) IN
      (SELECT manager_id,department_id
       FROM   employees
       WHERE  employee_id IN  (149,174))
AND     employee_id NOT IN (149,174)
```

EMPLOYEE_ID	MANAGER_ID	DEPARTMENT_ID
176	149	80

Tell Me / Show Me

MULTIPLE-COLUMN SUBQUERIES

A non-pair-wise multiple-column subquery also uses more than one column in the subquery, but it compares them one at a time, so the comparisons take place in different subqueries. You will need to write one subquery per column you want to compare against when performing non-pair-wise multiple column subqueries.

The example on the right shows a multiple-column non-pair-wise subquery with the subqueries highlighted in red.

This query is listing the employees who have either a manager_id or a department_id in common with employees 174 or 199.

```
SELECT  employee_id,
        manager_id,
        department_id
FROM    employees
WHERE   manager_id IN
          (SELECT  manager_id
           FROM    employees
           WHERE   employee_id IN
                        (174,199))
AND     department_id IN
          (SELECT  department_id
           FROM    employees
           WHERE   employee_id IN
                         (174,199))
AND     employee_id NOT IN(174,199);
```

EMPLOYEE_ID	MANAGER_ID	DEPARTMENT_ID
176	149	80
149	100	80

Tell Me / Show Me

Terminology

Key terms used in this lesson include:

Inner query

Multiple- row subquery

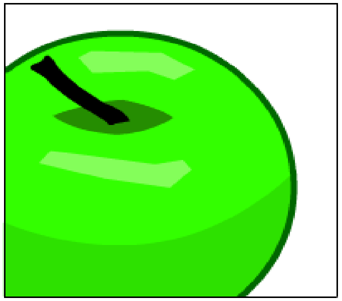
Outer subquery

Single row subquery

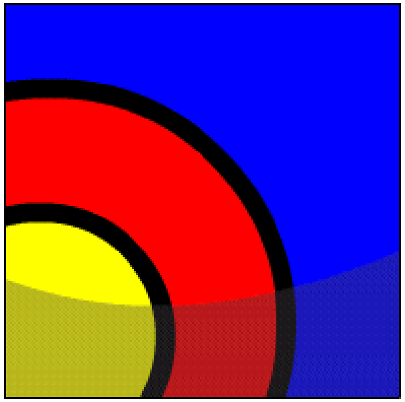
Subquery

Pair-wise multiple column subquery

Non-pair-wise multiple column subquery



Summary

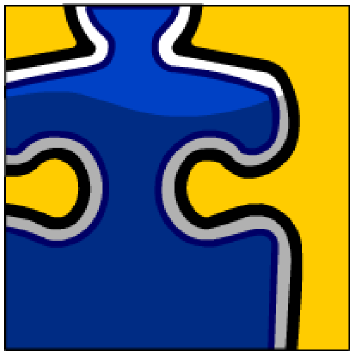


In this lesson you have learned to:

- Define and explain the purpose of subqueries for retrieving data
- Construct and execute a single-row subquery in the WHERE clause
- Distinguish between single-row and multiple-row subqueries
- Distinguish between pair-wise and non-pair-wise subqueries

Summary

Practice Guide



The link for the lesson practice guide can be found in the course resources in Section 0.