7

Subqueries

ORACLE®

Objectives

After completing this lesson, you should be able to do the following:

- **Describe the types of problems that subqueries can solve**
- **Define subqueries**
- **List the types of subqueries**
- **Write single-row and multiple-row subqueries**

6-2

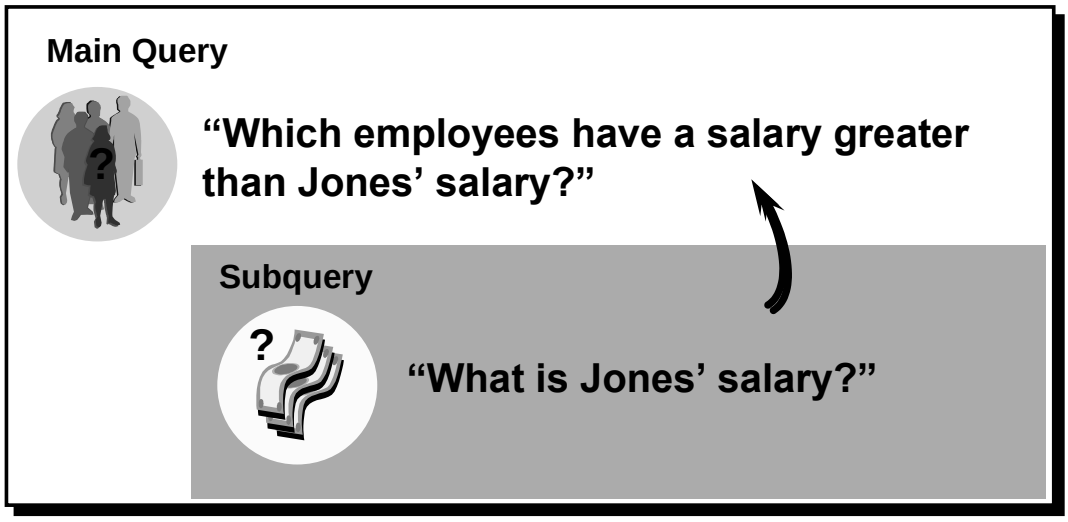
ORACLE®

Lesson Aim

In this lesson, you will learn about more advanced features of the SELECT statement. You can write subqueries in the WHERE clause of another SQL statement to obtain values based on an unknown conditional value. This lesson covers single-row subqueries and multiple-row subqueries.

Using a Subquery to Solve a Problem

"Who has a salary greater than Jones'?"



6-3

ORACLE®

Using aSubqueryto Solve a Problem

Suppose you want to write a query to find out who earns a salary greater than Jones' salary.

To solve this problem, you need *two* queries: one query to find what Jones earns and a second query to find who earns more than that amount.

You can solve this problem by combining the two queries, placing one query *inside* the other query.

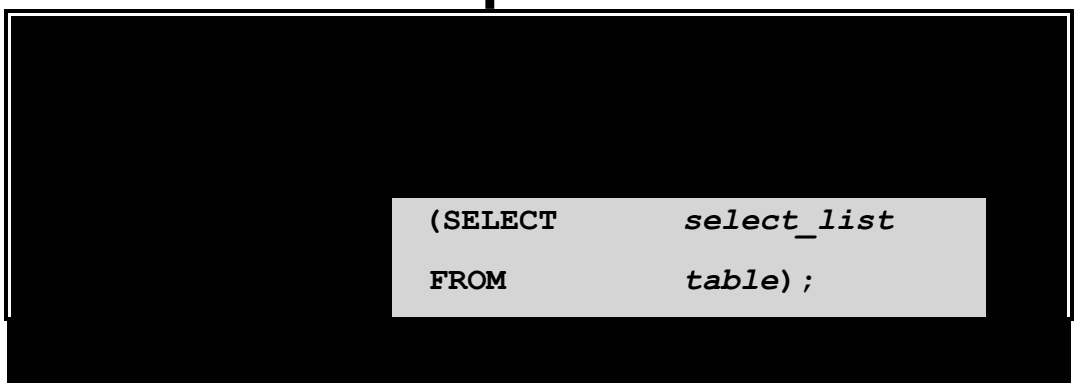
The inner query or the *subquery* returns a value that is used by the outer query or the main query. Using a subquery is equivalent to performing two sequential queries and using the result of the first query as the search value in the second query.

Subqueries

A subquery is a SELECT statement that is embedded in a clause of another SELECT statement. You can build powerful statements out of simple ones by using subqueries. They 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. You can place the subquery in a number of SQL clauses:

- WHERE clause
- HAVING clause

Subqueries



- **The subquery (inner query) executes once before the main query.**
- **The result of the subquery is used by the main query (outer query).**

ORACLE®

- FROM clause

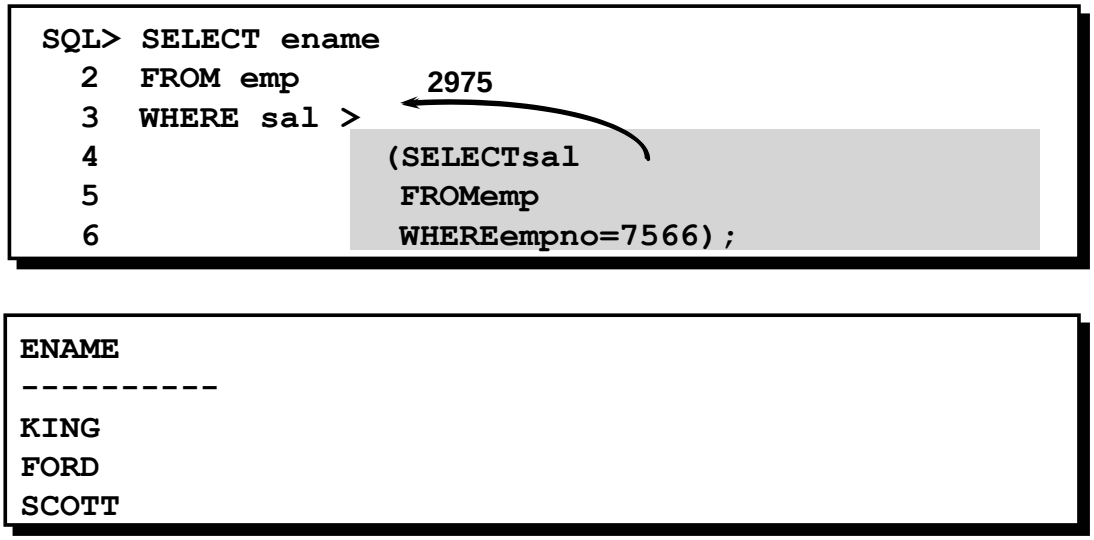
In the syntax:

operator includes a comparison operator such as >, =, or IN

Note: Comparison operators fall into two classes: single-row operators (>, =, >=, <, <>, <=) and multiplerow operators (IN, ANY, ALL).

The subquery is often referred to as a nested SELECT, sub-SELECT, or inner SELECT statement. The subquery generally executes first, and its output is used to complete the query condition for the main or outer query.

Using a Subquery



Using aSubquery

In the slide, the inner query determines the salary of employee 7566. The outer query takes the result of the inner query and uses this result to display all the employees who earn more than this amount.

Guidelines for Using Subqueries

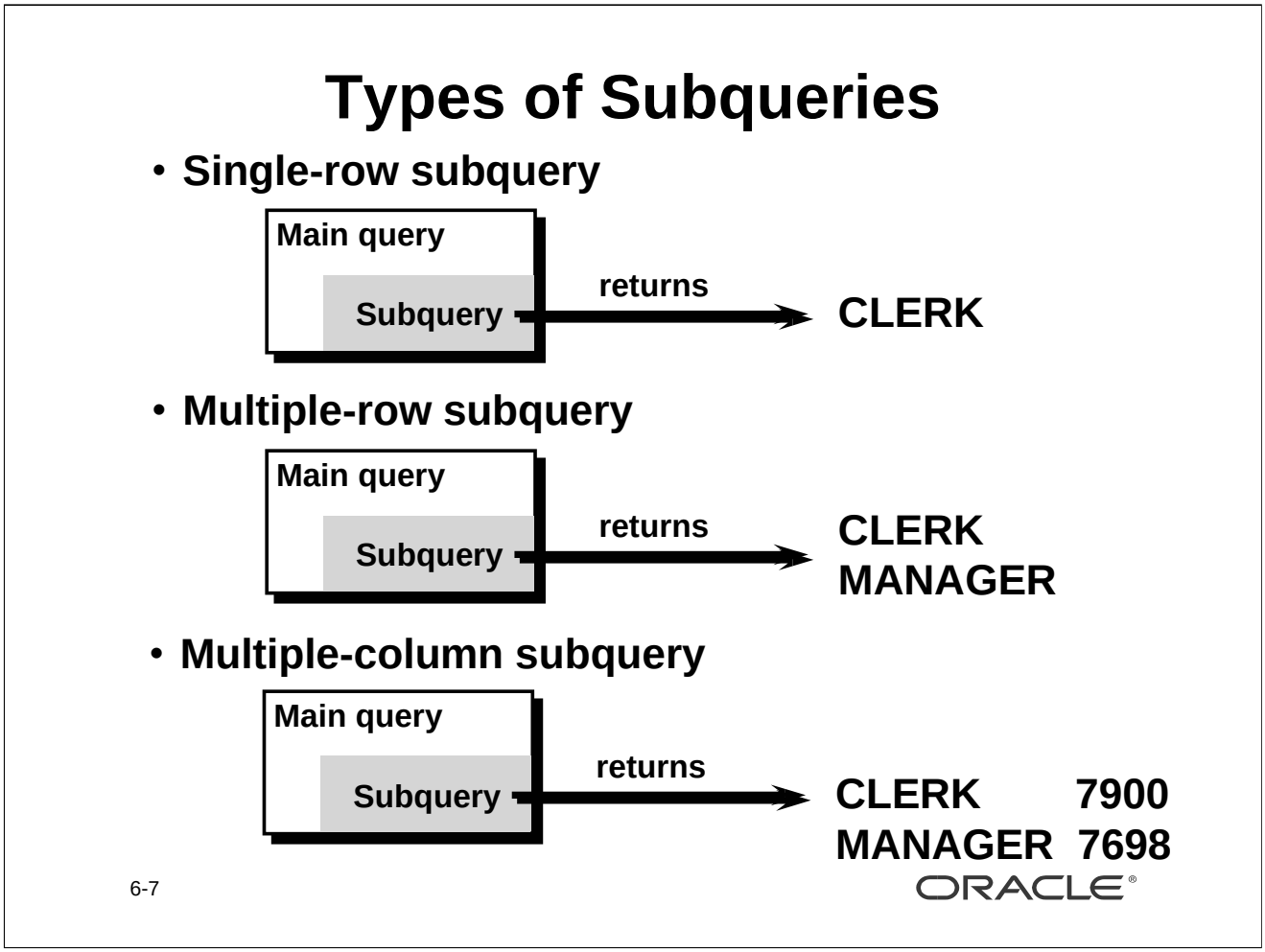
- A subquery must be enclosed in parentheses.
- A subquery must appear on the right side of the comparison operator.

Guidelines for Using Subqueries

- **Enclose subqueries in parentheses.**
- **Place subqueries on the right side of the comparison operator.**
- **Do not add an ORDER BY clause to a subquery.**
- **Use single-row operators with singlerow subqueries.**
- **Use multiple-row operators with multiple-row subqueries.**

ORACLE®

- Subqueries cannot contain an ORDER BY clause. You can have only one ORDER BY clause for a SELECT statement, and if specified it must be the last clause in the main SELECT statement.
- Two classes of comparison operators are used in subqueries: single-row operators and multiple-row operators.



Types ofSubqueries

- Single-row subqueries: Queries that return only one row from the inner SELECT statement
- Multiple-row subqueries: Queries that return more than one row from the inner SELECT statement
- Multiple-column subqueries: Queries that return more than one column from the inner SELECT statement

Single-Row Subqueries

A *single-row subquery* is one that returns one row from the inner SELECT statement. This type of subquery uses a single-row operator. The slide gives a list of single-row operators.

Example

Display the employees whose job title is the same as that of employee 7369.

```
SQL> SELECT   ename, job
  2                FROM    emp
  3                WHERE    job =
```

Single-Row Subqueries

- **Return only one row**
- **Use single-row comparison operators**

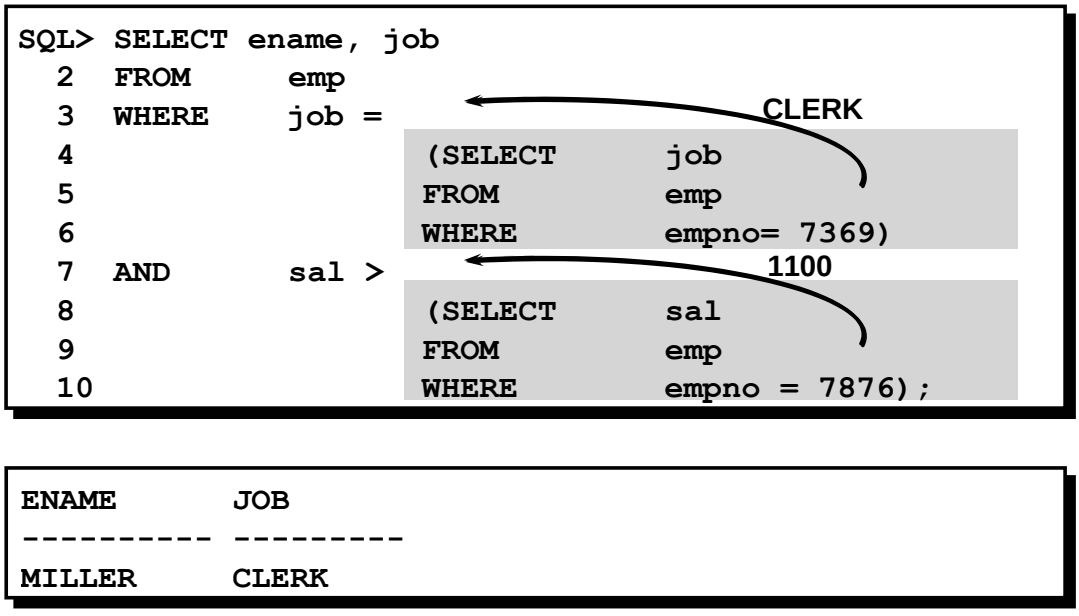
Operator	Meaning
=	Equal to
>	Greater than
>=	Greater than or equal to
<	Less than
<=	Less than or equal to
<>	Not equal to

ORACLE®

```
4                (SELECT  job
5                FROM    emp
6                WHERE   empno = 7369);
```

```
ENAME      JOB
---------- ---------
JAMES      CLERK
SMITH      CLERK
ADAMS      CLERK
MILLER     CLERK
```

Executing Single-Row Subqueries



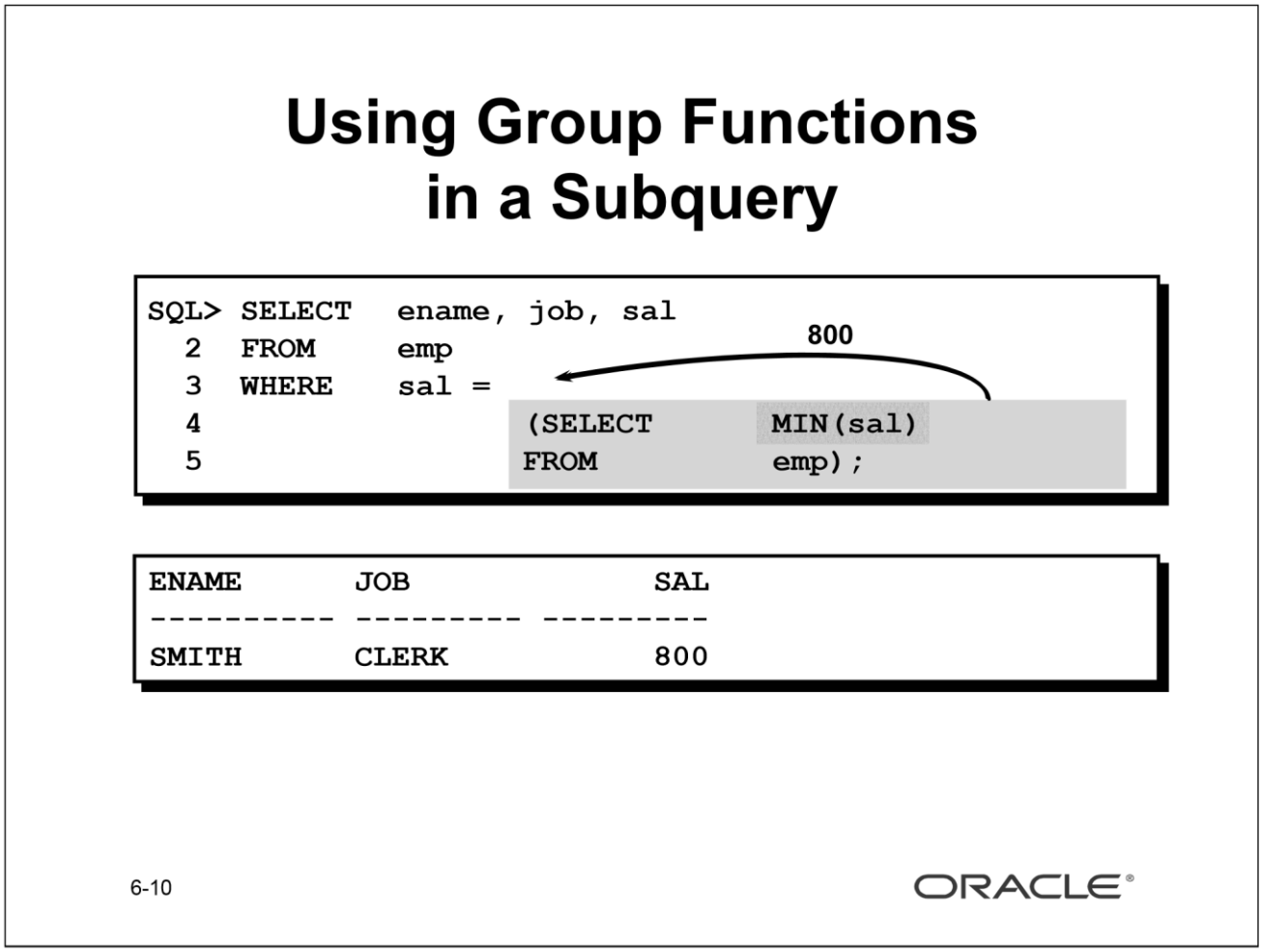
ORACLE®

Executing Single-RowSubqueries

A SELECT statement can be considered as a query block. The example on the slide displays employees whose job title is the same as that of employee 7369 and whose salary is greater than that of employee 7876. The example consists of three query blocks: the outer query and two inner queries. The inner query blocks are executed first, producing the query results: CLERK and 1100, respectively. The outer query block is then processed and uses the values returned by the inner queries to complete its search conditions.

Both inner queries return single values (CLERK and 1100, respectively), so this SQL statement is called a single-row subquery.

Note: The outer and inner queries can get data from different tables.



Using Group Functions in a Subquery

You can display data from a main query by using a group function in a subquery to return a single row. The subquery is in parentheses and is placed after the comparison operator.

The example on the slide displays the employee name, job title, and salary of all employees whose salary is equal to the minimum salary. The MIN group function returns a single value (800) to the outer query.

HAVING Clause with Subqueries

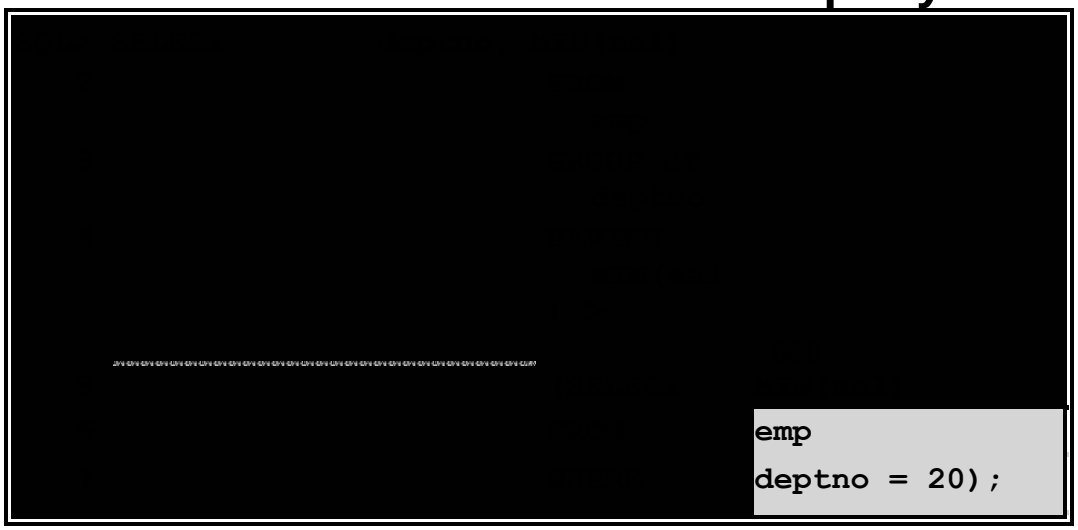
You can use subqueries not only in the WHERE clause, but also in the HAVING clause. The Oracle Server executes the subquery, and the results are returned into the HAVING clause of the main query.

The SQL statement on the slide displays all the departments that have a minimum salary greater than that of department 20.

```
DEPTNO   MIN(SAL)
```

HAVING Clause with Subqueries

- The Oracle Server executes subqueries first.
- The Oracle Server returns results into the HAVING clause of the main query.



6-11

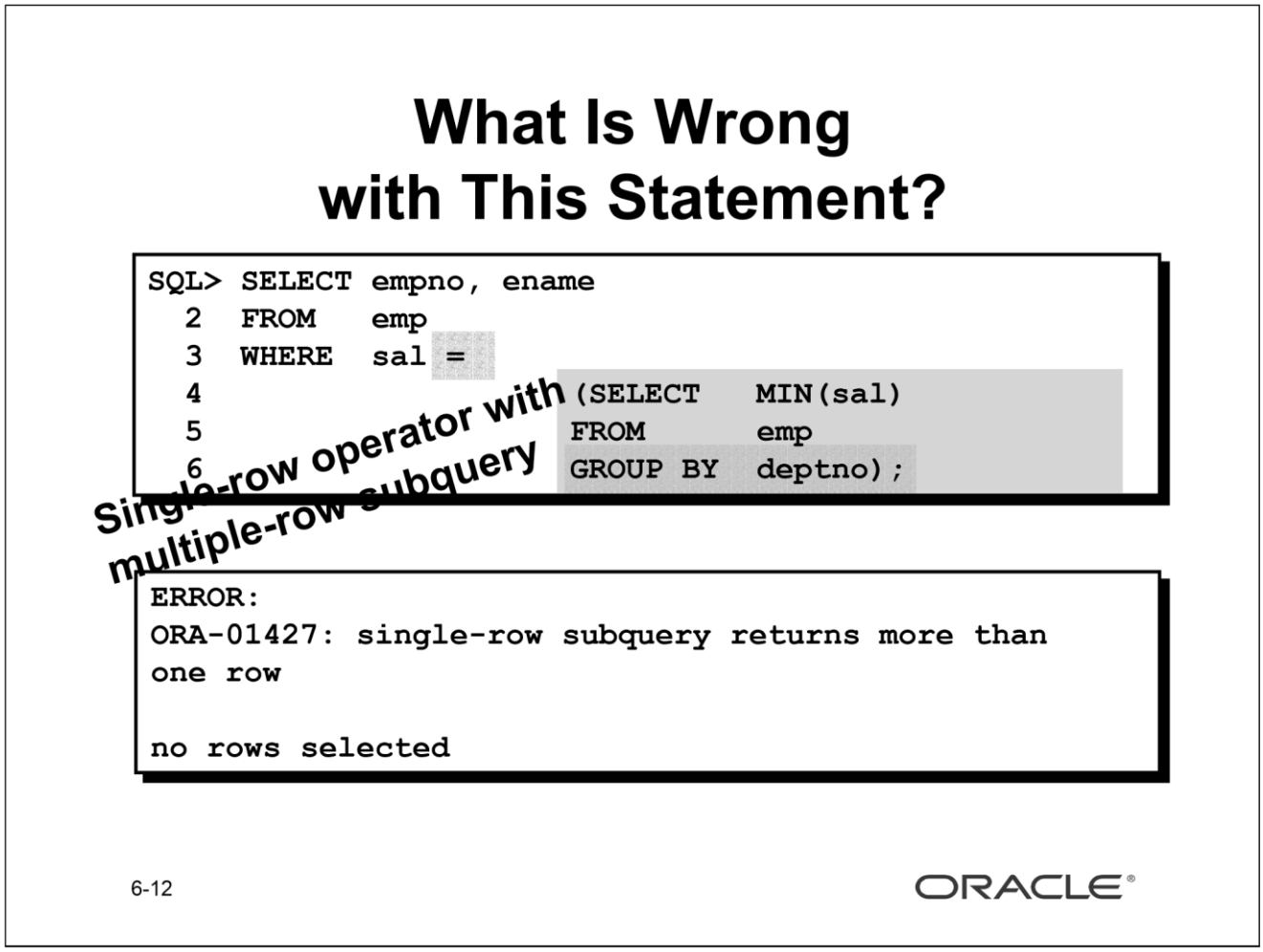
ORACLE®

```
--------- --------10
       1300 30
       950
```

Example

Find the job with the lowest average salary.

```
SQL> SELECT     job, AVG(sal)
  2                         FROM emp
  3                         GROUP BY job
  4                         HAVING    AVG(sal) = (SELECT
                            MIN(AVG(sal))
  5                         FROM         EMP
  6                         GROUP BY     job);
```



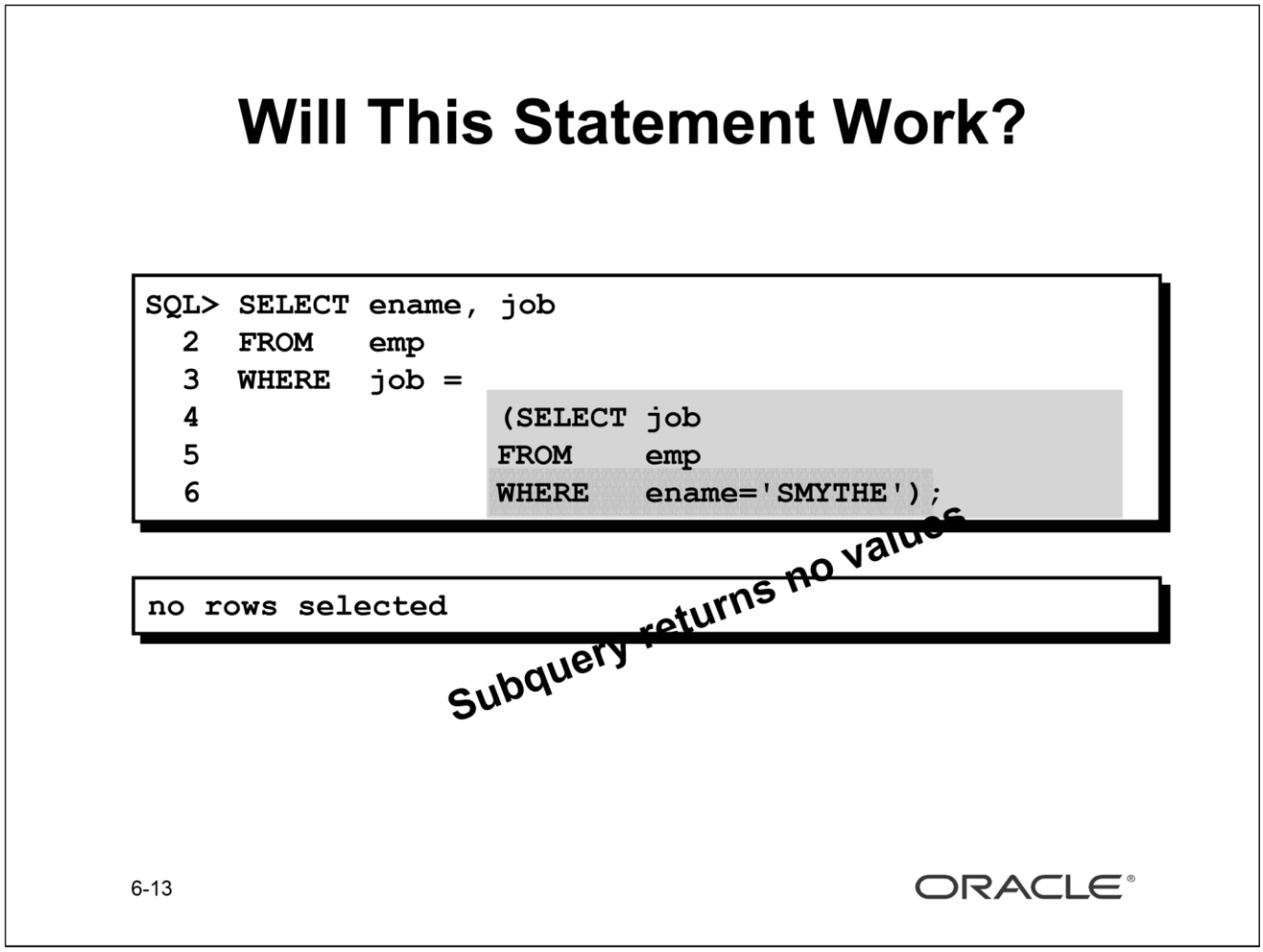
Errors with Subqueries

One common error with subqueries is more than one row returned for a single-row subquery.

In the SQL statement on the slide, the subquery contains a GROUP BY (deptno) clause, which implies that the subquery will return multiple rows, one for each group it finds. In this case, the result of the subquery will be 800, 1300, and 950.

The outer query takes the results of the subquery (800, 950, 1300) and uses these results in its WHERE clause. The WHERE clause contains an equal (=) operator, a single-row comparison operator expecting only one value. The = operator cannot accept more than one value from the subquery and hence generates the error.

To correct this error, change the = operator to IN.



Problems with Subqueries

A common problem with subqueries is no rows being returned by the inner query.

In the SQL statement on the slide, the subquery contains a WHERE (ename='SMYTHE') clause. Presumably, the intention is to find the employee whose name is Smythe. The statement seems to be correct but selects no rows when executed.

The problem is that Smythe is misspelled. There is no employee named Smythe. So the subquery returns no rows. The outer query takes the results of the subquery (null) and uses these results in its WHERE clause. The outer query finds no employee with a job title equal to null and so returns no rows.

Multiple-Row Subqueries

Subqueries that return more than one row are called *multiple-row subqueries*. You use a multiple-row operator, instead of a single-row operator, with a multiple-row subquery. The multiple-row operator expects one or more values.

```
SQL> SELECT    ename, sal, deptno
  2                FROM  emp
  3                WHERE sal IN (SELECT    MIN(sal)
  4                FROM  emp
```

Multiple-Row Subqueries

- **Return more than one row**
- **Use multiple-row comparison operators**

Operator	Meaning
IN	Equal to any member in the list
ANY	Compare value to each value returned by the subquery
ALL	Compare value to every value returned by the subquery

6-14

ORACLE®

```
5                       GROUP BY deptno);
```

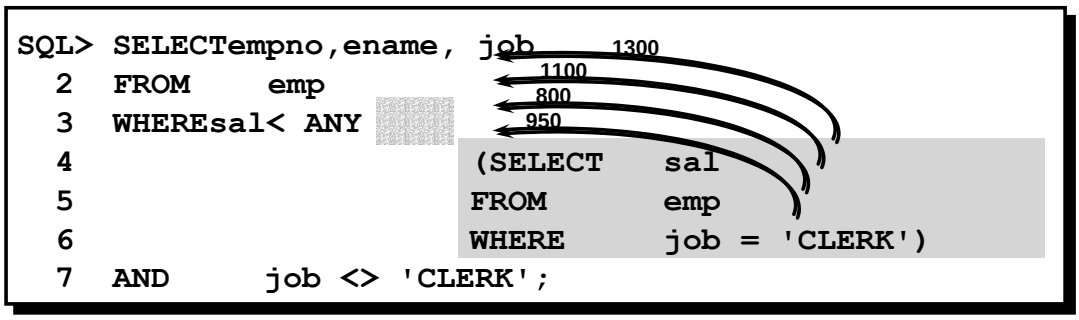
Example

Find the employees who earn the same salary as the minimum salary for departments.

The inner query is executed first, producing a query result containing three rows: 800, 950, 1300. The main query block is then processed and uses the values returned by the inner query to complete its search condition. In fact, the main query would look like the following to the Oracle Server:

```
SQL> SELECT ename, sal, deptno 2
  FROM emp
  3  WHERE    sal IN (800, 950, 1300);
```

Using ANY Operator in Multiple-Row Subqueries



EMPNO	ENAME	JOB
7654	MARTIN	SALESMAN
7521	WARD	SALESMAN

6-15

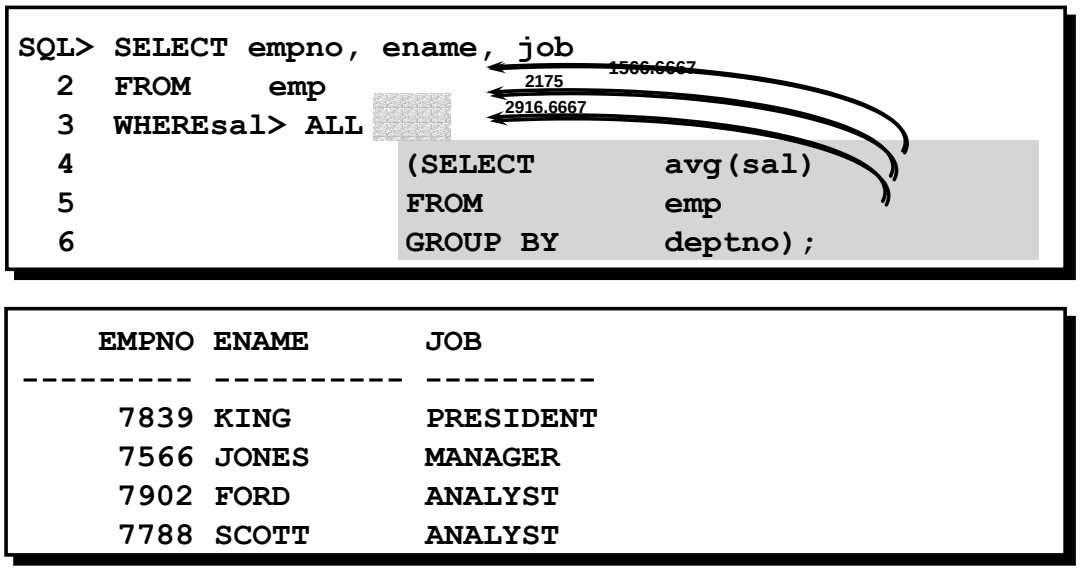
ORACLE®

Multiple-RowSubqueries(continued)

The ANY operator (and its synonym SOME operator) compares a value to *each* value returned by a subquery. The slide example displays employees whose salary is less than any clerk and who are not clerks. The maximum salary that a clerk earns is $1300. The SQL statement displays all the employees who are not clerks but earn less than $1300.

<ANY means less than the maximum. >ANY means more than the minimum. =ANY is equivalent to IN.

Using ALL Operator in Multiple-Row Subqueries



ORACLE®

Multiple-RowSubqueries(continued)

The ALL operator compares a value to *every* value returned by a subquery. The slide example displays employees whose salary is greater than the average salaries of all the departments. The highest average salary of a department is $2916.66, so the query returns those employees whose salary is greater than $2916.66.

>ALL means more than the maximum and <ALL means less than the minimum.

The NOT operator can be used with IN, ANY, and ALL operators.

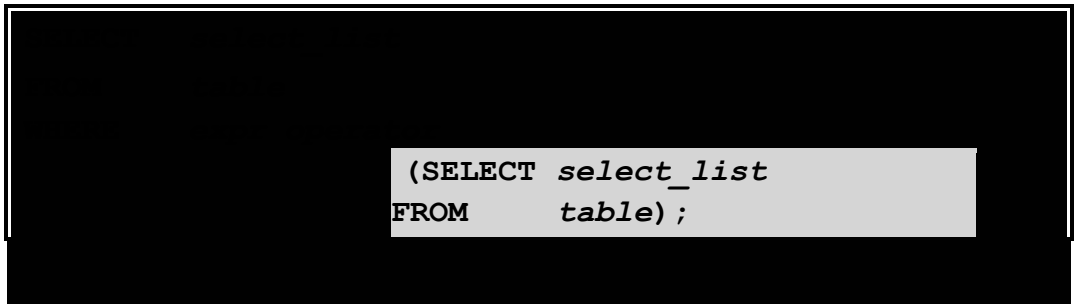
Summary

A subquery is a SELECT statement that is embedded in a clause of another SQL statement. Subqueries are useful when a query is based on a search criteria with unknown intermediate values.

Subqueries have the following characteristics:

Summary

Subqueries are useful when a query is based on unknown values.



6-17

ORACLE®

- Can pass one row of data to a main statement that contains a single-row operator, such as =, <>, >, >=, <, or <=
- Can pass multiple rows of data to a main statement that contains a multiple-row operator, such as IN
- Are processed first by the Oracle Server, and the WHERE or HAVING clause uses the results • Can contain group functions

Practice Overview

In this practice, you will write complex queries using nested SELECT statements.

Practice Overview

- **Creating subqueries to query values based on unknown criteria**
- **Using subqueries to find out what values exist in one set of data and not in another**

ORACLE®

Paper-Based Questions

You may want to consider creating the inner query first for these questions. Make sure that it runs and produces the data that you anticipate before coding the outer query.