6

Subqueries

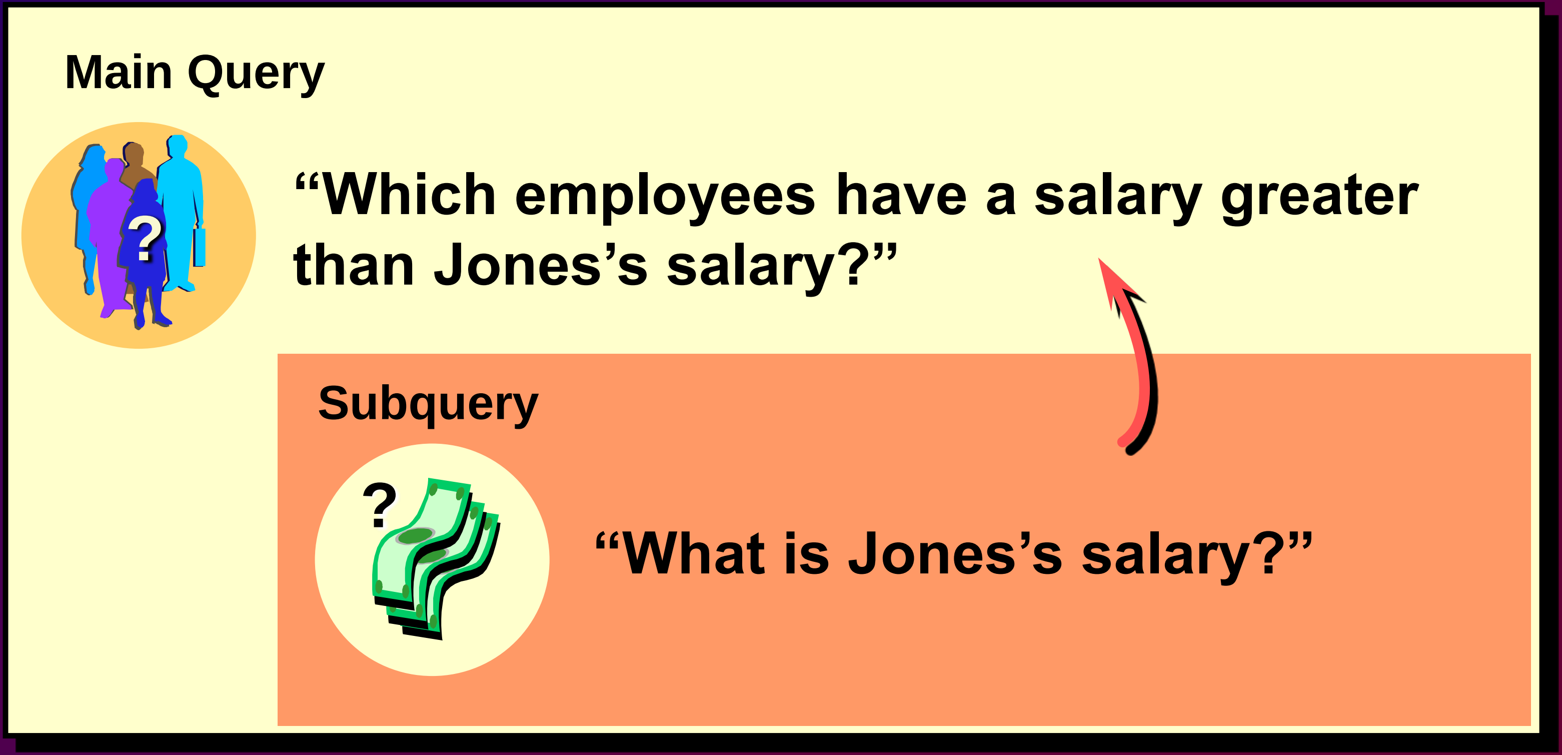
Objectives

At the end of this lesson, you should be able to:

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

Using a Subquery to Solve a Problem

“Who has a salary greater than Jones’s?”



Main Query

“Which employees have a salary greater than Jones’s salary?”

Subquery

“What is Jones’s salary?”

Subqueries

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

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

Using a Subquery

```
SQL> SELECT ename
  2  FROM   emp
  3  WHERE  sal >
  4                  (SELECT sal
  5                   FROM   emp
  6                   WHERE  empno=7566);
```

2975

```
ENAME
----------
KING
FORD
SCOTT
```

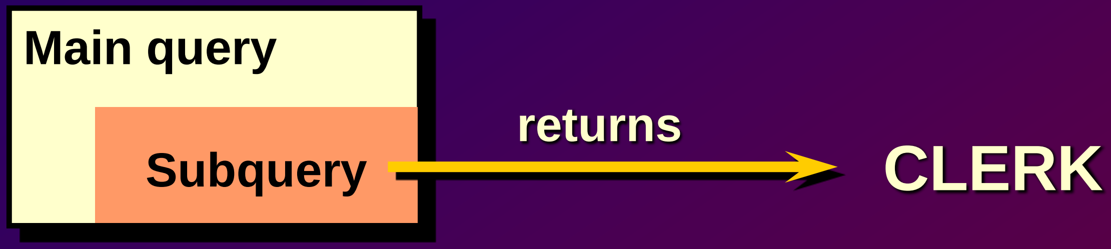
ORACLE®

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 single-row subqueries.
- Use multiple-row operators with multiple-row subqueries.

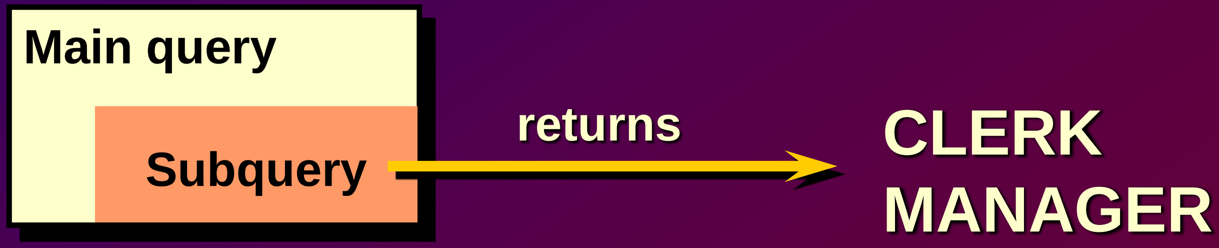
Types of Subqueries

- Single-row subquery



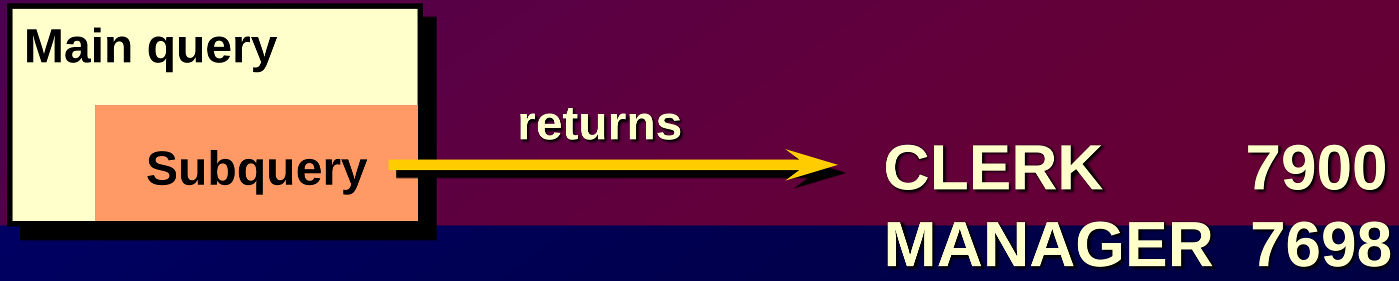
CLERK

- Multiple-row subquery



CLERK
MANAGER

- Multiple-column subquery



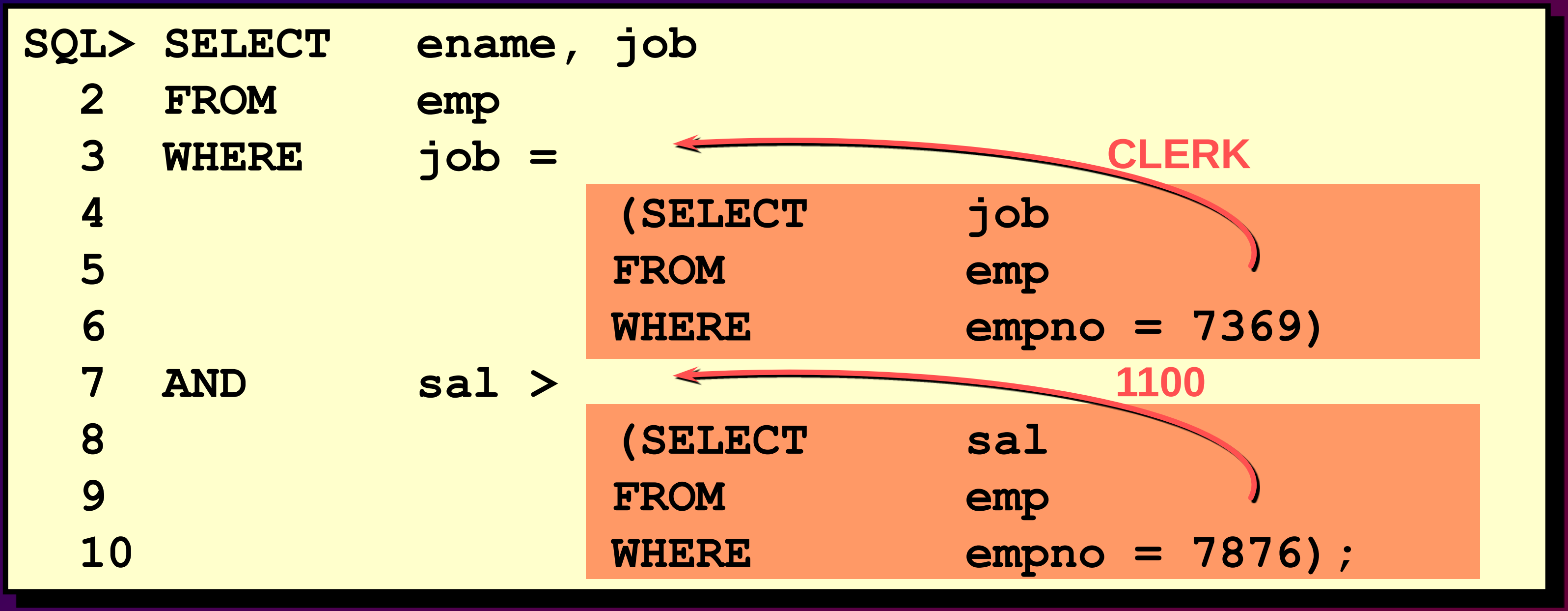
CLERK 7900
MANAGER 7698

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

Executing Single-Row Subqueries

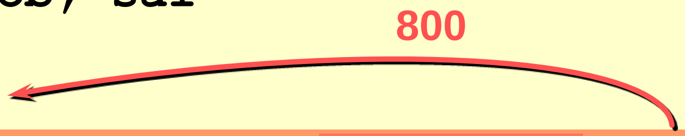


```
SQL> SELECT    ename, job
  2  FROM      emp
  3  WHERE     job =
  4                  (SELECT    job
  5                  FROM       emp
  6                  WHERE      empno = 7369)
  7  AND       sal >
  8                  (SELECT    sal
  9                  FROM       emp
 10                  WHERE      empno = 7876);
```

```
ENAME      JOB
---------- ---------
MILLER     CLERK
```

Using Group Functions in a Subquery

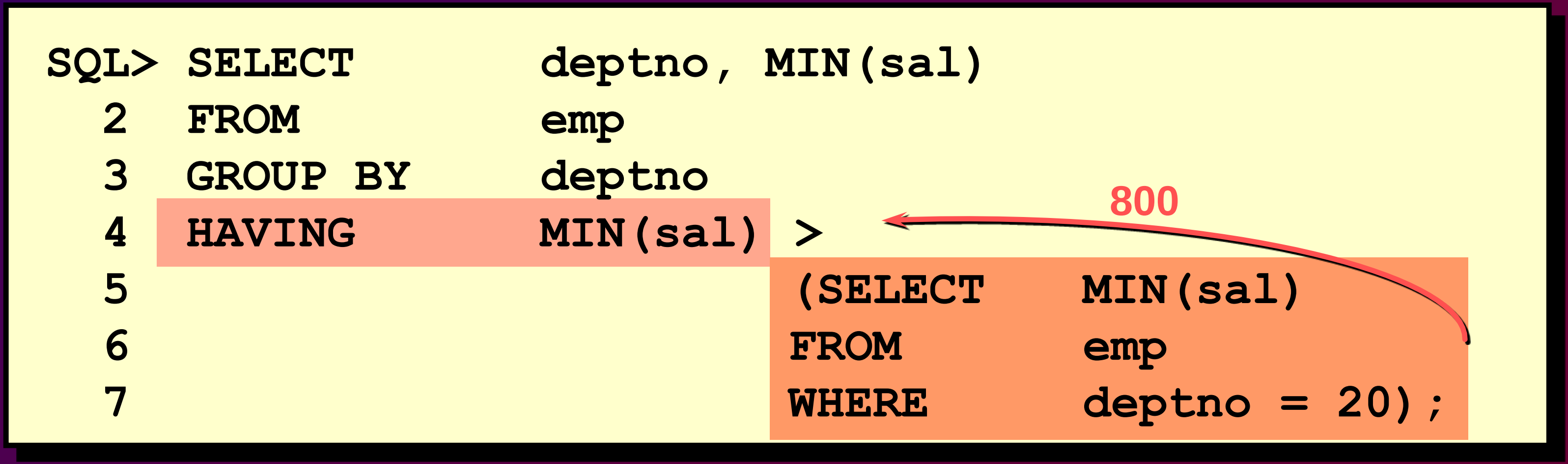
```
SQL> SELECT   ename, job, sal
  2  FROM     emp
  3  WHERE    sal =
  4                   (SELECT          MIN(sal)
  5                   FROM             emp);
```



```
ENAME      JOB              SAL
---------- --------- ----------
SMITH      CLERK            800
```

HAVING Clause with Subqueries

- The Oracle8 Server executes subqueries first.
- The Oracle8 Server returns results into the main query's HAVING clause.



```
SQL> SELECT        deptno, MIN(sal)
  2  FROM          emp
  3  GROUP BY      deptno
  4  HAVING        MIN(sal) >
  5                          (SELECT     MIN(sal)
  6                          FROM        emp
  7                          WHERE       deptno = 20);
```

What Is Wrong with This Statement?

```
SQL> SELECT empno, ename
  2  FROM   emp
  3  WHERE  sal =
  4                  (SELECT   MIN(sal)
  5                  FROM      emp
  6                  GROUP BY  deptno);
```

```
ERROR:
ORA-01427: single-row subquery returns more than
one row

no rows selected
```

Single-row operator with multiple-row subquery

Will This Statement Work?

```
SQL> SELECT ename, job
  2  FROM   emp
  3  WHERE  job =
  4                (SELECT job
  5                 FROM   emp
  6                 WHERE  ename='SMYTHE');
```

```
no rows selected
```

Subquery returns no values

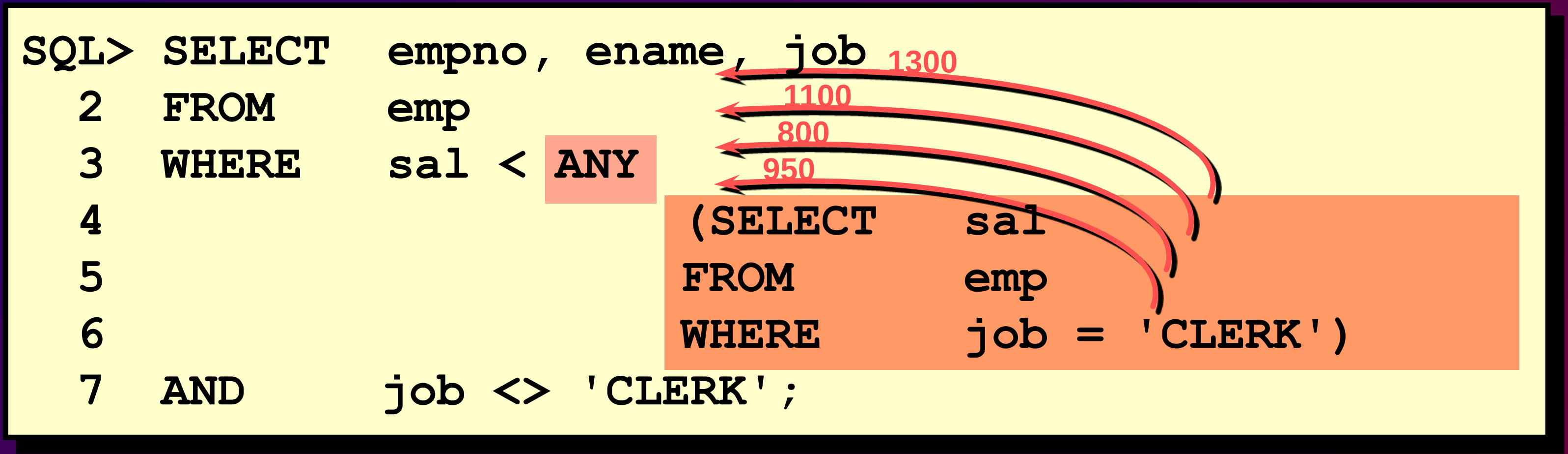
ORACLE®

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

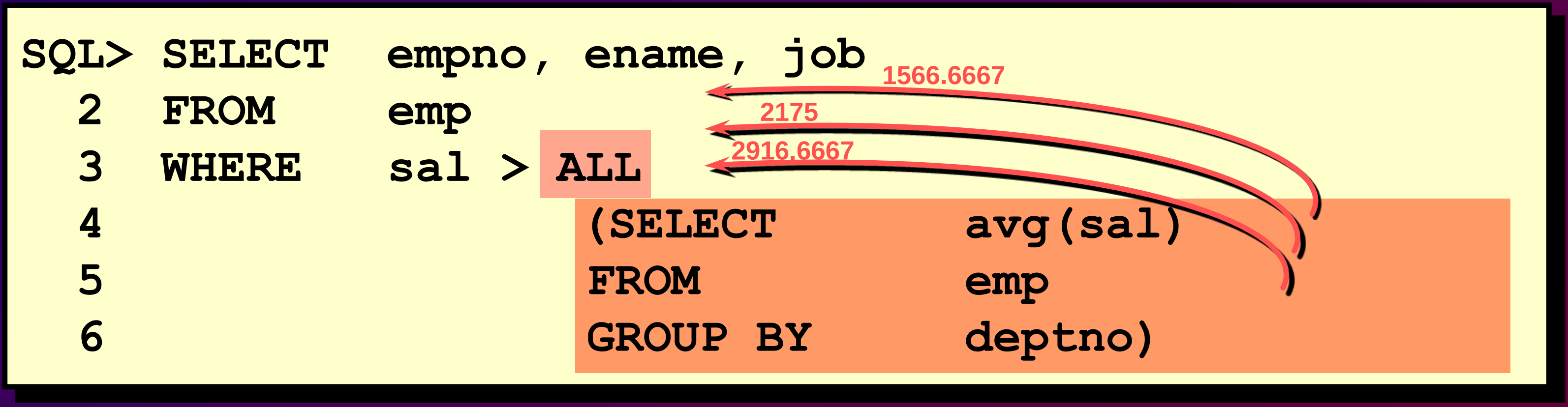
Using ANY Operator in Multiple-Row Subqueries



```
SQL> SELECT   empno, ename, job
  2  FROM     emp
  3  WHERE    sal < ANY
  4                        (SELECT    sal
  5                         FROM      emp
  6                         WHERE     job = 'CLERK')
  7  AND      job <> 'CLERK';
```

```
     EMPNO ENAME      JOB
---------- ---------- ---------
      7654 MARTIN     SALESMAN
      7521 WARD       SALESMAN
```

Using ALL Operator in Multiple-Row Subqueries



```
SQL> SELECT   empno, ename, job
  2  FROM     emp
  3  WHERE    sal > ALL
  4                     (SELECT         avg(sal)
  5                      FROM           emp
  6                      GROUP BY       deptno)
```

```
    EMPNO ENAME      JOB
--------- ---------- ---------
     7839 KING       PRESIDENT
     7566 JONES      MANAGER
     7902 FORD       ANALYST
     7788 SCOTT      ANALYST
```

Summary

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

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

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