5

Subqueries

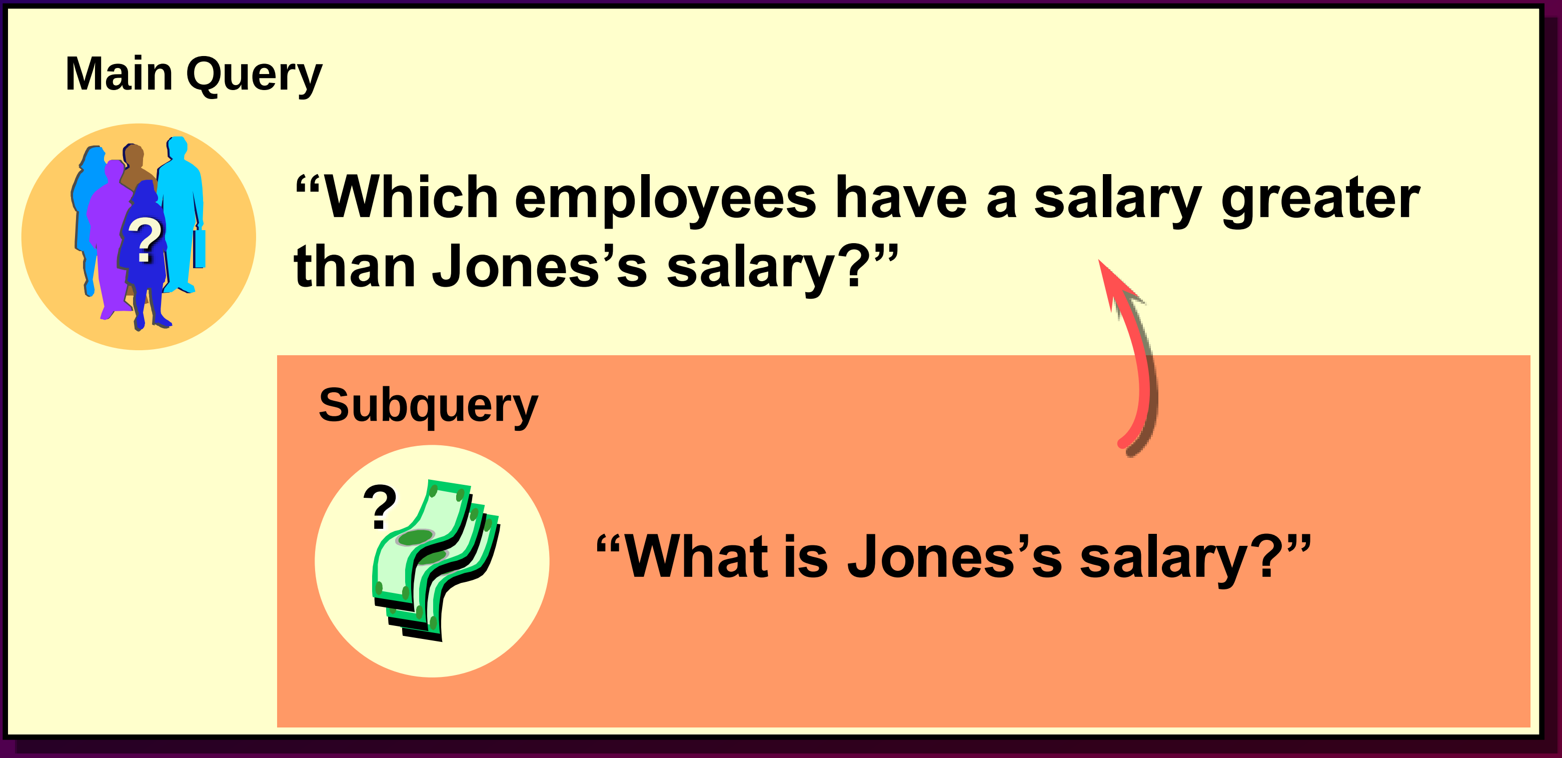
Objectives

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

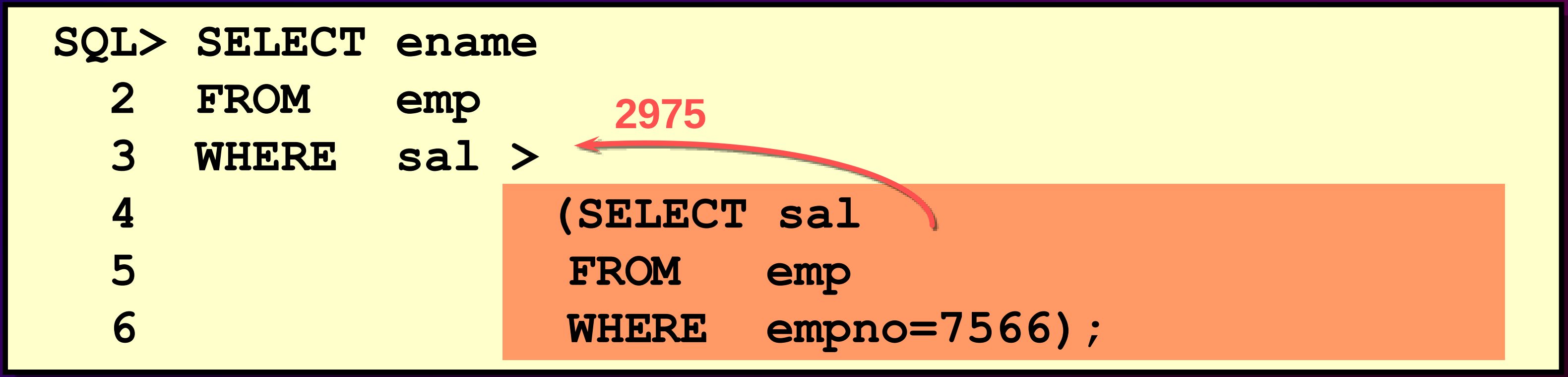
“What is Jones’s salary?”

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



2975

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

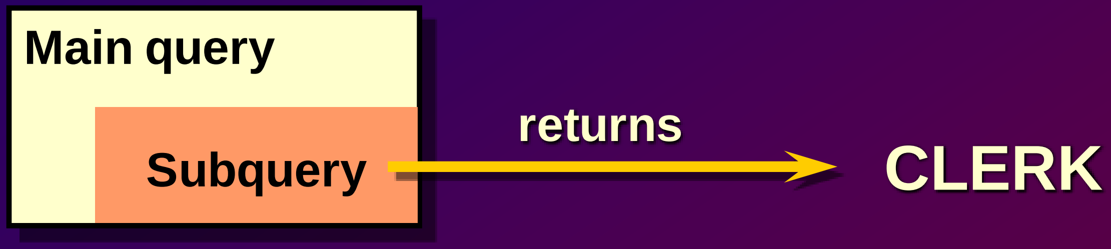
```
ENAME
----------
KING
FORD
SCOTT
```

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



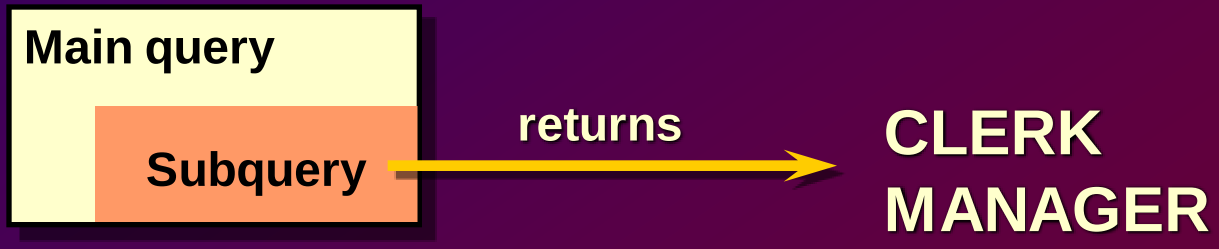
Main query

Subquery

returns

CLERK

- Multiple-row subquery



Main query

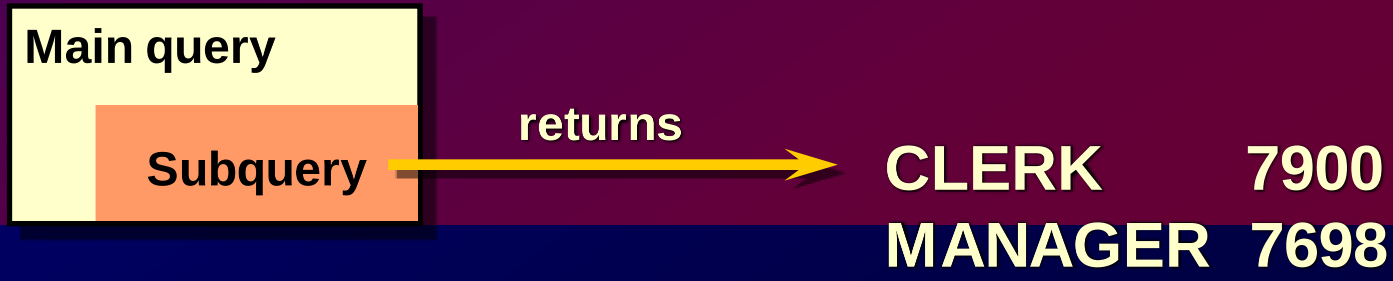
Subquery

returns

CLERK

MANAGER

- Multiple-column subquery



Main query

Subquery

returns

CLERK 7900

MANAGER 7698

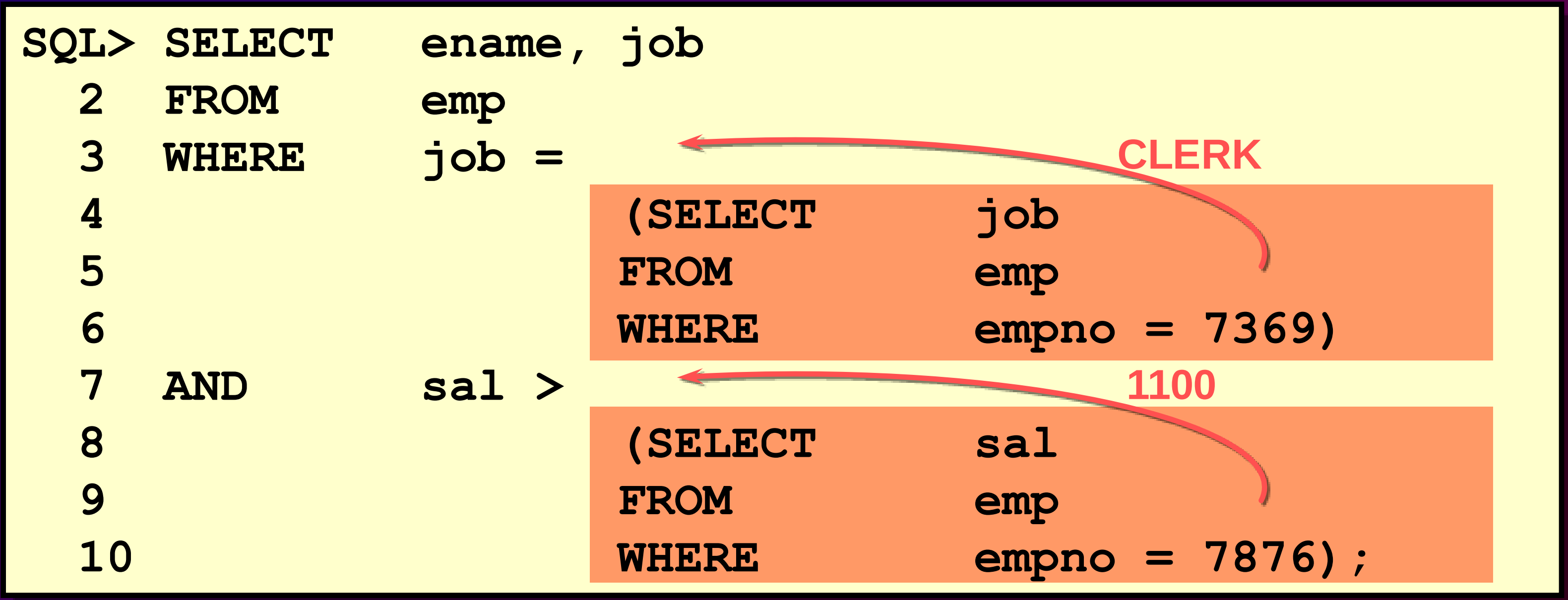
ORACLE®

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

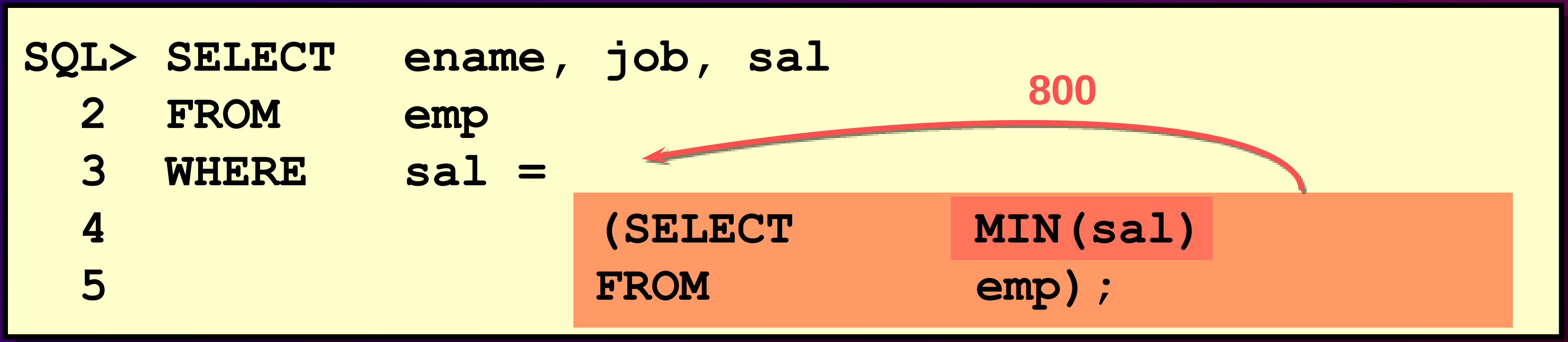
Executing Single-Row Subqueries



```
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 Oracle10 Server executes subqueries first.
- The Oracle10 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);
```

800

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);
```

Single-row operator with multiple-row subquery

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

no rows selected
```

ORACLE®

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
```

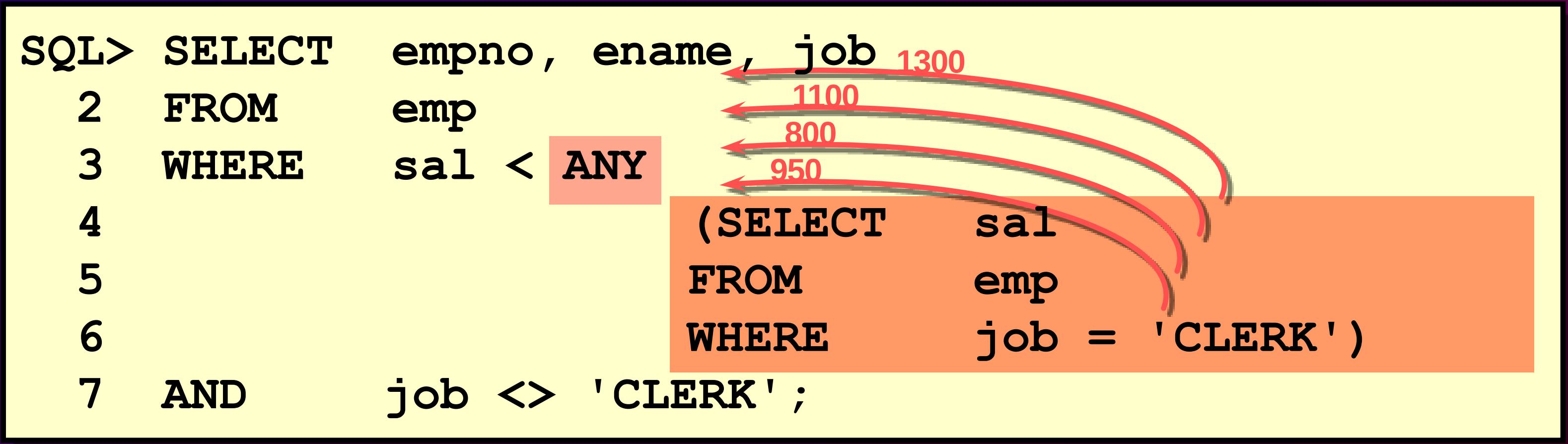
Subquery returns no values

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

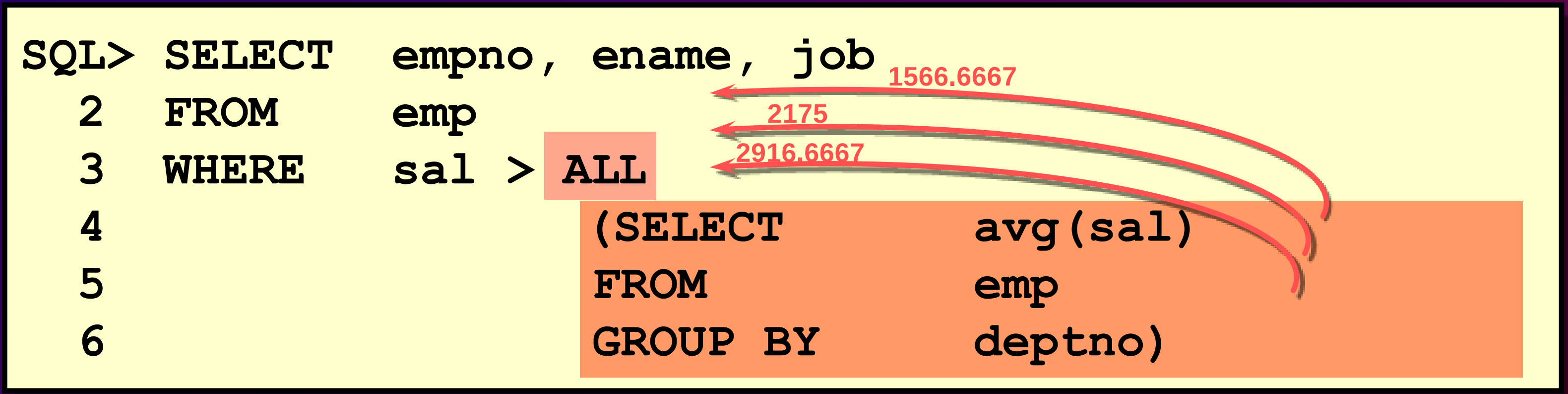
Using ANY Operator in Multiple-Row Subqueries



```
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
```

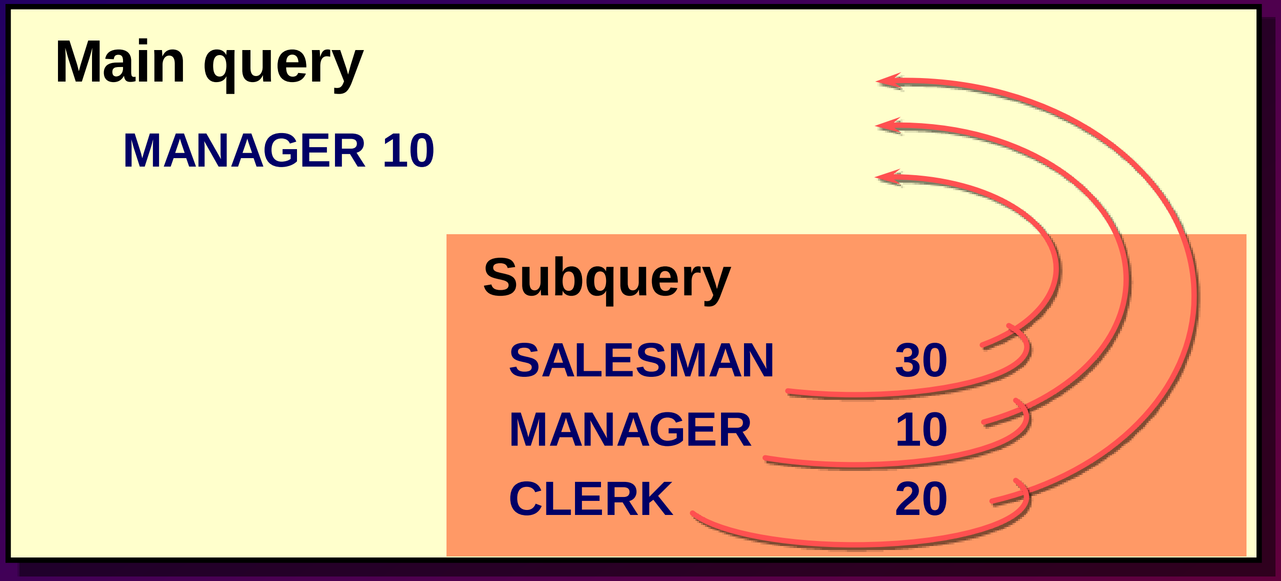
Using ALL Operator in Multiple-Row Subqueries



```
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
```

Multiple-Column Subqueries



Main query

MANAGER 10

Subquery

SALESMAN 30
MANAGER 10
CLERK 20

Main query compares to Values from a multiple-row and multiple-column subquery

MANAGER 10

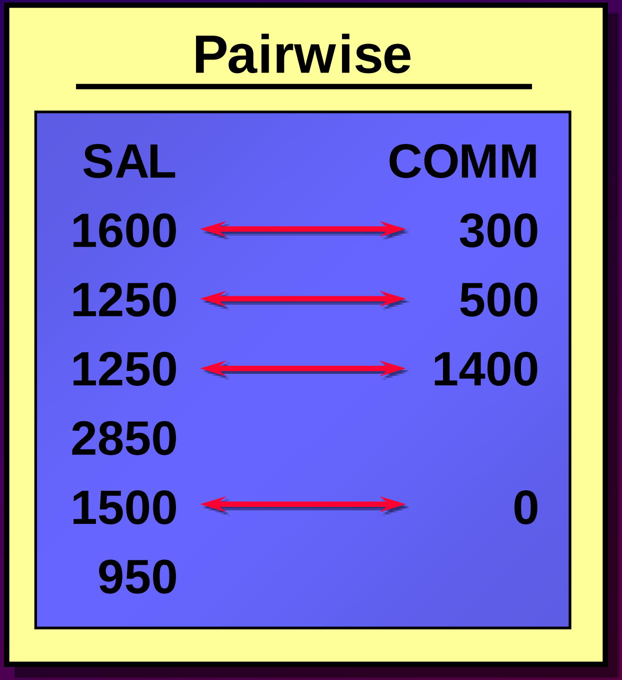
SALESMAN 30
MANAGER 10
CLERK 20

Using Multiple-Column Subqueries

Display the name, department number, salary, and commission of any employee whose salary and commission matches both the commission and salary of any employee in department 30.

```
SQL> SELECT   ename, deptno, sal, comm
  2  FROM     emp
  3  WHERE    (sal, NVL(comm,-1)) IN
  4                           (SELECT sal, NVL(comm,-1)
  5                            FROM   emp
  6                            WHERE  deptno = 30);
```

Column Comparisons



Pairwise

SAL COMM

1600 300

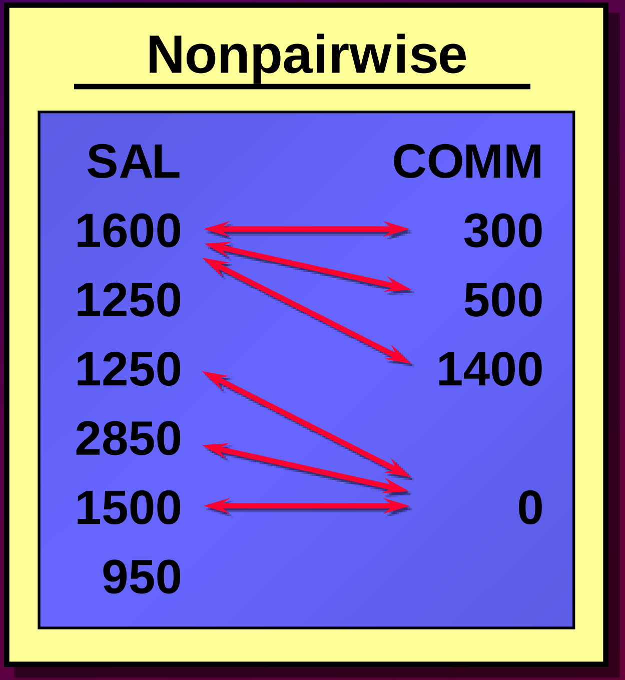
1250 500

1250 1400

2850

1500 0

950



Nonpairwise

SAL COMM

1600 300

1250 500

1250 1400

2850

1500 0

950

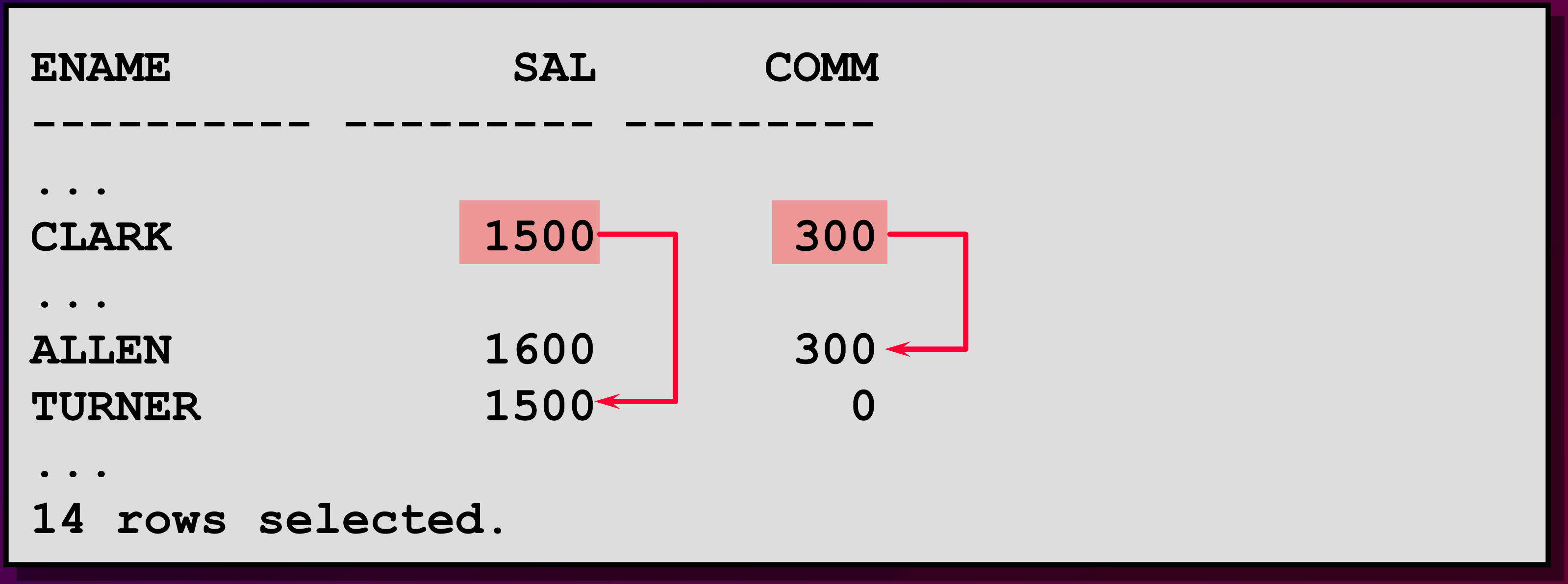
Nonpairwise Comparison Subquery

Display the name, department number, salary, and commission of any employee whose salary and commission matches the commission and salary of any employee in department 30.

```
SQL> SELECT   ename, deptno, sal, comm
  2  FROM     emp
  3  WHERE    sal IN                 (SELECT sal
  4                                   FROM    emp
  5                                   WHERE   deptno = 30)
  6  AND
  7           NVL(comm,-1) IN        (SELECT  NVL(comm,-1)
  8                                   FROM    emp
  9                                   WHERE   deptno = 30);
```

Modifying the EMP Table

- Assume that salary and commission for Clark are modified.
- Salary is changed to $1500 and commission to $300.



```
ENAME            SAL      COMM
---------- --------- ---------
...
CLARK           1500       300
...
ALLEN           1600       300
TURNER          1500         0
...
14 rows selected.
```

Pairwise Subquery

```
SQL> SELECT   ename, deptno, sal, comm
  2  FROM     emp
  3  WHERE    (sal, NVL(comm,-1)) IN
  4                           (SELECT sal, NVL(comm,-1)
  5                            FROM   emp
  6                            WHERE  deptno = 30);
```

```
ENAME         DEPTNO       SAL      COMM
---------- --------- --------- ---------
JAMES             30       950
WARD              30      1250       500
MARTIN            30      1250      1400
TURNER            30      1500         0
ALLEN             30      1600       300
BLAKE             30      2850

6 rows selected.
```

Nonpairwise Subquery

```
SQL> SELECT   ename,deptno, sal, comm
  2  FROM     emp
  3  WHERE    sal IN              (SELECT sal
  4                                FROM    emp
  5                                WHERE   deptno = 30)
  6  AND
  7           NVL(comm,-1) IN     (SELECT NVL(comm,-1)
  8                                FROM    emp
  9                                WHERE   deptno = 30);
```

```
ENAME         DEPTNO        SAL       COMM
---------- ---------- ---------- ----------
JAMES             30        950
BLAKE             30       2850
TURNER            30       1500          0
CLARK             10       1500        300
...
7 rows selected.
```

Null Values in a Subquery

```
SQL> SELECT   employee.ename
  2  FROM     emp employee
  3  WHERE    employee.empno NOT IN
                               (SELECT manager.mgr
                                FROM   emp manager);

no rows selected.
```

Using a Subquery in the FROM Clause

```
SQL> SELECT   a.ename, a.sal, a.deptno, b.salavg
  2  FROM     emp a, (SELECT   deptno, avg(sal) salavg
  3                   FROM     emp
  4                   GROUP BY deptno) b
  5  WHERE    a.deptno = b.deptno
  6  AND      a.sal > b.salavg;
```

```
ENAME             SAL    DEPTNO     SALAVG
---------- ---------- --------- ----------
KING             5000        10  2916.6667
JONES            2975        20       2175
SCOTT            3000        20       2175
...
6 rows selected.
```

Summary

- A multiple-column subquery returns more than one column.
- Column comparisons in a multiple-column comparisons can be pairwise or nonpairwise.
- A multiple-column subquery can also be used in the FROM clause of a SELECT statement.

Practice Overview

Creating subqueries