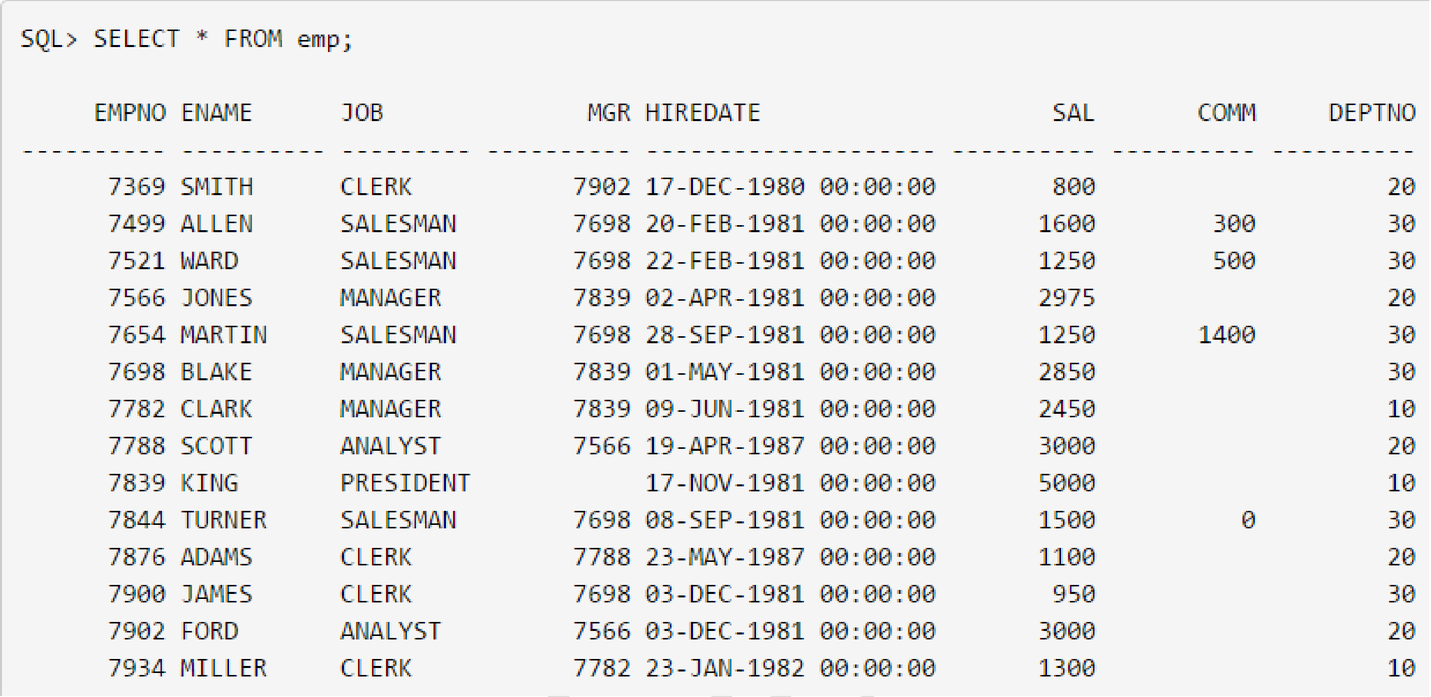
ALL, ANY and SOME Comparison Conditions in SQL

All queries would be explained on basis **emp** table. (*Data as below*)

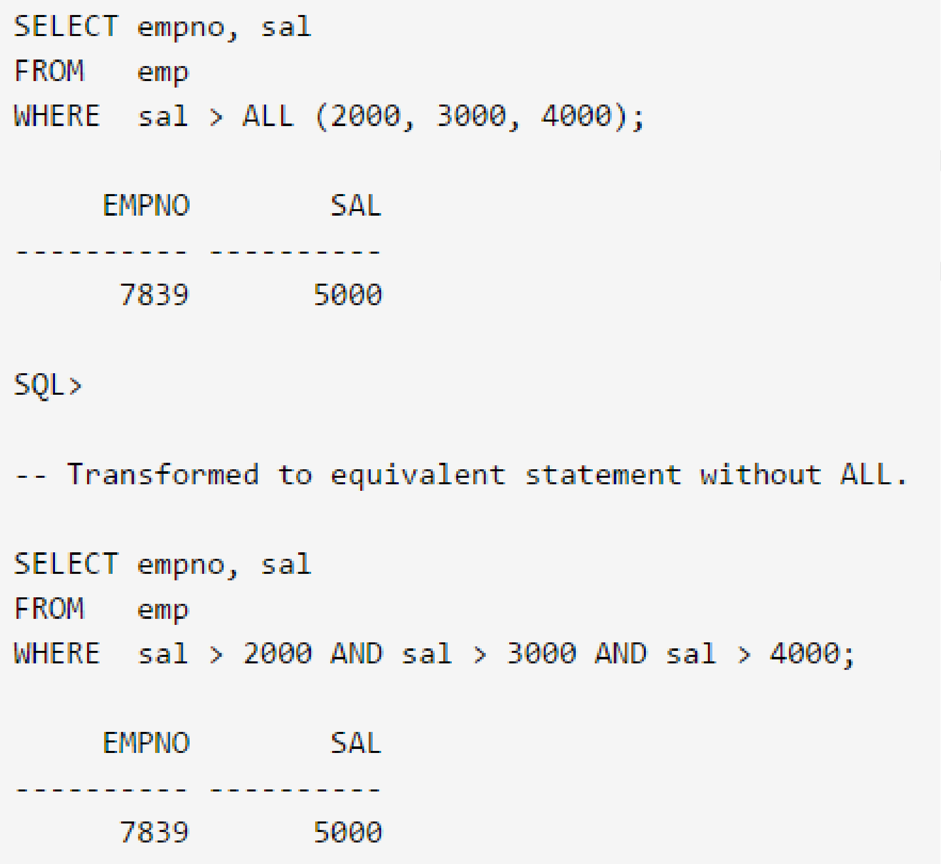


```
SQL> SELECT * FROM emp;

     EMPNO ENAME      JOB              MGR HIREDATE                    SAL       COMM     DEPTNO
---------- ---------- --------- ---------- ------------------- ---------- ---------- ----------
      7369 SMITH      CLERK           7902 17-DEC-1980 00:00:00        800                    20
      7499 ALLEN      SALESMAN        7698 20-FEB-1981 00:00:00       1600        300         30
      7521 WARD       SALESMAN        7698 22-FEB-1981 00:00:00       1250        500         30
      7566 JONES      MANAGER         7839 02-APR-1981 00:00:00       2975                    20
      7654 MARTIN     SALESMAN        7698 28-SEP-1981 00:00:00       1250       1400         30
      7698 BLAKE      MANAGER         7839 01-MAY-1981 00:00:00       2850                    30
      7782 CLARK      MANAGER         7839 09-JUN-1981 00:00:00       2450                    10
      7788 SCOTT      ANALYST         7566 19-APR-1987 00:00:00       3000                    20
      7839 KING       PRESIDENT            17-NOV-1981 00:00:00       5000                    10
      7844 TURNER     SALESMAN        7698 08-SEP-1981 00:00:00       1500          0         30
      7876 ADAMS      CLERK           7788 23-MAY-1987 00:00:00       1100                    20
      7900 JAMES      CLERK           7698 03-DEC-1981 00:00:00        950                    30
      7902 FORD       ANALYST         7566 03-DEC-1981 00:00:00       3000                    20
      7934 MILLER     CLERK           7782 23-JAN-1982 00:00:00       1300                    10
```

ALL

The ALL comparison condition is used to compare a value to a list or sub-query. It must be preceded by =, !=, >, <, <=, >= and followed by a list or sub-query.

When the ALL condition is followed by a list, the optimizer expands the initial condition to all elements of the list and strings them together with AND operators, as shown below.

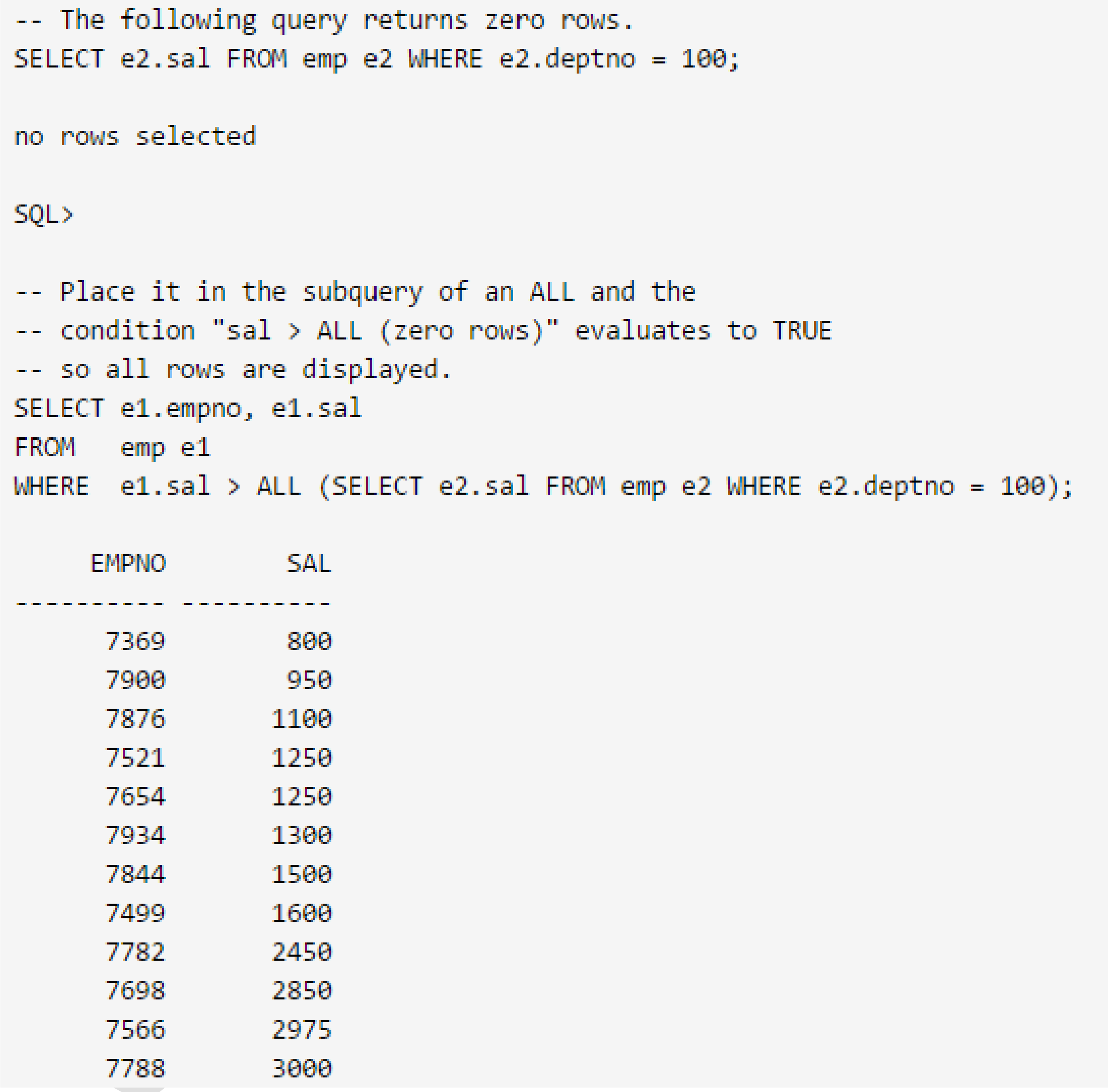


```
SELECT empno, sal
FROM   emp
WHERE  sal > ALL (2000, 3000, 4000);

     EMPNO        SAL
---------- ----------
      7839       5000

SQL>

-- Transformed to equivalent statement without ALL.

SELECT empno, sal
FROM   emp
WHERE  sal > 2000 AND sal > 3000 AND sal > 4000;

     EMPNO        SAL
---------- ----------
      7839       5000
```

Assuming sub-queries don't return zero rows, the following statements can be made for both list and sub-query versions:

- "x = ALL (...)": The value must match all the values in the list to evaluate to TRUE.
- "x != ALL (...)": The value must not match any values in the list to evaluate to TRUE.
- "x > ALL (...)": The value must be greater than the biggest value in the list to evaluate to TRUE.
- "x < ALL (...)": The value must be smaller than the smallest value in the list to evaluate to TRUE.
- "x >= ALL (...)": The value must be greater than or equal to the biggest value in the list to evaluate to TRUE.
- "x <= ALL (...)": The value must be smaller than or equal to the smallest value in the list to evaluate to TRUE.

If a subquery returns zero rows, the condition evaluates to TRUE. In the following example, the subquery returns zero rows, which means the whole expression "sal > ALL (zero rows)" evaluates to TRUE, so all rows are displayed.

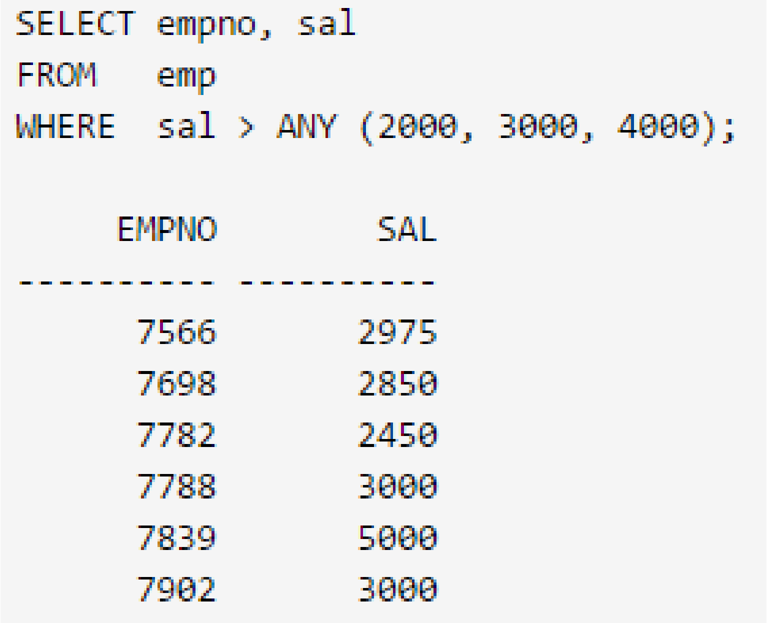


```
-- The following query returns zero rows.
SELECT e2.sal FROM emp e2 WHERE e2.deptno = 100;

no rows selected

SQL>

-- Place it in the subquery of an ALL and the
-- condition "sal > ALL (zero rows)" evaluates to TRUE
-- so all rows are displayed.
SELECT e1.empno, e1.sal
FROM   emp e1
WHERE  e1.sal > ALL (SELECT e2.sal FROM emp e2 WHERE e2.deptno = 100);

     EMPNO        SAL
---------- ----------
      7369        800
      7900        950
      7876       1100
      7521       1250
      7654       1250
      7934       1300
      7844       1500
      7499       1600
      7782       2450
      7698       2850
      7566       2975
      7788       3000
```

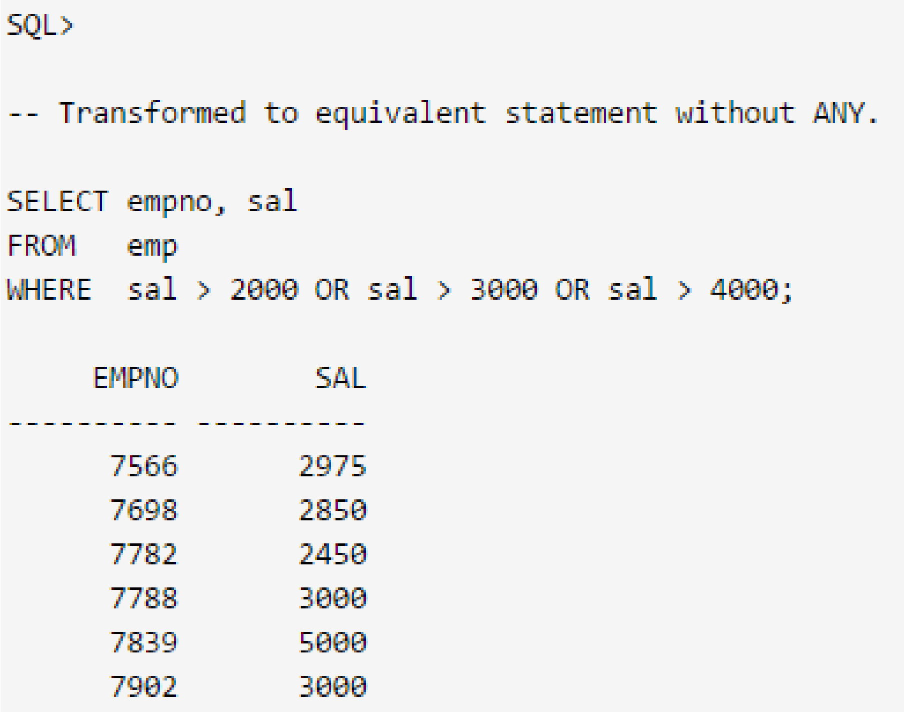
ANY

The ANY comparison condition is used to compare a value to a list or subquery. It must be preceded by =, !=, >, <, <=, >= and followed by a list or subquery.

When the ANY condition is followed by a list, the optimizer expands the initial condition to all elements of the list and strings them together with OR operators, as shown below.



```
SELECT empno, sal
FROM   emp
WHERE  sal > ANY (2000, 3000, 4000);

     EMPNO        SAL
---------- ----------
      7566       2975
      7698       2850
      7782       2450
      7788       3000
      7839       5000
      7902       3000
```

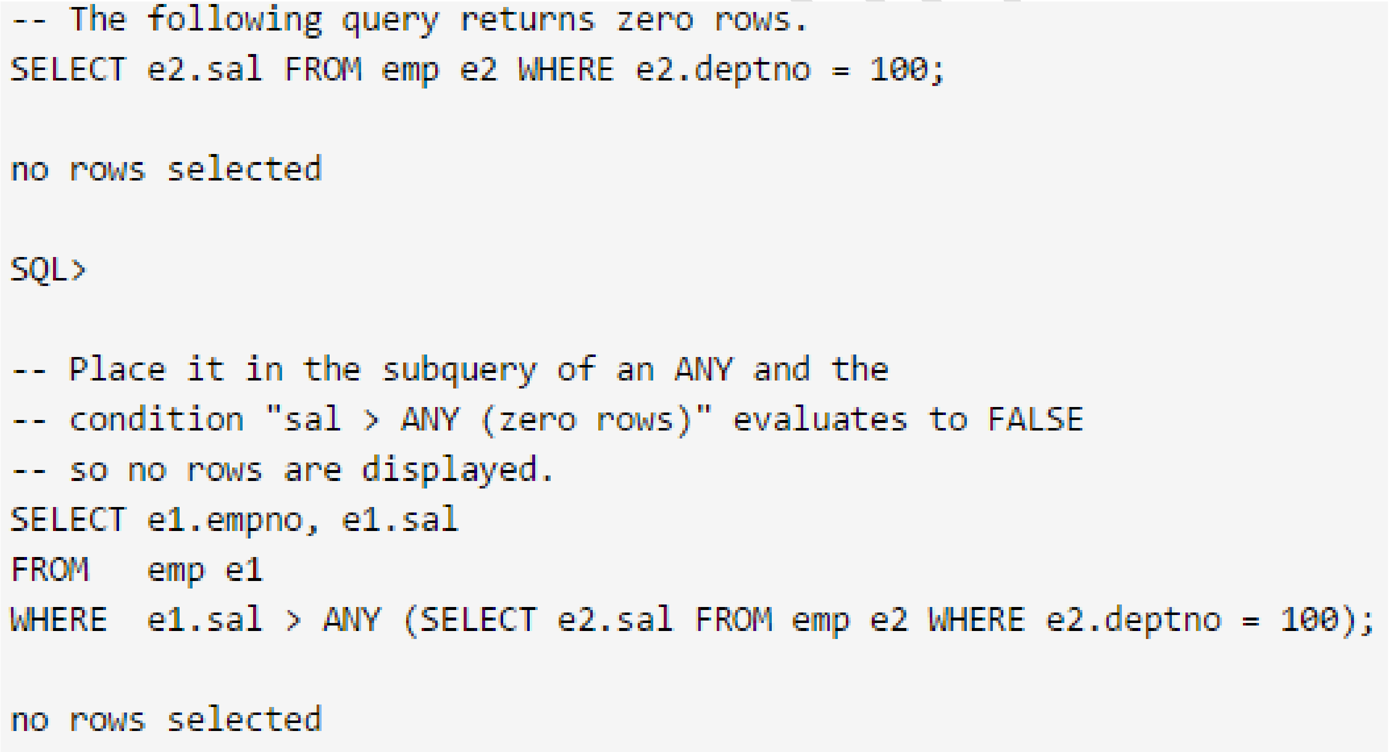


```
SQL>

-- Transformed to equivalent statement without ANY.

SELECT empno, sal
FROM   emp
WHERE  sal > 2000 OR sal > 3000 OR sal > 4000;

     EMPNO        SAL
---------- ----------
      7566       2975
      7698       2850
      7782       2450
      7788       3000
      7839       5000
      7902       3000
```

Assuming sub-queries don't return zero rows, the following statements can be made for both list and sub-query versions:

- "x = ANY (...)": The value must match one or more values in the list to evaluate to TRUE.
- "x != ANY (...)": The value must not match one or more values in the list to evaluate to TRUE.
- "x > ANY (...)": The value must be greater than the smallest value in the list to evaluate to TRUE.
- "x < ANY (...)": The value must be smaller than the biggest value in the list to evaluate to TRUE.
- "x >= ANY (...)": The value must be greater than or equal to the smallest value in the list to evaluate to TRUE.
- "x <= ANY (...)": The value must be smaller than or equal to the biggest value in the list to evaluate to TRUE.

If a sub-query returns zero rows, the condition evaluates to FALSE. In the following example, the sub-query returns zero rows, which means the whole expression "sal > ANY (zero rows)" evaluates to FALSE, so no rows are displayed.



```
-- The following query returns zero rows.
SELECT e2.sal FROM emp e2 WHERE e2.deptno = 100;

no rows selected

SQL>

-- Place it in the subquery of an ANY and the
-- condition "sal > ANY (zero rows)" evaluates to FALSE
-- so no rows are displayed.
SELECT e1.empno, e1.sal
FROM   emp e1
WHERE  e1.sal > ANY (SELECT e2.sal FROM emp e2 WHERE e2.deptno = 100);

no rows selected
```

More Examples:

```
SELECT e1.empno, e1.sal
FROM   emp e1
WHERE  e1.sal > ANY (SELECT e2.sal
                     FROM   emp e2
                     WHERE  e2.deptno = 10);
```

```
SELECT e1.empno, e1.sal
FROM   emp e1
WHERE  e1.sal > ALL (SELECT e2.sal
                     FROM   emp e2
                     WHERE  e2.deptno = 20);
```