SELECT \*

FROM employees

WHERE salary > (SELECT salary

FROM employees

WHERE employee\_id = 145);

Some Points to keep in mind regarding Subqueries:

1. A subquery can contain another subquery. Oracle allows us multiple levels for nesting subqueries.
2. If the subquery comes after FROM clause of SELECT statement then it is called an Inline View.
3. If the subquery is nested in the WHERE clause of SELECT statement then it is called as Nested Subquery.

**1. Single Row Subquery Example**

When the nested subquery or the inner query inside parenthesis returns just one row then it is an example of Single Row Subquery.

Suppose, we want to find the employees who are earning more than the highest earning employee in department=50, then we use this query:

SELECT first\_name,last\_name, salary

FROM employees

WHERE salary >

( SELECT MAX(salary)

FROM employees

WHERE department\_id=50);

**2. Multiple Row Subquery Example**

If the nested subquery returns more than one rows then it is an example of Multiple Row Subquery.

Let’s look at an example where we try to find employees whose salary is same as salary of any employee in department = 30. The IN operator determines whether a value matches any values in a list or a subquery.

SELECT first\_name, last\_name, salary

FROM employees

WHERE salary IN (SELECT salary

FROM employees

WHERE department\_id = 30);

When the inner nested subquery returns more than one row, then usual comparison operators ( =, <, > …) must be used with ANY or ALL. Here’s what these operators denote:

1. ANY – If used with comparison operator, the final outcome will be true if operator evaluates to TRUE for ANY values that a subquery returns.
2. ALL – If used with comparison operator, the outcome will be true only if operator evaluates to TRUE for ALL values that a subquery returns.

Let’s understand this through one query:

Suppose we want to find list of employees whose salary is ‘greater than any’ value of salary in department-30, then we use this query:

​

SELECT first\_name, last\_name, salary

FROM employees

WHERE salary > ANY (SELECT salary

FROM employees

WHERE department\_id = 30);

Similarly, we can use comparison operators with ANY and ALL operators to get different outcomes:

Employees whose salary is:

--Greater than minimum ( salary of employee in dept-30)

​

SELECT first\_name, last\_name, salary

FROM employees

WHERE salary > ANY (SELECT salary

FROM employees

WHERE department\_id = 30);

​

--Less than maximum ( salary of employee in dept-30)

​

SELECT first\_name, last\_name, salary

FROM employees

WHERE salary < ANY (SELECT salary

FROM employees

WHERE department\_id = 30);

​

--Greater than maximum ( salary of employee in dept-30)

​

SELECT first\_name, last\_name, salary

FROM employees

WHERE salary > ALL (SELECT salary

FROM employees

WHERE department\_id = 30);

​

--Less than minimum ( salary of employee in dept-30)

​

SELECT first\_name, last\_name, salary

FROM employees

WHERE salary < ALL (SELECT salary

FROM employees

WHERE department\_id = 30);