

# **Introduction to Oracle : SQL**

---

## Chapter5. 서브쿼리

- ▶ 서브쿼리를 사용해서 해결할 수 있는 문제 유형을 기술한다.
  - ▶ 치환변수를 사용하여 각종 쿼리문을 작성한다.
-

## 1. 서브쿼리의 정의

- 단독 쿼리문만으로는 해결할 수 없는 검색 수행 시 사용

Kochhar보다 많은 봉급을 받는 직원의 이름은?

**Main Query**

Kochhar보다 많은 봉급을 받는 직원의 이름은?

**Subquery**

Kochhar의 봉급은?

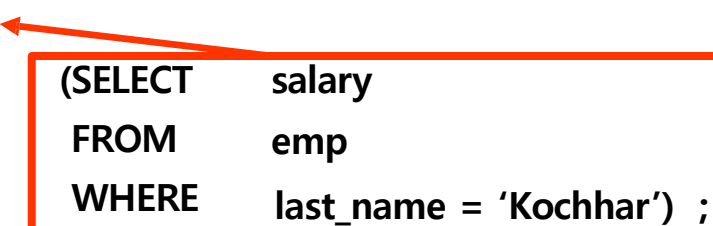
## 2. 서브쿼리의 구문

```
SELECT  select_list
FROM    table명
WHERE   표현식 연산자
              (SELECT  select_list
               FROM    table명) ;
```

- 서브쿼리가 메인쿼리보다 먼저 수행

### 3. 서브쿼리의 사용

```
SELECT last_name
FROM emp
WHERE salary > (SELECT salary
                 FROM emp
                 WHERE last_name = 'Kochhar') ;
```

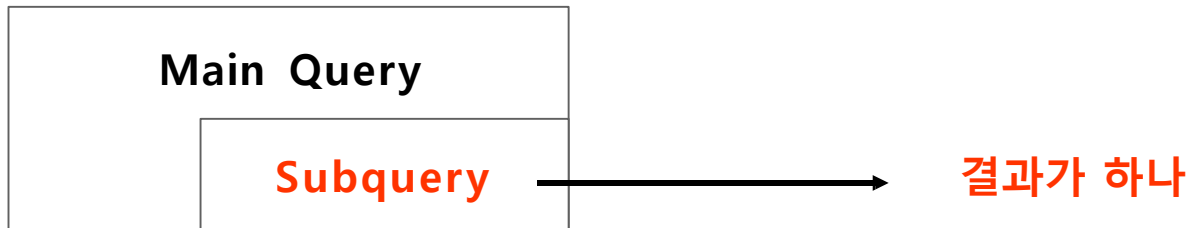
A diagram illustrating the use of a subquery in a SQL statement. The main query is a SELECT statement that retrieves the last\_name from the emp table where the salary is greater than a subquery result. The subquery is enclosed in a red rectangular box and returns the salary of the employee whose last\_name is 'Kochhar'. A red arrow points from the subquery box to the 'salary >' condition in the main query's WHERE clause, indicating that the subquery's result is used to compare the salary of other employees.

## 4. 서브쿼리 사용 시 고려사항

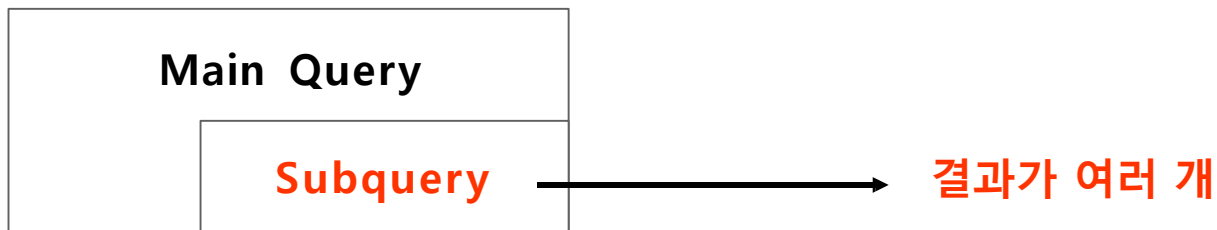
- 서브쿼리는 반드시 괄호에 감싸서 수행
  - 서브쿼리는 비교 연산자의 우측에 위치
  - TOP-N 분석 기능을 제외하고는 서브쿼리에 ORDER BY 절을 수행할 필요 없음
  - 결과값이 하나인 서브쿼리에 대해서는 단일행 연산자를, 결과값이 여러개인 다중행 서브쿼리에 대해서는 다중행연산자 를 사용
-

## 5. 서브쿼리의 유형

- 단일행 서브쿼리



- 다중행 서브쿼리



## 5. 서브쿼리의 유형

### (1) 단일행 서브쿼리

- 서브쿼리 결과값이 하나
- 단일행 비교연산자(=, >, >=, <, <=, <>)와 함께 작성

#### 1) 단일행 서브쿼리 사용

부서가 101번 사원과 같고, 월급이 141번 사원보다 많은 직원을 검색

```
SELECT last_name, department_id, salary
FROM employees
WHERE department_id = (SELECT department_id
                       FROM employees
                       WHERE employee_id = 101)
AND salary > (SELECT salary
              FROM employees
              WHERE employee_id = 141) ;
```




## 5. 서브쿼리의 유형

### (1) 단일행 서브쿼리

#### 2) 서브쿼리 내에서의 그룹함수 사용

가장 많은 월급을 받는 직원의 이름과 월급 검색

```
SELECT last_name, salary
FROM emp
WHERE salary = 
              (SELECT max(salary )
               FROM emp) ;
```

---



## 5. 서브쿼리의 유형

### (2) 다중행 서브쿼리

- 서브쿼리 결과값이 둘 이상
- 다중행 비교연산자(IN, ANY, ALL)와 함께 작성

연산자	의미
IN	같다
< ANY	최대값보다 작다
> ANY	최소값보다 크다
> ALL	최대값보다 크다
< ALL	최소값보다 작다

## 5. 서브쿼리의 유형

### (2) 다중행 서브쿼리

#### 1) IN 연산자 사용

```
SELECT last_name, salary
FROM emp
WHERE salary IN (SELECT MAX (salary)
                  FROM emp
                  GROUP BY department_id) ;
```

결과값이 여러 개

## 5. 서브쿼리의 유형

### (2) 다중행 서브쿼리

#### 2) ANY 연산자 사용

```
SELECT last_name, job_id, salary
FROM emp
WHERE salary < ANY (9000, 6000, 4200)
AND job_id <> 'IT_PROG' ;
```

(SELECT salary  
FROM emp  
WHERE job\_id = 'IT\_PROG' )

## 5. 서브쿼리의 유형

### (2) 다중행 서브쿼리 3

#### ) ALL 연산자 사용

```
SELECT last_name, job_id, salary
FROM emp
WHERE salary < ALL 9000, 6000, 4200
AND job_id <> 'IT_PROG' ;
```

(SELECT salary  
FROM emp  
WHERE job\_id = 'IT\_PROG' )

## 6. 서브쿼리의 NULL값 반환

- 서브쿼리 결과값이 NULL일 경우 메인쿼리의 결과도 NULL을 반환

```
SELECT last_name, department_id
FROM emp
WHERE department_id =
    (SELECT department_id
     FROM emp
     WHERE last_name = 'Hugo' ) ;
```

*NULL*