

# Introduction to Oracle9i : SQL

## Chapter5. 서브쿼리와 치환변수

- ▶ 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

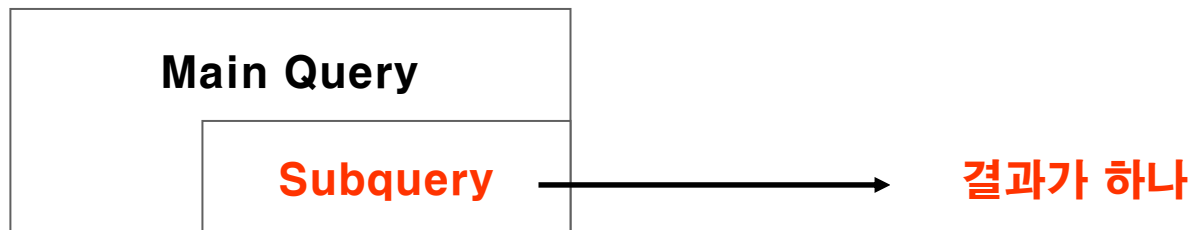
```
SELECT last_name  
FROM employees  
WHERE salary >   
    (SELECT salary  
     FROM employees  
     WHERE last_name = 'Kochhar') ;
```

## 4. 서브쿼리 사용 시 고려사항

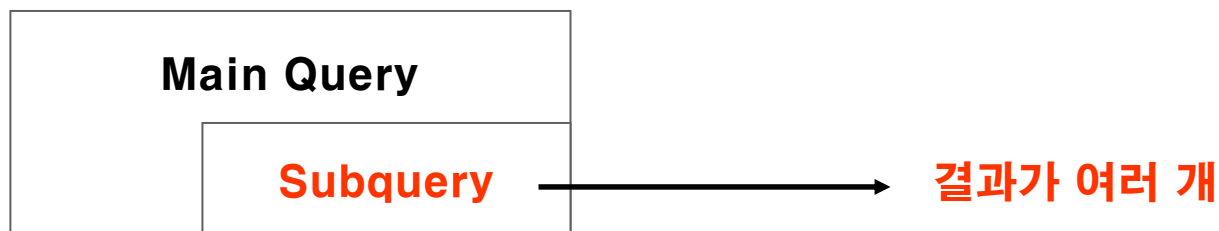
- 서브쿼리는 반드시 괄호에 감싸서 수행
- 서브쿼리는 비교 연산자의 우측에 위치
- TOP-N 분석 기능을 제외하고는 서브쿼리에 ORDER BY 절을 수행할 필요 없음
- 결과값이 하나인 서브쿼리에 대해서는 단일행 연산자를, 결과값이 여러개인 다중행 서브쿼리에 대해서는 다중행연산자를 사용

## 5. 서브쿼리의 유형

- 단일행 서브쿼리



- 다중행 서브쿼리



## 5. 서브쿼리의 유형

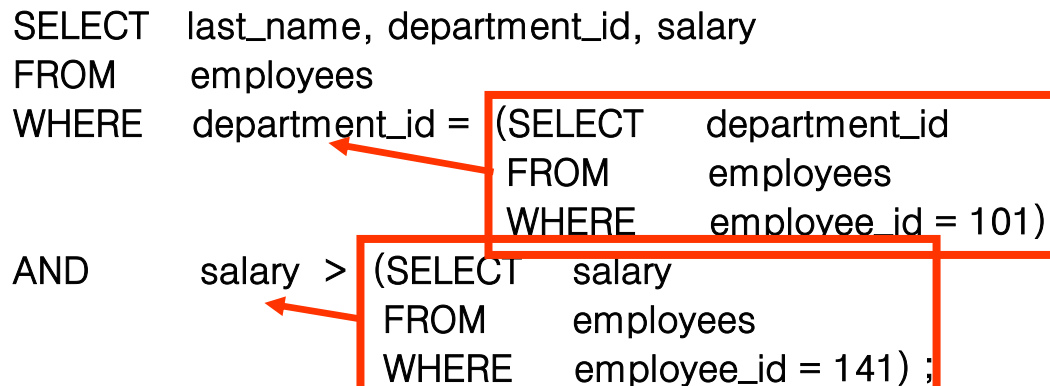
### (1) 단일행 서브쿼리

- 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 employees
WHERE salary = (SELECT MAX (salary)
                FROM employees) ;
```



## 5. 서브쿼리의 유형

### (1) 단일행 서브쿼리

### 3) HAVING 절에서의 서브쿼리 사용

부서별 최저 월급을 출력하되 60번 부서의 최저 월급보다는 큰 값만 검색

```
SELECT    department_id, MIN (salary)
FROM      employees
GROUP BY  department_id
HAVING    MIN (salary) > (SELECT    MIN (salary)
                           FROM      employees
                           WHERE     department_id = 60 ) ;
```

## 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 employees
WHERE salary IN (SELECT MAX (salary)
                  FROM employees
                  GROUP BY department_id) ;
```

결과값이 여러 개

## 5. 서브쿼리의 유형

### (2) 다중행 서브쿼리

#### 2) ANY 연산자 사용

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

## 5. 서브쿼리의 유형

### (2) 다중행 서브쿼리

#### 3) ALL 연산자 사용

```
SELECT last_name, job_id, salary
FROM employees
WHERE salary < ALL          9000, 6000, 4200
      (SELECT salary
        FROM employees
        WHERE job_id = 'IT_PROG' )
AND job_id <> 'IT_PROG' ;
```

## 6. 서브쿼리의 NULL값 반환

- 서브쿼리 결과값이 NULL일 경우 메인쿼리의 결과도 NULL을 반환

```
SELECT last_name, department_id  
FROM employees  
WHERE department_id =
```

```
(SELECT department_id  
FROM employees  
WHERE employee_name = 'Hugo' );
```

**NULL**

## 7. 치환변수

- 값의 임시적 저장을 위해 세 가지 치환 변수형태 제공
  - & : 값의 1회 사용을 위한 치환변수 사용 시
  - && : 후속 사용을 위한 치환변수 사용 시
  - DEFINE : 치환변수의 사전 선언 시 사용
- SELECT문의 모든 절에 사용 가능



## 7. 치환변수

### (1) &

```
SELECT last_name, salary
FROM employees
WHERE salary = &sal ;
```

```
SELECT last_name, salary
FROM employees
WHERE last_name = '&name' ;
```

```
SELECT last_name, salary, &col
FROM employees
WHERE &condition
ORDER BY &order ;
```

## 7. 치환변수

### (2) &&

```
SELECT last_name, salary, &&col  
FROM employees  
ORDER BY &col ;
```

## 7. 치환변수

### (3) DEFINE

```
DEFINE v_empid = 300
```

```
SELECT last_name, salary  
FROM employees  
WHERE employee_id = &v_empid ;
```