

서브쿼리

서브 쿼리의 기본 개념

- ④ LAST_NAME이 'Lee'인 사원의 부서명 조회
 - 필요한 SQL 문

```
SELECT department_id
FROM employees
WHERE last_name = 'Lee';
→ 80
```

```
SELECT department_name
FROM departments
WHERE department_id=80;
```

서브 쿼리의 기본 개념

- ④ LAST_NAME이 'Lee'인 사원의 부서명 조회
 - 서브 쿼리를 이용한 표현

메인 쿼리

```
SELECT department_name  
FROM departments  
WHERE department_id = (
```

서브 쿼리

```
SELECT department_id  
FROM employees  
WHERE last_name = 'Lee');
```

서브 쿼리의 기본 개념

❖ 서브 쿼리

- 하나의 SELECT 문장의 절 안에 포함된 또 하나의 SELECT 문장
- 비교 연산자의 오른쪽에 반드시 괄호를 사용하여 기술
- 메인 쿼리가 실행되기 이전에 한 번만 실행

- 단일 행 서브 쿼리
- 다중 행 서브 쿼리

단일행 서브 쿼리

❖ 단일행 서브 쿼리 예

```
SELECT department_name
FROM departments
WHERE department_id = ( SELECT department_id
                        FROM employees
                        WHERE last_name = 'Lee');
```

- 서브 쿼리의 결과가 하나의 로만 반환
- 메인 쿼리의 WHERE절에서는 단일행 비교 연산자인 =, >, >=, <, <=, <>를 사용

단일행 서브 쿼리

❖ 단일행 서브 쿼리 예

- 직원들의 평균 급여보다 더 많은 급여를 받는 직원 조회

```
SELECT first_name, last_name, salary
FROM employees
WHERE salary > (SELECT AVG(salary)
                FROM employees);
```

단일행 서브 쿼리

❖ 실습

- Jones와 같은 부서에서 근무하는 사원 이름과 부서의 번호를 출력하시오.
- Jones와 동일한 직급을 가진 사원의 이름과 부서 번호를 출력하시오.
- Jones의 급여와 동일하거나 더 적게 받는 사원명과 급여를 출력하시오.

단일행 서브 쿼리

```
SELECT last_name
FROM employees
WHERE department_id = (
    SELECT department_id
    FROM employees
    WHERE last_name='Jones')
```

```
SELECT last_name, job_id
from employees
WHERE job_id = (
    SELECT job_id
    FROM employees
    WHERE last_name='Jones')
```

```
SELECT last_name, salary
FROM employees
WHERE salary <= (
    SELECT salary
    FROM employees
    WHERE last_name='Jones')
```


단일행 서브 쿼리

❖ 실습

- Sales 부서에서 근무하는 모든 사원의 이름과 급여를 출력하시오.

```
SELECT last_name, salary
FROM employees
WHERE department_id = (
    SELECT department_id
    FROM department
    WHERE department_name = 'Sales' )
```

단일행 서브 쿼리

❖ 실습

- 자신의 직속 상관의 이름이 'Steven King'인 사원의 이름과 급여를 조회하시오.

```
SELECT last_name, salary
FROM employees
WHERE manager_id = (
    SELECT employee_id
    FROM employees
    WHERE first_name='Steven' and last_name = 'King' )
```

단일행 서브 쿼리

❖ Seattle에서 근무하는 사원의 이름, 부서 번호를 출력해 보시오

```
select department_id, last_name
from employees
where department_id in (
    select department_id
    from departments
    where location_id = (
        select location_id
        from locations
        where city='Seattle')
);
```

다중행 서브 쿼리

❖ 다중행 서브 쿼리

- 서브 쿼리에서 반환되는 결과가 하나 이상의 행일 때 사용하는 서브 쿼리
- 반드시 다중 연산자와 함께 사용되어야 한다

종류	설명
IN	메인 쿼리의 비교 조건이 서브 쿼리의 결과 중에서 하나라도 일치하면 참이다
ANY, SOME	메인 쿼리의 비교 조건이 서브 쿼리의 검색 결과와 하나 이상 일치하면 참이다
ALL	메인 쿼리의 비교 조건이 서브 쿼리의 검색 결과와 모든 값이 일치하면 참이다
EXIST	메인 쿼리의 비교 조건이 서브 쿼리의 검색 결과 중에서 만족하는 값이 하나라도 존재하면 참이다

다중행 서브 쿼리

❖ IN 연산자

- 연봉을 10000이상 받는 사원이 소속된 부서와 동일한 부서에 근무하는 사원들의 정보 조회

```
SELECT first_name, last_name, department_id
FROM employees
WHERE department_id IN (
    SELECT DISTINCT department_id
    FROM employees
    WHERE salary >= 10000);
```

다중행 서브 쿼리

❖ ALL 연산자

- 부서번호 30번 소속 직원들 중에서 급여를 가장 많이 받는 직원보다 더 많은 급여를 받는 사람의 이름과 급여를 조회
→ 단일행 서브 쿼리

```
SELECT first_name, last_name, salary
FROM employees
WHERE salary > (
    SELECT MAX(salary)
    FROM employees
    GROUP BY department_id
    HAVING department_id = 30);
```

다중행 서브 쿼리

❖ ALL 연산자

- 결과값이 여러 개로 나오는 쿼리문을 서브 쿼리로 사용할 경우 단순히 > 연산자만 사용하면 오류 발생
→ 다중행 서브 쿼리

```
SELECT first_name, last_name, salary
FROM employees
WHERE salary > ALL(
    SELECT salary
    FROM employees
    WHERE department_id = 30);
```

다중행 서브 쿼리

❖ ANY 연산자

- 부서 번호가 30번인 직원들의 급여 중 가장 낮은 값보다 높은 급여를 받는 직원의 이름, 급여를 출력

```
SELECT first_name, last_name, salary
FROM employees
WHERE salary > (
    SELECT MIN(salary)
    FROM employees
    GROUP BY department_id
    HAVING department_id=30);
```

```
SELECT first_name, last_name, salary
FROM employees
WHERE salary > ANY (
    SELECT salary
    FROM employees
    WHERE department_id=30);
```