suanlab

데이터베이스
Database

ORACLE®

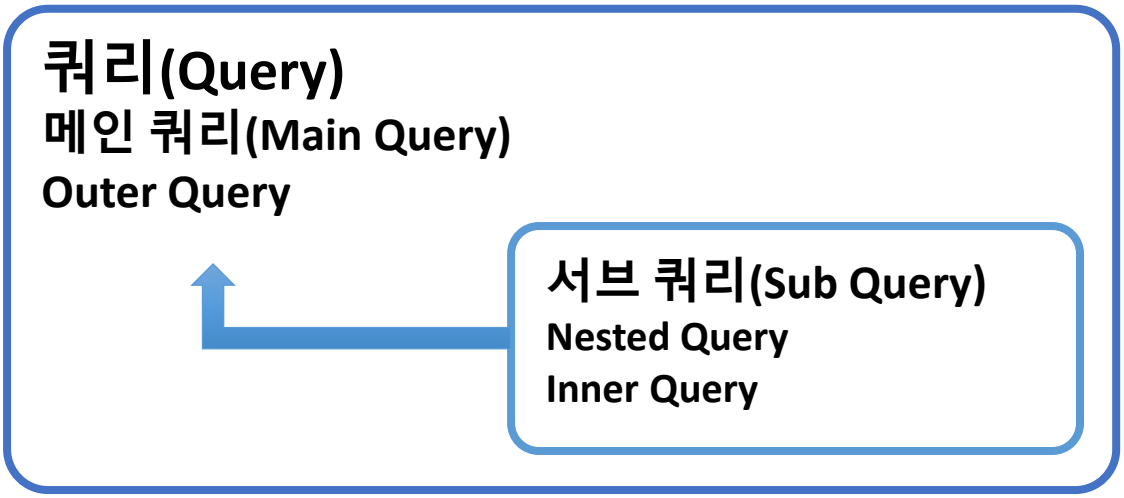
DATABASE

10
서브 쿼리

단일 행 서브 쿼리
다중 행 서브 쿼리
인라인 뷰

서브 쿼리(Sub Query)

- SELECT 문 안에 다시 SELECT 문이 기술된 쿼리
- 상위 SELECT 문 안에 하위 SELECT 문이 포함된 형태라 중첩된(nested) 쿼리라고도 부름
- 단일 SELECT 문 사용만으로는 복잡한 조건식을 만들 때 사용
- 다른 테이블에서 데이터 값을 조회한 후 조건으로 사용할 때 사용



- 서브 쿼리 종류
  - 단일 행 서브 쿼리: 하나의 행을 검색하는 서브 질의
  - 다중 행 서브 쿼리: 하나 이상의 행을 검색하는 서브 질의
  - 다중 열 서브 쿼리: 하나 이상의 열을 검색하는 서브 질의
- 서브 쿼리 규칙
  - 서브 쿼리는 괄호를 묶어서 사용
  - 단일 행 연산자 또는 다중 행 연산자로 서브 쿼리 연결
  - 서브 쿼리 실행 후 메인 쿼리 실행
  - 여러 서브 쿼리를 중첩해서 사용 가능

연산자 구분	종류	사용
단일 행 연산자	=, >, >=, <, <=, <>, !=	단일 행 서브 쿼리 다중 열 서브 쿼리
다중 행 연산자	IN, NOT IN, EXISTS, ANY, ALL	다중 행 서브 쿼리 다중 열 서브 쿼리

서브 쿼리(Sub Query)

- 단일 행 서브 쿼리
  - 서브쿼리 SELECT 문에서 단일 행 결과를 메인 쿼리에 전달
  - WHERE에 사용되는 열의 개수와 데이터 타입 일치 필요
  - 단일 행 연산자 사용: =, >, >=, <, <=, <>, !=

```
SELECT *
FROM employees
WHERE phone_number = ( SELECT phone_number
                       FROM employees
                       WHERE employee_id = 100 );
```

```
SELECT *
FROM employees
WHERE hire_date = ( SELECT hire_date
                    FROM employees
                    WHERE email = 'SKING' );
```

```
SELECT *
FROM employees
WHERE hire_date < ( SELECT hire_date
                    FROM employees
                    WHERE email = 'SKING' );
```

```
SELECT *
FROM employees
WHERE salary = ( SELECT salary
                 FROM employees
                 WHERE hire_date = '06/01/03' );
```

```
SELECT *
FROM employees
WHERE salary >= ( SELECT salary
                  FROM employees
                  WHERE hire_date = '06/01/03' );
```

서브 쿼리(Sub Query)

- 다중 행 서브 쿼리
  - 서브쿼리 SELECT 문에서 다중 행 결과를 메인 쿼리에 전달
  - 단일 행 연산자는 사용할 수 없고, 다중 행 연산자만 사용 가능

다중 행 연산자	설명	예
IN	서브 쿼리 결과 중 같은 값이 포함되어 있으면 TRUE	IN(100, 101)
NOT IN	서브 쿼리 결과 중 같은 값이 포함되지 않았으면 TRUE	NOT IN(100, 101)
EXISTS	서브 쿼리의 결과가 존재하면 TRUE	EXISTS(100)
ANY(SOME)	조건식을 하나라도 만족하면 TRUE	ANY(100, 101)
ALL	조건식을 모두 만족하면 TRUE	ALL(100, 101)

- IN / NOT IN 연산자

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

```
SELECT *
FROM employees
WHERE salary NOT IN ( SELECT MAX(salary)
                      FROM employees
                      GROUP BY department_id );
```

- EXISTS 연산자

```
SELECT *
FROM employees
WHERE EXISTS ( SELECT *
               FROM employees
               WHERE employee_id = 100 );
```

서브 쿼리(Sub Query)

- ANY / ALL 연산자

```
SELECT *
FROM employees
WHERE salary = ANY(6000, 10000, 12000);
```

```
SELECT *
FROM employees
WHERE salary <> ANY(6000, 10000, 12000);
```

```
SELECT *
FROM employees
WHERE salary < ANY(6000, 10000, 12000);
```

```
SELECT *
FROM employees
WHERE salary <= ALL(6000, 10000, 12000);
```

```
SELECT *
FROM employees
WHERE salary >= ALL(6000, 10000, 12000);
```

```
SELECT *
FROM employees
WHERE salary <> ALL(6000, 10000, 12000);
```

서브 쿼리(Sub Query)

- ANY / ALL 연산자

```
SELECT *
FROM employees
WHERE salary < ANY( SELECT salary
                    FROM employees
                    WHERE hire_date > '08/01/01' );
```

```
SELECT *
FROM employees
WHERE salary < ALL( SELECT salary
                    FROM employees
                    WHERE hire_date > '08/01/01' );
```

```
SELECT *
FROM employees
WHERE salary > ANY ( SELECT MAX(salary)
                     FROM employees
                     GROUP BY department_id );
```

```
SELECT *
FROM employees
WHERE salary < ALL ( SELECT MAX(salary)
                     FROM employees
                     GROUP BY department_id );
```

서브 쿼리(Sub Query)

- 인라인 뷰(Inline View)
  - FROM 절에 있는 서브쿼리가 인라인 뷰를 생성
  - FROM 절에 직접 기술하여 효율적인 검색 가능
  - FROM 절에 있는 서브쿼리에는 자주 별칭을 사용

```
SELECT *
FROM employees E, ( SELECT department_id
                    FROM departments
                    WHERE department_name = 'IT') D
WHERE E.department_id = D.department_id;
```

```
SELECT E.last_name, E.salary, D.avg_sal
FROM employees E, ( SELECT department_id, AVG(salary) avg_sal
                    FROM employees
                    GROUP BY department_id) D
WHERE E.department_id = D.department_id
AND E.salary > D.avg_sal;
```

```
SELECT department_name, ( SELECT AVG(salary)
                          FROM employees
                          GROUP BY department_name )
FROM departments;
```

```
SELECT ROWNUM, salary
FROM ( SELECT salary
       FROM employees
       ORDER BY salary DESC )
WHERE ROWNUM <= 10;
```

[실습] 서브 쿼리

- departments 테이블에서 department_name이 'IT'인 department_id와 일치하는 employees 테이블의 first_name, last_name, job_id, salary를 조회

- locations 테이블에서 state_province이 'California'인 location_id와 일치하는 departments 테이블의 department_id, department_name을 조회

- countries 테이블에서 region_id가 3인 country_id가 포함된 locations 테이블의 city, state_province, street_address를 조회

[실습] 서브 쿼리

- departments 테이블에서 manager_id가 null이 아닌 department_id와 일치하는 employees 테이블의 first_name, last_name, job_id, salary를 조회

- locations 테이블에서 city가 'Seattle'를 포함하지 않는 location_id와 일치하는 departments 테이블의 department_id, department_name을 조회

- regions 테이블에서 region_name이 'Europe'인 region_id가 일치하는 countries 테이블에서 country_id가 포함된 locations 테이블의 city, state_province, street_address를 조회


suanlab
이수안 컴퓨터 연구소
suan computer laboratory