**서브쿼리**

메인 쿼리 안에 있는 쿼리 > 보조, 하위 쿼리

서브쿼리는 괄호로 둘러싸인 형태

여러 개의 서브쿼리 가능

서브쿼리의 위치에 따라 **- 스칼라 서브쿼리 (SELECT 절) > 컬럼, 표현식**

**- 인라인 뷰 (FROM 절) > 테이블**

**- 중첩 서브쿼리 (WHERE 절) > 조건절**

**1. 스칼라 쿼리**

메인 쿼리의 **SELECT 절**에 위치 > 하나의 **컬럼**이나 **표현식**처럼 사용

서브쿼리 수행 결과가 하나의 값이 되는 쿼리

-최종 반환하는 로우 수 1개 / 컬럼이나 표현식도 1개

별칭(ALIAS)을 주는 것이 일반적임

서브쿼리 내에서 메인쿼리와 JOIN 하는 것이 일반적. 안하면 여러 건이 조회될 수 있음

스칼라 서브쿼리는 성능을 위해서는 사용을 자제하는 편이 좋다

예시\_) 부서명 가져오기

**SELECT** a.employee\_id

, a.first\_name || a.last\_name emp\_name

, a.department\_id

, ( **SELECT** b.department\_name

**FROM** departments b

**WHERE** a.department\_id = b.department\_id ) dept\_name

**FROM** employees a

**ORDER BY** 1;

예시2\_) 여러 개의 스칼라 서브쿼리

**SELECT** a.employee\_id, a.first\_name || ' ' || a.last\_name emp\_name

, a.department\_id

, ( **SELECT** b.department\_name

**FROM** departments b

**WHERE** a.department\_id = b.department\_id ) dept\_name

, ( **SELECT** d.country\_name

**FROM** departments b ,locations c ,countries d

**WHERE** a.department\_id = b.department\_id

**AND** b.location\_id = c.location\_id

**AND** c.country\_id = d.country\_id ) country\_name

**FROM** employees a

**ORDER BY** 1;

**2. 인라인 뷰**

메인쿼리의 FROM절에 위치 > 서브쿼리 자체가 하나의 테이블처럼 동작

서브쿼리가 반환하는 로우/컬럼/표현식의 수는 1개 이상 가능

별칭(ALIAS)은 반드시 명시 > 테이블처럼 사용하기 때문에

메인쿼리와의 조인은 메인쿼리의 WHERE 절에서 처리하는게 일반적

필요 이유

- 기존 단일 테이블만 읽어서는 필요한 정보를 가져오기가 어려울 때

예\_) 특정 조건으로 집계한 결과와 조인을 해야할 시

- 복잡한 쿼리의 경우, 좀 더 직관적으로 쿼리를 작성할 수 있음

LATERAL 키워드 사용시 서브쿼리 내에서 조인 가능 > 스칼라 서브쿼리 같이 동작

인라인 뷰 쿼리 앞에 LATERAL 명시하면 메인쿼리 컬럼 참조 가능

예\_)

**SELECT** a.employee\_id

, a.first\_name || a.last\_name emp\_name

, a.department\_id

, c.dept\_name

**FROM** employees a

, ( **SELECT** b.department\_id

, b.department\_name dept\_name

**FROM** departments b ) c

**WHERE** a.department\_id = c.department\_id

**ORDER BY** 1;

**LATERAL** 사용 예\_)

**SELECT** a.employee\_id

, a.first\_name || a.last\_name emp\_name

, a.department\_id

, c.dept\_name

**FROM** employees a

, **LATERAL**

( **SELECT** b.department\_name dept\_name

**FROM** departments b

**WHERE** a.department\_id = b.department\_id ) c

**ORDER BY** 1;

**3. 중첩 서브쿼리**

메인쿼리의 WHERE 절에 위치 > 조건절의 일부로 사용됨

서브쿼리 최종 반환 값과 메인쿼리 테이블의 특정 컬럼 값을 비교 시 사용

서브쿼리가 반환하는 로우/컬럼/표현식의 수는 1개 이상 가능

별칭(ALIAS)는 사용 불가

서브쿼리 내에서 메인쿼리와 조인 가능

예\_)

**SELECT** \*

FROM departments a

**WHERE** **EXISTS** ( **SELECT** 'A'

**FROM** employees b

**WHERE** a.department\_id = b.department\_id

**AND** b.salary > 10000 ) ;

**SELECT** employee\_id

, first\_name || ' ' || last\_name emp\_name

, job\_id, salary

**FROM** employees

**WHERE** (job\_id, salary ) **IN** ( **SELECT** job\_id

, min\_salary

**FROM** jobs)

**ORDER BY** 1;

**SELECT** department\_id

, employee\_id, last\_name

, salary

**FROM** employees a

**WHERE** salary **>** (**SELECT** AVG(salary)

**FROM** employees b

**WHERE** a.department\_id = b.department\_id)

**ORDER BY** department\_id ;

**4. 세미 조인**

서브 쿼리의 있는 로우와 조건이 맞는 메인쿼리의 로우 반환

메인쿼리와 중첩 서브쿼리를 사용할 때 사용하는 조인

(서브쿼리안에 메인쿼리와 조인 조건절이 있음)

EXIST, IN, 등등

**4. 안티 조인**

세미 조인에서 NOT 연산자 사용