

서브 쿼리(Subquery) 개념서

이 문서는 SQLD 시험을 위한 서브 쿼리의 개념과 활용에 대해 상세히 설명합니다. 서브 쿼리의 기본 개념, 종류, 위치에 따른 특징, 다중행 연산자, 작성 시 주의사항, 고급 활용 방법, 그리고 SQLD 시험 대비 팁을 포함하고 있습니다.

1. 서브 쿼리의 기본 개념

1.1 정의

- 하나의 SQL문 안에 포함되어 있는 또 다른 SQL문
- 메인 쿼리가 서브 쿼리를 포함하는 종속적인 관계
- 괄호()로 감싸서 표현

1.2 특징

- 독립적인 실행 가능**
 - 서브 쿼리는 단독으로도 실행 가능한 완전한 SELECT문
- 실행 순서**
 - 일반적으로 서브 쿼리가 먼저 실행된 후 메인 쿼리 실행
 - 상호연관 서브쿼리는 예외
- 사용 위치에 따른 구분**
 - SELECT절, FROM절, WHERE절, HAVING절 등에서 사용 가능

2. 서브 쿼리의 종류

2.1 반환되는 데이터 형태에 따른 분류

- 단일행 서브쿼리**
 - 하나의 행만 반환
 - 단일행 비교 연산자 사용 (=, >, <, >=, <=, <>)

```
SELECT 사원명, 급여
FROM 사원
WHERE 급여 > (SELECT AVG(급여) FROM 사원);
```

- 다중행 서브쿼리**
 - 여러 행 반환
 - 다중행 비교 연산자 사용 (IN, ANY, ALL, EXISTS)

```
SELECT 사원명
FROM 사원
WHERE 부서번호 IN (SELECT 부서번호
                   FROM 부서
                   WHERE 지역 = '서울');
```

- 다중컬럼 서브쿼리**
 - 여러 컬럼 반환

```
SELECT *
FROM 사원
WHERE (부서번호, 급여) IN (SELECT 부서번호, MAX(급여)
                          FROM 사원
                          GROUP BY 부서번호);
```

2.2 동작하는 방식에 따른 분류

- 비연관 서브쿼리**
 - 메인 쿼리와 독립적으로 실행
 - 서브쿼리가 먼저 실행되고 그 결과를 메인 쿼리에서 사용

```
SELECT *
FROM 사원
WHERE 급여 >= (SELECT AVG(급여) FROM 사원);
```

- 연관 서브쿼리**
 - 메인 쿼리의 컬럼을 서브쿼리에서 참조
 - 메인 쿼리의 값에 따라 서브쿼리의 결과가 달라짐

```
SELECT *
FROM 부서 d
WHERE EXISTS (SELECT 1
              FROM 사원 e
              WHERE e.부서번호 = d.부서번호);
```

3. 위치에 따른 서브 쿼리

3.1 SELECT절 서브쿼리 (스칼라 서브쿼리)

- 하나의 행과 하나의 컬럼만 반환해야 함

```
SELECT 사원명,
       (SELECT 부서명
        FROM 부서
        WHERE 부서.부서번호 = 사원.부서번호) as 부서명
FROM 사원;
```

3.2 FROM절 서브쿼리 (인라인 뷰)

- 테이블 역할을 하는 서브쿼리
- 반드시 별칭을 가져야 함

```
SELECT a.사원명, a.부서명
FROM (SELECT 사원명, 부서번호
      FROM 사원 WHERE 입사일 >= '2020-01-01') a
JOIN 부서 b ON a.부서번호 = b.부서번호;
```

3.3 WHERE절 서브쿼리

- 조건절에서 사용되는 서브쿼리

```
SELECT *
FROM 사원
WHERE 부서번호 IN (SELECT 부서번호
                  FROM 부서
                  WHERE 지역 = '서울');
```

3.4 HAVING절 서브쿼리

- 그룹핑된 결과에 대한 조건을 지정할 때 사용

```
SELECT 부서번호, AVG(급여)
FROM 사원
GROUP BY 부서번호
HAVING AVG(급여) > (SELECT AVG(급여) FROM 사원);
```

4. 다중행 서브쿼리 연산자

4.1 IN

- 서브쿼리의 결과 중 하나와 일치하면 참

```
SELECT *
FROM 사원
WHERE 부서번호 IN (SELECT 부서번호
                  FROM 부서
                  WHERE 지역 = '서울');
```

4.2 ANY/SOME

- 서브쿼리의 결과 중 하나라도 만족하면 참

```
SELECT *
FROM 사원
WHERE 급여 > ANY (SELECT 급여
                  FROM 사원
                  WHERE 부서번호 = 10);
```

4.3 ALL

- 서브쿼리의 모든 결과를 만족하면 참

```
SELECT *
FROM 사원
WHERE 급여 > ALL (SELECT 급여
                  FROM 사원
                  WHERE 부서번호 = 20);
```

4.4 EXISTS

- 서브쿼리의 결과가 존재하면 참

```
SELECT *
FROM 부서
WHERE EXISTS (SELECT 1
              FROM 사원
              WHERE 사원.부서번호 = 부서.부서번호);
```

5. 서브 쿼리 작성 시 주의사항

5.1 구문 규칙

- 괄호 사용: 서브쿼리는 반드시 괄호로 감싸야 함
- Order By 제한: 서브쿼리에서는 Order By 사용 불가 (단, 인라인 뷰에서는 사용 가능)

5.2 성능 고려사항

- 실행 계획 확인: 서브쿼리의 실행 순서와 방식 검토
- 조인으로 대체 가능성 검토: 가능한 경우 조인으로 변환하여 성능 개선

5.3 NULL 처리

- 서브쿼리 결과에 NULL이 포함된 경우 주의
- NOT IN 사용시 특히 주의 필요

6. 고급 서브 쿼리 활용

6.1 중첩 서브쿼리

```
SELECT *
FROM 사원
WHERE 부서번호 IN (SELECT 부서번호
                  FROM 부서
                  WHERE 상위부서번호 IN (SELECT 부서번호
                                        FROM 부서
                                        WHERE 부서명 = '영업부'));
```

6.2 스칼라 서브쿼리의 활용

```
SELECT a.사원명,
       (SELECT 부서명 FROM 부서 WHERE 부서번호 = a.부서번호) as 부서명,
       (SELECT 직급명 FROM 직급 WHERE 직급코드 = a.직급코드) as 직급명
FROM 사원 a;
```

6.3 TOP-N 쿼리

```
SELECT *
FROM (SELECT *
      FROM 사원
      ORDER BY 급여 DESC)
WHERE ROWNUM <= 5;
```

SQLD 시험 대비 TIP

주요 출제 포인트

- 서브쿼리의 종류와 특징 - 단일행/다중행/다중컬럼 서브쿼리 - 연관/비연관 서브쿼리
- 다중행 연산자의 활용 - IN, ANY, ALL, EXISTS의 차이 - 각 연산자의 특징과 사용법
- 위치별 서브쿼리 - SELECT/FROM/WHERE/HAVING절의 서브쿼리 - 각 위치별 특징과 제약사항

학습 전략

- 서브쿼리의 기본 개념 이해
- 다양한 유형의 서브쿼리 실습
- 실행 결과 예측 훈련
- 성능 관련 내용 숙지