

# SQL로 데이터 다루기

5장 서브쿼리

황지영 선생님

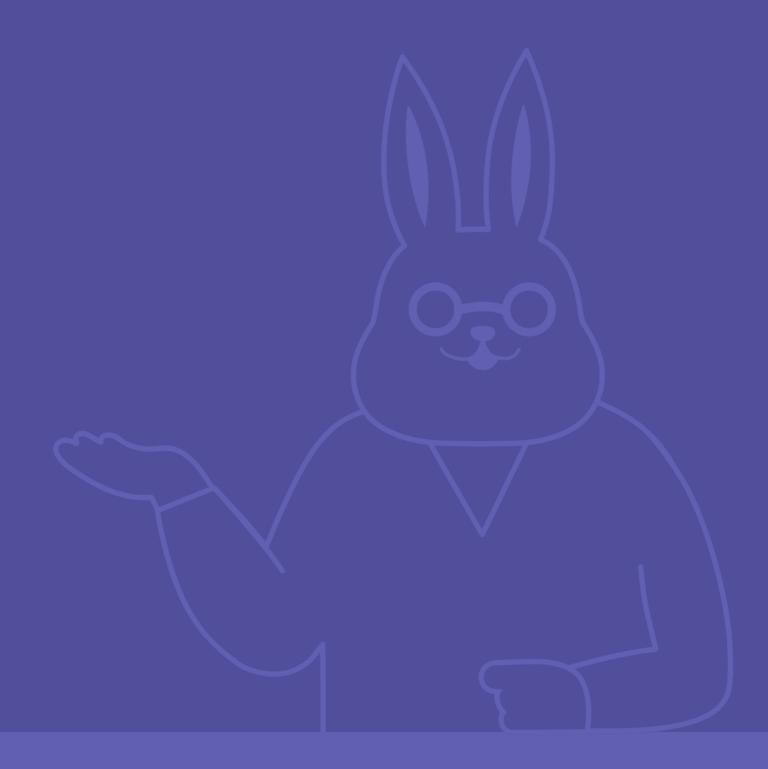


## Contents

- 01. 서브쿼리
- 02. 반환에 따른 분류
- 03. 위치에 따른 분류

01

# 서브쿼리



Confidential all right reserved

❷ 서브쿼리 정의

하나의 쿼리 안에 포함된 또 하나의 쿼리

메인 쿼리가 서브쿼리를 포함하는 종속적인 관계

#### ♥ 서브쿼리의 특징

- 알려지지 않은 기준을 이용한 검색에 유용
- 메인 쿼리가 실행되기 이전에 한 번만 실행
- 한 문장에서 여러 번 사용 가능

#### ❷ 기존의 검색 방법

```
SELECT * FROM employee
WHERE 급여 > 2500;
```

사원 elice의 급여를 알고 있는 상태에서 더 높은 급여를 받는 사원을 조회

#### ❷ 서브쿼리를 사용한 예시

### Example

```
SELECT * FROM employee
WHERE 급여 >
(SELECT 급여 FROM employee WHERE 이름='elice');
```

사원 elice의 급여를 알지 못해도 검색 가능

#### ♥ 서브쿼리 사용시 주의사항

- 1. 서브쿼리는 괄호와 함께 사용되어야 한다.
- 2. 서브쿼리 안에서 ORDER BY 절은 사용할 수 없다.
- 3. 서브쿼리는 연산자의 오른쪽에 사용되어야 한다.
- 4. 서브쿼리는 오로지 SELECT문으로만 작성 할 수 있다.

#### ❷ 서브쿼리 기본 문법

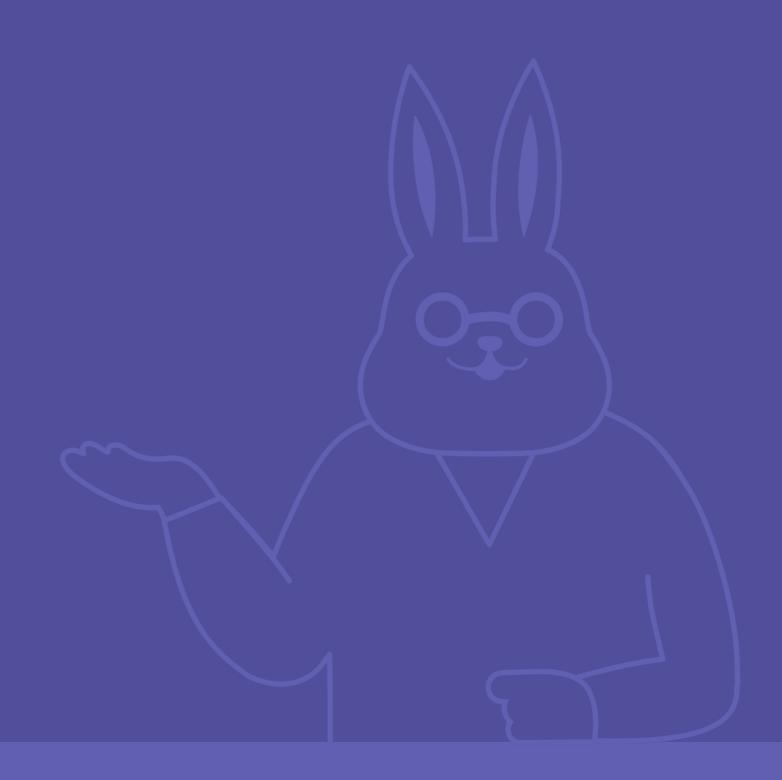
#### **Example**

```
SELECT * FROM employee
WHERE 급여 > 으른쪽

(SELECT 급여 FROM employee WHERE 이름='elice')
```

02

# 반환에 따른 분류



Confidential all right reserved

❷ 단일 행 서브쿼리 정의

결과가 한 행만 나오는 서브쿼리

서브쿼리가 결과를 1개의 값만 반환하고,

이 결과를 메인쿼리로 전달하는 쿼리

#### ❷ 단일 행 서브쿼리 기본 문법

#### **Example**

```
SELECT * FROM employee
WHERE 급여 > 단일행연산자

(SELECT 급여 FROM employee WHERE 사원번호 = 1);
```

사원번호는 기본적으로 1개만 있으므로 한 개의 행만 반환함 = 단일 행

### ❷ 단일 행 서브쿼리 연산자

기호	뜻
<b>=</b>	같다
<>>	같지 않다
	크다
>=	크거나 같다
<	작다
<=	작거나 같다

#### ♥ 다중 행 서브쿼리 정의

결과가 한 행만 나오는 단일 행 서브쿼리와는 다르게

서브쿼리가 결과를 2개 이상 반환하고,

이 결과를 메인쿼리로 전달하는 쿼리

#### ♥ 다중 행 서브쿼리 기본 문법

#### **Example**

```
SELECT * FROM employee
WHERE 급여 (IN <sup>다중행연산자</sup>
SELECT max(급여) FROM employee GROUP BY 부서번호);
```

### ♥ 다중 행 연산자

기호	뜻
IN	하나라도 만족하면 반환
ANY	하나라도 만족하면 반환 비교 연산 가능
ALL	모두 만족하면 반환 비교 연산 가능

#### ♥ 다중 행 연산자 사용 예시

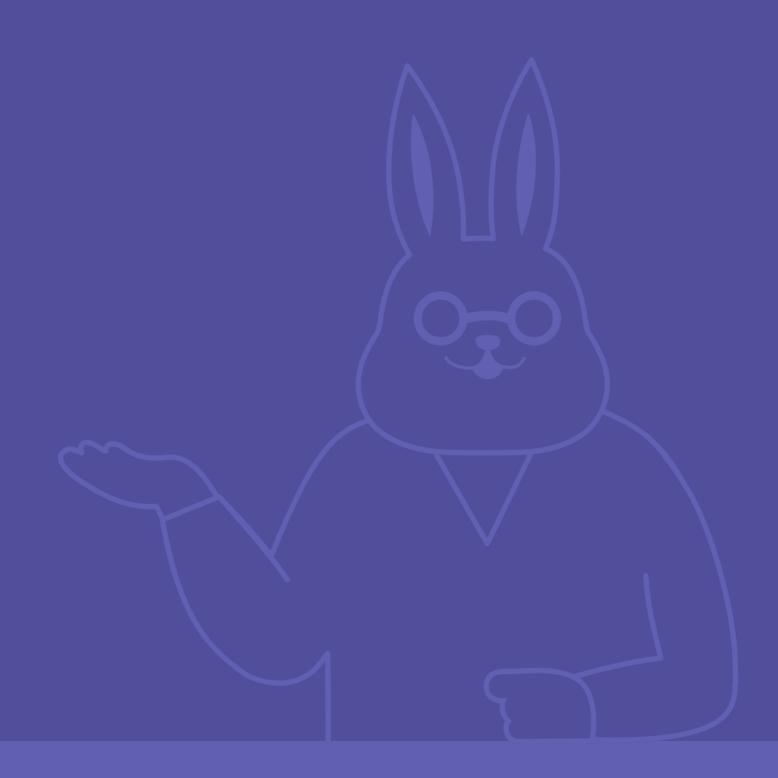
```
1 in (1, 2, 3, 4)

10 <any (1, 2, 3, 4)

99 >= all (99, 100, 101)
```

03

## 위치에 따른 분류



Confidential all right reserved

#### 03 위치에 따른 분류

#### ❷ 일반적인 서브쿼리의 형태

#### **Example**

```
SELECT * FROM employee
WHERE 급여 >
(SELECT 급여 FROM employee WHERE 이름='elice');
```

WHERE 절에 사용되는 서브쿼리 가장 일반적인 형태의 서브쿼리

#### 03 위치에 따른 분류

#### ❷ 스칼라 서브쿼리 정의

SELECT 절에서 사용하는 서브쿼리 스칼라 서브쿼리는 오로지 한 행만 반환 마치 JOIN을 사용한 것과 같은 결과를 나타낸다.

#### 03 위치에 따른 분류

#### ❷ 스칼라 서브쿼리 사용 방법

#### **Example**

```
SELECT students.name, (
    SELECT math
    FROM middle_test as m
    WHERE m.student_id = students.student_id
) AS middle_avg
FROM students;
```

## Credit

/\* elice \*/

코스 매니저 강윤수

콘텐츠 제작자 최찬환, 이재성

강사 황지영

감수자 장석준

디자인 황보영

## Contact

#### TEL

070-4633-2015

#### WEB

https://elice.io

#### E-MAIL

contact@elice.io

