



# SQL로 데이터 다루기 2

## 02 JOIN 심화



# 목차

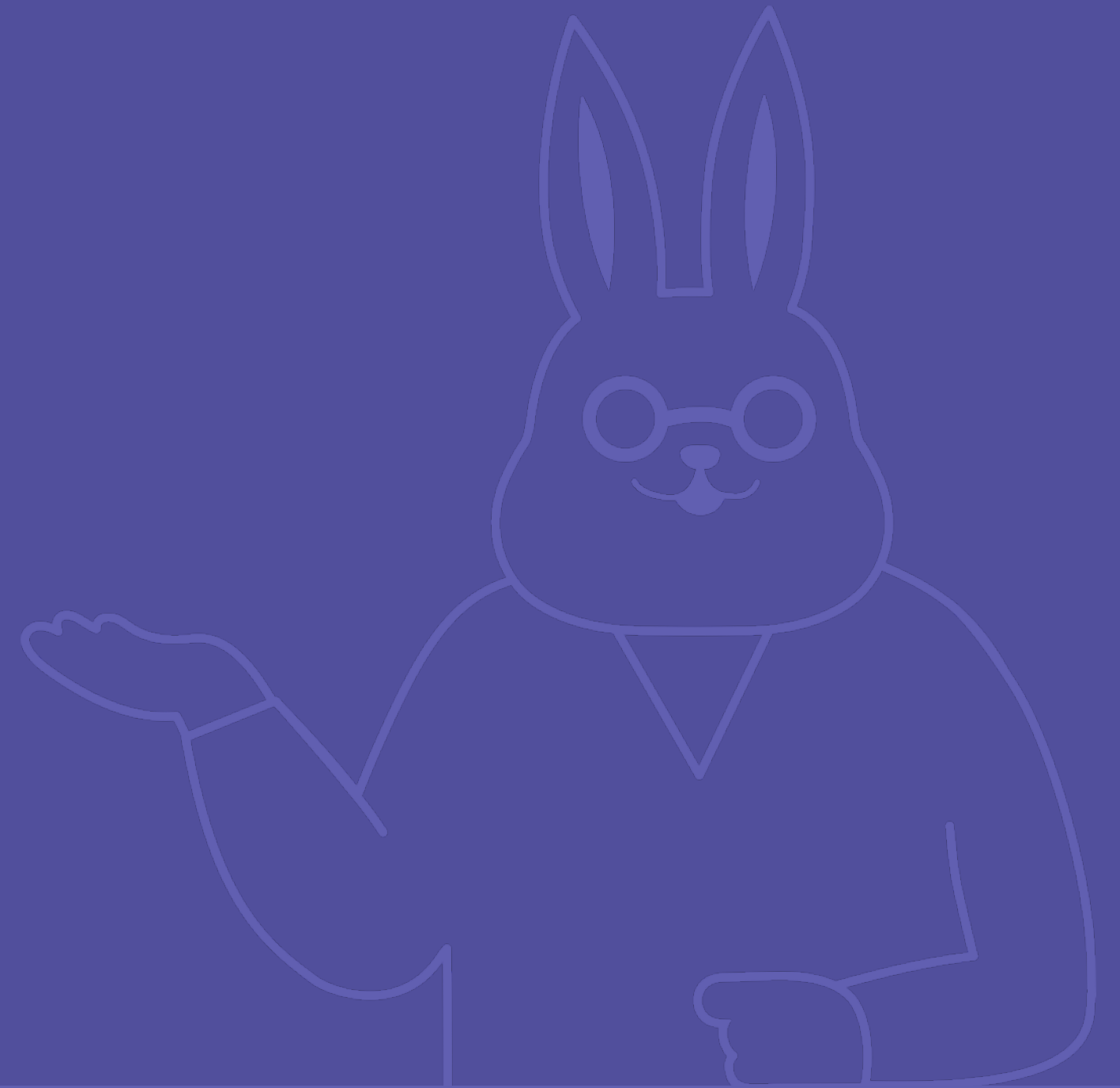
01. JOIN

02. FROM절 JOIN 형태

03. 셀프 조인

01

# JOIN



## ✓ JOIN(교집합)

두 개 이상의 테이블들을  
연결 또는 결합하여 데이터를 출력하는 것

연산자에 따라 JOIN 방식 분류 시

EQUI JOIN

Non EQUI JOIN

## ✓ EQUI JOIN(등가 교집합)

두 개의 테이블 간에 서로 정확하게 일치하는 경우를 활용하는 조인

간단히 말해,

등가 연산자를 사용한 조인을 의미

=

대부분 기본키-외래키 관계를 기반으로 발생하나,

모든 조인이 그런 것은 아니다

## ✓ Non EQUI JOIN(비등가 교집합)

두 개의 테이블 간에 서로 정확하게 일치하지 않는 경우를 활용하는 조인

간단히 말해,

등가 연산자 이외의 연산자들을 사용한 조인을 의미

&gt;

&gt;=

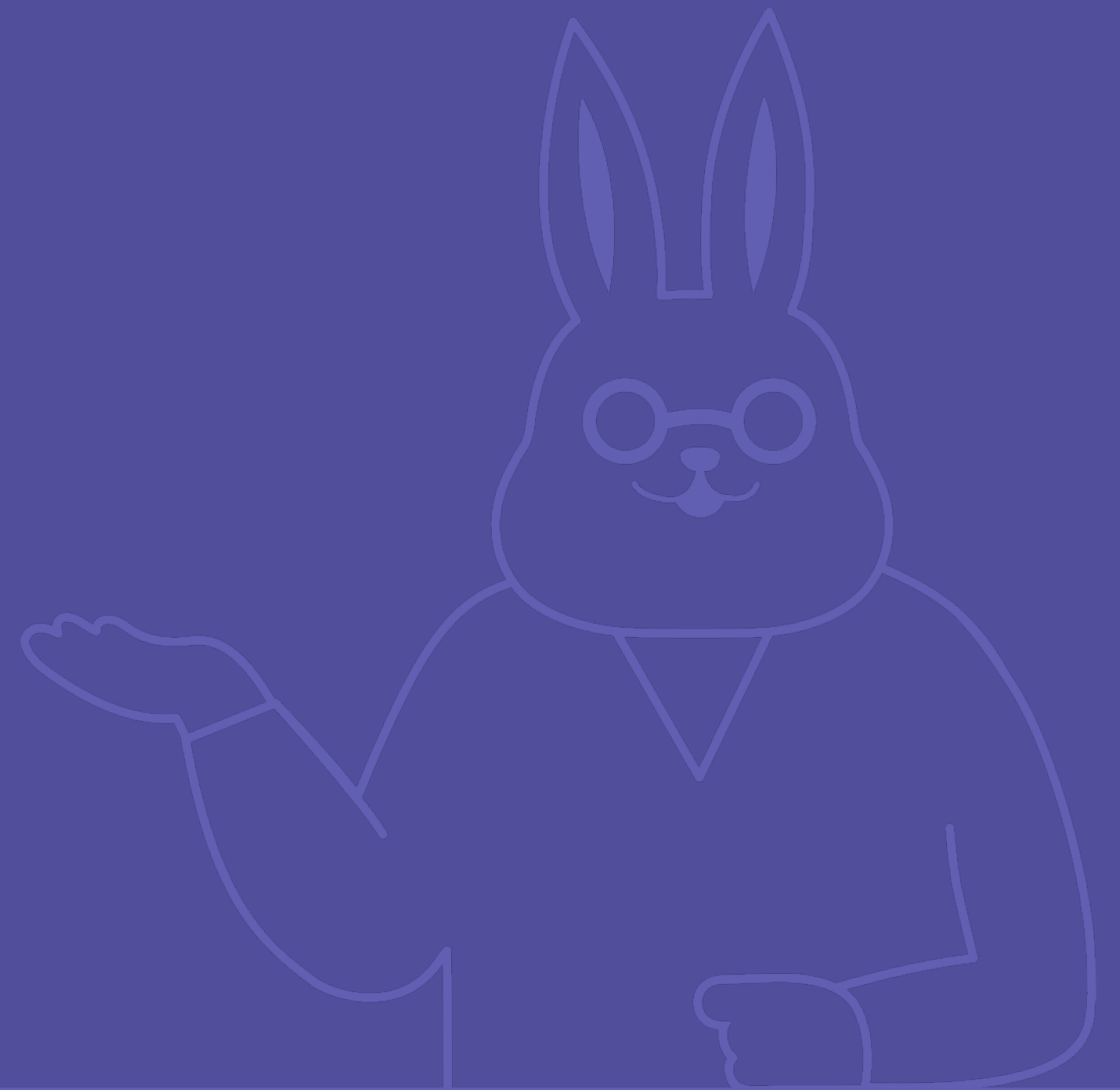
&lt;=

&lt;

BETWEEN

02

# FROM절 JOIN 형태



## ✓ INNER JOIN

내부 JOIN 이라고 하며 JOIN 조건에서

동일한 값이 있는 행만 반환

INNER JOIN은 JOIN의 기본값으로

‘INNER’ 생략 가능

예시

```
SELECT * FROM 테이블1 [INNER] JOIN 테이블2
```

INNER JOIN구로 테이블 정의

```
ON 테이블1.컬럼명 = 테이블2.컬럼명;
```

ON구를 사용해 조인 조건 지정



✓ INNER JOIN

USER

USER_ID	NAME	CLASS_ID
1000	모자장수	001
1001	엘리스 토끼	002
1002	체셔	001
1003	도도새	002

CLASS

ID	CLASS
001	A
002	B



결과값

USER_ID	NAME	CLASS_ID	ID	CLASS
1000	모자장수	001	001	A
1001	엘리스 토끼	002	002	B
1002	체셔	001	001	A
1003	도도새	002	002	B

예시

```
SELECT * FROM USER a JOIN CLASS b
```

```
ON a.CALSS_ID = b.ID;
```

ON 조건절을 활용하면 칼럼명이  
다르더라도 JOIN 조건을 사용 가능

## ✓ USING 조건절

같은 이름을 가진 칼럼들 중 원하는 칼럼에

대해서만 선택적으로 등가 조인 가능

SQL Server에서는 지원X

예시

```
SELECT * FROM 테이블1 JOIN 테이블2
```

```
USING(기준칼럼);
```

USNIG 조건절 사용시에는  
칼럼이나 테이블에 별칭을 붙일 수 없음

## ✓ NATURAL JOIN

두 테이블 간의 **동일한 이름**을 갖는 모든  
칼럼들에 대해 **등가 조인**을 수행

예시

```
SELECT * FROM 테이블1 NATURAL JOIN 테이블2;
```

추가로 ON 조건절이나 USING 조건절,  
WHERE절에서 JOIN 조건 정의 불가

✓ NATURAL JOIN

USER

ID	NAME	CLASS_ID
1000	모자장수	001
1001	엘리스 토끼	002
1002	체셔	001
1003	도도새	002

CLASS

CLASS_ID	CLASS
001	A
002	B



결과값

ID	NAME	CLASS_ID	CLASS
1000	모자장수	001	A
1001	엘리스 토끼	002	B
1002	체셔	001	A
1003	도도새	002	B

예시

```
SELECT * FROM USER NATURAL JOIN CLASS;
```

- INNER JOIN과 달리 별칭 지정 불가

✓ CROSS JOIN

JOIN 조건이 없는 경우 생길 수 있는

모든 데이터의 조합을 조회

USER

ID	NAME	AGE
1000	모자장수	25
1001	엘리스 토끼	10
1002	체셔	15
1003	도도새	20

CLASS

CLASS_ID	CLASS
001	A
002	B



결과값

ID	NAME	AGE	CLASS_ID	CLASS
1000	모자장수	25	001	A
1000	모자장수	25	002	B
1001	엘리스 토끼	10	001	A
1001	엘리스 토끼	10	002	B
1002	체셔	15	001	A
1002	체셔	15	002	B
1003	도도새	20	001	A
1003	도도새	20	002	B

✓ CROSS JOIN

예시

```
SELECT * FROM PERSON
(CROSS) JOIN PUBLIC_TRANSPORT;
```

PERSON

ID	NAME	PHONE
1	kim	010
2	moon	011

PUBLIC\_TRANSPORT

TRANSPORT_ID	TYPE
01A	bus
01B	subway
02A	cab



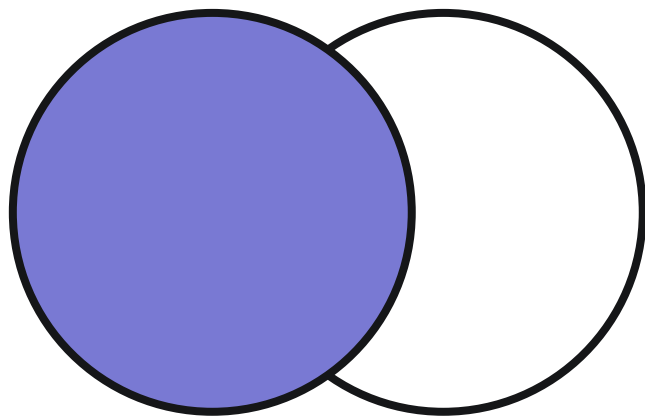
결과값

ID	NAME	PHONE	TRANSPORT_ID	TYPE
1	kim	010	01A	bus
1	kim	010	01B	subway
1	kim	010	02A	cab
2	moon	011	01A	bus
2	moon	011	01B	subway
2	moon	011	02A	cab

## ✓ OUTER JOIN

두 개의 테이블 간에 교집합을 조회하고  
한쪽 테이블에만 있는 데이터도 포함시켜서 조회  
빈 곳은 NULL 값으로 출력  
WHERE 조건절에서 한쪽에만 있는 데이터를  
포함시킬 테이블 쪽으로 (+)를 위치

✓ OUTER JOIN



예시

```
SELECT * FROM USER, CLASS
WHERE USER.CLASS_ID (+) = CLASS.CLASS_ID;
```

USER

ID	NAME	CLASS_ID
1000	모자장수	001
1001	엘리스 토끼	002
1002	체셔	002
1003	도도새	003

CLASS

CLASS_ID	CLASS
001	A
002	B

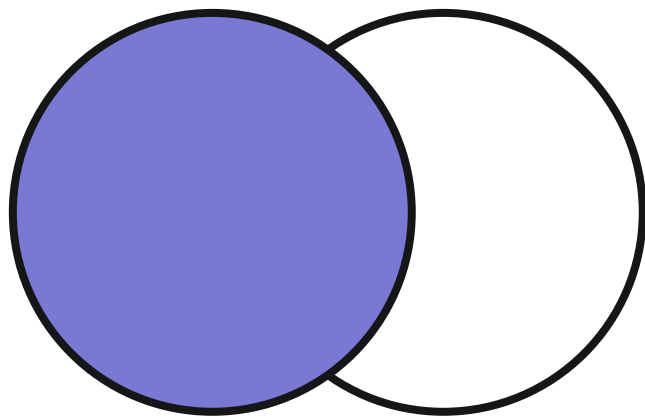


결과값

ID	NAME	CLASS_ID	CLASS_ID_1	CLASS
1000	모자장수	001	001	A
1001	엘리스 토끼	002	002	B
1002	체셔	002	002	B
1003	도도새	003		



✓ 표준 OUTER JOIN (LEFT JOIN)



예시

```
SELECT * FROM USER LEFT [OUTER] JOIN CLASS
ON USER.CLASS_ID = CLASS.CLASS_ID;
```

USER

ID	NAME	CLASS_ID
1000	모자장수	001
1001	엘리스 토끼	002
1002	체셔	002
1003	도도새	003

CLASS

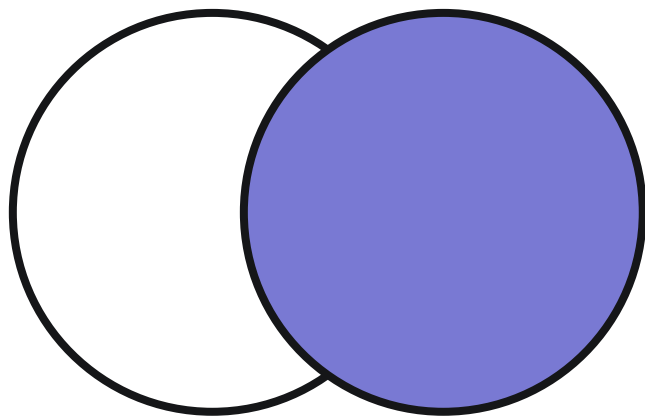
CLASS_ID	CLASS
001	A
002	B



결과값

ID	NAME	CLASS_ID	CLASS_ID_1	CLASS
1000	모자장수	001	001	A
1001	엘리스 토끼	002	002	B
1002	체셔	002	002	B
1003	도도새	003		

✔ 표준 OUTER JOIN (RIGHT JOIN)



예시

```
SELECT * FROM USER RIGHT [OUTER] JOIN CLASS
ON USER.CLASS_ID = CLASS.CLASS_ID;
```

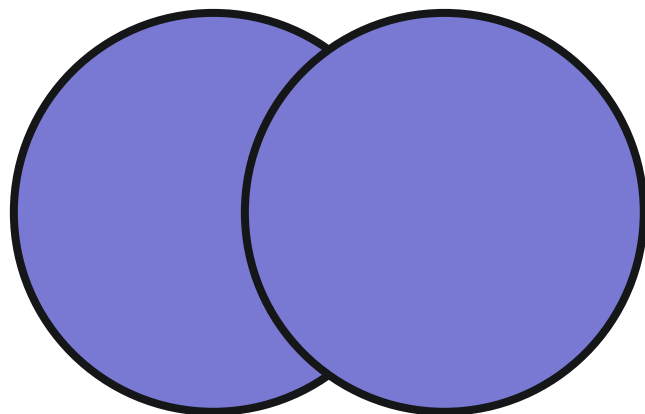
USER		
ID	NAME	CLASS_ID
1000	모자장수	001
1001	엘리스 토끼	002
1002	체셔	002

CLASS	
CLASS_ID	CLASS
001	A
002	B
003	C



결과값				
ID	NAME	CLASS_ID	CLASS_ID_1	CLASS
1000	모자장수	001	001	A
1001	엘리스 토끼	002	002	B
1002	체셔	002	002	B
			003	C

✓ 표준 OUTER JOIN (FULL OUTER JOIN)



예시

```
SELECT * FROM CLASS FULL OUTER JOIN USER
ON USER.CLASS_ID = CLASS.CLASS_ID;
```

USER

ID	NAME	CLASS_ID
1000	모자장수	001
1001	엘리스 토끼	002
1002	체셔	002
1003	도도새	003

CLASS

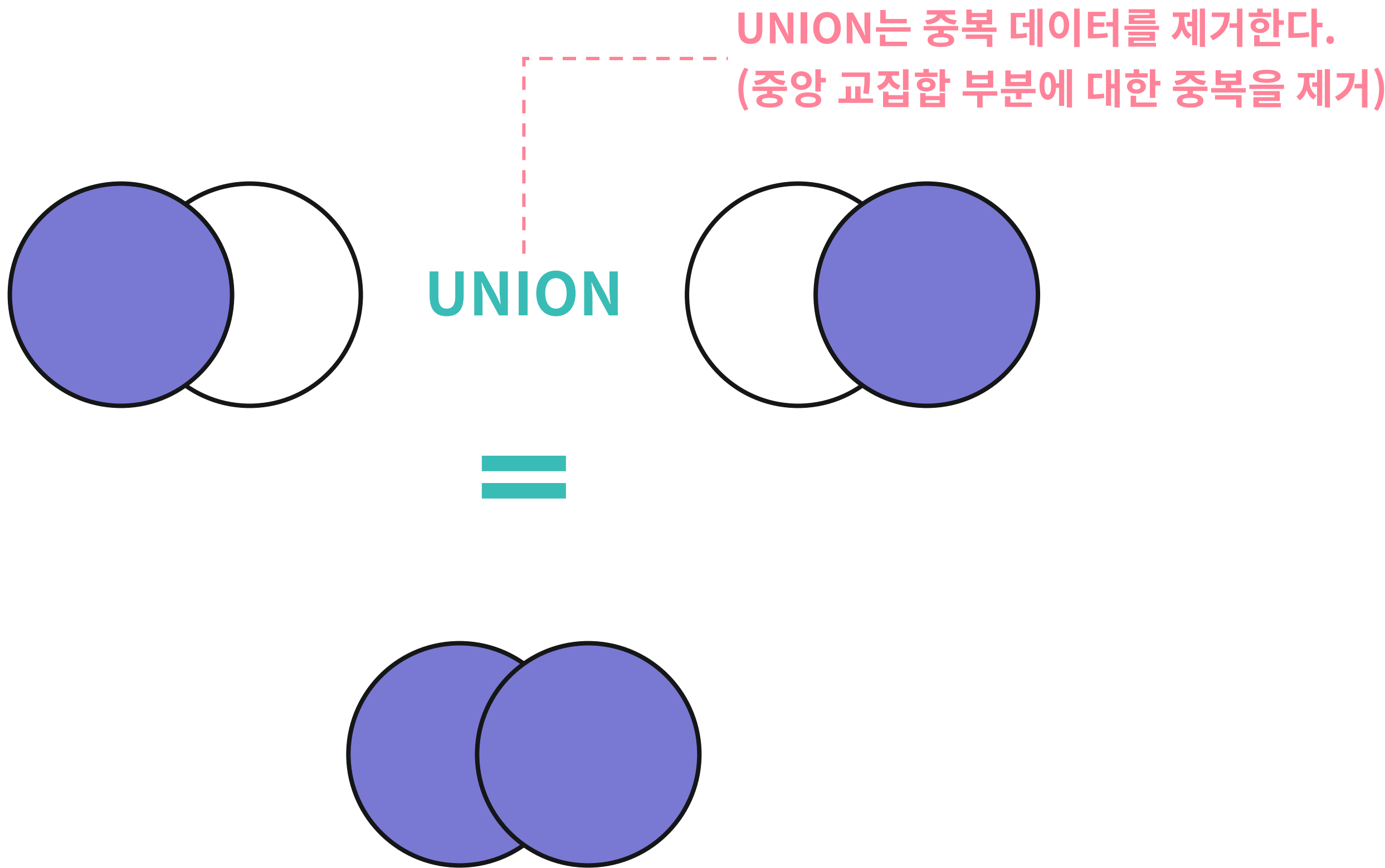
CLASS_ID	CLASS
001	A
002	B
003	C



결과값

ID	NAME	CLASS_ID	CLASS_ID_1	CLASS
1000	모자장수	001	001	A
1001	엘리스 토끼	002	002	B
1002	체셔	002	002	B
1003	도도새	003		
			003	C

✓ 표준 OUTER JOIN (FULL OUTER JOIN) in MySQL



✔ 표준 OUTER JOIN (FULL OUTER JOIN) in MySQL

예시

```
SELECT * FROM CLASS LEFT OUTER JOIN USER
ON USER.CLASS_ID = CLASS.CLASS_ID
UNION
SELECT * FROM CLASS RIGHT OUTER JOIN USER
ON USER.CLASS_ID = CLASS.CLASS_ID;
```

USER

ID	NAME	CLASS_ID
1000	모자장수	001
1001	엘리스 토끼	002
1002	체셔	002
1003	도도새	003

CLASS

CLASS_ID	CLASS
001	A
002	B
003	C



결과값

ID	NAME	CLASS_ID	CLASS_ID_1	CLASS
1000	모자장수	001	001	A
1001	엘리스 토끼	002	002	B
1002	체셔	002	002	B
1003	도도새	003		
			003	C

✓ INNER vs OUTER vs CROSS JOIN 비교

USER

ID	NAME	CLASS_ID
1000	모자장수	002
1001	엘리스 토끼	003
1002	체셔	004
1003	도도새	005

USER\_KEY

CLASS

CLASS_ID	CLASS
001	A
002	B
003	C

CLASS\_KEY

Q.각 JOIN에 따른 키 값의 조합과 출력되는 행의 개수?

INNER JOIN

A.

LEFT OUTER JOIN

A.

FULL OUTER JOIN

A.

CROSS JOIN

A.

RIGTH OUTER JOIN

A.

## ✓ INNER JOIN

JOIN을 활용한 쿼리에서도

**WHERE문**을 이용하여 조건을 걸 수 있음

예시

```
SELECT * FROM 테이블1 [INNER] JOIN 테이블2  
ON 테이블1.컬럼명 = 테이블2.컬럼명;  
WHERE [조건];
```

기존과 같이 WHERE문을 이용

✓ INNER JOIN

USER

USER_ID	NAME	CLASS_ID
1000	모자장수	001
1001	엘리스 토끼	002
1002	체셔	001
1003	도도새	002

CLASS

ID	CLASS
001	A
002	B



결과값

USER_ID	NAME	CLASS_ID	ID	CLASS
1000	모자장수	001	001	A

예시

```
SELECT * FROM USER a JOIN CLASS b
ON a.CALSS_ID = b.ID;
WHERE NAME = '모자장수';
```



03

# 셀프 조인



## ✓ 셀프 조인이란?

### 동일 테이블 사이의 조인

동일 테이블 사이의 조인을 수행하면 테이블과 칼럼 이름이  
모두 동일하기 때문에 식별을 위해 별칭 필수

## ✓ 셀프 조인 예시

예시

```
SELECT ALPHA.칼럼명, BETA.칼럼명, ...  
FROM 테이블1 ALPHA, 테이블1 BETA  
WHERE ALPHA.컬럼명2 = BETA.컬럼명1;
```

----- 동일 테이블 조인 시 별칭을 지정해줘야 구분 가능

✓ 셀프 조인 예시 - 계층형 질의

예시

```
SELECT ALPHA.사원번호, ALPHA.관리자, BETA.관리자 차상위;  
FROM 직원 ALPHA, 직원 BETA WHERE ALPHA.관리자 = BETA.사원번호;
```



# 크레딧

/\* elice \*/

코스 매니저

하주희

콘텐츠 제작자

하주희, 문범우

강사

문범우

감수자

장석준

디자이너

강혜정

# 연락처

TEL

070-4633-2015

WEB

<https://elice.io>

E-MAIL

[contact@elice.io](mailto:contact@elice.io)

