

자주 읽는 책을 데이터베이스에 꼼꼼히 정리하려고 한다.

책이름, 책의 발행년도, 책 가격, 책 분야,  
저자명, 저자의 출생년도, 저자소개, 저자의 이메일, 저자의 인스타그램주소,  
출판사명, 출판설립연도, 출판사소개설명, 출판사 이메일, 출판사 인스타그램 주소

**위 정보를 모두 데이터 베이스에 저장하고 싶은데...!!**  
**너무 하나의 행에 많은 것이 저장될 것 같다☹**

ex) Library(서재) 테이블에  
Book\_name, book\_year, book\_price, book\_type, autor\_name, author\_year....  
....

**컬럼이 10개가 넘는 테이블이 되어버리고 만다.**  
(\* 테이블이라 어렵게 생각말자. 엑셀에서도 컬럼과 행이 있다.)

일단 테이블을 대분류에 맞게 쪼개준다.

책이름, 발행년도, 책가격, 분야 → 책 테이블

저자명, 저자의 출생년도, 저자소개설명, 저자의 이메일

→ 저자 테이블

출판사명, 출판설립년도, 출판사소개설명, 출판사 이메일

→ 출판사 테이블

Publisher (출판사)
출판사 고유번호 ID
출판사명
출판설립년도
출판사소개설명
출판사 이메일
출판사 인스타

[출판사 테이블]

Book (책)
책 고유번호 ID
책이름
발행년도
책가격
분야

[책 테이블]

Author (저자)
저자 고유번호 ID
저자출생년도
저자소개설명
저자의 이메일
저자의 인스타

[저자 테이블]

잘 나누기는 했는데... 이 애들을 어떻게 연결하지..?  
이렇게만 두면, 그저 뿔뿔히 흩어진 정보들

출판사 - 책 - 저자

이렇게 책이 가운데 있고 이를 연결해주는 형태로 해주고 싶다.

책 테이블에서 출판사, 저자 테이블을 참조하기 쉽게  
ID값(출판사ID, 저자ID를)을 책 테이블에 연결해줄까..?

짜잔! 이렇게 책 테이블에 출판사ID를 넣어 출판사 테이블과 연결했고  
책 테이블에 저자ID도 넣어 저자 테이블과 연결할 수 있었다. **이게 조인이다.**



\*출판사ID는 출판사테이블의 마치 아이디 역할을 하는 기본키(PK)가 되지만  
책 테이블 내에서는 참조하기 쉽게 해주는 외래키(FK)가 된다.  
마찬가지로, 저자ID는 저자 테이블의 기본키(PK)가 되지만 책 테이블 내에서는 FK이다.

그렇다면 조인을 왜 해야되는지 알아보자.

**단순히 하나의 테이블만 바라보는 경우라면 조인이 필요가 없다.**

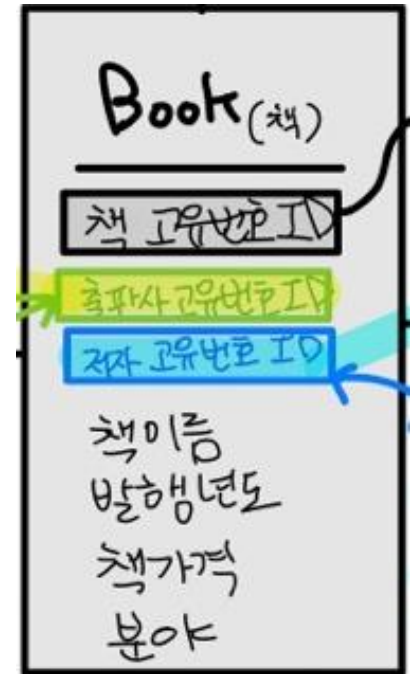
예시로 이런 경우는 필요가 없다.

Q. 책 이름이 <세상에서 가장 쉬운 코딩책>인 책을 전체적으로 보기

A.

```
1 -- [1]. 책 이름이 <세상에서 가장 쉬운 코딩책>인 책을 전체적으로 보기
2 SELECT *
3 FROM BOOK
4 WHERE BOOK_NAME = '세상에서 가장 쉬운 코딩책';
5
```

왜냐면, BOOK 테이블만 조회해도 충분히 다 볼 수 있기 때문에.



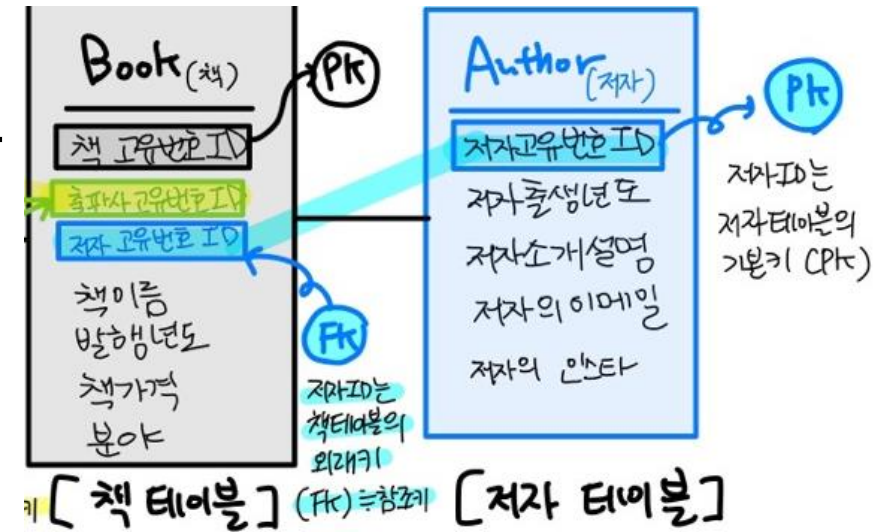
Query #1 Execution time: 9ms

book_id	publisher_id	author_id	book_name	book_pub_year	book_price	book_type
B1001	P1001	A1001	세상에서 가장 쉬운 코딩책	2023	25920	CODING

그렇다면 이건 어떤가?

Q. 책 이름이 <세상에서 가장 쉬운 코딩책>인 책의 저자의 소개 설명을 출력하기.  
흐음... BOOK 테이블과 AUTHOR 테이블을 두개다 봐야한다고 직감이 온다.

이럴때 쓰는것이 조인이고 조인에는 대표적으로 종류가  
이너조인, 아우터조인(왼쪽,오른쪽) 이렇게 있는 것이다.



일단 가장 직관적인 이너조인을 해본다.

이너조인은 같이 겹치는 데이터만 출력하는데, 두 테이블의 교집합이라고 생각하면 편하다.  
조인조건에는 BOOK테이블에 있는 AUTHOR\_ID와 AUTHOR테이블에 있는 AUTHOR\_ID를  
연결하면 될 것 같다.

```
7 -- [2]. 이번에는 책 이름이 <세상에서 가장 쉬운 코딩책>인 책의 저자의 소개 설명을 출력하기.  
8 -- BOOK 테이블만 보면 저자에 대한 내용이 없다.  
9 -- AUTHOR 테이블과 조인해야한다.  
10 -- 공통된 것만 출력하는 이너조인을 걸어본다.  
11 -- 이너조인은 교집합만 출력한다.  
12 SELECT *  
13 FROM BOOK B  
14 INNER JOIN AUTHOR A  
15 ON A.AUTHOR_ID = B.AUTHOR_ID  
16 WHERE B.BOOK_NAME = '세상에서 가장 쉬운 코딩책';
```

조인조건

```

7 -- [2]. 이번에는 책 이름이 <세상에서 가장 쉬운 코딩책>인 책의 저자의 소개 설명을 출력하기.
8 -- BOOK 테이블만 보면 저자에 대한 내용이 없다.
9 -- AUTHOR 테이블과 조인해야한다.
10 -- 공통된 것만 출력하는 이너조인을 걸어본다.
11 -- 이너조인은 교집합만 출력한다.
12 SELECT *
13 FROM BOOK B
14 INNER JOIN AUTHOR A
15 ON A.AUTHOR_ID = B.AUTHOR_ID
16 WHERE B.BOOK_NAME = '세상에서 가장 쉬운 코딩책';

```

오... **AUTHOR\_ID를 조인조건**으로 걸어서 두 개의 테이블이 아주 잘 연결된 것을 볼 수 있다.

Query #2 Execution time: 1ms

**조인조건**

book_id	publisher_id	author_id	book_name	book_pub_year	book_price	book_type	author_id	author_name	author_year	author_desc	author_email	author_insta
B1001	P1001	A1001	세상에서 가장 쉬운 코딩책	2023	25920	CODING	A1001	위캔코딩	1995	패션디자인에서 6개월만에 개발자가 되었다.	WECANCODING@GMAIL.COM	null

BOOK(책)테이블 컬럼들

AUTHOR(저자)테이블 컬럼들

A.

```
12 SELECT A.AUTHOR_DESC  
13 FROM BOOK B  
14 INNER JOIN AUTHOR A  
15 ON A.AUTHOR_ID = B.AUTHOR_ID  
16 WHERE B.BOOK_NAME = '세상에서 가장 쉬운 코딩책';
```

Query #2

Execution time: 1ms

author\_desc

패션디자인에서 6개월만에 개발자가 되었다.

자 이제부터 모두가 헛갈려하는 아우터조인에 대해 설명하려한다.

**이것을 명심하라. FROM 뒤에 오는 테이블**

**이 테이블이 기준이 된다.**

그리고 이 기준 테이블은 왼쪽이라 생각하고 조인테이블은 오른쪽에있다 생각해본다.

```
25 SELECT *
26 FROM BOOK B
27 RIGHT OUTER JOIN AUTHOR A
28 ON A.AUTHOR_ID = B.AUTHOR_ID;
29
```

기준테이블(L)

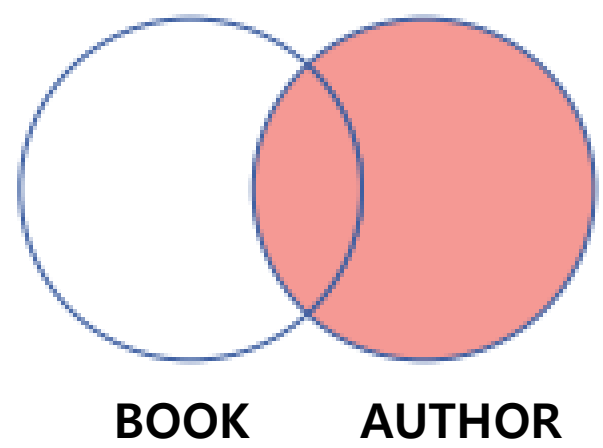
조인테이블(R)



```
18
19 -- [3]. 마우터조인은 FROM 바로 뒤에나오는 테이블을 기준으로 두고(여기서는 BOOK이 기준)
20 -- LEFT 조인이면 왼쪽 테이블 위주(기준테이블)로 출력하고
21 -- RIGHT 조인이면 오른쪽 테이블 위주(조인하는테이블)로 출력된다.
22 -- 예시로, 하단의 RIGHT 쿼리를 실행하면 AUTHOR 테이블의 데이터는 모두 나온다.
23 -- 하지만 BOOK 테이블은 AUTHOR와 겹치는 데이터가 존재하지 않을수가 있다.
24 -- 그럴경우 NULL로 보여진다.
```

```
25 SELECT *
26 FROM BOOK B
27 RIGHT OUTER JOIN AUTHOR A
28 ON A.AUTHOR_ID = B.AUTHOR_ID;
29
```

RIGHT OUTER TABLE이니 오른쪽 AUTHOR의 데이터는 모두 출력하되,  
BOOK과 겹치지 않는 데이터의 경우 NULL로 표기한다.

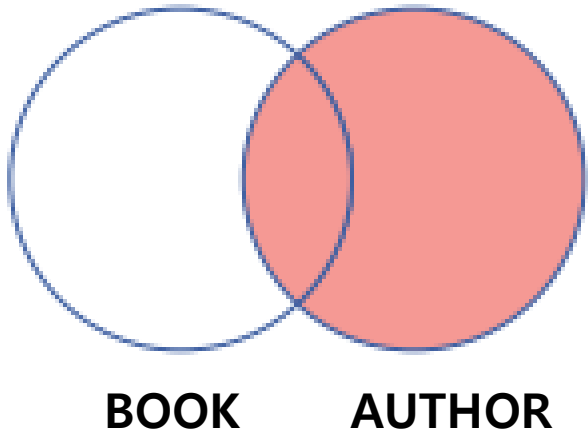


Query #3 Execution time: 1ms

book_id	publisher_id	author_id	book_name	book_pub_year	book_price	book_type	author_id	author_name	author_year	author_desc	author_email	author_insta
B1001	P1001	A1001	세상에서 가장 쉬운 코딩책	2023	25920	CODING	A1001	위캔코딩	1995	패션디자이너에서 6개월만에 개발자가 되었다.	WECANCODING@GMAIL.COM	null
B1002	P1001	A1002	달러 투자 무작정 따라하기	2023	20700	ECONOMY	A1002	박성현	1980	소설가로 데뷔한 지 20년이 넘는 베테랑 작가입니다.	author2@gmail.com	https://www.instagram.com/author2
B1003	P1002	A1003	전지적 푸바오 시점	2023	19800	ESSAY	A1003	송영관	1992	푸바오 할부지	author3@yahoo.com	https://www.instagram.com/author3
B1004	P1003	A1004	트렌드 코리아 2020	2020	17100	ECONOMY	A1004	김난도	1985	인류의 본질을 탐구하는 철학적인 시각으로 작품을 창작하는 작가입니다.	author4@gmail.com	https://www.instagram.com/author4
B1005	P1004	A1004	아프리카 청춘이다	2010	12600	ESSAY	A1004	김난도	1985	인류의 본질을 탐구하는 철학적인 시각으로 작품을 창작하는 작가입니다.	author4@gmail.com	https://www.instagram.com/author4
null	null	null	null	null	null	null	A1005	각두기데이터	1978	자연과 환경에 대한 경각심을 높이는 책으로 유명한 환경 운동가 겸 작가입니다.	author5@yahoo.com	https://www.instagram.com/author5

그러니까, 우측의 A1001~A1004까지는 모두 BOOK테이블과 조인조건으로 매칭이되는 테이블이다.

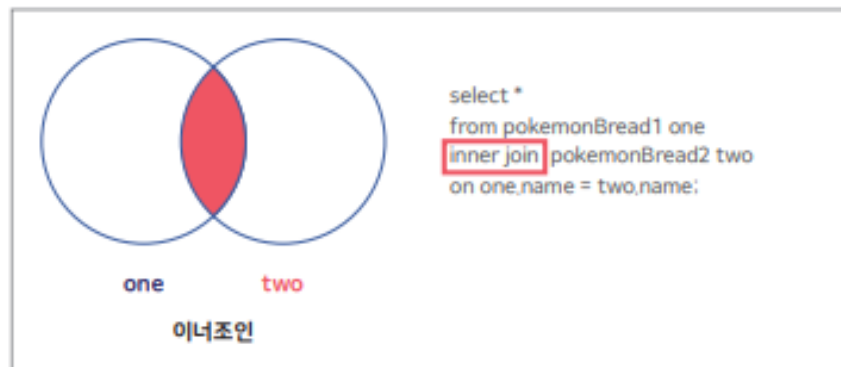
하지만 저자명 '깍두기데이터'인 A1005인 는 AUTHOR에는 있지만 BOOK에는 없다.  
그래서 우측은 온전히 나오지만 A1005가 없는 BOOK부분에는 null로 삽입이 된 것이다.



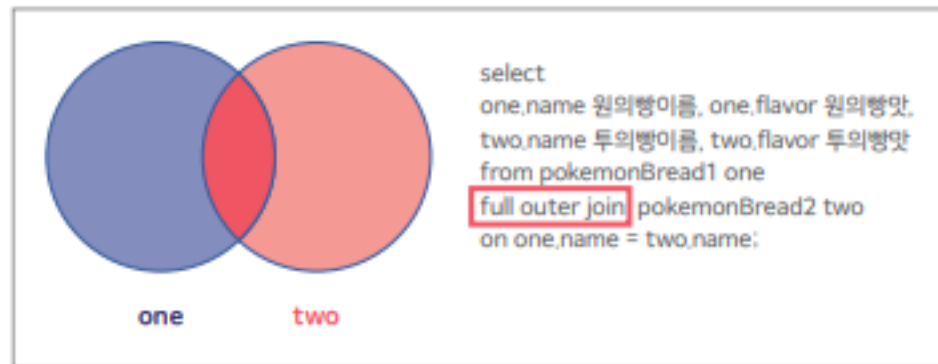
Query #3 Execution time: 1ms

book_id	publisher_id	author_id	book_name	book_pub_year	book_price	book_type	author_id	author_name	author_year	author_desc	author_email	author_insta
B1001	P1001	A1001	세상에서 가장 쉬운 코딩책	2023	25920	CODING	A1001	위캔코딩	1995	패션디자이너에서 6개월만에 개발자가 되었다.	WECANCODING@GMAIL.COM	null
B1002	P1001	A1002	달러 투자 무작정 따라하기	2023	20700	ECONOMY	A1002	박성현	1980	소설가로 데뷔한 지 20년이 넘는 베테랑 작가입니다.	author2@gmail.com	https://www.instagram.com/author2
B1003	P1002	A1003	전지적 푸바오 시점	2023	19800	ESSAY	A1003	송영관	1992	푸바오 할부지	author3@yahoo.com	https://www.instagram.com/author3
B1004	P1003	A1004	트렌드 코리아 2020	2020	17100	ECONOMY	A1004	김난도	1985	인류의 본질을 탐구하는 철학적인 시각으로 작품을 창작하는 작가입니다.	author4@gmail.com	https://www.instagram.com/author4
B1005	P1004	A1004	아프리카 청춘이다	2010	12600	ESSAY	A1004	김난도	1985	인류의 본질을 탐구하는 철학적인 시각으로 작품을 창작하는 작가입니다.	author4@gmail.com	https://www.instagram.com/author4
null	null	null	null	null	null	null	A1005	깍두기데이터	1978	자연과 환경에 대한 경각심을 높이는 책으로 유명한 환경 운동가 겸 작가입니다.	author5@yahoo.com	https://www.instagram.com/author5

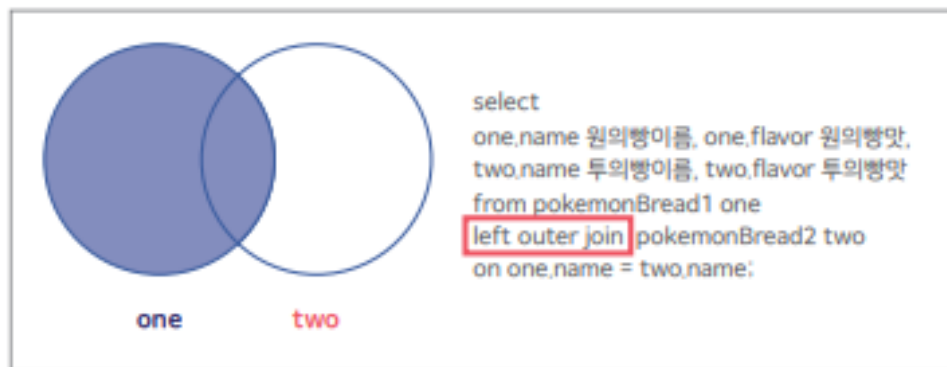
[그림 5-25] 교집합을 출력하는 이너조인



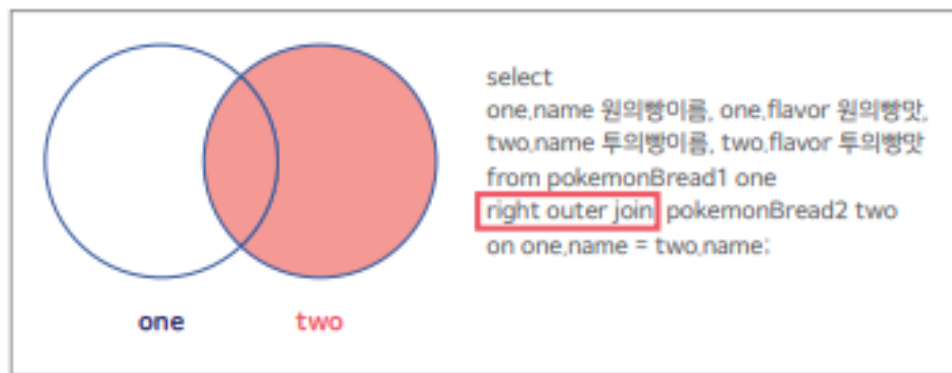
[그림 5-36] 데이터를 모두 출력하는 풀 아우터조인



[그림 5-28] 왼쪽 테이블 위주로 출력하는 왼쪽 아우터조인



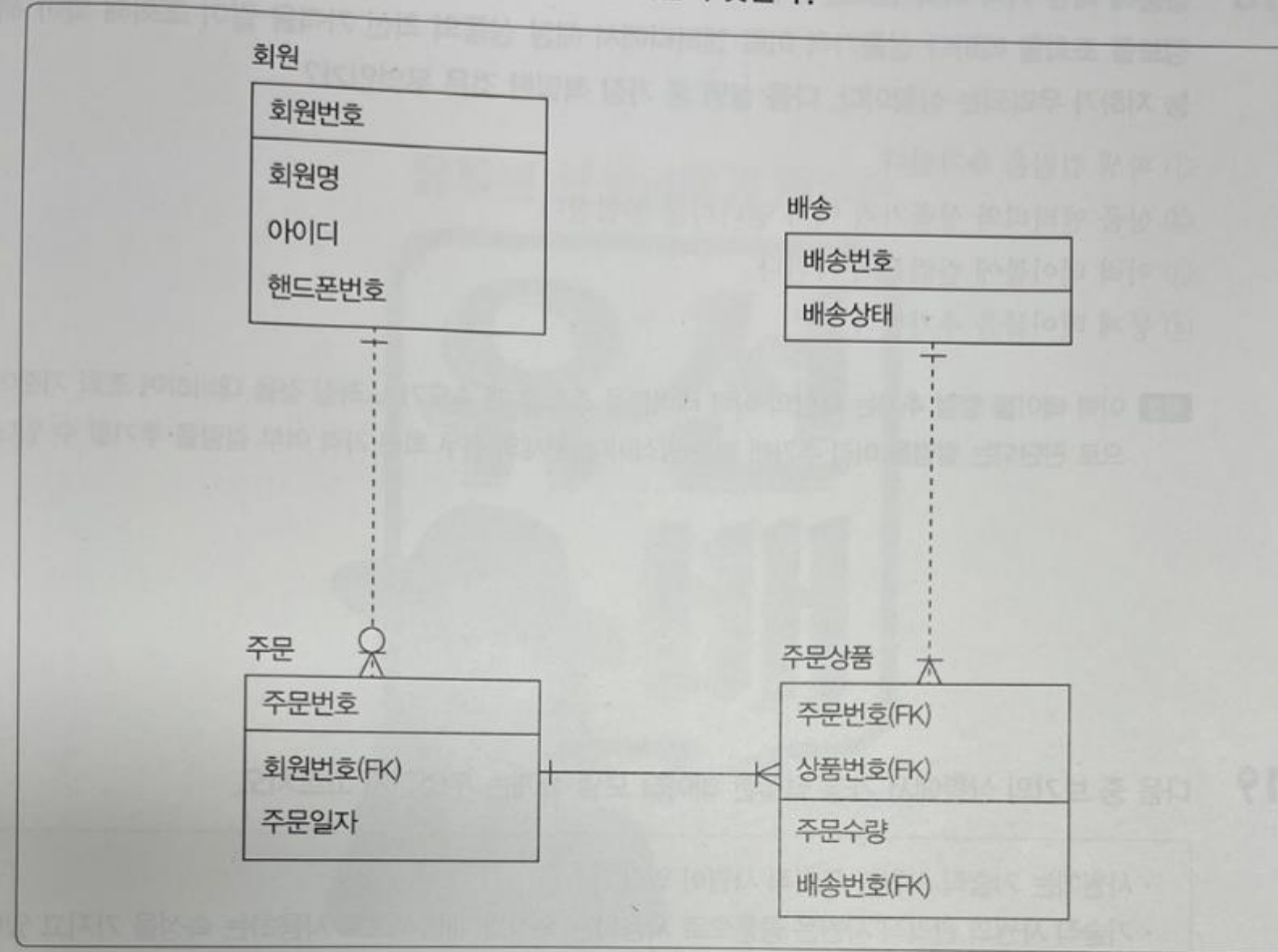
[그림 5-32] 오른쪽 테이블 위주로 출력하는 오른쪽 아우터조인



조인이 어려운거같다면 제가 드리는 데이터에 여러가지 시도 해보면서 직접 경험해보기 ~.  
더미데이터 만드느라고도 시간걸리네우..

# QUIZ.

17 다음 데이터 모델에 대한 설명으로 가장 적절한 것은 무엇인가?



- ① 회원의 핸드폰 번호와 배송상태를 함께 조회하려면 2번의 조인이 필요하다.
- ② 회원의 핸드폰 번호와 배송상태를 함께 조회하려면 3번의 조인이 필요하다.
- ③ 배송 엔터티에 주문번호를 추가하면 1번의 조인으로 회원의 핸드폰 번호와 배송상태를 함께 조회할 수 있다.
- ④ 배송 엔터티에 회원번호를 추가하면 조인하지 않아도 회원의 핸드폰 번호와 배송상태를 함께 조회할 수 있다.

# ANSWER.

회원의 핸드폰번호만 보고 싶다면 회원테이블만 보면 됩니다.

하지만, 회원이 주문한 주문목록을 보고 싶을때 회원 테이블만 보면 단지 회원 목록만 볼 수 있겠죠?!

주문 테이블에 있는 회원번호와 회원의 회원번호를 함께 조인(연결)해서 해당 회원번호에 맞는 주문을 볼 수 있습니다. 이렇게 하면 조인을 한 번 하게 되죠~

따라서, 회원의 아이디와 주문 일자를 보고 싶다면 한 번의 조인이 필요하다 볼 수 있습니다.

그런데 회원의 핸드폰번호와 배송상태를 함께 조회한다 해볼게요.

회원 테이블과 주문 테이블을 연결하고(PK이자 FK인 회원번호로)

+ 주문 테이블과 주문 상품 테이블을 연결하고(동일한 방식으로 주문번호로)

+ 주문상품테이블과 배송상태 테이블(동일한 방식으로 배송번호)을 연결하여야지만 볼 수 있습니다.

그럼 **총 세 번의 조인**이 필요할 것으로 보여지네요.