



## @March 28, 2022 SQL CLASS

### 내부JOIN 외부 JOIN 다중질의문

#### | 3개의 테이블을 조인하기

##### 내부 조인 기본

```
SELECT
*
FROM
    orders, book, customer
where
    orders.custid = customer.custid
    and
    orders.bookid = book.bookid;
```

```
SELECT
*
FROM
    orders
INNER JOIN book
ON orders.bookid = book.bookid
INNER JOIN customer
ON orders.custid = customer.custid;
```

##### 위 2개의 코드는 같은 결과를 출력한다

첫번째것이 간결하고 회사에서 많이 쓰인다.

##### 외부 조인 기본

만약 탈퇴한 회원이 있어서 그 회원이 customer 테이블에 없다면 조인을 해도 그 행이 뜨지 않는다.

- 만약 서점에서 하루매출을 알아내야할 때, 탈퇴한 고객이 있다고해서 그 order 리스트를 없애면 안되기 때문에 아웃 조인
- 이 때에는 손님의 정보가 없어도 order 리스트에 있다면 출력이 된다

```
SELECT
*
FROM
    orders, customer
WHERE
    orders.custid = customer.custid(+); //이곳 뒤에 (+)를 붙여주면 된다.
```

11	11	(null)	1	100000	28-MAR-22	(null)	(null)	(null)	(null)
----	----	--------	---	--------	-----------	--------	--------	--------	--------

위와 같이 customer이 지워져도 주문 리스트는 그대로 나오게 된다.

```
SELECT
*
FROM
orders
LEFT OUTER JOIN
customer
ON
orders.custid = customer.custid;
```

위 두 코드는 같은 결과를 출력한다

부속질의문

```
-- 부속 질의문 --

SELECT bookname
FROM book
WHERE price=(SELECT MAX(price) FROM BOOK);

--bookname이 정렬된 채로 bookname을 출력하기
SELECT bookname
FROM (select * from book order by book.bookname desc);

--Orders 안에 있는 즉, 주문한 사람들의 이름을 출력
SELECT name
FROM Customer WHERE custid IN (SELECT custid FROM Orders);

--아래 두개 똑같은 결과 --
--가격이 가장 비싼 책 이름을 출력하기 --

SELECT bookname
FROM book
where price = (SELECT MAX(price) FROM book);

SELECT bookname
FROM (SELECT * FROM BOOK ORDER BY price desc)
where rownum<2 and price is not null;

--대한미디어에서 출판한 도서를 구매한 고객의 이름을 보이기 --

SELECT bookid FROM Book WHERE publisher='대한미디어'; --먼저 대한미디어에서 출판한 bookid를 추출
SELECT custid FROM Orders WHERE bookid IN(SELECT bookid FROM Book WHERE publisher='대한미디어'); -- 주문 리스트에서 대한미디어에서 출판 책들이 포함되는 c
SELECT name FROM Customer WHERE custid IN (SELECT custid FROM Orders WHERE bookid IN(SELECT bookid FROM Book WHERE publisher='대한미디어'));

--주문하지 않은 책의 이름을 구하기--
SELECT bookname FROM BOOK WHERE bookid IN(SELECT bookid FROM BOOK WHERE bookid NOT IN (SELECT bookid FROM Orders));

-----

--SELECT distinct bookid, bookname
--FROM BOOK
--WHERE
--bookid not in(select distinct bookid from orders);

--SELECT name, adress
--FROM Customer cs
--WHERE EXISTS(SELECT * FROM Orders od WHERE cs.custid = od.custid);

--SELECT *
--FROM orders, customer
--WHERE orders.custid = customer.custid;
```

## 테이블 생성

```
CREATE TABLE TEMP (  
    temp_id number(2) primary key, //NOT NULL에 unique  
    test1 number default 0, //기본값은 0  
    test2 number CHECK (test2 >= 0), //0보다 커야 insert한다  
    test3 number not null, //비울 수 없음  
    test4 number unique //중복되면 안된다.  
);
```