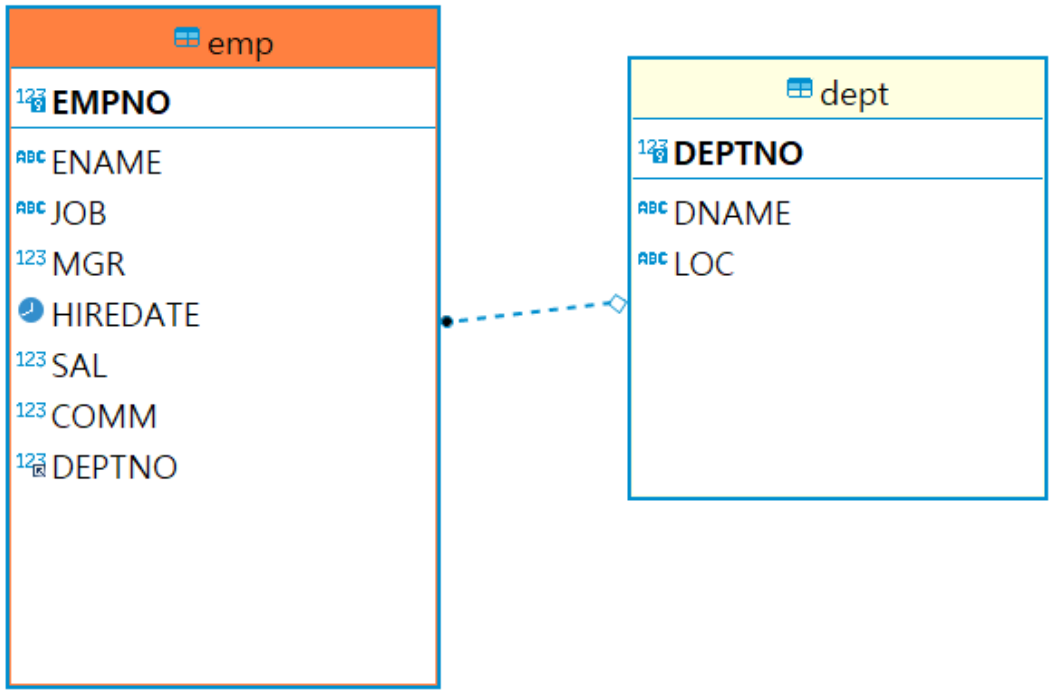
JOIN 보충 자료 ☺

현 상 원

조인은 두 개 이상 테이블을 조회할 때 사용.

하나의 테이블에 원하는 데이터가 모두 있다면 참 좋겠지만,
두 개의 테이블을 엮어야 원하는 결과가 나올 때가 있음.



즉, 조인을 안하면 emp테이블에서
사원들의 부서 번호는 아는데
부서 이름은 조회할 수 가 없음...

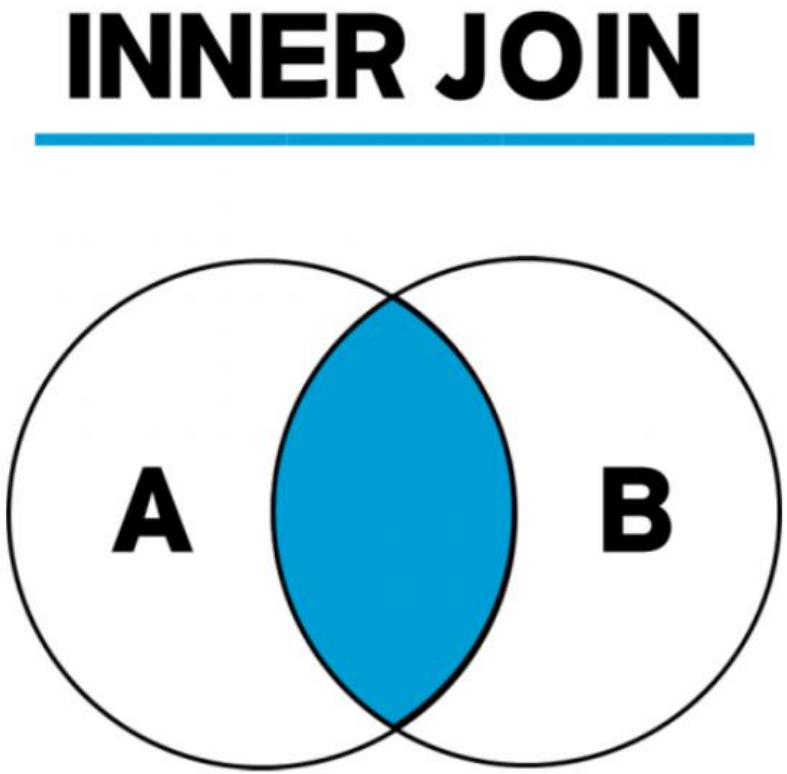
모든 사원의
부서 이름을
조회해 와!

A 사수

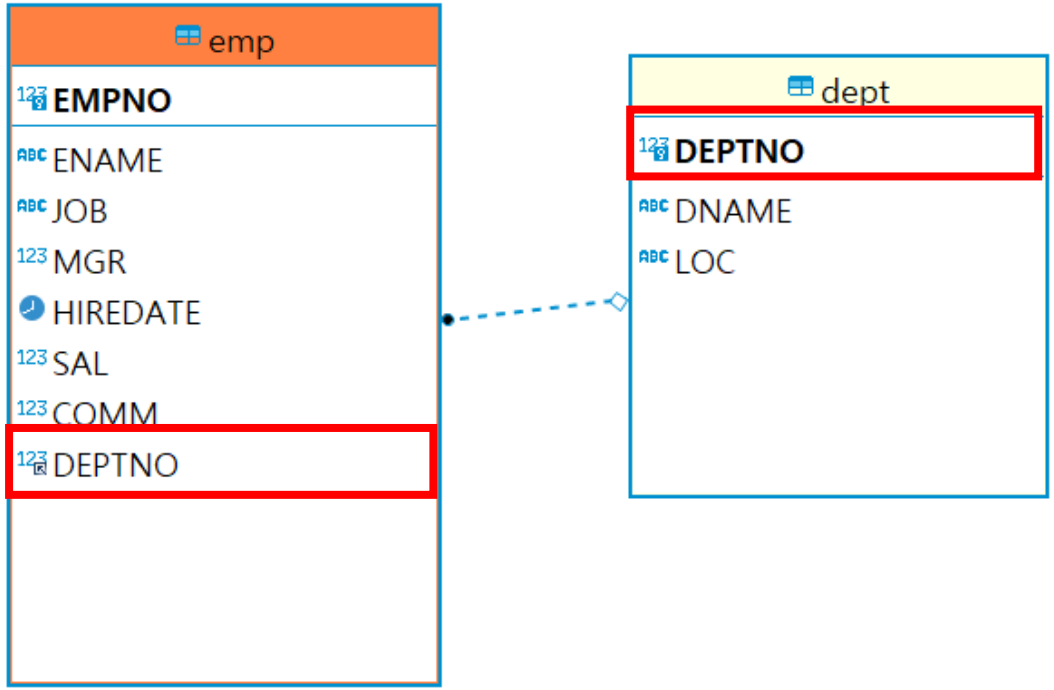
조인을 모르면 문제를 해결할 수 없음!

B 개발자

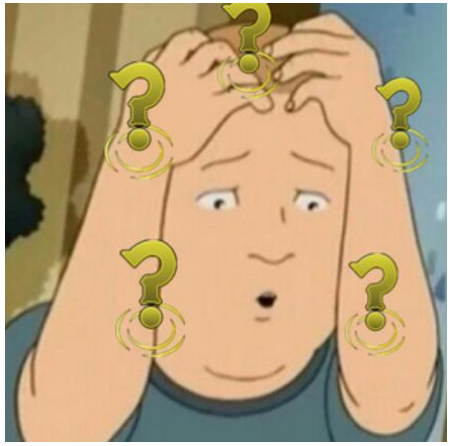
조인을 하려면 테이블들의 **교집합 데이터**를 찾아야함!



emp(사원)테이블과 dept(부서)테이블의 교집합은 무엇일까?



정답은 deptno(부서 번호)
즉, deptno(부서번호)를 이용해서
dept테이블에 있는 dname(부서이름)을 조회!



잠깐!, deptno가 교집합이라는 사실을 어떻게 알 수 있을까?

단순히 컬럼 이름이 같아서?..

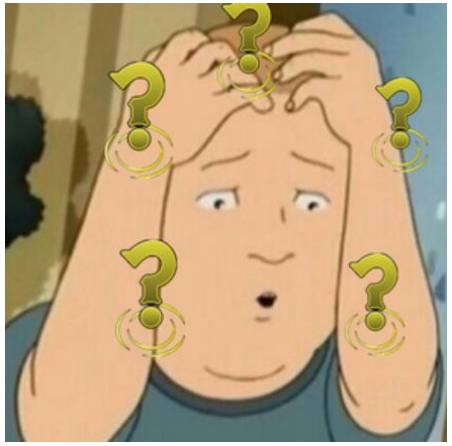
emp테이블을 create(만들때)할 때 dept테이블과 *관계를 형성하는 설정을 입력했기 때문.
DDL을 배우지 않았기 때문에 해당 설정은 다음에 배워보자!

```
CREATE TABLE `emp` (
  `EMPNO` decimal(4,0) NOT NULL,
  `ENAME` varchar(10) DEFAULT NULL,
  `JOB` varchar(9) DEFAULT NULL,
  `MGR` decimal(4,0) DEFAULT NULL,
  `HIREDATE` date DEFAULT NULL,
  `SAL` decimal(7,2) DEFAULT NULL,
  `COMM` decimal(7,2) DEFAULT NULL,
  `DEPTNO` decimal(2,0) DEFAULT NULL,
  PRIMARY KEY (`EMPNO`),
  KEY `FK_DEPTNO` (`DEPTNO`),
  CONSTRAINT `FK_DEPTNO` FOREIGN KEY (`DEPTNO`) REFERENCES `dept` (`DEPTNO`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

JOIN 하는 방법

```
SELECT
        컬럼 이름 …
FROM 테이블 이름
        INNER JOIN 두번째 테이블
        ON 교집합 컬럼 연결
[WHERE 검색 조건]
…
```

* FROM절과 WHERE절 사이에 위치한다.
* INNER는 생략 가능 (디폴트 값 : inner join)
* 내부 조인이라고 하며, 혹은 조인이라고 한다.



잠깐!, FROM뒤에 오는 테이블과 JOIN뒤에 오는 테이블 순서가 있을까?..

정답: 내부 조인은 순서가 없다.

다시 말해 emp와 dept 순서는 없다!
단, 내부 조인을 사용할 때.

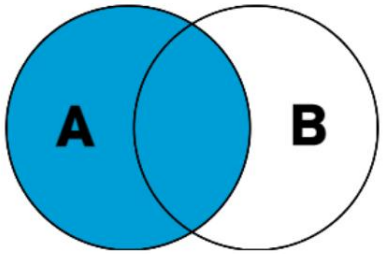
```sql
select
    e.ename,
    d.dname
from emp as e
inner join dept as d
on e.deptno = d.deptno
```

```sql
select
    e.ename,
    d.dname
from dept as d
inner join emp as e
on d.deptno = e.deptno
```

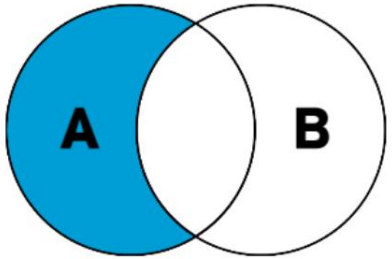
왼쪽 SQL코드는 동일한 결과값을 가져옴.

쌤, 조인 종류가 더 있나요...?

LEFT OUTER JOIN

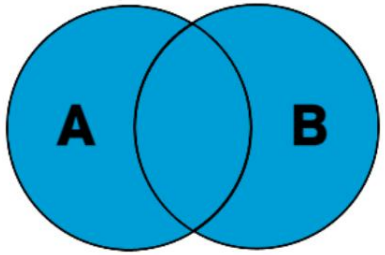



```
SELECT *
FROM A a
    LEFT JOIN B b
    ON a.KEY = b.KEY
```

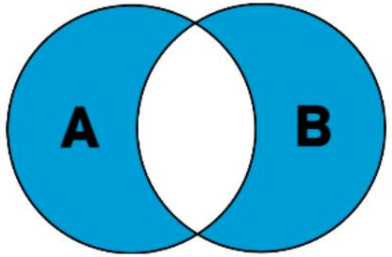



```
SELECT *
FROM A a
    LEFT JOIN B b
    ON a.KEY = b.KEY
WHERE b.KEY IS NULL
```

FULL OUTER JOIN

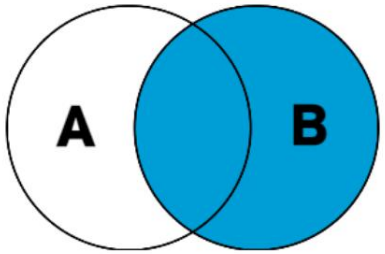



```
SELECT *
FROM A a
    FULL OUTER JOIN B b
    ON a.KEY = b.KEY
```

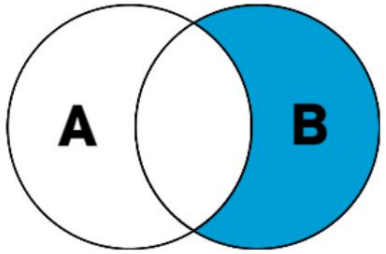



```
SELECT *
FROM A a
    FULL OUTER JOIN B b
    ON a.KEY = b.KEY
WHERE a.KEY IS NULL
    OR b.KEY IS NULL
```

RIGHT OUTER JOIN




```
SELECT *
FROM A a
    RIGHT OUTER JOIN B b
    ON a.KEY = b.KEY
```




```
SELECT *
FROM A a
    RIGHT OUTER JOIN B b
    ON a.KEY = b.KEY
WHERE a.KEY IS NULL
```

To be continued..