

2025년 상반기 K-디지털 트레이닝

SQL 고급 - 조인

[KB] IT's Your Life

✓ 조인(Join)

○ 조인

- 두 개 이상의 테이블을 서로 묶어서 하나의 결과 집합으로 만들어 내는 것
- 종류 : INNER JOIN, OUTER JOIN, CROSS JOIN, SELF JOIN

조인되고 나서는 하나의
테이블이기에
그렇게 다루면 된다

○ 데이터베이스의 테이블

- 중복과 공간 낭비를 피하고 데이터의 무결성을 위해서 여러 개의 테이블로 분리하여 저장
- 분리된 테이블들은 서로 관계(Relation)를 가짐
- 1대 다 관계 보편적

하지만 데이터가 흩어짐
조인으로 대처

✓ INNER JOIN(내부 조인) ✓

○ 조인 중에서 가장 많이 사용되는 조인

- 대개의 업무에서 조인은 INNER JOIN 사용 우리가 할 업무에서는 outer을 더 많이 쓰긴함
- 일반적으로 JOIN이라고 얘기하는 것이 이 INNER JOIN 지칭
- 사용 형식

```
SELECT <열 목록>  
FROM <첫 번째 테이블>  
INNER JOIN <두 번째 테이블>  
    ON <조인될 조건> ✓  
[WHERE 검색조건];
```

- JOIN만 써도 INNER JOIN으로 인식함

조인이 완료된 결과에서
필터링을 한번더 하겠다

✓ INNER JOIN(내부 조인)

```
USE sqldb;
```

```
SELECT *
```

```
FROM buytbl
```

```
INNER JOIN usertbl
```

```
ON buytbl.userID = usertbl.userID ✓
```

```
WHERE buytbl.userID = 'JYP';
```

on조건에서 비교되는 칼럼들을
공통 칼럼이라고 한다. 같은 의미여야함

행들을 순회하며 on조건에 부합하면
해당 행을 결합시켜라

	num	userID	prodName	groupName	price	amount	userID	name	birthYear	addr	mobile1	mobile2	height	mDate
▶	3	JYP	모니터	전자	400	1	JYP	조용필	1950	경기	011	4444444	166	2009-04-04

한쪽 경우에는 있고 다른 한쪽에서 없는 경우도 있다.

내부 조인은 둘다 있는 결과만을 표시한다.

한쪽이 없어도 보고 싶다면 -> 아우터 조인

✓ INNER JOIN(내부 조인)

공통칼럼 == 아이디



✓ OUTER JOIN(외부 조인)

기준이 되는 테이블의 행들이 꼭 하나 이상은 나오는 테이블

○ 조인의 조건에 만족되지 않는 행까지도 포함시키는 것

```
SELECT <열 목록>
FROM <첫 번째 테이블(LEFT 테이블)>
    <LEFT | RIGHT | FULL> OUTER JOIN <두 번째 테이블(RIGHT 테이블)>
    ON <조인될 조건>
[WHERE 검색조건];
```

○ LEFT OUTER JOIN

- 왼쪽 테이블의 것은 모두 출력되어야 한다면 이해
- 줄여서 LEFT JOIN으로 쓸 수 있음

user테이블과 buy테이블을 아우터 조인할때
기준 테이블을 user로 하면
buy하지 않은 사람들의 정보도 표시가 되겠네

○ RIGHT OUTER JOIN

- 오른쪽 테이블의 것은 모두 출력되어야 한다면 이해

left right full은 기준이 되는 테이블

left outer join은 일단 기준이 left. -> | left는 일단 다 나와야한다. 기준이기에

full은 둘다 일단 다 나와야 한다.

라고 생각하면 쉽다

✓ OUTER JOIN(외부 조인)

○ LEFT OUTER JOIN

```
USE sqldb;
```

```
SELECT U.userID, U.name, B.prodName, U.addr,  
       CONCAT(U.mobile1, U.mobile2) AS '연락처'  
FROM usertbl U  
     LEFT OUTER JOIN buytbl B  
       ON U.userID = B.userID  
ORDER BY U.userID;
```

유저테이블을 기준으로 (유저테이블을 다 보여주고)
조건에 해당되는게 있으면 합병해서 보여줘

물건을 사지 않은
조관우 김경호 임재범 이승기 윤종신도
한번씩은 나오는 것을 확인할 수 있다
물건을 많이 산 바비킴 경우, 산 물건 만큼 합병된 행들이 표시되고 있다

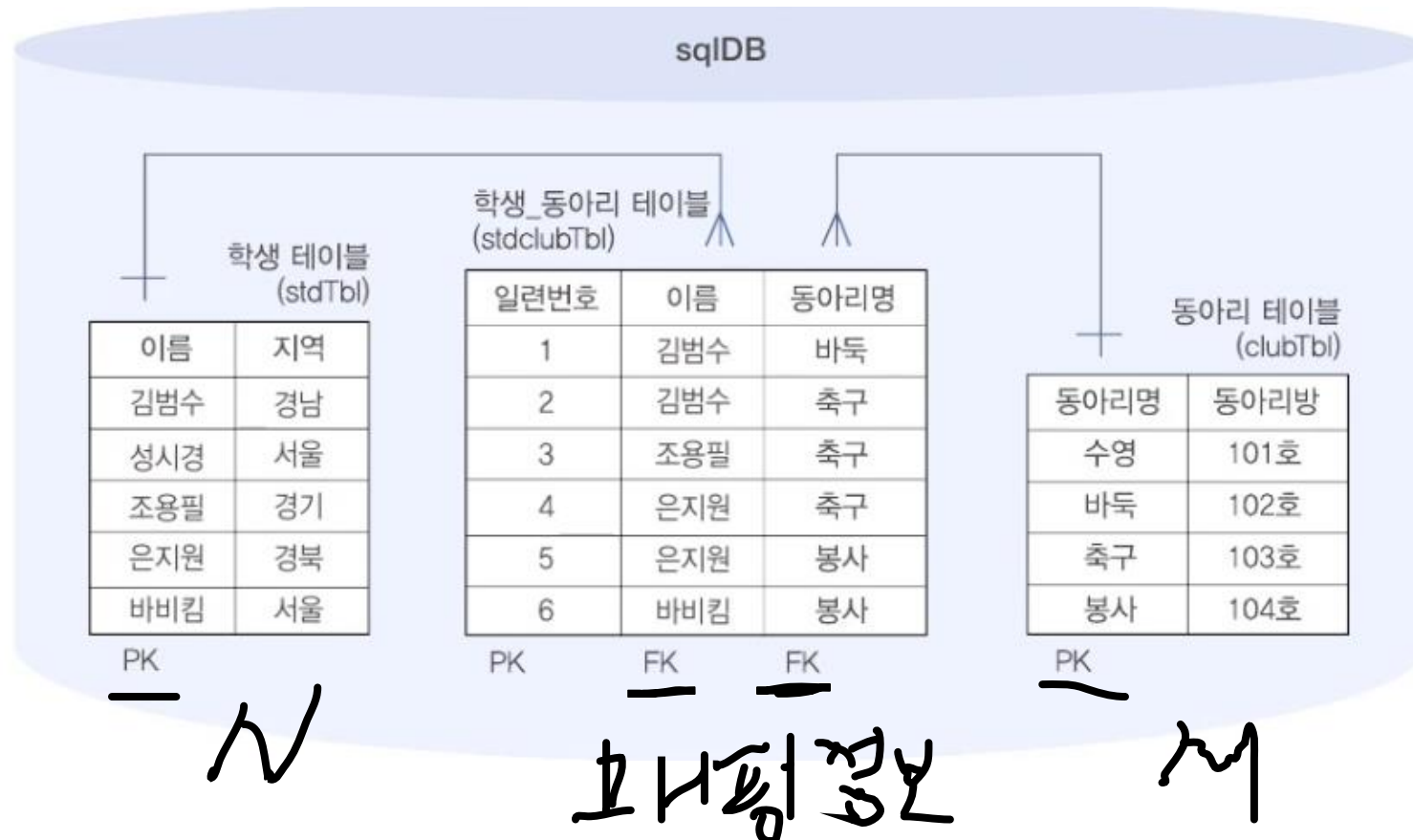
	userID	name	prodName	addr	연락처
▶	BBK	바비킴	모니터	서울	0100000000
	BBK	바비킴	메모리	서울	0100000000
	BBK	바비킴	운동화	서울	0100000000
	BBK	바비킴	운동화	서울	0100000000
	EJW	은지원	책	경북	0118888888
	EJW	은지원	청바지	경북	0118888888
	EJW	은지원	책	경북	0118888888
	JKW	조관우	NULL	경기	0189999999
	JYP	조용필	모니터	경기	0114444444
	KBS	김범수	운동화	경남	0112222222
	KBS	김범수	노트북	경남	0112222222
	KBS	김범수	청바지	경남	0112222222
	KKH	김경호	NULL	전남	0193333333
	LJB	임재범	NULL	서울	0166666666
	LSG	이승기	NULL	서울	0111111111
	SSK	성시경	책	서울	NULL
	YJS	윤종신	NULL	경남	NULL

✓ 3개의 테이블 조인



- 다대다(N:M) 관계에서 주로 사용
- 학생과 동아리의 관계
 - 한 학생은 여러 동아리에 가입 가능
 - 한 동아리는 여러 학생으로 구성됨

N:M관계는 테이블이 이때는 N:1, 1:1(중간 매핑테이블), 1:M 테이블로 나뉘어져 있다.



✓ 3개의 테이블 조인

○ 학생과 동아리의 관계

USE sqldb;

```
CREATE TABLE stdtbl (  
    stdName VARCHAR(10) NOT NULL PRIMARY KEY,  
    addr CHAR(4) NOT NULL  
);
```

1

```
CREATE TABLE clubtbl (  
    clubName VARCHAR(10) NOT NULL PRIMARY KEY,  
    roomNo CHAR(4) NOT NULL  
);
```

2

```
CREATE TABLE stdclubtbl(  
    num int AUTO_INCREMENT NOT NULL PRIMARY KEY,  
    stdName VARCHAR(10) NOT NULL,  
    clubName VARCHAR(10) NOT NULL,  
    FOREIGN KEY(stdName) REFERENCES stdtbl(stdName),  
    FOREIGN KEY(clubName) REFERENCES clubtbl(clubName)  
);
```

3

✓ 3개의 테이블 조인

문제 상황을 보고 이너 조인이 필요한지 아우터가 필요한지 결정부터 하자

○ 학생을 기준으로 학생 이름/지역/가입한 동아리/동아리방 정보 추출

```
SELECT S.stdName, S.addr, SC.clubName, C.roomNo
FROM stdtbl S
    INNER JOIN stdclubtbl SC
        ON S.stdName = SC.stdName
    INNER JOIN clubtbl C
        ON SC.clubName = C.clubName
ORDER BY S.stdName;
```

학생정보와 학생클럽매핑정보 이너조인
후
클럽정보 이너조인

이너조인 순서에 따라 결과가 달라질 수 있다. 어떤 것을 기준으로 하냐에 따라

○ 동아리를 기준으로 가입한 학생의 목록 추출

```
SELECT C.clubName, C.roomNo, S.stdName, S.addr
FROM stdtbl S
    INNER JOIN stdclubtbl SC
        ON SC.stdName = S.stdName
    INNER JOIN clubtbl C
        ON SC.clubName = C.clubName
ORDER BY C.clubName;
```

번외 주의 null과 연산자가 만나면 결과가 unknown임
where절에서는 true값만 표시함. 아래처리하면안됨

```
ON b.userID = u.usi
WHERE b.userID = NULL
```

아래처럼 해야함 혹은 is null 써야함

```
IFNULL(b.userID, TRUE)
```

✓ CROSS JOIN(상호 조인) 모든 조합 다나옴 nXM 개의 행

- 한쪽 테이블의 모든 행들과 다른 쪽 테이블의 모든 행을 조인시키는 기능
- CROSS JOIN의 결과 개수 = 두 테이블 개수를 곱한 개수

회원 테이블(userTbl)

아이디	이름	생년	지역	국번	전화번호	키	가입일
LSG	이승기	1987	서울	011	1111111	182	2008.8.8
KBS	김범수	1979	경남	011	2222222	173	2012.4.4
KKH	김경호	1971	전남	019	3333333	177	2007.7.7
JYP	조용필	1950	경기	011	4444444	166	2009.4.4
SSK	성시경	1979	서울			186	2013.12.12
LJB	임재범	1963	서울	016	6666666	182	2009.9.9
YJS	윤종신	1969	경남			170	2005.5.5
EJW	은지원	1978	경북	011	8888888	174	2014.3.3
JKW	조관우	1965	경기	018	9999999	172	2010.10.10
BBK	바비킴	1973	서울	010	0000000	176	2013.5.5

PK

구매 테이블(buyTbl)

순번	아이디	물품명	분류	단가	수량
1	KBS	운동화		30	2
2	KBS	노트북	전자	1000	1
3	JYP	모니터	전자	200	1
4	BBK	모니터	전자	200	5
5	KBS	청바지	의류	50	3
6	BBK	메모리	전자	80	10
7	SSK	책	서적	15	5
8	EJW	책	서적	15	2
9	EJW	청바지	의류	50	1
10	BBK	운동화		30	2
11	EJW	책	서적	15	1
12	BBK	운동화		30	2

PK FK

✓ CROSS JOIN(상호 조인)

- 테스트로 사용할 많은 용량의 데이터를 생성할 때 주로 사용 ✓
- ON 구문을 사용할 수 없음
- 대량의 데이터를 생성하면 시스템이 다운되거나 디스크 용량이 모두 찰 수 있어 COUNT(*) 함수로 개수만 카운트

```
USE employees;
```

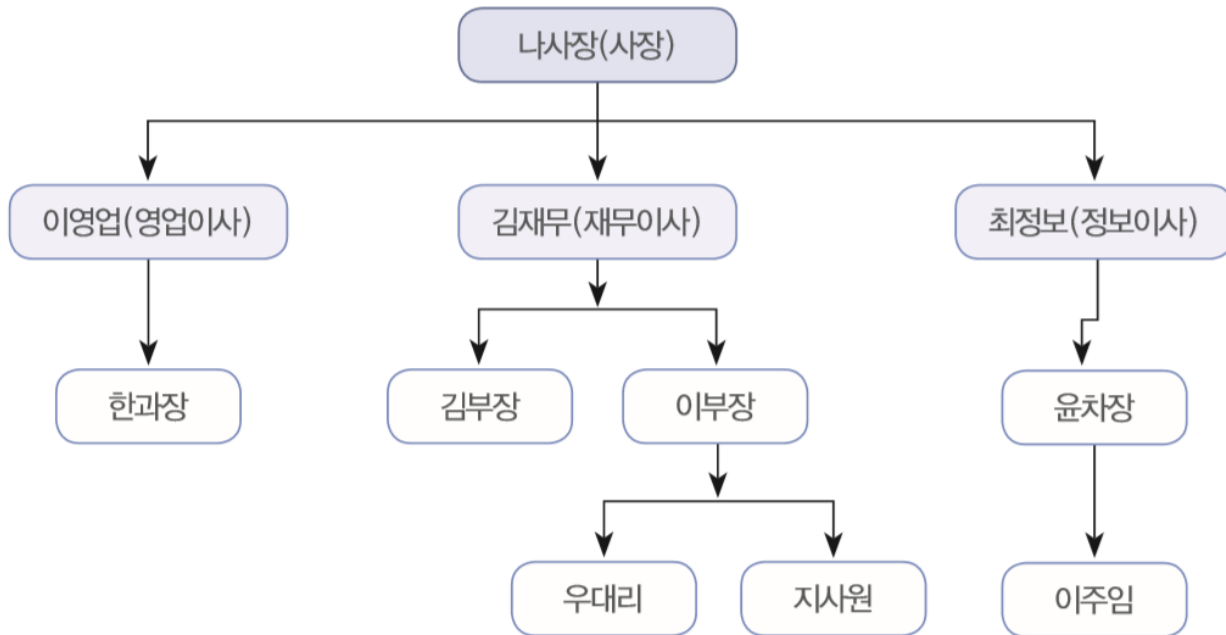
```
SELECT COUNT(*) AS '데이터개수'  
FROM employees  
CROSS JOIN titles;
```

	데이터개수
▶	133003039392

✓ SELF JOIN(자체 조인) ✓ 아주 가끔 쓰이는 조인

○ 자기 자신과 자기 자신이 조인한다는 의미

- 대표적인 예
 - 조직도와 관련된 테이블



테이블이 다른 정보를 필요로 하는데
하나 밖에 없고 해당 테이블(본인)이
필요할때
셀프 조인 ㄱㄱ

직원 이름(EMP) - 기본 키	상관 이름(MANAGER)	구내 번호
나사장	없음 (NULL)	0000
김재무	나사장	2222
김부장	김재무	2222-1
이부장	김재무	2222-2
우대리	이부장	2222-2-1
지사원	이부장	2222-2-2
이영업	나사장	1111
한과장	이영업	1111-1
최정보	나사장	3333
윤차장	최정보	3333-1
이주임	윤차장	3333-1-1

✓ SELF JOIN(자체 조인)

- 하나의 테이블을 별칭을 이용해 2개의 테이블로 처리

USE sqlldb;

```
CREATE TABLE empTbl(emp CHAR(3), manager CHAR(3), empTel VARCHAR(8));
INSERT INTO empTbl VALUES('나사장', NULL, '0000');
INSERT INTO empTbl VALUES('김재무', '나사장', '2222');
INSERT INTO empTbl VALUES('김부장', '김재무', '2222-1');
INSERT INTO empTbl VALUES('이부장', '김재무', '2222-2');
INSERT INTO empTbl VALUES('우대리', '이부장', '2222-2-1');
INSERT INTO empTbl VALUES('지사원', '이부장', '2222-2-2');
INSERT INTO empTbl VALUES('이영업', '나사장', '1111');
INSERT INTO empTbl VALUES('한과장', '이영업', '1111-1');
INSERT INTO empTbl VALUES('최정보', '나사장', '5355');
INSERT INTO empTbl VALUES('윤차장', '최정보', '3355-1');
INSERT INTO empTbl VALUES('이주임', '윤차장', '5335-1-1');
```

```
SELECT * FROM empTbl;
```

✓ SELF JOIN(자체 조인)

○ 우대리 상관의 연락처 확인

```
SELECT A.emp AS '부하직원', B.emp AS '직속상관', B.empTel AS '직속상관연락처'
FROM empTbl A
  INNER JOIN empTbl B
    ON A.manager = B.emp
WHERE A.emp = '우대리';
```

물리적으로는 하나지만
별칭을 2개이상 부여하여
2개의 테이블처럼 처리하는 조인
=> 셀프 조인

	부하직원	직속상관	직속상관연락처
▶	우대리	이부장	2222-2

```
SELECT u.userID, u.name, GROUP_CONCAT(b.prodName SEPARATOR ',') AS '상품목록'
FROM usertbl u
  LEFT OUTER JOIN buytbl b
    ON u.userID = b.userID
GROUP BY u.userID, u.name
ORDER BY u.userID;
```

	userID	name	상품목록
▶	BBK	바베킨	모니터, 메모리, 운동화, 운동화
	EJW	은지원	책, 청바지, 책
	JKW	조관우	NULL
	JYP	조용필	모니터
	KBS	김범수	운동화, 노트북, 청바지
	KKH	김강호	NULL
	LJB	임재범	NULL
	LSG	이승기	NULL
	SSK	성시경	책
	YJS	윤종신	NULL

만약 이너조인했다면

상품 목록이 null인 행들은 없음

✓ UNION / UNION ALL / NOT IN / IN

- 두 쿼리의 결과를 행으로 합치는 것

```
SELECT 문장1
      UNION [ALL]
SELECT 문장2
```

컬럼명이 다른디?

SELECT stdName, addr FROM stdTbl

stdName	addr
김범수	경남
성시성	서울
조용필	경기
은지원	경북
바비킴	서울

SELECT clubName, roomNo FROM clubTbl

clubName	roomNo
수영	101호
바둑	102호
축구	103호
봉사	104호

앞에 쿼리 결과의 컬럼명이 남음

UNION ALL

stdName	addr
김범수	경남
성시성	서울
조용필	경기
은지원	경북
바비킴	서울
수영	101호
바둑	102호
축구	103호
봉사	104호

✓ UNION ALL

- 중복된 열까지 모두 출력

USE sqlldb;

SELECT stdName, addr FROM stdtbl

UNION ALL

SELECT clubName, roomNo FROM clubtbl;

cf) UNION 만 사용하는 경우

- 중복된 열은 제거됨

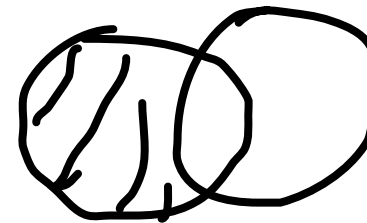
world
employees
sakila
,') AS '상품목록'

조인 연습은 이 순서대로

sakila는 많아서 3개 조인이 할게 많다

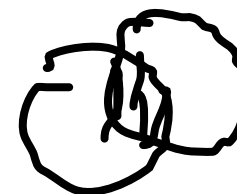
✓ NOT IN 차집합

- 첫 번째 쿼리의 결과 중에서, 두 번째 쿼리에 해당하는 것을 제외
- sqldb의 사용자를 모두 조회하되 전화가 없는 사람은 제외
USE sqldb;



```
SELECT name, CONCAT(mobile1, mobile2) AS '전화번호' FROM usertbl
WHERE name NOT IN (SELECT name FROM usertbl WHERE mobile1 IS NULL);
```

- 두 번째 쿼리에 해당되는 것만 조회
USE sqldb;



```
SELECT name, CONCAT(mobile1, mobile2) AS '전화번호' FROM usertbl
WHERE name IN (SELECT name FROM usertbl WHERE mobile1 IS NULL);
```

교집합