

---

# SQL 고급

## - 조인 -

# 조인

---

## ❖ 조인(join)

- 두 개 이상의 테이블을 서로 묶어서 하나의 결과 집합으로 만들어 내는 것
- 관계형 테이블의 가장 큰 특징
- 테이블 간의 관계
  - 1:1 관계
  - 1:N 관계
  - N:M 관계

# 조인

---

## ❖ INNER JOIN(내부 조인)

- 가장 많이 사용되는 조인
- 공통 컬럼을 기반으로 결합
  - 부모 테이블 : Primary Key
  - 자식 테이블 : Foreign Key(부모 테이블의 PK 값 참조)

### [형식]

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

# 조인

---

## ❖ INNER JOIN(내부 조인)

```
USE sqlDB;

SELECT *
FROM buyTBL
  INNER JOIN userTBL
    ON buyTBL.userID = userTBL.userID
WHERE buyTBL.userID = 'JYP';
```

# 조인

## ❖ INNER JOIN(내부 조인)

❷ JYP와 동일한 아이디 검색

❶ JYP를 추출

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				

❸ 두 행을  
결합

# 조인

---

## ❖ INNER JOIN(내부 조인)

```
USE sqlDB;
```

```
SELECT *  
FROM buyTBL  
    INNER JOIN userTBL  
    ON buyTBL.userID = userTBL.userID;
```

```
SELECT userID, name, prodName, addr, CONCAT(mobile1, mobile2) AS '연락처'  
FROM buyTBL  
    INNER JOIN userTBL  
    ON buyTBL.userID = userTBL.userID ;
```

```
SELECT buyTBL.userID, name, prodName, addr, CONCAT(mobile1,mobile2) AS '연락처'  
FROM buyTBL  
    INNER JOIN userTBL  
    ON buyTBL.userID = userTBL.userID ;
```

# 조인

## ❖ INNER JOIN(내부 조인)

```
SELECT buyTBL.userID, userTBL.name, buyTBL.prodName, userTBL.addr,  
       CONCAT(userTBL.mobile1, userTBL.mobile2) AS '연락처'  
FROM buyTBL  
     INNER JOIN userTBL  
     ON buyTBL.userID = userTBL.userID;
```

```
SELECT B.userID, U.name, B.prodName, U.addr,  
       CONCAT(U.mobile1, U.mobile2) AS '연락처'  
FROM buyTBL B  
     INNER JOIN userTBL U  
     ON B.userID = U.userID;
```

```
SELECT B.userID, U.name, B.prodName, U.addr,  
       CONCAT(U.mobile1, U.mobile2) AS '연락처'  
FROM buyTBL B  
     INNER JOIN userTBL U  
     ON B.userID = U.userID  
WHERE B.userID = 'JYP';
```

# 조인

---

## ❖ INNER JOIN(내부 조인)

```
SELECT B.userID, U.name, B.prodName, U.addr,  
       CONCAT(U.mobile1, U.mobile2) AS '연락처'  
FROM buyTBL B  
     INNER JOIN userTBL U  
     ON B.userID = U.userID  
WHERE B.userID = 'JYP'  
ORDER BY U.userID;
```

→ 전체 회원이 다 나오는가?  
구매 이력이 있는 회원만 출력됨



# 조인

---

## ❖ INNER JOIN(내부 조인)

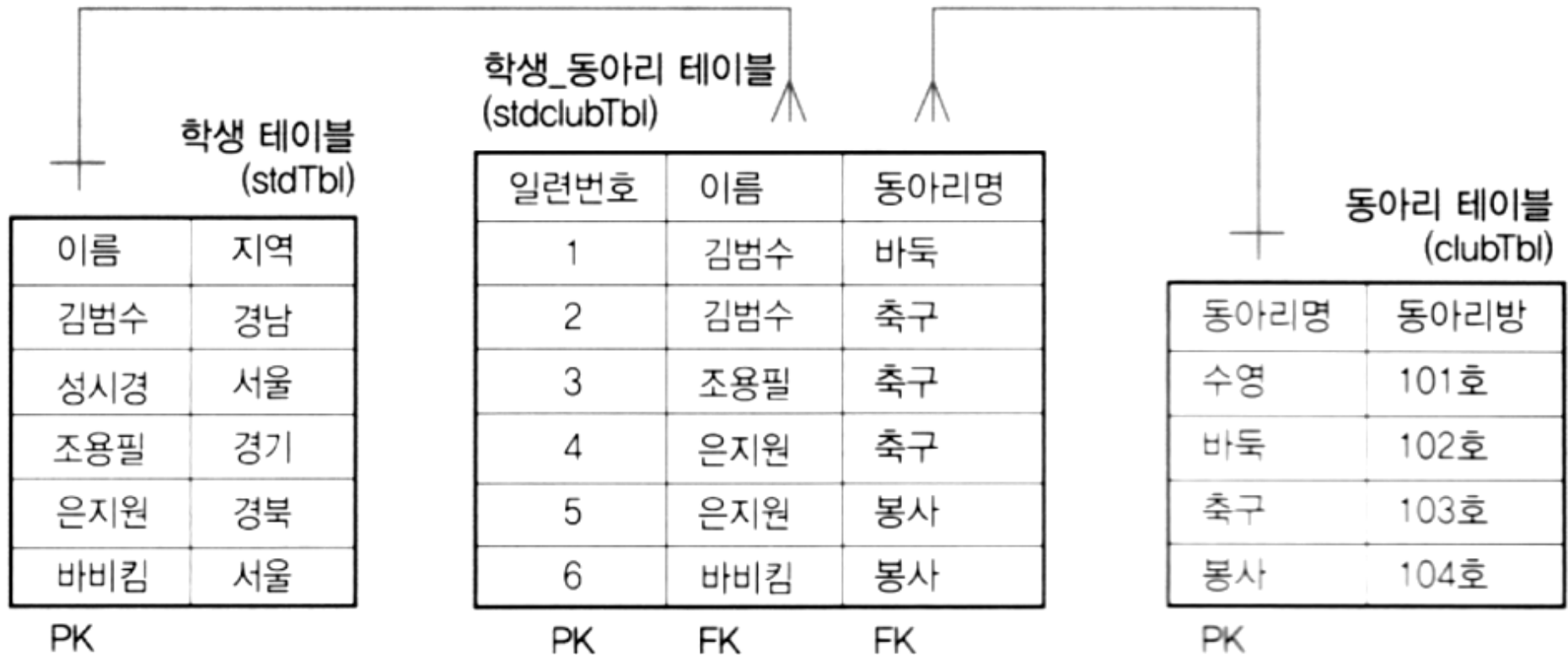
```
SELECT DISTINCT U.userID, U.name, U.addr  
FROM userTBL U  
     INNER JOIN buyTBL B  
     ON U.userID = B.userID  
ORDER BY U.userID ;
```

```
SELECT U.userID, U.name, U.addr  
FROM userTBL U  
WHERE EXISTS (  
    SELECT *  
    FROM buyTBL B  
    WHERE U.userID = B.userID );
```

# 조인

## ❖ INNER JOIN(내부 조인)

- 3개의 테이블 조인



# 조인

---

## ❖ 테스트 데이터

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

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

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)
);
```

# 조인

---

## ❖ 테스트 데이터

```
INSERT INTO stdTBL VALUES  
( '김범수', '경남' ), ( '성시경', '서울' ), ( '조용필', '경기' ),  
( '은지원', '경북' ), ( '바비킴', '서울' );
```

```
INSERT INTO clubTBL VALUES  
( '수영', '101호' ), ( '바둑', '102호' ), ( '축구', '103호' ),  
( '봉사', '104호' );
```

```
INSERT INTO stdclubTBL VALUES  
( NULL, '김범수', '바둑' ), ( NULL, '김범수', '축구' ),  
( NULL, '조용필', '축구' ), ( NULL, '은지원', '축구' ),  
( NULL, '은지원', '봉사' ), ( NULL, '바비킴', '봉사' );
```

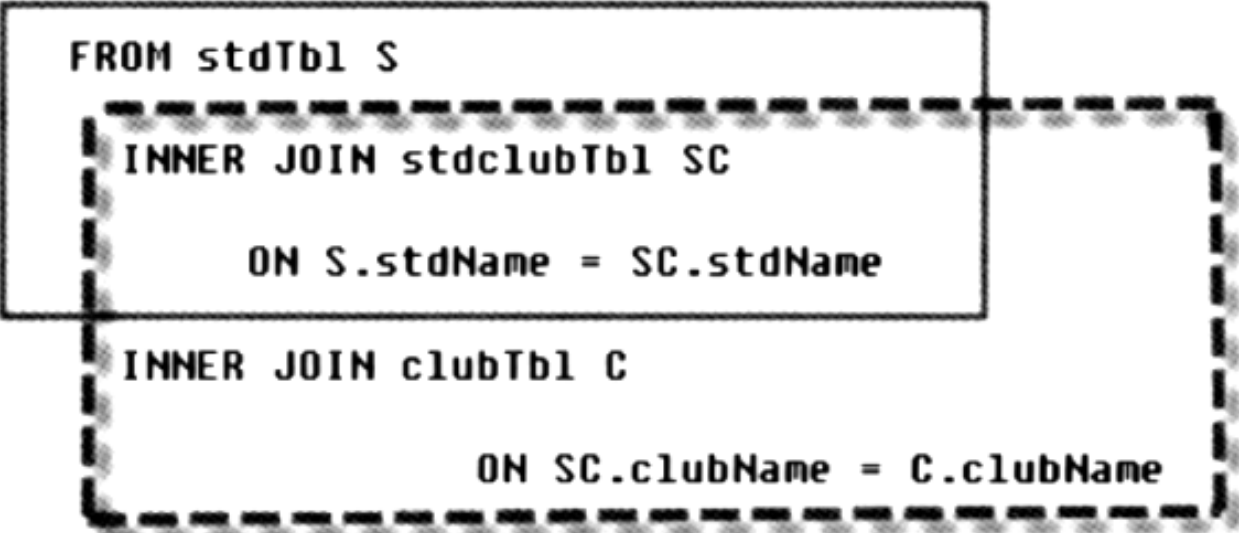
# 조인

## ❖ INNER JOIN(내부 조인)

- 3개의 테이블 조인

```
SELECT S.stdName, S.addr, C.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;
```

```
SELECT S.stdName, S.addr, C.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
```



# 조인

## ❖ OUTER JOIN(외부조인)

- 내부 조인은 값이 있는 행 만을 대상으로 함
- OUTER JOIN은 모든 행에 대해 조인을 함

[형식]

SELECT <열 목록>

FROM <첫 번째 테이블(LEFT 테이블)>

**<LEFT | RIGHT | FULL> OUTER JOIN** <두 번째 테이블(RIGHT 테이블)>

ON <조인될 조건>

[WHERE 검색 조건]

- LEFT JOIN : 왼쪽 테이블의 모든 행을 기준으로 조인
- RIGHT JOIN : 오른쪽 테이블의 모든 행을 기준으로 조인
- FULL : 두 테이블의 모든 행에 대해 조인

# 조인

---

## ❖ OUTER JOIN(외부조인)

```
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;
```

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

# 조인

---

## ❖ OUTER JOIN(외부조인)

```
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  
WHERE B.prodName IS NULL  
ORDER BY U.userID;
```



# 조인

## ❖ CROSS JOIN(상호 조인)

- 두 테이블의 가능한 모든 조합을 구성

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

```
USE sqlDB;
```

```
SELECT *
```

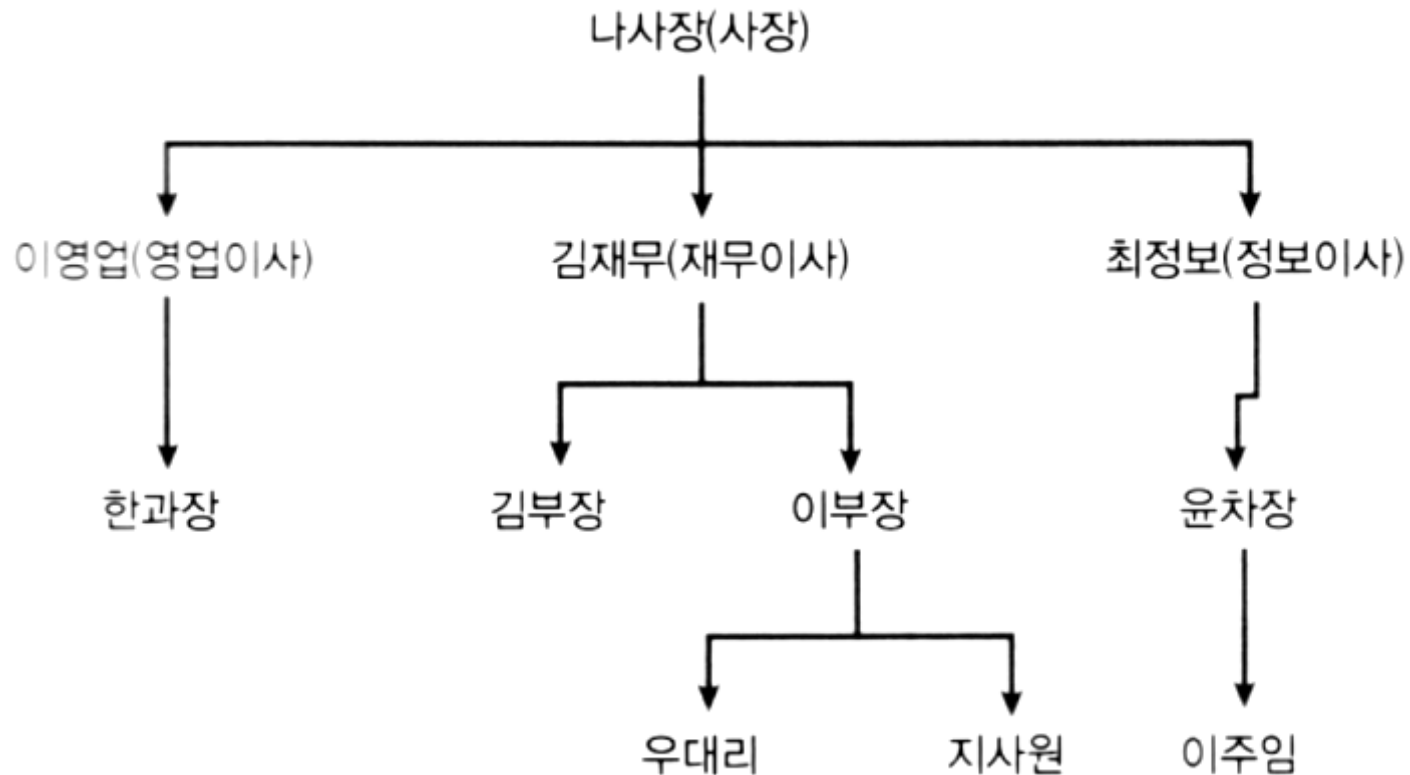
```
FROM buyTBL
```

```
CROSS JOIN userTBL ;
```

# 조인

## ❖ SELF JOIN(자체 조인)

- 한 개의 테이블에 대해 자체 조인을 수행하는 것
- 한 개의 테이블을 두 개의 테이블로 간주
  - Alias를 두개 배정하여 구분
- 대표적인 예
  - 조직도



# 조인

## ❖ SELF JOIN(자체 조인)

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

# 조인

---

## ❖ SELF JOIN(자체 조인)

```
USE sqlDB;
```

```
CREATE TABLE empTbl (emp CHAR(3), manager CHAR(3), empTel VARCHAR(8));
```

```
INSERT INTO empTbl VALUES  
( '나사장' ,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' );
```

# 조인

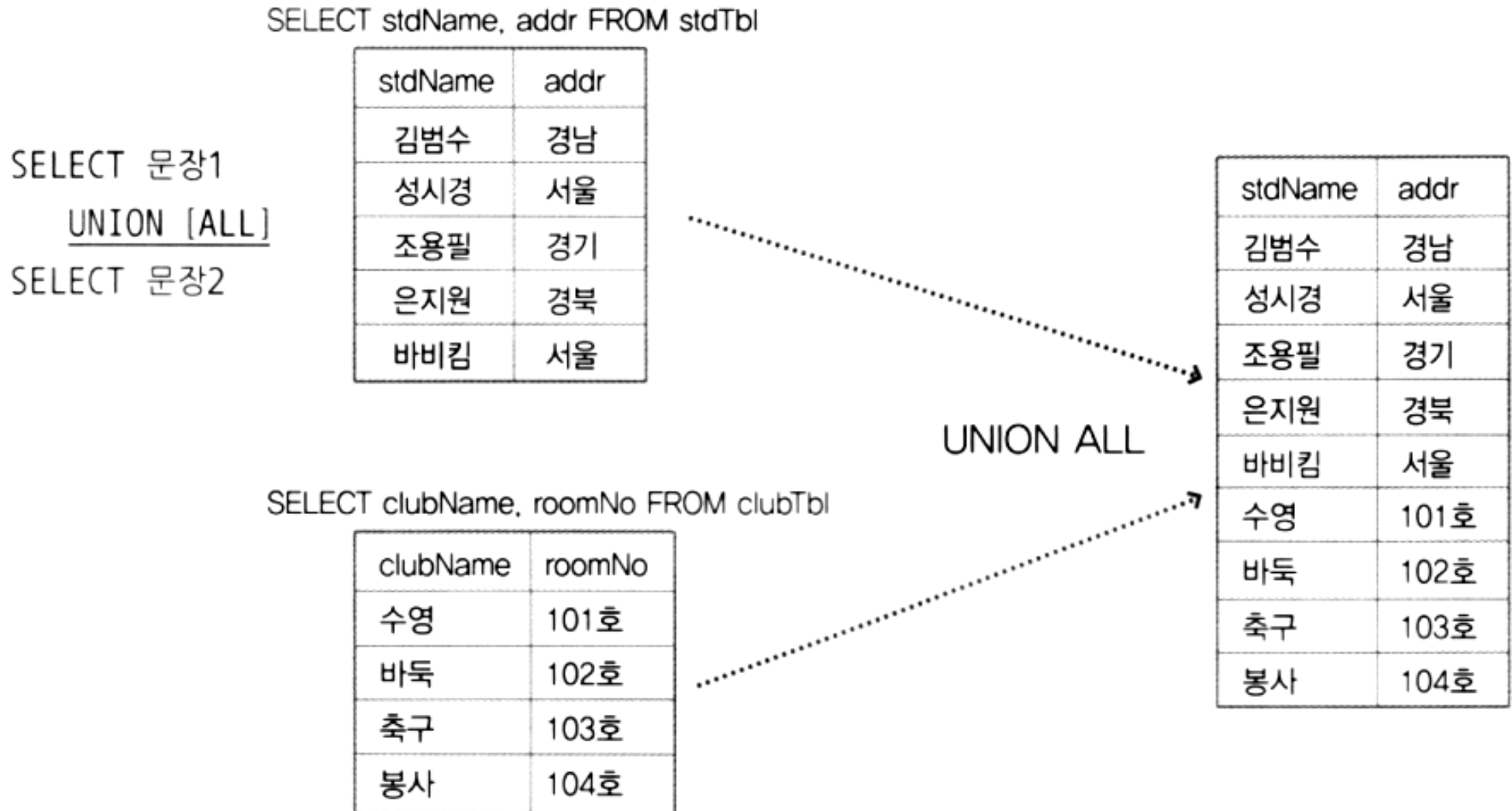
---

## ❖ SELF JOIN(자체 조인)

```
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';
```

# 조인

## ❖ UNION / UNION ALL / NOT IN / IN



# 조인

## ❖ UNION / UNION ALL / NOT IN / IN

```
USE sqlDB;  
SELECT stdName, addr FROM stdTBL  
UNION ALL  
SELECT clubName, roomNo FROM clubTBL;
```

→ UNION : 중복 제거, UNION ALL : 중복 모두 포함

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

→ 차 집합

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

→ 교 집합