

제6장 조인(join)



4. 외부 조인(outer join)

- 외부 조인(outer join)
 - 조인 조건을 만족하지 않는 행까지 포함하여 출력하는 조인
- 외부 조인의 형식

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

4. 외부 조인(outer join)

- 전체 회원의 구매 기록을 출력하되 구매 기록이 없는 회원도 출력

```
USE sqldb;  
SELECT U.userID, U.userName, 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;
```

	userID	userName	prodName	addr	연락처
▶	KHD	강호동	운동화	경북	01122222222
	KHD	강호동	노트북	경북	01122222222
	KHD	강호동	청바지	경북	01122222222
	KJD	김제동	책	경남	NULL
	KKJ	김국진	NULL	서울	01933333333
	KYM	김용만	모니터	서울	01044444444
	LHJ	이현희	책	경기	010-8888888
	PL	박수홍	청바지	경기	01000000000
	PSH	박수홍	운동화	서울	01000000000
	SDY	신동엽	NULL	경기	NULL
	YJS	유재석	NULL	서울	01011111111

4. 외부 조인(outer join)

- 동일한 결과를 출력 하되 쿼리문에 RIGHT OUTER JOIN 사용

```
SELECT U.userID, U.userName, 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;
```

- 물건을 한 번도 구매한 적이 없는 회원의 목록 출력

```
SELECT U.userID, U.userName, 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;
```

	userID	userName	prodName	addr	연락처
▶	KKJ	김국진	NULL	서울	01933333333
	LKK	이경규	NULL	경남	01899999999
	NHS	남희석	NULL	충남	01666666666
	SDY	신동엽	NULL	경기	NULL
	YJS	유재석	NULL	서울	01011111111

6. 상호 조인(cross join)

- 상호 조인(cross join)
 - 한쪽 테이블의 모든 행과 다른 쪽 테이블의 모든 행을 조인하는 것
 - 상호 조인 결과 테이블의 행수는 두 테이블의 행수를 곱한 값
 - 상호 조인은 카티션곱(cartesian product)이라고도 함

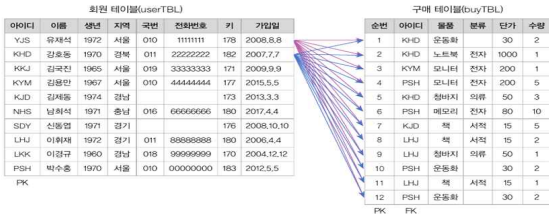


그림 8-18 상호 조인 방식

6. 상호 조인(cross join)

- 회원 테이블과 구매 테이블을 상호 조인

```
USE sqldb;  
SELECT *  
FROM buyTBL  
CROSS JOIN userTBL;
```

- COUNT(*) 함수로 상호 조인한 결과의 개수만 알아본 예

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

	데이터개수
▶	133003039392

7. 셀프 조인(self join)

- 자체 조인(self join)
 - 자기 자신과 자기 자신을 조인하는 것
 - 자체 조인을 활용하는 대표적인 예는 조직도와 관련된 테이블



그림 8-20 간단한 조직도의 예

7. 셀프 조인(self join)

- 자체 조인(self join)

표 8-1 조직도 테이블

직원 이름(emp): 기본키	상관 이름(manager)	구내 번호(empTel)
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

8. 셀프 조인(self join)-실습

1 테이블 만들기

1-1 조직도 테이블 만들기

```
USE sqldb;  
CREATE TABLE empTBL(emp CHAR(3), manager CHAR(3), empTel VARCHAR(8));
```

1-2 조직도 테이블에 데이터 입력

```
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', '3333');  
INSERT INTO empTBL VALUES ('윤차장', '최정보', '3333-1');  
INSERT INTO empTBL VALUES ('이주임', '윤차장', '3333-1-1');
```

8. 셀프 조인(self join)-실습

2 자체 조인하기

2-1 우대리 상관의 구내 번호 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';
```

	부하직원	직속상관	직속상관연락처
▶	우대리	이부장	2222-2

하나의 테이블을 찢어서 두 개로 분리한다고 생각하고, 조건을 설정할 때 manager필드와 emp필드가 같고 where절을 우대리로 주면 될 것이다.

9. UNION과 UNIONALL

■ UNION

- 두 쿼리의 결과를 행으로 합치는 연산자

SELECT stdName, addr FROM stdTBL

stdName	addr
강호동	경북
김제동	경남
김용만	서울
이휘재	경기
박수홍	서울

SELECT clubName, roomNo FROM clubTBL

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

union은 두 쿼리문의 결과를 출력하는데 중복된 것을 제외를 한다.
하지만, union all은 두 쿼리문의 결과를 출력하는 것은 똑같지만 중복된
것도 다 출력된다는 것이 차이점이다.

UNION ALL

stdName	addr
강호동	경북
김제동	경남
김용만	서울
이휘재	경기
박수홍	서울
수영	101호
바둑	102호
축구	103호
봉사	104호

그림 8-22 UNION의 결합 과정

9. UNION과 UNIONALL

- UNION 연산자를 사용하는 형식

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

- 중복된 열까지 모두 출력하려면 UNION ALL 사용

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

	stdName	addr
▶	강호동	경북
	김용만	서울
	김제동	경남
	박수홍	서울
	이회재	경기
	바둑	102호
	봉사	104호
	수영	101호
	축구	103호

10. NOT IN과 IN

- NOT IN

- 첫 번째 쿼리의 결과 중에서 두 번째 쿼리에 해당하는 것을 제외하고 출력

- sqldb의 사용자를 모두 출력하되 전화번호가 없는 사람을 제외

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

	userName	전화번호
▶	강호동	011-22222222
	김국진	019-33333333
	김용만	010-44444444
	이희재	011-88888888
	이경규	018-99999999
	남궁석	016-66666666
	박수홍	010-00000000
	유재석	010-11111111

- 전화번호가 없는 사람만 조회

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

	userName	전화번호
	김제동	NULL
	신동엽	NULL

감사합니다.

