

제6장 조인(join)



1. 조인(join)이란?

- 조인(JOIN)

- 2개 이상의 테이블을 묶어서 하나의 결과 테이블을 만드는 것

- sqlldb의 구조

데이터베이스의 테이블

1. 여러 개의 테이블로 분리하여 저장

- 중복과 공간낭비를 피하고 데이터의 무결성을 위함

2. 분리된 테이블은 서로 관계(Relation)를 가짐

- 1대 다 관계에서 일어나는 데이터 처리 필요



1. 조인(join)이란?

- 일대다 관계
 - sqldb의 회원 테이블(userTBL)과 구매 테이블(buyTBL)이 맺고 있는 관계
 - 한쪽 테이블에는 하나의 값만 존재하고 그 값과 대응되는 다른 쪽 테이블의 값은 여러 개인 관계
- 일대다 관계 예
 - 기업의 직원 테이블과 급여 테이블, 학교의 학생 테이블과 학점 테이블

2. 내부 조인(inner join)

- 내부 조인의 형식

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

inner join을 통상 조인이라 칭한다.
조인은 2개 이상의 테이블을 결합하여 쿼리하여
결과를 출력하는 sql고급 부분에 속한다.

- KYM이라는 아이디를 가진 회원이 구매한 물건을 발송하려면?

```
USE sqlldb;  
SELECT *  
  FROM buyTBL  
  INNER JOIN userTBL  
    ON buyTBL.userID = userTBL.userID  
 WHERE buyTBL.userID = 'KYM';
```

	num	userID	prodName	groupName	price	amount	userID	userName	birthYear	addr	mobile1	mobile2	height	mDate
	3	KYM	모니터	전자	200	1	KYM	김용만	1967	서울	010	44444444	177	2015-05-05

2. 내부 조인(inner join)

■ 내부 조인의 작동

- 구매 테이블의 회원 아이디(buyTBL.userID)인 KYM을 추출
- KYM과 동일한 값을 회원 테이블의 아이디 (userTBL.userID) 열에서 검색
- 아이디 KYM을 찾으면 구매 테이블과 회원 테이블의 두 행을 결합(조인)



그림 8-3 내부 조인의 작동

2. 내부 조인(inner join)

- 만약 WHERE buyTBL.userID = 'KYM'을 생략하면?

	num	userID	prodName	groupName	price	amount	userID	userName	birthYear	addr	mobile1	mobile2	height	mDate
▶	1	KHD	운동화	신발	30	2	KHD	강포동	1970	경북	011	22222222	182	2007-07-07
	2	KHD	노트북	전자	1000	1	KHD	강포동	1970	경북	011	22222222	182	2007-07-07
	5	KHD	정바지	의류	50	3	KHD	강포동	1970	경북	011	22222222	182	2007-07-07
	7	KJD	책	서적	15	5	KJD	김재동	1974	경남	010	00000000	173	2013-03-03
	3	KYM	모니터	전자	200	1	KYM	김승만	1967	서울	010	44444444	177	2015-05-05
	8	LHJ	책	서적	15	2	LHJ	이희재	1972	경기	011	88888888	180	2006-04-04
	9	LHJ	정바지	의류	50	1	LHJ	이희재	1972	경기	011	88888888	180	2006-04-04
	11	LHJ	책	서적	15	1	LHJ	이희재	1972	경기	011	88888888	180	2006-04-04
	4	PSH	모니터	전자	200	5	PSH	박수홍	1970	서울	010	00000000	183	2012-05-05
	6	PSH	메모리	전자	80	10	PSH	박수홍	1970	서울	010	00000000	183	2012-05-05
	10	PSH	운동화	신발	30	2	PSH	박수홍	1970	서울	010	00000000	183	2012-05-05
	12	PSH	운동화	신발	30	2	PSH	박수홍	1970	서울	010	00000000	183	2012-05-05

그림 8-4 내부 조인 결과 2

where절을 주지 아니하면 당연히 2개의 테이블 데이터를 다 가져올 것이다.

2. 내부 조인(inner join)

- 아이디, 이름, 구매 물품, 주소, 연락처만 추출

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

실행 결과

알리아스를 주지 아니하면 두 테이블다 userID가 존재하여 필드의 모호성 발생

Error Code: 1052. Column 'userid' in field list is ambiguous

- 어느 테이블의 userID를 추출할지 선택(구매 테이블의 userID라고 명시)

```
SELECT buyTBL.userID, userName,  
prodName, addr, CONCAT(mobile1, mobile2)  
AS '연락처'  
FROM buyTBL  
INNER JOIN userTBL  
ON buyTBL.userID = userTBL.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
	KYM	김용만	모니터	서울	01044444444
	LHJ	이희재	책	경기	01188888888
	LHJ	이희재	청바지	경기	01188888888
	LHJ	이희재	책	경기	01188888888
	PSH	박수홍	모니터	서울	01000000000
	PSH	박수홍	메모리	서울	01000000000
	PSH	박수홍	운동화	서울	01000000000
	PSH	박수홍	운동화	서울	01000000000

2. 내부 조인(inner join)

- SELECT 다음의 열 이름도 모두 '테이블이름.열이름' 형식으로 작성한 예

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

- 각 테이블에 별칭 부여

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


2. 내부 조인(inner join)

- KYM이라는 아이디를 가진 회원이 구매한 물건과 회원 정보 조인(아이디, 이름, 물품, 주소, 연락처만 출력되게 하고 코드도 간결하게 수정)

```
SELECT B.userID, U.userName, 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 = 'KYM';
```

	userID	userName	prodName	addr	연락처
	KYM	김용만	모니터	서울	01044444444

- 회원 테이블(userTBL)을 기준으로 KYM이 구매한 물건의 목록 조회

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

2. 내부 조인(inner join)

- 전체 회원이 구매한 목록 모두 출력(회원 아이디 순으로 정렬)

```
SELECT U.userID, U.userName, B.prodName, U.addr, CONCAT(U.mobile1, U.mobile2) AS '연락처'
FROM userTBL U
     INNER 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	01099999999
	KYM	김용만	모니터	서울	01044444444
	LHJ	이회재	책	경기	01188888888
	LHJ	이회재	청바지	경기	01188888888
	LHJ	이회재	책	경기	01188888888
	PSH	박수홍	모니터	서울	01000000000
	PSH	박수홍	메모리	서울	01000000000
	PSH	박수홍	운동화	서울	01000000000
	PSH	박수홍	운동화	서울	01000000000

2. 내부 조인(inner join)

- 쇼핑몰에서 한 번이라도 구매한 기록이 있는 우수 회원에게 감사의 안내문을 발송하려 할 때 (DISTINCT 키워드 활용)

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

	userID	userName	addr
▶	KHD	강호동	경북
	KJD	김제동	경남
	KYM	김용만	서울
	LHJ	이회재	경기
	PSH	박수홍	서울

- EXISTS 구문을 사용하면 앞과 동일한 결과를 만들 수 있음

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

exists구문은 서브쿼리에 필드가 존재하는지만 확인하여 리턴하여 준다.
즉, boolean값을 리턴하는데 순서는 먼저 첫 번째 select문을 실행하여
그 결과를 토대로 exists구문의 select와 비교하여 일치하는 행이 있다면
리턴하여 출력을 하게 된다.결과론적으로 distinct와 같은 역할을 한다고
보면 될 것이다.

감사합니다.

