

조인

이번 장에서는 새로운 테이블을 만들어 실습한다.

먼저 **노래** 테이블을 만들어보자.

```
CREATE TABLE 노래 (  
  ID INTEGER NOT NULL PRIMARY KEY,  
  제목 TEXT NOT NULL  
);
```

다음으로 **음반** 테이블을 만들고,

```
CREATE TABLE 음반 (  
  ID INTEGER NOT NULL PRIMARY KEY,  
  제목 TEXT NOT NULL,  
  연도 INTEGER  
);
```

음반과 노래의 관계를 표현하는 **수록곡** 테이블도 만들자.

```
CREATE TABLE 수록곡 (  
  음반ID INTEGER NOT NULL,  
  노래ID INTEGER NOT NULL  
);
```

다음 **다이어그램**은 세 테이블의 관계를 나타낸다.



이제 **노래** 테이블에 데이터를 넣어보자.

INSERT INTO 노래 VALUES

```
(1, '가우똥'),
(2, 'Shuppy Shuppy'),
(3, 'Control'),
(4, '영러브'),
(5, '한번만 안아줘'),
(6, '반짝반짝'),
(7, '기대해'),
(8, 'I Don''t Mind'),
(9, 'Easy go'),
(10, '여자대통령');
```

음반 테이블에도 데이터를 넣어보자.

INSERT INTO 음반 VALUES

```
(1, 'Girl''s Day Party #1', 2010),
(2, 'Everyday', 2011),
(3, 'Expectation', 2013),
(4, '여자대통령', 2013);
```

음반과 노래의 관계를 표현하는 수록곡 테이블에 데이터를 넣어보자. **Everyday** 앨범의 **반짝반짝** 이라는 곡은 **Expectation** 앨범에도 수록됐다.

INSERT INTO 수록곡 VALUES

```
(1, 1),
(1, 2),
(1, 3),
(2, 4),
(2, 5),
(2, 6), -- Everyday - 반짝반짝
(3, 7),
(3, 8),
(3, 9),
(3, 6), -- Expectation - 반짝반짝
(3, 5),
(4, 10);
```

데이터를 조회해보자.

```
SELECT 음반.제목 앨범명, 음반.연도 발매년도, 노래.제목 곡명
FROM 수록곡
INNER JOIN 음반 ON 수록곡.음반ID = 음반.ID
INNER JOIN 노래 ON 수록곡.노래ID = 노래.ID
```

위의 SQL 문은 **INNER JOIN** 구문을 명시적으로 사용했다. 다음 문장은 조인 구문을 명시하지 않았지만 위의 것과 같은 결과를 얻을 수 있다.

```
SELECT 음반.제목 앨범명, 음반.연도 발매년도, 노래.제목 곡명
FROM 노래, 음반, 수록곡
WHERE 음반.ID = 수록곡.음반ID AND 노래.ID = 수록곡.노래ID;
```

수행 결과는 다음과 같다.

앨범명	발매년도	곡명
Girl's Day Party #1	2010	가우똥
Girl's Day Party #1	2010	Shuppy Shuppy
Girl's Day Party #1	2010	Control
Everyday	2011	영러브
Everyday	2011	한번만 안아줘
Everyday	2011	반짝반짝
Expectation	2013	기대해
Expectation	2013	I Don't Mind
Expectation	2013	Easy go
Expectation	2013	반짝반짝
Expectation	2013	한번만 안아줘
여자대통령	2013	여자대통령

2011년에 발매된 앨범만 조회하려면 WHERE 절을 추가하면 된다.

```
SELECT 음반.제목 앨범명, 음반.연도 발매년도, 노래.제목 곡명
FROM 수록곡
INNER JOIN 음반 ON 수록곡.음반ID = 음반.ID
INNER JOIN 노래 ON 수록곡.노래ID = 노래.ID
WHERE 음반.연도 = 2011;
```

결과는 다음과 같다.

앨범명	발매년도	곡명
Everyday	2011	영러브
Everyday	2011	한번만 안아줘
Everyday	2011	반짝반짝