[MySQL] 조인 : JOIN (INNER, LEFT, RIGHT)

조인은 관계형 데이터 베이스에서 일반적으로 많이 사용하며, 충분한 이해 없이 사용하기 쉬운 내용이다.

이번 시간에는 여러 조인의 종류가 그 사용하는 예에 대해서 공부해보자.

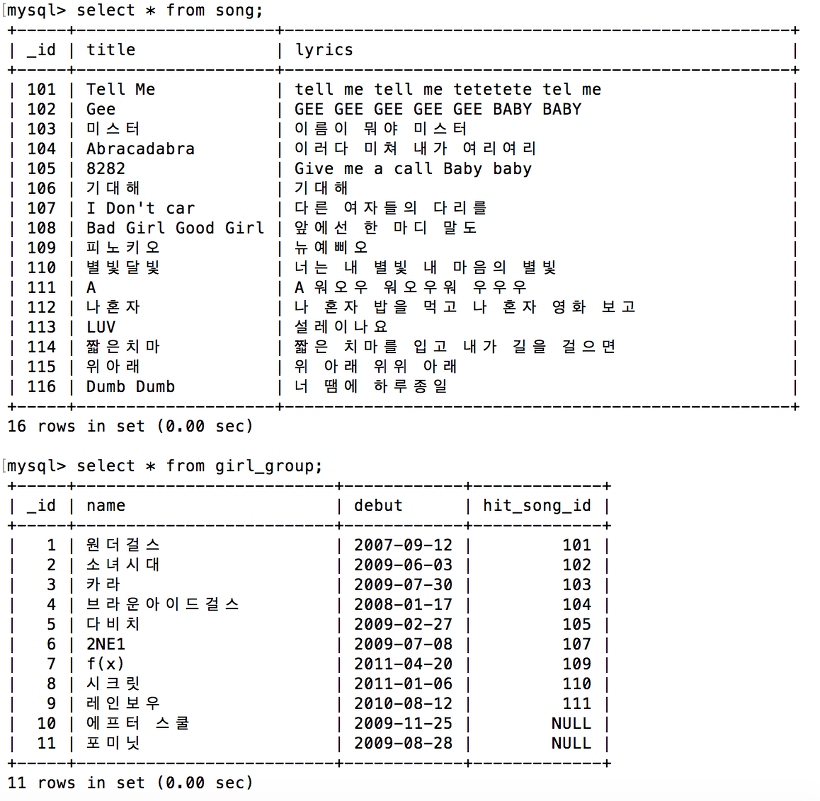
1. INNER 조인
2. LEFT OUTER, RIGHT OUTER, OUTER 조인
3. 카티전 조인 (CROSS 조인)
4. 셀프 조인

우선 JOIN을 할 두 테이블을 생성하자.

|  |  |
| --- | --- |
|  | CREATE TABLE girl\_group |
|  | ( |
|  | \_id INT PRIMARY KEY AUTO\_INCREMENT, |
|  | name VARCHAR(32) NOT NULL, |
|  | debut DATE NOT NULL, |
|  | hit\_song\_id INT |
|  | ); |
|  |  |
|  | CREATE TABLE song |
|  | ( |
|  | \_id INT PRIMARY KEY AUTO\_INCREMENT, |
|  | title VARCHAR(32) NOT NULL, |
|  | lyrics VARCHAR(32) |
|  | ); |
|  | INSERT INTO song (\_id, title, lyrics) VALUES (101, 'Tell Me', 'tell me tell me tetetete tel me'); |
|  | INSERT INTO song (title, lyrics) VALUES ('Gee', 'GEE GEE GEE GEE GEE BABY BABY'); |
|  | INSERT INTO song (title, lyrics) VALUES ('미스터', '이름이 뭐야 미스터'); |
|  | INSERT INTO song (title, lyrics) VALUES ('Abracadabra', '이러다 미쳐 내가 여리여리'); |
|  | INSERT INTO song (title, lyrics) VALUES ('8282', 'Give me a call Baby baby'); |
|  | INSERT INTO song (title, lyrics) VALUES ('기대해', '기대해'); |
|  | INSERT INTO song (title, lyrics) VALUES ('I Don\'t car', '다른 여자들의 다리를'); |
|  | INSERT INTO song (title, lyrics) VALUES ('Bad Girl Good Girl', '앞에선 한 마디 말도'); |
|  | INSERT INTO song (title, lyrics) VALUES ('피노키오', '뉴예삐오'); |
|  | INSERT INTO song (title, lyrics) VALUES ('별빛달빛', '너는 내 별빛 내 마음의 별빛'); |
|  | INSERT INTO song (title, lyrics) VALUES ('A', 'A 워오우 워오우워 우우우'); |
|  | INSERT INTO song (title, lyrics) VALUES ('나혼자', '나 혼자 밥을 먹고 나 혼자 영화 보고'); |
|  | INSERT INTO song (title, lyrics) VALUES ('LUV', '설레이나요 '); |
|  | INSERT INTO song (title, lyrics) VALUES ('짧은치마', '짧은 치마를 입고 내가 길을 걸으면'); |
|  | INSERT INTO song (title, lyrics) VALUES ('위아래', '위 아래 위위 아래'); |
|  | INSERT INTO song (title, lyrics) VALUES ('Dumb Dumb' , '너 땜에 하루종일'); |
|  |  |
|  |  |
|  | INSERT INTO girl\_group (name, debut, hit\_song\_id) |
|  | VALUES ('원더걸스', '2007-09-12', 101); |
|  | INSERT INTO girl\_group (name, debut, hit\_song\_id) |
|  | VALUES ('소녀시대', '2009-06-03', 102); |
|  | INSERT INTO girl\_group (name, debut, hit\_song\_id) |
|  | VALUES ('카라', '2009-07-30', 103); |
|  | INSERT INTO girl\_group (name, debut, hit\_song\_id) |
|  | VALUES ('브라운아이드걸스', '2008-01-17', 104); |
|  | INSERT INTO girl\_group (name, debut, hit\_song\_id) |
|  | VALUES ('다비치', '2009-02-27', 105); |
|  | INSERT INTO girl\_group (name, debut, hit\_song\_id) |
|  | VALUES ('2NE1', '2009-07-08', 107); |
|  | INSERT INTO girl\_group (name, debut, hit\_song\_id) |
|  | VALUES ('f(x)', '2011-04-20', 109); |
|  | INSERT INTO girl\_group (name, debut, hit\_song\_id) |
|  | VALUES ('시크릿', '2011-01-06', 110); |
|  | INSERT INTO girl\_group (name, debut, hit\_song\_id) |
|  | VALUES ('레인보우', '2010-08-12', 111); |
|  | INSERT INTO girl\_group (name, debut) |
|  | VALUES ('에프터 스쿨', '2009-11-25'); |
|  | INSERT INTO girl\_group (name, debut) |
|  | VALUES ('포미닛', '2009-08-28'); |

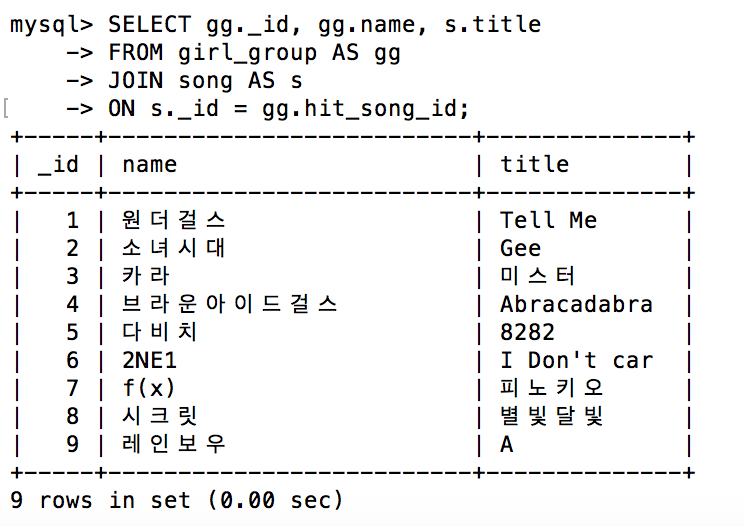
[view raw](https://gist.github.com/tmmoond8/fc8ba6f4d046aec117968bb80e97a294/raw/7777ccd0c42a0134d6ac693f7dc305a0797c004b/mysql_36.sql)[mysql\_36.sql](https://gist.github.com/tmmoond8/fc8ba6f4d046aec117968bb80e97a294#file-mysql_36-sql)hosted with ❤ by [GitHub](https://github.com/)

아래 사진과 같이 테이블이 만들어 진다.



**1. INNER 조인**

INNER 조인은 우리가 생각하는 일반적인 용도에 사용한다.



|  |  |
| --- | --- |
|  | SELECT gg.\_id, gg.name, s.title |
|  | FROM girl\_group AS gg |
|  | JOIN song AS s |
|  | ON s.\_id = gg.hit\_song\_id; |

[view raw](https://gist.github.com/tmmoond8/19b38c368566382e170eef83bafa6fcb/raw/4693fd562552a7b14745080de6e20f2bc82ac34a/mysql_37.sql)[mysql\_37.sql](https://gist.github.com/tmmoond8/19b38c368566382e170eef83bafa6fcb#file-mysql_37-sql)hosted with ❤ by [GitHub](https://github.com/)

\* ON 대신 WHERE를 쓸 수 있다.

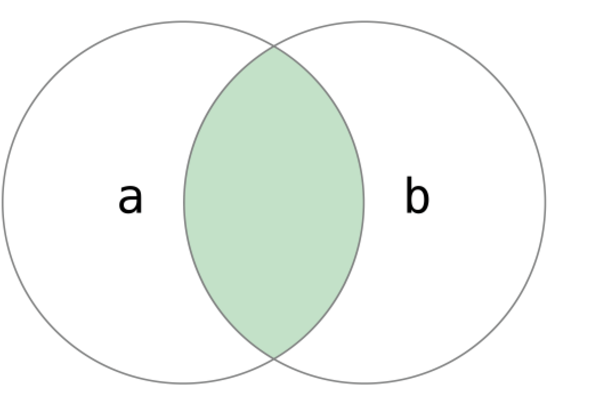
INNER 조인은 MySQL에서는 간략히 JOIN으로 나타낸다. 일반적으로 사용하는 JOIN 이다.

 핵심은 JOIN뒤에 ON인데, 두 테이블이 결합하는 조건을 나타낸다.

두 테이블 모두 노래의 \_id를 가지고 있으며, 서로 모두 포함하는 레코드를 합쳐서 표현한다.

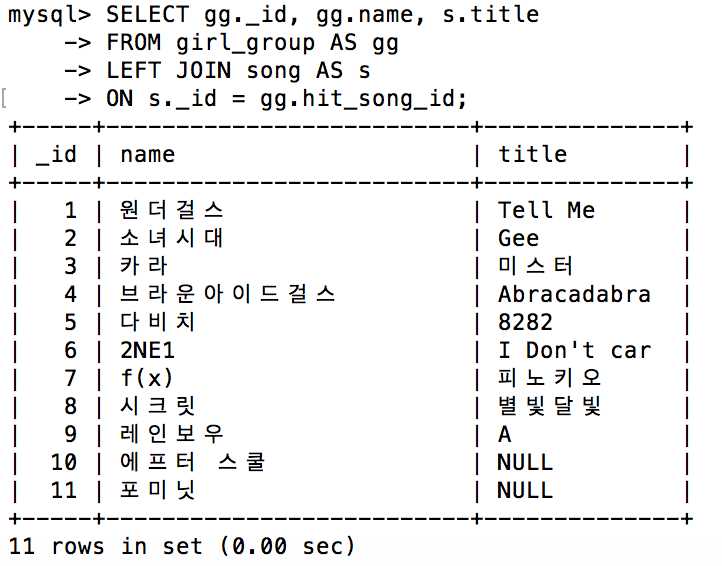
 '위 아래' 라는 곡의 id 는 115인데 이 값은 girl\_group 테이블에 존재하지 않기 때문에 나오지 않는다.

집합으로 표현하자면 아래와 같다.



**2. LEFT OUTER, RIGHT OUTER 조인**

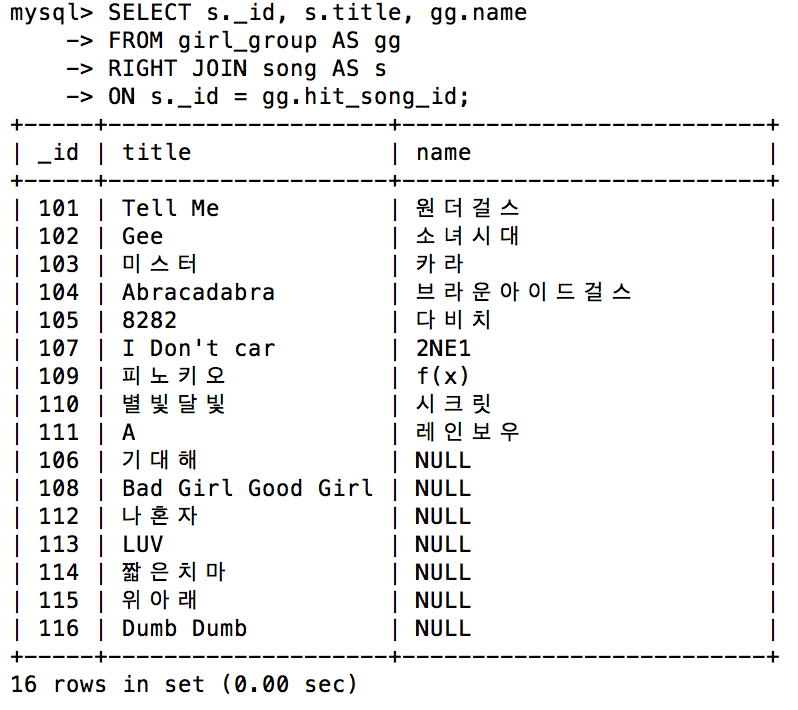
히트 곡이 저장되어 있지 않은 에프터 스쿨, 포미닛의 경우, 곡이 표시 되지 않더라도 보이고 싶을 때는 하나의 테이블 기준으로 합치는 조인을 사용할 수 있다.



위 사진과 같이 일치 하지 않는 값을 가지고 있더라도 표시할 수 있게 된다.

|  |  |
| --- | --- |
|  | SELECT gg.\_id, gg.name, s.title |
|  | FROM girl\_group AS gg |
|  | LEFT OUTER JOIN song AS s |
|  | ON s.\_id = gg.hit\_song\_id; |
|  |  |
|  | SELECT s.\_id, s.title, gg.name |
|  | FROM girl\_group AS gg |
|  | RIGHT OUTER JOIN song AS s |
|  | ON s.\_id = gg.hit\_song\_id; |

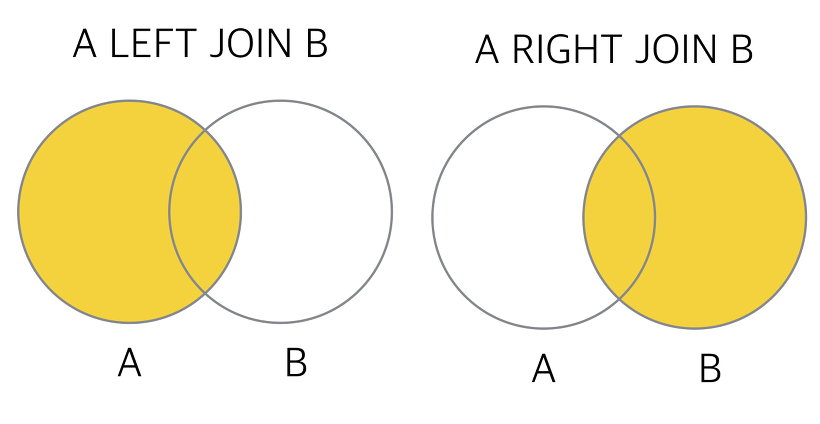
두 번째 쿼리는 song table을 기준으로 조인을 하였다.



\* A LEFT JOIN B 와 B RIGHT JOIN A는 완전히 같은 식이다.

\* LEFT OUTER 대신 LEFT, RIGHT OUTER 대신 RIGHT만 입력해도 같은 기능을 수행한다.

LEFT 조인 과 RIGHT 조인을 집합으로 표현하면



MySQL에서는 OUTER JOIN을 지원하지 않지만, 유사한 처리를 할 수 있다. 아래는 OUTER JOIN의 범위다.

