

# sql 독학 강의 # Cartesian Product 카티션 곱 ansi SQL 문법 12편 - sTricky

sTricky 2020. 4. 20. 12:32

## sql 독학 강의 # Cartesian Product 카티션 곱 ansi SQL 문법 12편 - sTricky



### [컨텐츠 index](#)

0. Cartesian Product, 카티션곱 의 정의

1. 카티션곱 활용방법

SQL 독학 강의 오늘은 join의 기법 중 하나인 **카티션 곱**에 관한 내용을 다루겠습니다.

앞선 시간에 소개해 드린 join의 기법을 각각 한편씩 나누어서 포스팅하도록 하겠습니다.

**mysql SQL문법**과 함께 **Ansi SQL 문법**도 같이 소개해 드리도록 하겠습니다.

Ansi SQL이란 앞에서도 안내를 한번 드리긴 했는데, 간단하게 다시 한번 말씀드리자면, 각각의 RDBMS가 서로 조금씩 다른 SQL문법을 사용하는데, 모든 RDBMS에서 사용될 수 있는 공통 문법이라고 생각하시면 쉽게 이해하실 수 있습니다.

바로 시작 하겠습니다.

오늘도 파이팅!!!!

#전편 보러 가기#

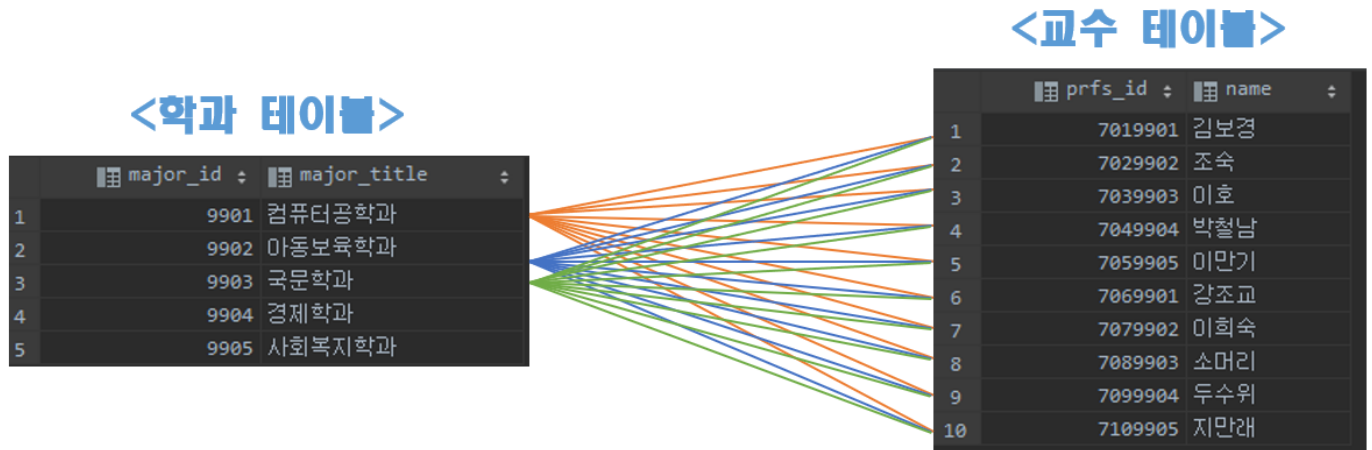
[2020/04/16 - \[Database/sql 강의\] - sql 독학 강의 # mysql join \(정의 및 종류\) 11편 -sTricky](#)

	sql 독학 강의 # mysql join (정의 및 종류) 11편 -sTricky stricky.tistory.com
--	--

## 0. Cartesian Product, 카티션곱의 정의

카티션 곱은 RDBMS에서 사용되는 join의 한 기법 중 하나입니다. where 절이나 on 절에 join 조건을 주지 않고 join을 수행하게 되면 카티션 곱이 수행됩니다.

Ansi SQL에서는 cross join 이라고도 합니다. 카티션 곱 join이 일어나게 되면 from 절에서 참조한 테이블들의 행 개수를 각각 모두 곱한 값만큼의 결과가 출력됩니다.



## 1. 카티션곱 활용방법

카티션곱은 사실 RDBMS의 개념과 상충하는 개념이 됩니다. RDBMS는 각각의 테이블이 키를 가지고 있고, 해당 키를 이용해서 다른 테이블과 관계를 형성하는 개념을 가지고 있는데, 이 카티션곱 join은 키를 이용하지 않고 그냥 모든 데이터를 1:1로 연결하는 join 방법이기 때문이죠.

저도 실무에서 일을 하면서 카티션곱 조인을 사용할 경우가 자주 없었습니다만, 간혹 꼭 필요할 때가 있습니다.

- 1) 데이터를 대량으로 복제해야 할 때
- 2) 특정 데이터 튜플만 복제되어야 할 때
- 3) 연결고리가 없는 두 테이블의 데이터를 무작위로 합쳐야 할 때

이 정도 경우에 사용했던 것 같습니다. 이것 말고도 다른 케이스도 있겠지만 제가 필요로 했던 경우는 위 경우로 설명할 수 있을 것 같습니다.

## 2. 카티션 곱 SQL 작성방법

예를 들어보겠습니다. 학과 테이블을 우선 보겠습니다.

※ 포스팅에서 사용되는 데이터는 <SQL 독학 강의 11편>을 참고하시면 됩니다.

```
select * from class.major;
```

	major_id	major_title	major_prfs_cnt	major_student_cnt	tel
1	9901	컴퓨터공학과	7	123	023454321
2	9902	마동보육학과	8	345	023456676
3	9903	국문학과	6	213	023456567
4	9904	경제학과	5	432	023456987
5	9905	사회복지학과	9	312	023454534

총데이터가 5건이 출력된 것을 확인할 수 있습니다.

다음은, 교수 테이블의 데이터를 확인해 보도록 하겠습니다.

```
select * from class.professor;
```

	prfs_id	bl_major_id	name	tel
1	7019901	9901	김보경	023445678
2	7029902	9902	조숙	023446789
3	7039903	9903	이호	023449584
4	7049904	9904	박철남	023449588
5	7059905	9905	이만기	023443443
6	7069901	9901	강조교	023449994
7	7079902	9902	이희숙	023443321
8	7089903	9903	소머리	023440123
9	7099904	9904	두수위	023443327
10	7109905	9905	지만래	023449995

모두 10건의 데이터가 출력이 되었습니다.

위 두 테이블을 카티션 곱으로 조인한다면 위에서 설명드린 바와 같이 **5 \* 10, 즉, 50건의 데이터가** 출력되어야 합니다.

위 두 테이블을 카티션 곱으로 조인해보도록 하겠습니다. 아래와 같이 mysql 문법과 ansi SQL 문법으로 각각 SQL을 작성했습니다.

### <mysql 문법>

```
select m.major_id,
       m.major_title,
       p.prfs_id,
```

```
p.name  
from class.major m, class.professor p;
```

### <ansi SQL 문법>

```
select m.major_id,  
       m.major_title,  
       p.prfs_id,  
       p.name  
from class.major m cross join class.professor p;
```

mysql문법을 사용할 때는 major 테이블과 professor 테이블 사이에 ", "를 넣어서 구분했지만 ansi SQL에서는 "cross join"이라는 명령어를 사용했습니다.

참고로 각 테이블 명 뒤에 붙어있는 m과 p는 테이블의 alias 명입니다. select, where 절에서 컬럼명을 사용해야 하는 경우 어떤 테이블의 컬럼명인지를 알수 있도록 테이블 명을 컬럼명 앞에 붙여야 하는데, 그때마다 테이블 명을 다 쓰기엔 너무 길기 때문에 간단하게 alias 명을 대신 사용을 합니다.

위 SQL문을 잘 보시면 이해가 가실 겁니다.

결과는 아래 그림과 같이 두 SQL 모두 같습니다.

	major_id	major_title	prfs_id	name
15	9905	사회복지학과	7039903	이호
16	9901	컴퓨터공학과	7049904	박철남
17	9902	마동보육학과	7049904	박철남
18	9903	국문학과	7049904	박철남
19	9904	경제학과	7049904	박철남
20	9905	사회복지학과	7049904	박철남
21	9901	컴퓨터공학과	7059905	이만기
22	9902	마동보육학과	7059905	이만기
23	9903	국문학과	7059905	이만기
24	9904	경제학과	7059905	이만기
25	9905	사회복지학과	7059905	이만기
26	9901	컴퓨터공학과	7069901	강조교
27	9902	마동보육학과	7069901	강조교
28	9903	국문학과	7069901	강조교
29	9904	경제학과	7069901	강조교
30	9905	사회복지학과	7069901	강조교
31	9901	컴퓨터공학과	7079902	이희숙
32	9902	마동보육학과	7079902	이희숙
33	9903	국문학과	7079902	이희숙
34	9904	경제학과	7079902	이희숙
35	9905	사회복지학과	7079902	이희숙
36	9901	컴퓨터공학과	7089903	소머리
37	9902	마동보육학과	7089903	소머리
38	9903	국문학과	7089903	소머리
39	9904	경제학과	7089903	소머리
40	9905	사회복지학과	7089903	소머리
41	9901	컴퓨터공학과	7099904	두수위
42	9902	마동보육학과	7099904	두수위
43	9903	국문학과	7099904	두수위
44	9904	경제학과	7099904	두수위
45	9905	사회복지학과	7099904	두수위
46	9901	컴퓨터공학과	7109905	지만래
47	9902	마동보육학과	7109905	지만래
48	9903	국문학과	7109905	지만래
49	9904	경제학과	7109905	지만래
50	9905	사회복지학과	7109905	지만래

그림과 같이 50건의 데이터가 출력이 되었음을 알 수 있습니다.

major\_title 칼럼을 보시면 학과명 5개가 반복되면서 마지막 칼럼인 name에 있는 교수 이름이 하나씩 연결되는 것을 확인하실 수 있습니다.

실제로 한번 실행을 해보시면 이해가 더 빠르실 겁니다.

간단하죠?

카티션곱을 금방 쉽게 이해하신다면 앞으로 나오는 다양한 join 기법들을 조금 더 편안하게 이해하실 수 있으리라 생각합니다.

