sql 독학 강의 # outer join SQL 15편 -sTricky

sTricky 2020. 5. 6. 12:07

sql 독학 강의 # outer join SQL 15편 -sTricky

컨텐츠 index

안녕하세요.

오늘은 지난 join 관련 공부하던것과 이어서 outer join에 관해서 배워 보도록 하겠습니다.

전편 강의 보러 가기

2020/04/27 - [Database/sql 강의] - sql 독학 강의 # 비등가 join with ansi SQL 14편 -sTricky

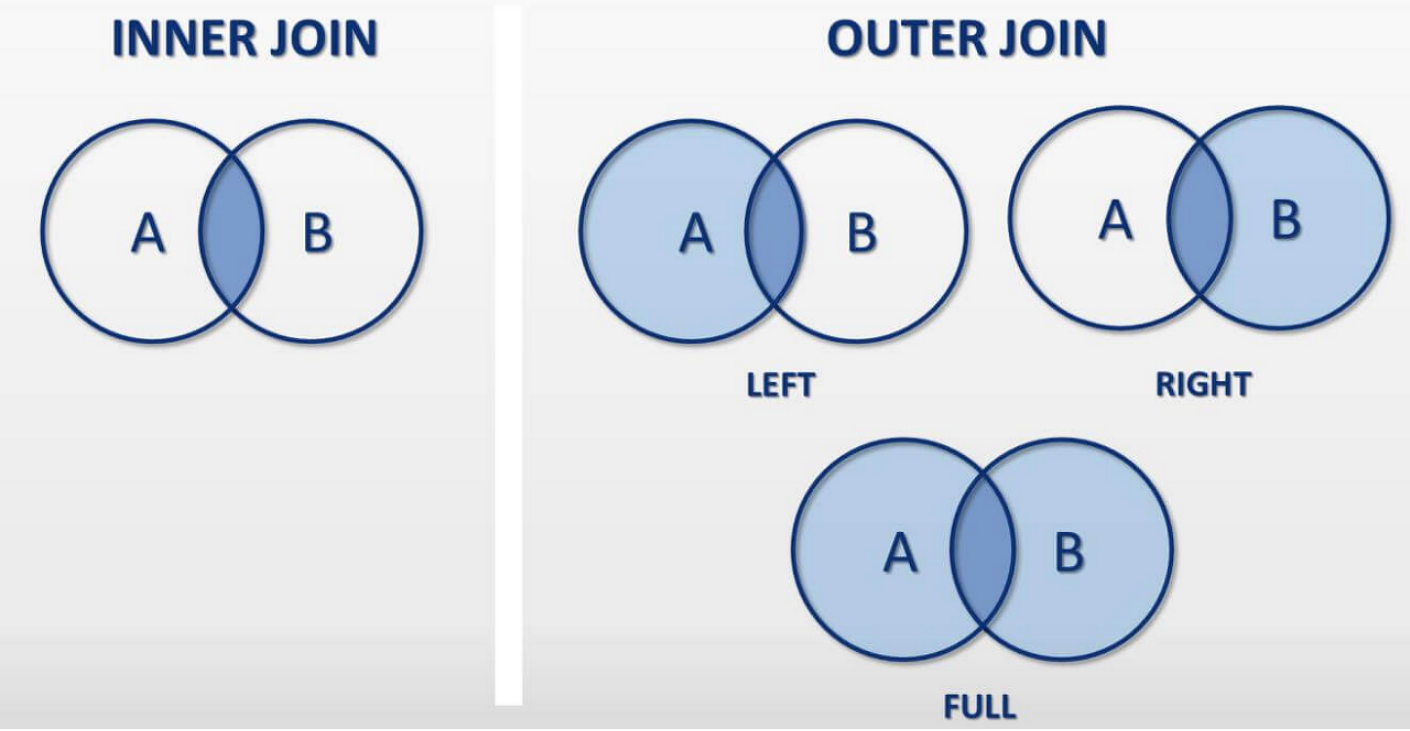
	sql 독학 강의 # 비등가 join with ansi SQL 14편 -sTricky stricky.tistory.com

0. outer join의 정의

outer join은 RDBMS에서 join을 할 때 inner join을 빼면 가장 많이 사용하는 join 기법입니다.

inner join의 경우 A 와 B 두 테이블을 join 할시에 양쪽에 key값을 기준으로 모두 존재하는 데이터만 출력이 되지만, outer join은 한쪽을 기준으로 하여 다른 쪽에 key값이 일치하는 게 없더라도 모두 출력을 하는 join 기법입니다.

outer join의 종류에는 left outer join, right outer join, full outer join이 있는데 mysql에서는 full outer join을 지원하지 않고 있습니다.



left outer join과 right outer join은 어떤 방식이 다르다기 보단 왼쪽과 오른쪽 어디다가 기준을 둘 것이냐에 따라 다릅니다. 사용방법은 같다고 볼 수 있습니다.

mysql에서는 full outer join이 필요할 때는 union으로 우회적으로 사용할 수 있습니다.

```
## full outer join 을 union all로 구현하기##
select *
from A full outer join B
on A.a = B.b;

select *
from A left outer join B
on A.a = B.b
union
select *
from B left outer join A
on A.a = B.b;
```

필요시에 꼭 써야 하는 outer join이지만 필요 없을 땐 쓰지 않아야 합니다. outer join은 모든 데이터를 다 가지고 올 때 full scan을 하기 때문에 DB에 무리를 가할 수 있기 때문입니다.

적절할 때 알맞게 사용하시길 바랍니다.

outer join을 하기 테스트하기 위해서 기존 class.student 테이블에 일부 데이터를 좀 더 추가하겠습니다.

```
INSERT INTO class.student (student_id, major_id, bl_prfs_id, name, tel) VALUES (1031, 9901, null, '신채령', '01044755564');
INSERT INTO class.student (student_id, major_id, bl_prfs_id, name, tel) VALUES (1032, 9902, null, '이만도', '01022287777');
INSERT INTO class.student (student_id, major_id, bl_prfs_id, name, tel) VALUES (1033, 9903, null, '박만호', '01099972253');
INSERT INTO class.student (student_id, major_id, bl_prfs_id, name, tel) VALUES (1034, 9904, null, '최이강', '01029386577');
INSERT INTO class.student (student_id, major_id, bl_prfs_id, name, tel) VALUES (1035, 9905, null, '강이민', '01033334444');
INSERT INTO class.student (student_id, major_id, bl_prfs_id, name, tel) VALUES (1036, 9901, null, '민형도', '01099973331');
INSERT INTO class.student (student_id, major_id, bl_prfs_id, name, tel) VALUES (1037, 9902, null, '도지란', '01055567774');
```

※원본 데이터가 필요하신 분은 아래 링크로 가셔서 테이블을 생성하고 데이터를 insert 하시기 바랍니다.

https://stricky.tistory.com/243?category=1108324

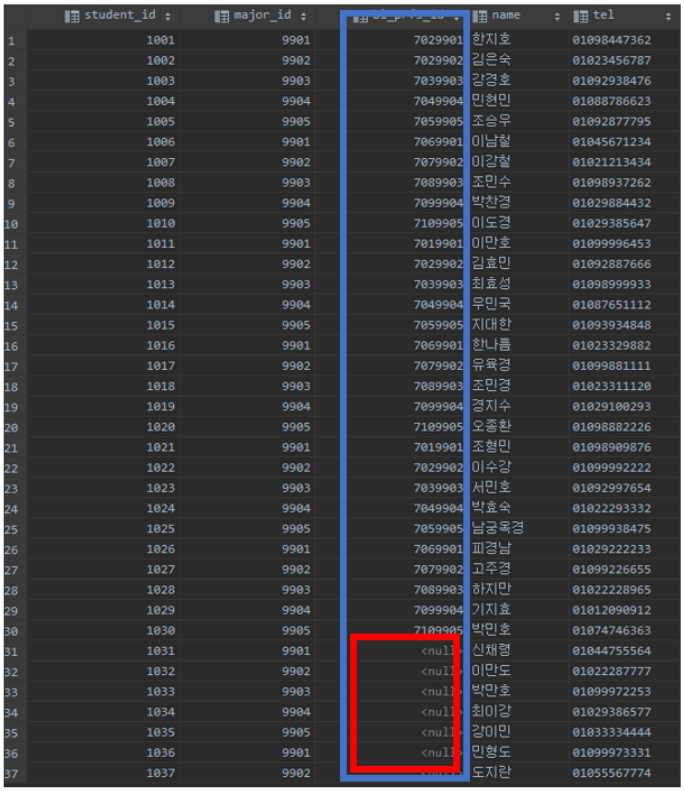
	sql 독학 강의 # mysql join (정의 및 종류) 11편 -sTricky stricky.tistory.com

위 데이터를 모두 추가하셨으면 다시 진행하겠습니다.

1. outer join 사용 예제

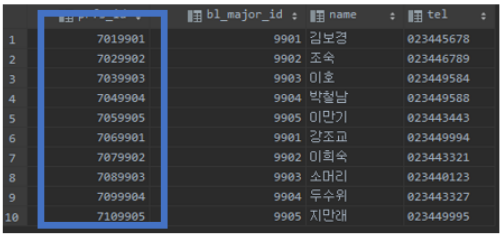
아래와 같이 student, professor 테이블이 각각 있을 때 그림에서 보면 student 테이블에 파란색 박스로 표시된 칼럼이 담당 교수 ID인데, 일부 담당 교수 ID가 없는 학생이 빨간 박스로 표시되어 있습니다.

Student 테이블



	student_id	major_id	[prof_id]	name	tel
1	1001	9901	7029901	한지호	01098447362
2	1002	9902	7029902	김은숙	01023456787
3	1003	9903	7039903	강경호	01092938476
4	1004	9904	7049904	민현민	01088786623
5	1005	9905	7059905	조승우	01092877795
6	1006	9901	7069901	이남철	01045671234
7	1007	9902	7079902	이강철	01021213434
8	1008	9903	7089903	조민수	01098937262
9	1009	9904	7099904	박찬경	01029884432
10	1010	9905	7109905	이도경	01029385647
11	1011	9901	7019901	이만호	01099996453
12	1012	9902	7029902	김효민	01092887666
13	1013	9903	7039903	최효성	01098999933
14	1014	9904	7049904	우민국	01087651112
15	1015	9905	7059905	지대한	01093934848
16	1016	9901	7069901	한나름	01023329882
17	1017	9902	7079902	유육경	01099881111
18	1018	9903	7089903	조민경	01023311120
19	1019	9904	7099904	경지수	01029100293
20	1020	9905	7109905	오종환	01098882226
21	1021	9901	7019901	조형민	01098909876
22	1022	9902	7029902	이수강	01099992222
23	1023	9903	7039903	서민호	01092997654
24	1024	9904	7049904	박효숙	01022293332
25	1025	9905	7059905	남궁옥경	01099938475
26	1026	9901	7069901	피경남	01029222233
27	1027	9902	7079902	고주경	01099226655
28	1028	9903	7089903	하지만	01022228965
29	1029	9904	7099904	기지효	01012090912
30	1030	9905	7109905	박민호	01074746363
31	1031	9901	<null>	신채경	01044755564
32	1032	9902	<null>	이만도	01022287777
33	1033	9903	<null>	박만호	01099972253
34	1034	9904	<null>	최이강	01029386577
35	1035	9905	<null>	강이민	01033334444
36	1036	9901	<null>	민형도	01099973331
37	1037	9902	<null>	도지란	01055567774

Professor 테이블



	[prof_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

이런 경우를 아직 담당교수 배정이 안된 학생이라고 가정한다면 두 테이블을 일반 inner join으로 join 한다면 전체 학생에서 아직 담당교수가 배정되지 않은 학생들은 빠지게 됩니다.

이런 누락 없이 데이터를 모두 보고자 할 때 outer join을 쓰게 됩니다.

위 두 테이블을 left outer join 하면 아래와 같이 결과가 나오게 됩니다.

Student 테이블, Professor 테이블
Left outer join 결과

	s.name	bl_prfs_id	p.name	prfs_id
1	이만호	7019901	김보경	7019901
2	조형민	7019901	김보경	7019901
3	김은숙	7029902	조숙	7029902
4	김효민	7029902	조숙	7029902
5	이수강	7029902	조숙	7029902
6	강경호	7039903	이호	7039903
7	최효성	7039903	이호	7039903
8	서민호	7039903	이호	7039903
9	민현민	7049904	박철남	7049904
10	우민국	7049904	박철남	7049904
11	박효숙	7049904	박철남	7049904
12	조승우	7059905	이만기	7059905
13	지대한	7059905	이만기	7059905
14	남궁옥경	7059905	이만기	7059905
15	이남철	7069901	강조교	7069901
16	한나름	7069901	강조교	7069901
17	피경남	7069901	강조교	7069901
18	이강철	7079902	이희숙	7079902
19	유육경	7079902	이희숙	7079902
20	고주경	7079902	이희숙	7079902
21	조민수	7089903	소머리	7089903
22	조민경	7089903	소머리	7089903
23	하지만	7089903	소머리	7089903
24	박찬경	7099904	두수위	7099904
25	경지수	7099904	두수위	7099904
26	기지효	7099904	두수위	7099904
27	이도경	7109905	이만래	7109905
28	오종환	7109905	이만래	7109905
29	박민호	7109905	이만래	7109905
30	한지호	7029901	null>	<null>
31	신채형	<null>	null>	<null>
32	이만도	<null>	null>	<null>
33	박만호	<null>	null>	<null>
34	최이강	<null>	null>	<null>
35	강이민	<null>	null>	<null>
36	민형도	<null>	null>	<null>
37	도지란	<null>	null>	<null>

파란색 상자 안의 두 칼럼은 student 테이블에서, 노란 상자 안의 두 컬럼은 professor 테이블에서 가지고 온 데이터를 student 테이블의 bl_prfs_id와 professor 테이블의 prfs_id 두 칼럼을 키로 연결하여 left outer join을 한 건데,

결과를 보시면 아시겠지만, 아래 30번째 행부터는 오른쪽 professor 테이블에 데이터가 없습니다.

물론 좌측의 student 테이블에 더 bl_profs_id는 31번 하아부터 데이터가 null로 표시되어 있지만 students 테이블의 name 칼럼에는 데이터가 표시되고 있으니, 데이터가 있다고 봐야겠죠.

"한지호"라는 학생은 bl_prfs_id에 값이 있지만, 연결이 되지 않아 professor 테이블 데이터가 null로 표시되어 있습니다. "한지호" 학생은 bl_prfs_id에 저장된 id 값이 교수 테이블에 존재하지 않는 것을 의미하고, 나머지 "신 채령"부터 "도지란" 까지는 아예 bl_prfs_id 값이 없기 때문에 professor 테이블과 연결이 되지 않은 것입니다.

이렇게 연결되지 않은 데이터까지 left outer join을 이용해서 출력해낼 수 있습니다.

2. outer join SQL 작성방법

오라클의 경우는 (+)라는 기호를 이용해서 오 라클식(?) SQL로 작성을 할 수 있지만 mysql은 outer join의 경우 ANSI SQL 형태로 작성을 해야 합니다.

그 작성법은 아래와 같습니다.

```
select s.name, s.bl_prfs_id, p.name, p.prfs_id
from class.student s
        left outer join class.professor p
                        on s.bl_prfs_id = p.prfs_id;
```

이전에 함께 공부했던 inner join의 ANSI SQL 작성 방법과 똑같다고 볼 수 있습니다.

여러분들께서는 left를 right로도 바꿔 보시고, 테이블 칼럼 순서도 바꿔보시면서 연습을 해보시면 될 것 같습니다. 학과 테이블과도 outer join을 작성하셔서 실습을 직접 많이 해보시길 바랍니다.

오늘 함께 공부한 outer join과 관련해서 궁금증은 언제든지 댓글로 문의하시기 바랍니다.