



[스파르타코딩클럽] 엑셀보다 쉬운 SQL - 3주차



매 주차 강의자료 시작에 PDF파일을 올려두었어요!

▼ PDF 파일

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/26862c81-bb58-4ad5-92b8-dbd1e2428a7c/_SQL_3_updated_210621.pdf

[수업 목표]

1. 여러 테이블의 정보를 연결하는 Join을 이해한다.
2. 연결된 정보를 바탕으로 보다 풍부한 데이터분석을 연습한다.
3. 아래 위로 결과를 연결하는 Union을 공부한다.

[목차]

01. 오늘 배울 것
02. 여러 테이블을 연결해보자: Join 이란?
03. Join 본격 사용해보기
04. 배웠던 문법을 Join과 함께 사용해보기
05. 이제는 실전! 본격 쿼리 작성해보기
06. 이렇게 끝내면 아쉽죠? 한번 더 총복습!
07. Left Join - 안써보니까 섭섭했죠?
08. 결과물 합치기! Union 배우기
09. 끝 & 숙제 설명
10. 3주차 숙제 답안 코드



모든 토글을 열고 닫는 단축키

Windows : **Ctrl** + **alt** + **t**

Mac : **⌘** + **⌥** + **t**

01. 오늘 배울 것

▼ 1) Join: 여러 정보를 한 눈에 보고 싶다면

▼ '오늘의 다짐'을 남긴 회원의 이름을 알고싶어요



오늘의 미션!

'오늘의 다짐' 이벤트 당첨자를 선정하여 스타벅스 기프티콘을 지급해야
해요.

우선, 배운 내용을 사용해서 '오늘의 다짐' 테이블을 불러와 볼까요?

*오늘의 다짐 이벤트: 오늘의 다짐을 남겨준 10명 추천해서 기프티콘 지
급하는 이벤트.

	abc checkin_id	created_at	updated_at	abc course_id	abc user_id	123 week	abc comment	123 likes
1	5f2b821f1c03ab000904e862	2020-08-06 13:08:00.0	2020-08-06 18:21:52.0	5f0ae408765da	4b8a10e6	1	월 말에는 그럴 ...	4
2	5f2bca728cf86b0009520d7f	2020-08-06 18:16:34.0	2020-08-07 09:52:15.0	5f0ae408765da	fac91699	2	1주차 시작인데...	3
3	5f2bcafd8cf86b0009520dc7	2020-08-06 18:18:54.0	2020-08-07 09:52:13.0	5f0ae408765da	c9ea3998	2	개발 너무너무 ...	3
4	5f2bcc0d8cf86b0009520dd1	2020-08-06 18:23:25.0	2020-08-07 09:52:11.0	5f0ae408765da	c6752f58	2	우 싫지는 않을 ...	5
5	5f2bce748cf86b0009520e1d	2020-08-06 18:33:40.0	2020-08-06 21:00:43.0	5f0ae408765da	4b8a10e6	2	중해서 듣다보니 ...	3
6	5f2bd2368cf86b0009520e24	2020-08-06 18:49:42.0	2020-08-07 15:56:08.0	5f0ae408765da	c9ea3998	2	1주차 시작인데...	4
7	5f2beb3cb63cef00093e40d3	2020-08-06 20:36:29.0	2020-08-11 09:51:26.0	5f0ae408765da	95621907	1	고픈데 목요일 저...	3
8	5f2beb48b63cef00093e40d4	2020-08-06 20:36:41.0	2020-08-07 09:51:59.0	5f0ae408765da	deb54ce4	1	룩할 때 마음가짐...	4
9	5f2bf08153c92c00099be3ff	2020-08-06 20:58:57.0	2020-08-07 09:52:00.0	5f0ae408765da	12dd94fe	1	각보다 퇴근하고 ...	4
10	5f2bf21cb63cef00093e410f	2020-08-06 21:05:48.0	2020-08-11 09:51:25.0	5f0ae408765da	76f656d4	1	써 목요일이네요!...	3
11	5f2bf22953c92c00099be412	2020-08-06 21:06:01.0	2020-08-07 09:52:01.0	5f0ae408765da	deb54ce4	1	래 하나만 듣고 ...	4
12	5f2bf8cab63cef00093e415f	2020-08-06 21:34:18.0	2020-08-11 09:51:24.0	5f0ae408765da	f0fc6b71	2	있어서 진도 쑥쑥...	3
13	5f2bfb8b53c92c00099be470	2020-08-06 21:46:04.0	2020-08-11 09:51:24.0	5f0ae408765da	326fa9eb	2	주차는 무사히 클...	3
14	5f2bfd2bb63cef00093e4184	2020-08-06 21:52:59.0	2020-08-07 09:52:02.0	5f0ae408765da	dda1b48e	1	말까지 1주차 부...	2
15	5f2c037bb63cef00093e41b5	2020-08-06 22:19:56.0	2020-08-09 20:26:56.0	5f0ae408765da	ca285aff	1	단 하나라도 듣고...	2
16	5f2c04c053c92c00099be4ac	2020-08-06 22:25:21.0	2020-08-07 09:52:04.0	5f0ae408765da	941e53a3	1	위가 방해하고 있...	2
17	5f2c05d5b63cef00093e41df	2020-08-06 22:29:58.0	2020-08-11 09:51:22.0	5f0ae408765da	ef860677	1	요일에 시작해서 ...	3
18	5f2c06d8b63cef00093e41e4	2020-08-06 22:34:17.0	2020-08-07 09:52:05.0	5f0ae408765da	1c3fc9f4	1	빠도 빨리 빨리 ...	2
19	5f2c0f95b63cef00093e4245	2020-08-06 23:11:33.0	2020-08-09 20:26:55.0	5f0ae408765da	0db9835b	1	장다니면서 힘들겠...	2

• 이제는 테이블 정보를 쉽게 불러올 수 있죠?



그런데 문제가 생겼어요!

'오늘의 다짐' 이벤트 당첨자를 추첨하기 위해서는, 이름과 연락처 등의 정보를 알아야 하는데 여기에는 user_id라는 정보만 있어요.



뭔가 user_id라는 정보에 힌트가 담겨있을 것 같은데... 어떻게 하면 좋을까요?

▼ 기존의 방식으로 하면...



우선, 회원 정보가 필요하니 users 테이블을 한번 살펴볼까요?

	ABC user_id	created_at	updated_at	ABC name	ABC email
1	3b3eac9f	2020-07-12 19:21:46.0	2020-07-29 16:09:55.0	이**	abc3b3@hanmail.com
2	afb35ce0	2020-07-12 19:24:32.0	2020-07-12 19:24:32.0	이**	happyafb@naver.com
3	81bde520	2020-07-13 12:12:26.0	2020-07-13 12:12:26.0	신**	sparta81b@teamspartha.co
4	5302c5dc	2020-07-13 19:27:09.0	2020-07-13 19:27:09.0	신**	python530@gmail.com
5	49905ce9	2020-07-13 19:48:46.0	2020-07-13 19:48:46.0	강**	python499@yahoo.com
6	122a2530	2020-07-13 20:11:13.0	2020-07-13 20:11:13.0	이**	happy122@gmail.com
7	86459d74	2020-07-13 20:14:23.0	2020-07-13 20:14:23.0	정**	happy864@yahoo.com
8	f3c4a33a	2020-07-13 21:13:33.0	2020-07-13 21:13:33.0	남**	abcf3c@naver.com
9	16e64944	2020-07-13 22:03:42.0	2020-07-13 22:03:42.0	남**	mysql16e@yahoo.com
10	3af5bc14	2020-07-13 22:30:00.0	2020-07-13 22:30:00.0	고**	sparta3af@hanmail.com
11	856ff83f	2020-07-13 22:38:55.0	2020-08-04 22:07:03.0	문**	hello856@hanmail.com
12	1edf9111	2020-07-14 00:26:02.0	2020-07-14 00:26:02.0	강**	python1ed@gmail.com
13	1078f508	2020-07-14 11:51:37.0	2020-07-14 11:51:37.0	방**	coolcode107@yahoo.com
14	5088448c	2020-07-14 12:41:51.0	2020-07-14 12:41:51.0	노**	python508@naver.com
15	7b55e828	2020-07-14 12:42:09.0	2020-07-14 12:42:09.0	김**	qwerty7b5@gmail.com
16	28611f56	2020-07-14 12:43:01.0	2020-07-14 12:43:01.0	이**	qwerty286@naver.com
17	efaf5439	2020-07-14 12:43:08.0	2020-07-14 12:43:08.0	임**	sqlgoodefa@naver.com
18	f1903dd4	2020-07-14 12:45:17.0	2020-07-14 12:45:17.0	김**	happyf19@daum.net
19	4a6309f9	2020-07-14 12:45:46.0	2020-07-14 12:45:46.0	이**	hello4a6@hanmail.com
20	deb54ce4	2020-07-14 12:46:42.0	2020-07-14 12:46:42.0	임**	sqlgooddeb@naver.com
21	a6b7ac89	2020-07-14 12:47:17.0	2020-07-14 12:47:17.0	김**	abca6b@naver.com
22	24h83a8a	2020-07-14 12:49:18.0	2020-07-14 12:49:18.0	김**	sparta24h@yahoo.com

• 마찬가지로 select 쿼리문을 사용해서 잘 불러오셨죠?



오! 뭔가, 똑같은 이름의 필드를 발견했어요. users 테이블의 user_id 필드와, checkins 테이블의 user_id 필드의 이름이 같아요. 이것 잘 연결시키면 될 것 같지 않나요?

▼ 자, 실제로 맞는지 확인해 봅시다! where를 잘 쓰면 되겠죠?

	asc checkin_id	created_at	updated_at	asc course_id	asc user_id	week	comment	likes
1	5f2b821f1c03ab000904e862	2020-08-06 13:08:00.0	2020-08-06 18:21:52.0	5f0ae408765da	4b8a10e6	1	월 말에는 그럴 ...	4
2	5f2bca728cf86b0009520d7f	2020-08-06 18:16:34.0	2020-08-07 09:52:15.0	5f0ae408765da	fac91699	2	1주차 시작인데...	3
3	5f2bcafd8cf86b0009520dc7	2020-08-06 18:18:54.0	2020-08-07 09:52:13.0	5f0ae408765da	c9ea3998	2	개발 너무너무 ...	3
4	5f2bcc0d8cf86b0009520dd1	2020-08-06 18:23:25.0	2020-08-07 09:52:11.0	5f0ae408765da	c6752f58	2	우 심지어 앎을 ...	5
5	5f2bce748cf86b0009520e1d	2020-08-06 18:33:40.0	2020-08-06 21:00:43.0	5f0ae408765da	4b8a10e6	2	중해서 듣다보니 ...	3
6	5f2bd2368cf86b0009520e24	2020-08-06 18:49:42.0	2020-08-07 15:56:08.0	5f0ae408765da	c9ea3998	2	1주차 시작인데...	4
7	5f2beb3cb63cef00093e40d3	2020-08-06 20:36:29.0	2020-08-11 09:51:26.0	5f0ae408765da	95621907	1	고픈데 목요일 저...	3
8	5f2beb48b63cef00093e40d4	2020-08-06 20:36:41.0	2020-08-07 09:51:59.0	5f0ae408765da	deb54ce4	1	룩할 때 마음가짐...	4
9	5f2bf08153c92c00099be3ff	2020-08-06 20:58:57.0	2020-08-07 09:52:00.0	5f0ae408765da	12dd94fe	1	간보다 퇴근하고 ...	4
10	5f2bf21cb63cef00093e410f	2020-08-06 21:05:48.0	2020-08-11 09:51:25.0	5f0ae408765da	76f656d4	1	써 목요일이네요!...	3
11	5f2bf22953c92c00099be412	2020-08-06 21:06:01.0	2020-08-07 09:52:01.0	5f0ae408765da	deb54ce4	1	래 하나만 듣고 ...	4
12	5f2bf8cab63cef00093e415f	2020-08-06 21:34:18.0	2020-08-11 09:51:24.0	5f0ae408765da	f0fc6b71	2	있어서 진도 쪽쪽...	3
13	5f2bf8b8b63c92c00099be470	2020-08-06 21:46:04.0	2020-08-11 09:51:24.0	5f0ae408765da	326fa9eb	2	주차는 무사히 클...	3
14	5f2bfd2bb63cef00093e4184	2020-08-06 21:52:59.0	2020-08-07 09:52:02.0	5f0ae408765da	dda1b48e	1	알까지 1주차 뽐...	2
15	5f2c037bb63cef00093e41b5	2020-08-06 22:19:56.0	2020-08-09 20:26:56.0	5f0ae408765da	ca285aff	1	단 하나라도 듣고...	2
16	5f2c04c053c92c00099be4ac	2020-08-06 22:25:21.0	2020-08-07 09:52:04.0	5f0ae408765da	941e53a3	1	위가 방해하고 있...	2
17	5f2c05d5b63cef00093e41df	2020-08-06 22:29:58.0	2020-08-11 09:51:22.0	5f0ae408765da	ef860677	1	요일에 시작해서 ...	3
18	5f2c06dbb63cef00093e41e4	2020-08-06 22:34:17.0	2020-08-07 09:52:05.0	5f0ae408765da	1c3fc9f4	1	빠도 빨리 빨리 ...	2
19	5f2c0f95b63cef00093e4245	2020-08-06 23:11:33.0	2020-08-09 20:26:55.0	5f0ae408765da	0db9835b	1	장다니면서 힘들것...	2

- 우선 checkins 테이블의 맨 첫 데이터 user_id인 4b8a10e6를 복사해 줍니다.

```
select * from users
where user_id = "4b8a10e6"
```

users

select * from users where user_id = "4b8a10e6" Enter a SQL expression to filter results (use Ctrl+Space)

	asc user_id	created_at	updated_at	asc name	asc email
1	4b8a10e6	2020-08-02 18:53:11.0	2020-08-02 19:19:49.0	이**	python4b8@naver.com

- 그리고, user_id가 4b8a10e6인 데이터를 users 테이블에서 불러와보니 이렇게 잘 나오네요!



대충 감을 잡았어요!

1. checkins 테이블의 user_id를 복사!
2. users 테이블에서 해당 user_id를 갖는 데이터를 가져오기!
3. 작성자를 조회하고 싶은 '오늘의 다짐' 갯수대로 1, 2 과정을 반복!



오늘의 다짐을 남긴 유저 100명을 얻어야 하면, 100번을 쿼리를 날려야겠죠? 우선 이렇게 하는데... 더 좋은 방법이 없을까요?



[오늘의 꿀팁!]

한 테이블에 모든 정보를 담을 수도 있겠지만, 불필요하게 테이블의 크기가 커져 불편해집니다.

그래서, 데이터를 종류별로 쪼개 다른 테이블에 담아놓고 연결이 필요한 경우 연결할 수 있도록 만들어놓습니다.

예를 들면, users와 checkins 테이블에 동시에 존재하는 user_id 처럼요.

이런 필드를 두 테이블을 연결시켜주는 열쇠라는 의미로 'key'라고 부릅니다.

▼ 우리 모두의 시간은 소중하니까. Join을 사용하면?



소중한 여러분의 시간이 낭비되게 만들도록 개발자 분들은 호락호락하지 않습니다.

두 테이블을 연결한다는 의미인 Join을 사용하면 이 고민을 한 방에 해결할 수 있어요!

▼ 직접 눈으로 한번 봐볼까요?



우선 지금은, 눈으로만!

이번 주에 열심히 배울 내용입니다.

id	user_id	week	comment	likes	user_id	created_at	updated_at	name	email
iae0006002817	49905ce9	1	열심히!	3	49905ce9	2020-07-13 19:48:46.0	2020-07-13 19:48:46.0	강**	python499@yahoo.com
iae0006002817	deb54ce4	1	복합 때 마음가짐...	4	deb54ce4	2020-07-14 12:46:42.0	2020-07-14 12:46:42.0	임**	sqlgooddeb@naver.com
iae0006002817	deb54ce4	1	레 하나만 들고 ...	4	deb54ce4	2020-07-14 12:46:42.0	2020-07-14 12:46:42.0	임**	sqlgooddeb@naver.com
iae0006002817	deb54ce4	1	일의 내가 부끄러...	2	deb54ce4	2020-07-14 12:46:42.0	2020-07-14 12:46:42.0	임**	sqlgooddeb@naver.com
iae0006002817	0fe500cc	2	주차 강의 듣고 ...	2	0fe500cc	2020-07-14 13:15:54.0	2020-07-14 13:15:54.0	배**	coolcode0fe@naver.com
iae0006002817	0fe500cc	2	택근무 야근 후 ...	2	0fe500cc	2020-07-14 13:15:54.0	2020-07-14 13:15:54.0	배**	coolcode0fe@naver.com
iae0006002817	1f301966	1	다 하기 벅치긴 ...	3	1f301966	2020-07-14 13:22:11.0	2020-07-14 13:22:11.0	정**	python1f3@hanmail.com
iae0006002817	1f301966	1	늘 1주차 끝내자...	2	1f301966	2020-07-14 13:22:11.0	2020-07-14 13:22:11.0	정**	python1f3@hanmail.com
iae0006002817	1f301966	1	뿐지만 페이스를 ...	3	1f301966	2020-07-14 13:22:11.0	2020-07-14 13:22:11.0	정**	python1f3@hanmail.com
iae0006002817	775725f7	2	늘은 1주차 마무리...	2	775725f7	2020-07-14 21:16:46.0	2020-07-14 21:16:46.0	정**	hello775@hanmail.com
iae0006002817	3100f720	1	심히 해야겠어요...	3	3100f720	2020-07-14 22:42:51.0	2020-07-14 22:42:51.0	임**	python310@naver.com
iae0006002817	a2c78b69	2	년전에 독학 포기...	2	a2c78b69	2020-07-15 01:50:05.0	2020-07-15 01:50:05.0	임**	pythona2c@daum.net
iae0006002816	7e9b4584	1	늘 1주차 끝내고...	1	7e9b4584	2020-07-15 09:15:45.0	2020-07-15 09:15:45.0	윤**	mysql7e9@teamsparta.co
iae0006002817	dda1b48e	1	말까지 1주차 부...	2	dda1b48e	2020-07-15 13:35:37.0	2020-07-15 13:35:37.0	박**	mysqidda@daum.net
iae0006002817	dda1b48e	1	직 늦지 않았다 ...	2	dda1b48e	2020-07-15 13:35:37.0	2020-07-15 13:35:37.0	박**	mysqidda@daum.net
iae0006002816	ed12149c	1	아자아자 화이팅!	0	ed12149c	2020-07-16 00:48:07.0	2020-07-16 00:48:07.0	신**	spartaed1@qmail.com

- Join이라는 기능을 사용하면, 이렇게 한 눈에 두 개의 테이블을 연결해서 볼 수 있어요.



짧은 Join 쿼리 한 줄만 더 쓰면, 한번의 쿼리로 이렇게 정보를 한 눈에 볼 수 있어요!

▼ 2) Join을 사용한 여러가지 연습을 진행하기



지금까지 배운 내용을 사용해 실전과 같은 데이터분석을 진행해봅시다!

회사에서 일을 하다보면, 저장된 데이터로부터 얻어내고 싶은 인사이트가 있겠죠?

스파르타 데이터베이스를 사용해 실전 데이터분석을 함께 해봅시다.

02. 여러 테이블을 연결해보자: Join 이란?

▼ 3) Join 이란?



Join이란?

두 테이블의 공통된 정보 (key값)를 기준으로 테이블을 연결해서 한 테이블처럼 보는 것을 의미해요.

예) user_id 필드를 기준으로 users 테이블과 orders 테이블을 연결해서 한 눈에 보고 싶어요!



위의 예시와 같이, 두 테이블의 정보를 연결해서 함께 보고싶을 때가 있겠죠?

그럴 때를 대비해서 무언가 연결된 정보가 있을 때, user_id 처럼 동일한 이름과 정보가 담긴 필드를 두 테이블에 똑같이 담아놓는답니다. 이런 필드를 두 테이블을 연결시켜주는 열쇠라는 의미로 'key'라고 불러요.

▼ 직접 데이터로 살펴볼까요?



깜짝 퀴즈! 아래 테이블에서 (checkins 테이블) key값에 해당하는 필드를 찾아보세요.

	asc checkin_id	created_at	updated_at	asc course_id	asc user_id	123 week	asc comment	123 likes
1	5f2cd8e9b63cef00093e447c	2020-08-07 13:30:34.0	2020-08-09 20:26:46.0	5f0ae408765dae0006002817	49905ce9	1	열심히!	3
2	5f2beb48b63cef00093e40d4	2020-08-06 20:36:41.0	2020-08-07 09:51:59.0	5f0ae408765dae0006002817	deb54ce4	1	복합 때 마음가짐...	4
3	5f2bf22953c92c00099be412	2020-08-06 21:06:01.0	2020-08-07 09:52:01.0	5f0ae408765dae0006002817	deb54ce4	1	레 하나만 들고 ...	4
4	5f312304390108000936b933	2020-08-10 19:35:49.0	2020-08-11 09:50:11.0	5f0ae408765dae0006002817	deb54ce4	1	일의 내가 부끄러...	2
5	5f314f56390108000936bad2	2020-08-10 22:44:55.0	2020-08-12 20:05:29.0	5f0ae408765dae0006002817	0fe500cc	2	주차 강의 듣고 ...	2
6	5f3280b3171df600097d996a	2020-08-11 20:27:47.0	2020-08-12 21:53:04.0	5f0ae408765dae0006002817	0fe500cc	2	택근로 야근 후 ...	2
7	5f2d3c28a9ff0a0009b4ac13	2020-08-07 20:34:01.0	2020-08-10 09:45:57.0	5f0ae408765dae0006002817	1f301966	1	다 하기 벅차긴 ...	3
8	5f2e072224177900090af162	2020-08-08 11:00:02.0	2020-08-09 20:26:06.0	5f0ae408765dae0006002817	1f301966	1	늘 1주차 끝내자...	2
9	5f30bc8aa65cce0009353450	2020-08-10 12:18:35.0	2020-08-12 17:12:23.0	5f0ae408765dae0006002816	1f301966	2	쁘지만 페이스를 ...	3
10	5f2c1bb553c92c00099be518	2020-08-07 00:03:17.0	2020-08-09 20:26:53.0	5f0ae408765dae0006002817	775725f7	1	늘은 1주차 마무...	2
11	5f2cf8feb63cef00093e45ca	2020-08-07 15:47:26.0	2020-08-10 21:31:42.0	5f0ae408765dae0006002817	3100f720	1	심히 해야겠어요...	3
12	5f32d3c3390108000936c1af	2020-08-12 02:22:11.0	2020-08-12 21:53:07.0	5f0ae408765dae0006002817	a2c78b69	2	년전에 독학 포기...	2
13	5f2e60b0a9ff0a0009b4af37	2020-08-08 17:22:09.0	2020-08-09 20:28:19.0	5f0ae408765dae0006002816	7e9b4584	1	늘 1주차 끝내고...	1
14	5f2bfd2bb63cef00093e4184	2020-08-06 21:52:59.0	2020-08-07 09:52:02.0	5f0ae408765dae0006002817	dda1b48e	1	말까지 1주차 루...	2
15	5f320aae390108000936bd08	2020-08-11 12:04:15.0	2020-08-12 21:53:01.0	5f0ae408765dae0006002817	dda1b48e	1	직 늦지 않았다 ...	2
16	5f2d658424177900090aefe9	2020-08-07 23:30:28.0	2020-08-07 23:30:28.0	5f0ae408765dae0006002816	ed12149c	1	아자아자 화이팅!	0
17	5f2df40024177900090af0d5	2020-08-08 09:38:25.0	2020-08-09 20:28:27.0	5f0ae408765dae0006002816	ed12149c	1	어나자마자 공부시...	1
18	5f2c16bfb63cef00093e425e	2020-08-06 23:42:08.0	2020-08-07 09:52:06.0	5f0ae408765dae0006002817	a2526635	1	완강하기	2
19	5f33aabc56fffe00094f216a	2020-08-12 17:39:25.0	2020-08-12 17:39:25.0	5f0ae408765dae0006002816	9d669255	1	주차를 조금 늦게...	0
20	5f2c05d5b63cef00093e41df	2020-08-06 22:29:58.0	2020-08-11 09:51:22.0	5f0ae408765dae0006002816	ef860677	1	요일에 시작해서 ...	3
21	5f2c027bb63cef00093e41b5	2020-08-06 22:19:56.0	2020-08-09 20:26:56.0	5f0ae408765dae0006002817	ca285aff	1	단 하나라도 들고...	2

- 눈치채신 분도 계시겠지만, 빨간색으로 표시해놓은 필드가 모두 key값에 해당하는 필드입니다.



[오늘의 꿀팁!]

병원에서 의사선생님이 '환자번호 101번님 진료받으러 들어오세요' 라고 불렀는데 같은 환자번호를 가진 사람이 여러명이 있으면 누가 들어와야 할지 환자번호만으로 알 수 없겠죠?

SQL에서의 Join도 마찬가지로요.

key값을 사용해 연결하고 싶은 테이블에 찾아가서 똑같은 값을 가지는 key를 찾게 되는데, 똑같은 key를 가지는 데이터가 여러개 있으면 어느 데이터를 가져와서 연결해야 할지 알 수 없어요.

▼ [코드스니펫] Join을 사용해서 Key값으로 두 테이블 연결해보기



자세한 내용은 곧장 설명해드릴테니, 눈으로만 따라오세요!

```
select * from point_users
left join users
on point_users.user_id = users.user_id
```




대충 감이 잡히나요? 자, 이제 본격 시동을 걸어보죠!

	updated_at	user_id	point		user_id	created_at	updated_at	name
7.0	2020-08-11 17:46:54.0	00b567a0	3,151		1	005ccfad	2020-07-30 23:07:43.0	곽**
9.0	2020-08-09 12:22:11.0	01b472fa	170		2	00b567a0	2020-07-29 23:43:18.0	오**
9.0	2020-08-10 20:29:38.0	01c76e41	650		3	01b472fa	2020-07-27 17:23:00.0	조**
8.0	2020-08-09 23:30:59.0	01deb108	2,406		4	01c76e41	2020-07-29 14:08:27.0	김**
8.0	2020-08-12 17:27:22.0	03667091	5,895		5	01deb108	2020-07-19 22:20:17.0	오**
3.0	2020-08-10 19:51:35.0	0644d41f	6,003		6	0342fde3	2020-07-16 13:30:48.0	테**
4.0	2020-08-10 21:54:50.0	080133db	1,179		7	03508c31	2020-07-24 23:07:10.0	김**
8.0	2020-08-03 12:19:08.0	081e2577	0		8	03667091	2020-07-14 13:11:48.0	최**
2.0	2020-08-07 17:56:23.0	093f520a	2,734		9	0493607e	2020-07-20 17:50:32.0	김**
1.0	2020-08-11 22:32:26.0	09d75364	7,763		10	0570bb2b	2020-07-16 13:43:26.0	김**
0.0	2020-08-09 20:39:18.0	0b47bcbc	4,837		11	0644d41f	2020-07-28 13:50:11.0	박**
4.0	2020-08-09 17:41:27.0	0b9ca920	6,479		12	0673d09d	2020-07-20 21:37:38.0	송**
8.0	2020-08-11 21:40:44.0	0bfe0771	7,992					
3.0	2020-08-11 02:03:29.0	0caeffd0	1,585					



[오늘의 꿀팁!] 혹시 엑셀을 잘 쓰신다면?

SQL의 Join은 엑셀의 vlookup과 동일하다고 생각하시면 됩니다 :-)

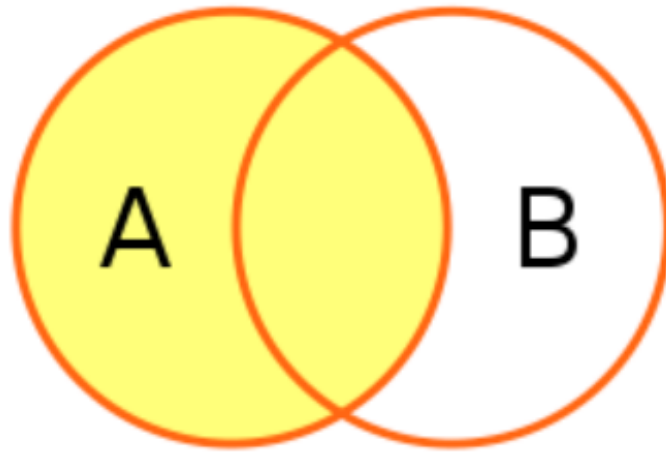
▼ 4) Join의 종류: Left Join, Inner Join

▼ Left Join: 유저 데이터로 Left Join 이해해보기



앗! 어디서 많이 본 그림 아닌가요?

생각하는 그 그림이 맞아요! SQL에서의 Join은 두 집합 사이의 관계와 같습니다.



- 여기서 A와 B는 각각의 테이블을 의미합니다. 둘 사이의 겹치는 부분은, 뭔가 테이블 A와 B의 key 값이 연결되는 부분일 것 같지 않나요?

▼ [코드스니펫] 유저 데이터로 Left Join 이해해보기

```
select * from users u
left join point_users p
on u.user_id = p.user_id;
```



어떤 데이터는 모든 필드가 채워져있지만, 어떤 데이터는 비어있는 필드가 있습니다.

꼭찬 데이터: 해당 데이터의 user_id 필드값이 point_users 테이블에 존재해서 연결한 경우

비어있는 데이터: 해당 데이터의 user_id 필드값이 point_users 테이블에 존재하지 않는 경우

- 비어있는 데이터의 경우, 회원이지만 수강을 등록/시작하지 않아 포인트를 획득하지 않은 회원인 경우예요!

▼ Where를 사용해서 직접 확인해보기!



꼭찬 데이터, 비어있는 데이터에서 하나씩 user_id를 뽑아서 orders 테이블에서 조회해볼까요?

꼭찬 데이터의 user_id값 예시: d90e7626

비어있는 데이터의 user_id값 예시: 3b3eac9f

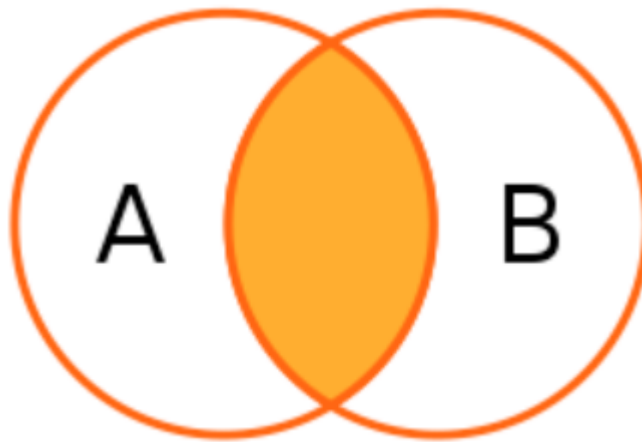


위의 user_id값으로 조회하면 데이터가 정상적으로 한 개의 데이터가 출력되지만, 아래의 user_id값으로 조회하면 데이터가 출력되지 않는 것을 알 수 있습니다!

▼ Inner Join: 유저 데이터로 Inner Join 이해해보기



이것도 어디서 많이 본 그림이에요!



- 여기서 A와 B는 각각의 테이블을 의미합니다. 이 그림은 뭔가, 두 테이블의 교집합을 이야기하고 있는 것 같지 않나요?

▼ [코드스니펫] 유저 데이터로 Inner Join 이해해보기

```
select * from users u
inner join point_users p
on u.user_id = p.user_id;
```



앗, 여기서는 비어있는 필드가 있는 데이터가 없어요!

그 이유는, 같은 user_id를 두 테이블에서 모두 가지고 있는 데이터만 출력했기 때문이에요.

▼ 진짜 그런지 확인해 볼까요?



Left Join을 했을 때 빈 필드가 없는 데이터의 갯수와, Inner Join을 했을때의 전체 데이터의 갯수가 같은지 확인해보면 되겠죠?



저랑 같이 해볼까요?

03. Join 본격 사용해보기

▼ 5) Join 함께 연습해보기



자, 이제 본격 여러 테이블을 연결해 볼까요? 두근두근!
SQL 쿼리는 직접 짜면서 가장 빠르게 배운답니다.

▼ [실습] orders 테이블에 users 테이블 연결해보기



주문 정보에 유저 정보를 연결해 분석을 위한 준비를 하려고 합니다! 우선 inner Join을 사용해서 주문 정보에, 유저 정보를 붙여서 볼까요?

▼ 정답 쿼리 살펴보기

```
select * from orders o
inner join users u
on o.user_id = u.user_id;
```



[오늘의 팁!]

주문을 하기 위해서는 회원정보가 있어야 할테니, orders 테이블에 담긴 user_id는 모두 users 테이블에 존재하겠죠?

▼ [실습] checkins 테이블에 users 테이블 연결해보기



'오늘의 다짐' 테이블에 유저 정보를 연결해 분석을 위한 준비를 하려고 합니다! 우선 Inner Join을 사용해서 '오늘의 다짐' 테이블에, 유저 테이블을 붙여서 볼까요?

▼ 정답 쿼리 살펴보기

```
select * from checkins c
inner join users u
on c.user_id = u.user_id;
```



[오늘의 팁!]

연결의 기준이 되고싶은 테이블을 from 절에,
기준이 되는 테이블에 붙이고 싶은 테이블을 Join 절에 위치해 놓습니다.

▼ [실습] enrolleds 테이블에 courses 테이블 연결해보기



'수강 등록' 테이블에 과목 정보를 연결해 분석을 위한 준비를 하려고 합니다! 우선 Inner Join을 사용해서 '수강 등록' 테이블에, 과목 테이블을 붙여서 볼까요?

▼ 정답 쿼리 살펴보기

```
select * from enrolleds e
inner join courses c
on e.course_id = c.course_id;
```

▼ 6) SQL 쿼리가 실행되는 순서

```
select * from enrolleds e
inner join courses c
on e.course_id = c.course_id;
```



위 쿼리가 실행되는 순서: from → join → select

1. from enrolleds: enrolleds 테이블 데이터 전체를 가져옵니다.
2. inner join courses on e.course_id = c.course_id: courses를 enrolleds 테이블에 붙이는데, enrolleds 테이블의 course_id와 동일한 course_id를 갖는 courses의 테이블을 붙입니다.
3. select * : 붙여진 모든 데이터를 출력합니다.



항상 from에 들어간 테이블을 기준으로, 다른 테이블이 붙는다고 생각하면 편합니다!

04. 배웠던 문법을 Join과 함께 사용해보기

▼ 7) 배웠던 문법 Join과 함께 연습해보기

▼ checkins 테이블에 courses 테이블 연결해서 통계치 내보기



'오늘의 다짐' 정보에 과목 정보를 연결해 과목별 '오늘의 다짐' 갯수를 세어보자!

▼ [코드스니펫] 과목별 오늘의 다짐 갯수 세어보기

```
select co.title, count(co.title) as checkin_count from checkins ci
inner join courses co
on ci.course_id = co.course_id
group by co.title
```



[오늘의 팁!]

2주차에 배운 alias는 이렇게 사용하면 편합니다. 연결되는 테이블이 많아지면서 필드명과 테이블명이 헷갈려 실수할 수 있는데, 이렇게 alias를 지정해 주면 편하고 깔끔하게 SQL 쿼리를 작성할 수 있어요.

▼ point_users 테이블에 users 테이블 연결해서 순서대로 정렬해보기



유저의 포인트 정보가 담긴 테이블에 유저 정보를 연결해서, 많은 포인트를 얻은 순서대로 유저의 데이터를 뽑아보자!

▼ [코드스니펫] 많은 포인트를 얻은 순서대로 유저 데이터 정렬해서 보기

```
select * from point_users p
inner join users u
on p.user_id = u.user_id
order by p.point desc
```

▼ orders 테이블에 users 테이블 연결해서 통계치 내보기



주문 정보에 유저 정보를 연결해 네이버 이메일을 사용하는 유저 중, 성씨별 주문건수를 세어보자!

▼ [코드스니펫] 네이버 이메일 사용하는 유저의 성씨별 주문건수 세어보기

```
select u.name, count(u.name) as count_name from orders o
inner join users u
on o.user_id = u.user_id
where u.email like '%naver.com'
group by u.name
```

▼ 8) SQL 쿼리가 실행되는 순서

```
select u.name, count(u.name) as count_name from orders o
inner join users u
on o.user_id = u.user_id
where u.email like '%naver.com'
group by u.name
```



위 쿼리가 실행되는 순서: from → join → where → group by → select

1. from orders o: orders 테이블 데이터 전체를 가져오고 o라는 별칭을 붙입니다.
2. inner join users u on o.user_id = u.user_id : users 테이블을 orders 테이블에 붙이는데, orders 테이블의 user_id와 동일한 user_id를 갖는 users 테이블 데이터를 붙입니다. (*users 테이블에 u라는 별칭을 붙입니다)
3. where u.email like '%naver.com': users 테이블 email 필드값이 naver.com으로 끝나는 값만 가져옵니다.
4. group by u.name: users 테이블의 name값이 같은 값들을 뭉쳐줍니다.
5. select u.name, count(u.name) as count_name : users 테이블의 name필드와 name 필드를 기준으로 뭉쳐진 갯수를 세어서 출력해줍니다.



Join의 실행 순서는 항상 from 과 붙어다닌다고 생각하면 편해요!

05. 이제는 실전! 본격 쿼리 작성해보기

▼ 9) [퀴즈] Join 연습1



결제 수단 별 유저 포인트의 평균값 구해보기
(어느 결제수단이 가장 열심히 듣고 있나~)

join 할 테이블: `point_users` 에, `orders` 를 붙이기

꿀팁! → `round(숫자, 자릿수)` 를 이용해서 반올림을 해볼까요?

	ABC payment_method ↑↓	123 avg_point ↑↓
1	kakaopay	7,729
2	CARD	5,134
3	TAXBILL	4,588
4	MONEY	3,684

▼ 정답 쿼리 살펴보기!

```
select o.payment_method, round(AVG(p.point)) from point_users p
inner join orders o
on p.user_id = o.user_id
group by o.payment_method
```



이해가 안 되신다고요? 저랑 같이 해봐요!

▼ 10) [퀴즈] Join 연습2



결제하고 시작하지 않은 유저들을 성씨별로 세어보기
(어느 성이 가장 시작을 안하였는가~)

join 할 테이블: `enrolleds` 에, `users` 를 붙이기

꿀팁! → `is_registered = 0` 인 사람들을 세어보아요!

꿀팁! → `order by` 를 이용해서 내림차순으로 정렬하면 보기 좋겠죠?

	ABC name 🔼🔼	123 cnt_name 🔼🔼
1	남**	5
2	김**	4
3	이**	3
4	신**	2
5	임**	2
6	강**	1
7	최**	1
8	미**	1
9	테**	1
10	황**	1
11	류**	1
12	박**	1
13	곽**	1

▼ 정답 쿼리 살펴보기!

```
select name, count(*) as cnt_name from enrolleds e
inner join users u
on e.user_id = u.user_id
where is_registered = 0
group by name
order by cnt_name desc
```



이해가 안 되신다고요? 저랑 같이 해봐요!

▼ 11) [퀴즈] Join 연습3



과목 별로 시작하지 않은 유저들을 세어보기

join 할 테이블: `courses` 에, `enrolleds` 를 붙이기

꿀팁! → `is_registered = 0` 인 사람들을 세어보아요!

	ABC course_id	ABC title	123 cnt_notstart
1	5f0ae408765dae0006002816	웹개발 종합반	14
2	5f0ae408765dae0006002817	앱개발 종합반	10

▼ 정답 쿼리 살펴보기!

```
select c.course_id, c.title, count(*) as cnt_notstart from courses c
inner join enrolleds e
on c.course_id = e.course_id
where is_registered = 0
group by c.course_id
```



이해가 안 되신다고요? 저랑 같이 해봐요!

06. 이렇게 끝내면 아쉽죠? 한번 더 총복습!

▼ 12) [퀴즈] Join 연습4



웹개발, 앱개발 종합반의 week 별 체크인 수를 세어볼까요? 보기 좋게 정리해보기!

join 할 테이블: `courses` 에 `checkins` 를 붙이기

꿀팁! → group by, order by에 `coma` 로 이어서 두 개 필드를 걸어보세요!

	<code>title</code>	<code>week</code>	<code>cnt</code>
1	앱개발 종합반	1	52
2	앱개발 종합반	2	18
3	앱개발 종합반	3	2
4	웹개발 종합반	1	44
5	웹개발 종합반	2	11
6	웹개발 종합반	3	7

▼ 정답 쿼리 살펴보기!

```
select c1.title, c2.week, count(*) as cnt from checkins c2
inner join courses c1 on c2.course_id = c1.course_id
group by c1.title, c2.week
order by c1.title, c2.week
```



이해가 안 되신다고요? 저랑 같이 해봐요!

▼ 13) [퀴즈] Join 연습5



연습4번에서, 8월 1일 이후에 구매한 고객들만 발라내어 보세요!

join 할 테이블: `courses` 에, `checkins` 를 붙이고!
+ `checkins` 에, `orders` 를 한번 더 붙이기!

꿀팁! → `orders` 테이블에 inner join을 한번 더 걸고, where 절로 마무리!

	ABC title	123 week	123 cnt
1	앱개발 종합반	1	6
2	앱개발 종합반	2	3
3	웹개발 종합반	1	14
4	웹개발 종합반	2	2

▼ 정답 쿼리 살펴보기!

```
select c1.title, c2.week, count(*) as cnt from courses c1
inner join checkins c2 on c1.course_id = c2.course_id
inner join orders o on c2.user_id = o.user_id
where o.created_at >= '2020-08-01'
group by c1.title, c2.week
order by c1.title, c2.week
```



이해가 안 되신다고요? 저랑 같이 해봐요!

07. Left Join - 안써보니까 섭섭했죠?

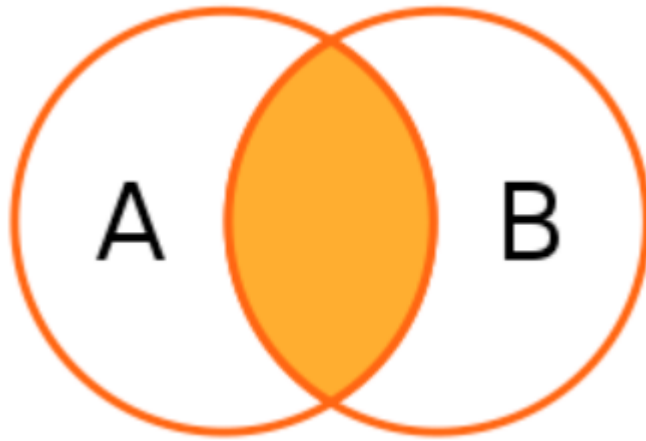
▼ 14) 다시한번 복습! inner join vs left join



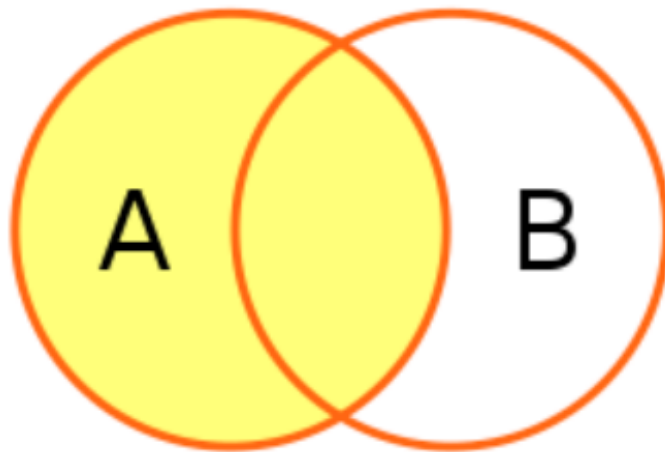
inner join 은 교집합, left join 은 첫번째 원에 붙이는 것!

- 그림으로 복습하기

▼ inner join



▼ left join



- 그래서! left join은 어디에 → 뭐를 붙일건지, 순서가 중요 하답니다!

▼ 15) Left join 언제쓸까요? (복습)



바로 한번 볼까요?

예를 들면 모든 유저가 포인트를 갖고 있지를 알 수 있잖아요!

- users 테이블과 ↔ point_users 테이블을 left join 해봅시다.

```
select * from users u
left join point_users pu on u.user_id = pu.user_id
```

	user_id	created_at	updated_at	name	email	point_user_id	created_at	updated_at	user_id	point
1	3b3eac9f	2020-07-12 19:21:46	2020-07-29 16:09:55	이**	abc3b3@hanmail.com	[NULL]			[NULL]	[NULL]
2	afb35ce0	2020-07-12 19:24:32	2020-07-12 19:24:32	이**	happyafb@naver.com	[NULL]			[NULL]	[NULL]
3	81bde520	2020-07-13 12:12:26	2020-07-13 12:12:26	신**	sparta81b@teamsparta.co	[NULL]			[NULL]	[NULL]
4	5302c5dc	2020-07-13 19:27:09	2020-07-13 19:27:09	신**	python530@gmail.com	[NULL]			[NULL]	[NULL]
5	49905ce9	2020-07-13 19:48:46	2020-07-13 19:48:46	강**	python499@yahoo.com	5f28c46dee305500095e5740	2020-08-04 11:14:05	2020-08-07 15:09:51	49905ce9	450
6	122a2530	2020-07-13 20:11:13	2020-07-13 20:11:13	이**	happy122@gmail.com	[NULL]			[NULL]	[NULL]
7	86459d74	2020-07-13 20:14:23	2020-07-13 20:14:23	정**	happy864@yahoo.com	[NULL]			[NULL]	[NULL]
8	f3c4a33a	2020-07-13 21:13:33	2020-07-13 21:13:33	남**	abc3c@naver.com	[NULL]			[NULL]	[NULL]
9	16e64944	2020-07-13 22:03:42	2020-07-13 22:03:42	남**	mysql16e@yahoo.com	[NULL]			[NULL]	[NULL]
10	3af5bc14	2020-07-13 22:30:00	2020-07-13 22:30:00	고**	sparta3af@hanmail.com	5f27d3f3371f2b0009bbb6fc	2020-08-03 18:08:03	2020-08-03 18:10:19	3af5bc14	50
11	856ff83f	2020-07-13 22:38:55	2020-08-04 22:07:03	문**	hello856@hanmail.com	5f2a3a52e268350009a6a811	2020-08-05 13:49:22	2020-08-12 22:04:51	856ff83f	1,665
12	1edf9111	2020-07-14 00:26:02	2020-07-14 00:26:02	강**	python1ed@gmail.com	[NULL]			[NULL]	[NULL]
13	1078f508	2020-07-14 11:51:37	2020-07-14 11:51:37	방**	coolcode107@yahoo.com	5f2b5a0e202843000925d94d	2020-08-06 10:17:03	2020-08-11 14:00:16	1078f508	3,806
14	5088448c	2020-07-14 12:41:51	2020-07-14 12:41:51	노**	python508@naver.com	[NULL]			[NULL]	[NULL]
15	7b55e828	2020-07-14 12:42:09	2020-07-14 12:42:09	김**	qwerty7b5@gmail.com	5f294b64e268350009a6a4ed	2020-08-04 20:49:57	2020-08-11 20:57:55	7b55e828	3,986
16	28611f56	2020-07-14 12:43:01	2020-07-14 12:43:01	이**	qwerty286@naver.com	[NULL]			[NULL]	[NULL]
17	efaf5439	2020-07-14 12:43:08	2020-07-14 12:43:08	임**	sqlgoodefa@naver.com	5f275a4ad3943c0009e065fb	2020-08-03 09:28:59	2020-08-12 11:07:13	efaf5439	3,538
18	f1903dd4	2020-07-14 12:45:17	2020-07-14 12:45:17	김**	happyf19@daum.net	[NULL]			[NULL]	[NULL]
19	4a6309f9	2020-07-14 12:45:46	2020-07-14 12:45:46	이**	hello4a6@hanmail.com	[NULL]			[NULL]	[NULL]
20	deb54ce4	2020-07-14 12:46:42	2020-07-14 12:46:42	임**	sqlgoodeb@naver.com	5f27f250d80bcb000982b154	2020-08-03 20:17:37	2020-08-12 16:21:56	deb54ce4	4,087

- 이 상태에선, 이런 게 가능합니다.



유저 중에, 포인트가 없는 사람(=즉, 시작하지 않은 사람들)의 통계!
속닥속닥) **is NULL** , **is not NULL** 을 함께 배워보아요!

```
select name, count(*) from users u
left join point_users pu on u.user_id = pu.user_id
where pu.point_user_id is NULL
group by name
```

```
select name, count(*) from users u
left join point_users pu on u.user_id = pu.user_id
where pu.point_user_id is not NULL
group by name
```

▼ 16) [퀴즈] 여기서 퀴즈! - 막해보기



7월10일 ~ 7월19일에 가입한 고객 중,
포인트를 가진 고객의 숫자, 그리고 전체 숫자, 그리고 비율을 보고 싶어요!

아래와 같은 결과를 보고 싶다면 어떻게 해야할까요?

- 이렇게 저렇게 해볼까요?

힌트1 → **count** 은 NULL을 세지 않는답니다!

힌트2 → Alias(별칭)도 잘 붙여주세요!

힌트3 → 비율은 소수점 둘째자리에서 반올림!

	pnt_user_cnt	tot_user_cnt	ratio
1	82	202	0.41

▼ 정답 쿼리 살펴보기

```
select count(point_user_id) as pnt_user_cnt,
       count(*) as tot_user_cnt,
       round(count(point_user_id)/count(*),2) as ratio
from users u
left join point_users pu on u.user_id = pu.user_id
where u.created_at between '2020-07-10' and '2020-07-20'
```

08. 결과물 합치기! Union 배우기

▼ 17) Select를 두 번 할 게 아니라, 한번에 모아서 보고싶은 경우, 있을걸요!

- 예를 들면 이렇게!



근데, 그러려면 한 가지 조건이 있어요! 노란색과 파란색 박스의 필드명이 같아야 합니다. 😊 (당연하겠죠?)



▼ 18) Union을 이용해서 아래와 같은 모습을 만들어볼까요?

▼ [코드스니펫] - 퀴즈5 쿼리


```
select c1.title, c2.week, count(*) as cnt from courses c1
inner join checkins c2 on c1.course_id = c2.course_id
inner join orders o on c2.user_id = o.user_id
where o.created_at >= '2020-08-01'
group by c1.title, c2.week
order by c1.title, c2.week
```

- 보고 싶은 모습!

month	title	week	cnt
7월	앱개발 종합반	2	17
7월	웹개발 종합반	2	11
7월	앱개발 종합반	1	47
7월	웹개발 종합반	1	30
7월	웹개발 종합반	3	7
7월	앱개발 종합반	3	2
8월	앱개발 종합반	1	6
8월	앱개발 종합반	2	3
8월	웹개발 종합반	2	2
8월	웹개발 종합반	1	14

- 우선, 'month'를 붙여줘야겠네요!

```
select '7월' as month, c.title, c2.week, count(*) as cnt from checkins c2
inner join courses c on c2.course_id = c.course_id
inner join orders o on o.user_id = c2.user_id
where o.created_at < '2020-08-01'
group by c2.course_id, c2.week
order by c2.course_id, c2.week
```

- 여기에 아래 위로 **Union all** 을 사용해 붙여주면 끝!

```
(
  select '7월' as month, c.title, c2.week, count(*) as cnt from checkins c2
  inner join courses c on c2.course_id = c.course_id
  inner join orders o on o.user_id = c2.user_id
  where o.created_at < '2020-08-01'
  group by c2.course_id, c2.week
  order by c2.course_id, c2.week
)
union all
(
  select '8월' as month, c.title, c2.week, count(*) as cnt from checkins c2
```

```

inner join courses c on c2.course_id = c.course_id
inner join orders o on o.user_id = c2.user_id
where o.created_at > '2020-08-01'
group by c2.course_id, c2.week
order by c2.course_id, c2.week
)

```

- 앗, 그런데, 한 가지! 정렬이 깨졌네요!? 😅



네 맞습니다! union을 사용하면 내부 정렬이 먹지 않아요.
이 때 유용한 방법이 있지요. 바로, SubQuery(서브쿼리) !

- 그런데 이번주는 꼭 쉬구요! 4주차에 나올 거니까 기대해주세요!

09. 끝 & 숙제 설명



이번 3주차도 고생 많았습니다. 역시 한 주의 마무리는 숙제죠!

숙제: enrolled_id별 수강완료(done=1)한 강의 갯수를 세어보고, 완료한 강의 수가 많은 순서대로 정렬해보기. user_id도 같이 출력되어야 한다.

▼ 힌트!

- 조인해야 하는 테이블: `enrolleds`, `enrolleds_detail`
- 조인하는 필드: `enrolled_id`

▼ 결과

	enrolled_id	user_id	max_count
1	5f12644e3476b5000967d439	3e057f9e	43
2	5f1456ac088e090009bd6188	6a4e3c73	43
3	5f1980ea0a4f8f00098cb593	f0fc6b71	43
4	5f1a716d0a4f8f00098cb7f7	36b19951	43
5	5f1af6ba0583c40009c4947a	326fa9eb	43
6	5f2231fbd4ec030009e2f4a5	25c45557	43
7	5f2611a6e2218200098dbd6d	96556738	43
8	5f26943be462a000096e754e	c9ea3998	43
9	5f1aeb110583c40009c49426	60348256	41
10	5f16e31007475700090890ad	10d856a7	39
11	5f1698610747570009088a6a	bcf534af	38

😎 이런 결과가 나오면 정답! 🙌

10. 3주차 숙제 답안 코드

▼ [코드스니펫] - 3주차 숙제 답안 코드

```
select e.enrolled_id,
       e.user_id,
       count(*) as cnt
  from enrolleds e
 inner join enrolleds_detail ed on e.enrolled_id = ed.enrolled_id
 where ed.done = 1
 group by e.enrolled_id, e.user_id
 order by cnt desc
```

Copyright © TeamSparta All rights reserved.