



[스파르타코딩클럽] 엑셀보다 쉬운 SQL - 2주차



매 주차 강의자료 시작에 PDF파일을 올려두었어요!

▼ PDF 파일

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/683a9047-c016-4905-9c1e-5860b2bf9f1f/_SQL_2_updated_210621.pdf

[수업 목표]

1. 동일한 범주의 데이터를 묶어서 통계를 내주는 Group by를 이해한다.
2. 출력하는 데이터를 필드의 값으로 정렬하여 출력하는 Order by를 익힌다.
3. 조금 더 복잡한 분석을 위해 자주 사용되는 유용한 문법을 익힌다.

[목차]

01. 오늘 배울 것
02. 범주의 통계를 내주는 Group by
03. SQL 쿼리가 실행되는 순서
04. Group by 기능 알아보기
05. 깔끔한 정렬이 필요할 땐? Order by
06. Where와 함께 사용해보기
07. 같이 샅샅이 보기
08. Order by, Group by 같이 연습해보기
09. 이외 유용한 문법 배워보기
10. 끝 & 숙제 설명
11. 2주차 숙제 답안 코드



모든 토글을 열고 닫는 단축키

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

Mac : **⌘** + **⌥** + **t**

01. 오늘 배울 것

▼ 1) 우리는 데이터에서 무엇이 궁금할까?

▼ 통계: 최대 / 최소 / 평균 / 개수



데이터 분석의 목적: 쌓여있는 날것의 데이터 → 의미를 갖는 '정보'로의 변환

- 데이터베이스 테이블에 저장된 데이터: 쌓여있는 날것의 데이터
- 가장 많은 Like를 받은 사람의 이름, 전체 신청자수, 평균 연령: 의미있는 '정보'



더 나아가면? '범주 (category)' 각각의 정보가 궁금할 수 있습니다

- 예) 과목별 신청자 평균 연령, 과목별 신청자수, 성씨별 회원수 등

▼ 통계 구하기: 기존 방법의 한계



우선 과목별 신청자수를 구한다고 생각해봅시다! 자, 지금까지 배운 내용을 사용해볼까요?

- 쿼리를 어떻게 작성하면 좋을까요?

▼ 과목별 신청자 수 - 쿼리 예시 보기

```
select count(*) from orders
where course_title = "앱개발 종합반";
```

```
select count(*) from orders
where course_title = "웹개발 종합반";
```

- 총 두 개의 과목이 있으니, 두 개의 쿼리를 작성해서 각각 이렇게 구할 수 있겠죠?



이번에는 성씨별 회원수를 구하고 싶어요! 과목별 신청자수 구했던 것처럼 하면 될까요?

앗, 그런데 스파르타 회원의 성씨가 총 몇개였죠?

- 1주차에서 쿼리로 직접 확인했듯이, 스파르타 회원의 성씨는 총 54개예요.
- 음.. 그렇다면, 성씨별 회원수를 구하려면 총 54번의 쿼리를 작성해야 할까요?

▼ 동일한 범주의 데이터를 묶어주는 Group by



이렇게 불필요한 반복작업을 하도록 프로그래머들이 가만두지 않았겠죠?

그래서 SQL에는 Group by라는 문법이 있습니다.



Group by란?

동일한 범주를 갖는 데이터를 하나로 묶어서, 범주별 통계를 내주는 것을 의미해요.

Group by를 이용하면 1) 같은 성씨의 데이터를 하나로 묶고 2) 각 성씨의 회원수를 구할 수 있어요.

▼ 간단하게 맛만 볼까요?

```
select name, count(*) from users
group by name;
```

- 위와 같은 쿼리를 실행하면, 아래와 같은 결과물이 나옵니다.

	ABC name	count(*)
1	이**	85
2	신**	14
3	강**	11
4	정**	16
5	남**	21
6	고**	6
7	문**	6

- 성씨별로 회원이 몇 명인지, 세어진 것을 확인할 수 있죠?



이해가 안 된다고요? 괜찮아요! 이번주 저와 함께 하다보면 곧 이해 되실거예요.

▼ 깔끔하게 데이터를 정렬해보자: Order by



아까 봤던 데이터.. 뭔가 찝찝하지 않았나요?

	ABC name	count(*)
1	이**	85
2	신**	14
3	강**	11
4	정**	16
5	남**	21
6	고**	6
7	문**	6

- 85, 14, 11, 16, 21, 6, 6... 뭔가 정렬하고 싶은 욕구가 샘솟지 않나요?



이럴 때, order by라는 기능을 사용하면 이렇게 정렬이 가능합니다! 참 깔끔하죠?

	ABC name	123 count(*)
1	김**	100
2	이**	85
3	박**	32
4	최**	29
5	남**	21
6	임**	17
7	정**	16
8	유**	14
9	신**	14
10	황**	12
11	강**	11



자, 이번주 내용을 맛보니 설레지 않나요? 신발끈 꼭 묶고, 저와 함께 달려보아요!

02. 범주의 통계를 내주는 Group by

▼ 2) 스파르타 회원: 성씨별로 몇 명의 회원이 있는지 알아보자



성씨별로 몇 명이 회원이 있는지 구하려고 where 절을 사용해서 수십개의 쿼리를 작성하는 것은 너무 비효율적입니다. 이 문제를 Group by를 사용해서 어떻게 해결할 수 있을까요?



쿼리는 열 번 듣는 것보다, 한 번 따라해보는게 빨리 배웁니다! 바로 쿼리로 가볼까요?

▼ [코드스니펫] 성씨별 회원수를 Group by로 쉽게 구해보기

```
select name, count(*) from users
group by name;
```

- 아래와 같은 결과가 나옵니다!

	ABC name	123 count(*)
1	이**	85
2	신**	14
3	강**	11
4	정**	16
5	남**	21
6	고**	6
7	문**	6

▼ 어떻게 이렇게 쉽게 되었을까요?

👉 자, 차근차근 쿼리를 같이 볼까요?

```
select name, count(*) from users
group by name;
```

1. from users: users 테이블에서 데이터를 불러옵니다
2. group by name: name이라는 필드에서 동일한 값을 갖는 데이터를 하나로 합쳐줍니다
3. select name, count(*): 이름과 count(*)를 출력해 주는데, 여기서 count(*)는 group by로 합쳐진 데이터의 개수를 세어주는 것입니다!

👉 아직 잘 모르시겠다고요? 괜찮아요! 저와 함께 연습하다보면 어느샌가 익숙해질거예요.

03. SQL 쿼리가 실행되는 순서

▼ 3) Group by 제대로 알아보기: SQL 쿼리가 실행되는 순서

🔥 SQL에서 쿼리가 실행되는 순서를 아는 것은 정말 중요해요. 함께 단계별로 살펴봐요!

```
select name, count(*) from users
group by name;
```



위 쿼리가 실행되는 순서: from → group by → select

1. from users: users 테이블 데이터 전체를 가져옵니다.
2. group by name: users 테이블 데이터에서 같은 name을 갖는 데이터를 합쳐줍니다.
3. select name, count(*): name에 따라 합쳐진 데이터가 각각 몇 개가 합쳐진 것인지 세어줍니다.

예) 이**, 이**, 김**, 김**, 박** 이렇게 데이터가 있었다면, 이**는 2개, 김**은 2개, 박**은 1개겠죠!



코드스니펫과 함께, 예시로 살펴봐요!

▼ [코드스니펫] users 테이블 전체 불러오기

```
select * from users;
```

- 요렇게 나오면 성공!

▼ [코드스니펫] users 테이블에서 '신' 씨를 가진 데이터만 불러와서 개수 살펴보기

```
select * from users  
where name = "신**";
```

- 지난주 배웠던 where를 사용해서 뽑아봤어요. 14개가 나오네요.

▼ [코드스니펫] group by를 사용해서 '신'씨를 가진 데이터가 몇 개인지 살펴보기

```
select name, count(*) from users  
group by name;
```

- 이렇게 나오면 성공! 신씨를 가진 데이터가 14개로 동일하게 나오죠?

select name, count(*) from us | Enter

	ABC name	123 count(*)
1	이**	85
2	신**	14
3	강**	11
4	정**	16
5	남**	21
6	고**	6
7	문**	6



이제 조금 감이 잡히죠? 지금 당장 이해가 안돼도 괜찮아요, 같이 실습해보며 100% 이해해봐요!

04. Group by 기능 알아보기

▼ 4) Group by 기능 알아보기



Group by는 동일한 범주를 갖는 데이터를 하나로 묶어서, **범주별 통계를** 내 주는 것이라고 지난 시간에 배웠습니다. 이번 시간에는 Group by로 통계를 내는 기능을 알아보까요?

▼ 사용할 테이블 알아보기!



이번 수업에서는, 수강생 분들의 '오늘의 다짐'이 담겨있는 checkins 테이블을 사용할거예요.

- 자, select * from checkins limit 10 쿼리를 날려서 테이블 구조를 볼까요?

▼ checkins 테이블 보러가기

	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	5f0ae408765dae0006002817	4b8a10e6	1	월 말에는 그럴 ...	4
2	5f2bca728cf86b0009520d7f	2020-08-06 18:16:34.0	2020-08-07 09:52:15.0	5f0ae408765dae0006002817	fac91699	2	1주차 시작인데...	3
3	5f2bcafd8cf86b0009520dc7	2020-08-06 18:18:54.0	2020-08-07 09:52:13.0	5f0ae408765dae0006002817	c9ea3998	2	개발 너무너무 ...	3
4	5f2bcc0d8cf86b0009520dd1	2020-08-06 18:23:25.0	2020-08-07 09:52:11.0	5f0ae408765dae0006002817	c6752f58	2	우 심지어 앓을 ...	5
5	5f2bce748cf86b0009520e1d	2020-08-06 18:33:40.0	2020-08-06 21:00:43.0	5f0ae408765dae0006002817	4b8a10e6	2	중해서 들다보니 ...	3
6	5f2bd2368cf86b0009520e24	2020-08-06 18:49:42.0	2020-08-07 15:56:08.0	5f0ae408765dae0006002816	c9ea3998	2	1주차 시작인데...	4
7	5f2beb3cb63cef00093e40d3	2020-08-06 20:36:29.0	2020-08-11 09:51:26.0	5f0ae408765dae0006002816	95621907	1	고른데 목요일 저...	3
8	5f2beb48b63cef00093e40d4	2020-08-06 20:36:41.0	2020-08-07 09:51:59.0	5f0ae408765dae0006002817	deb54ce4	1	목할 때 마음가짐...	4
9	5f2bf081f3c92c00099be3ff	2020-08-06 20:58:57.0	2020-08-07 09:52:00.0	5f0ae408765dae0006002817	12dd94fe	1	강보다 퇴근하고 ...	4
10	5f2bf21cb63cef00093e410f	2020-08-06 21:05:48.0	2020-08-11 09:51:25.0	5f0ae408765dae0006002816	76f656d4	1	써 목요일이네요!	3



이번 강의에서 사용할 필드는 요거!

week: 수강생이 '오늘의 다짐'을 남긴 시점의 강의 주차를 의미합니다.

likes: 남긴 '오늘의 다짐' 게시물에 달린 좋아요의 수를 의미합니다.

▼ 동일한 범주의 개수 구하기



동일한 범주의 갯수는 count(*)를 사용해서 해요. 바로 쿼리 결과를 확인해 볼까요?

▼ [코드스니펫] 주차별 '오늘의 다짐' 개수 구하기

```
select week, count(*) from checkins
group by week;
```

- 1주차는 96개, 2주차는 29개, 3주차는 9개가 나와요!

```
select 범주별로 세어주고 싶은 필드명, count(*) from 테이블명
group by 범주별로 세어주고 싶은 필드명;
```

- 이 규칙으로 하면 된답니다. 참 쉽죠? (여기선, week가 범주별로 세어주고 싶은 필드명이죠!)

▼ 동일한 범주에서의 최솟값 구하기



동일한 범주 특정 필드의 최솟값은 min(필드명)을 사용해서 해요. 마찬가지로, 바로 쿼리로 고고!

▼ [코드스니펫] 주차별 '오늘의 다짐'의 좋아요 최솟값 구하기

```
select week, min(likes) from checkins
group by week;
```

```
select 범주가 담긴 필드명, min(최솟값을 알고 싶은 필드명) from 테이블명
group by 범주가 담긴 필드명;
```

- 이 규칙으로 하면 된답니다.
- 여기서는, 범주가 담긴 필드명은 week, 최솟값을 알고 싶은 필드명은 likes 겠죠.

▼ 동일한 범주에서의 최댓값 구하기



동일한 범주 특정 필드의 최댓값은 max(필드명)을 사용해서 해요. 마찬가지로, 바로 쿼리로 고고!

▼ [코드스니펫] 주차별 '오늘의 다짐'의 좋아요 최댓값 구하기

```
select week, max(likes) from checkins
group by week;
```

```
select 범주가 담긴 필드명, max(최댓값을 알고 싶은 필드명) from 테이블명
group by 범주가 담긴 필드명;
```

- 이 규칙으로 하면 된답니다. 최솟값 구할때와 달라진건 하나밖에 없어요.
- 여기서는, 범주가 담긴 필드명은 week, 최대값을 알고 싶은 필드명은 likes 겠죠.

▼ 동일한 범주의 평균 구하기



동일한 범주 특정 필드의 평균값은 avg(필드명)을 사용해서 해요. 마찬가지로, 바로 쿼리로 고고!

▼ [코드스니펫] 주차별 '오늘의 다짐'의 좋아요 평균값 구하기

```
select week, avg(likes) from checkins
group by week;
```

```
select 범주가 담긴 필드명, avg(평균값을 알고 싶은 필드명) from 테이블명
group by 범주가 담긴 필드명;
```

- 이 규칙으로 하면 된답니다. 최솟값 구할 때와 달라진건 하나밖에 없어요.
- 여기서는, 범주가 담긴 필드명은 week, 평균값을 알고 싶은 필드명은 likes 겠죠.

▼ 동일한 범주의 합계 구하기



동일한 범주 특정 필드의 합계는 sum(필드명)을 사용해서 해요. 마찬가지로, 바로 쿼리로 고고!

▼ [코드스니펫] 주차별 '오늘의 다짐'의 좋아요 합계 구하기

```
select week, sum(likes) from checkins
group by week;
```

```
select 범주가 담긴 필드명, sum(합계를 알고 싶은 필드명) from 테이블명
group by 범주가 담긴 필드명;
```

- 이 규칙으로 하면 된답니다. 최솟값 구할 때와 달라진건 하나밖에 없어요.
- 여기서는, 범주가 담긴 필드명은 week, 합계를 알고 싶은 필드명은 likes 겠죠.

05. 깔끔한 정렬이 필요할 땐? Order by

▼ 5) Order by로 앞의 결과를 정렬해보자



뭔가 찝찝했던 성씨별 회원수 데이터

85, 14, 11, 16, 21, 6, 6.. 뭔가 정렬해주고 싶다는 강한 욕구가 들지는 않았나요?

	ABC name	123 count(*)
1	이**	85
2	신**	14
3	강**	11
4	정**	16
5	남**	21
6	고**	6
7	문**	6



Order by를 사용하면 한 번에 정렬할 수 있어요. 같이 쿼리를 살펴볼까요?

▼ [코드스니펫] 원본 쿼리 살펴보기

```
select name, count(*) from users
group by name;
```

▼ [코드스니펫] 결과의 개수 오름차순으로 정렬해보기

```
select name, count(*) from users
group by name
order by count(*);
```

	ABC name	123 count(*)
1	염**	1
2	천**	1
3	경**	1
4	길**	1
5	도**	1
6	현**	1
7	나**	1
8	왕**	1
9	진**	1
10	노**	2

- 아까 코드에 order by count(*) 만 추가해줬어요. 갯수 (count(*) 값)을 기준으로 정렬해달라는 뜻이에요.



참 쉽죠? 그렇다면, 내림차순으로 정렬은 어떻게 할까요?

▼ [코드스니펫] 결과의 개수 내림차순으로 정렬해보기

```
select name, count(*) from users
group by name
order by count(*) desc;
```

	ABC name 🔽	123 count(*) 🔽
1	김**	100
2	이**	85
3	박**	32
4	최**	29
5	남**	21
6	임**	17
7	정**	16
8	유**	14
9	신**	14
10	황**	12
11	강**	11

- 이걸 더 쉽죠? order by count(*)에 desc만 붙여줬어요.



[꿀팁!] 여기서의 desc는 내림차순을 의미하는 영단어 descending의 약자입니다.

▼ 6) Order by 사용해보기



Order by는 모든 SQL 쿼리에 적용될 수 있는 기능입니다.

▼ 바로 이렇게요!



like를 많이 받은 순서대로 '오늘의 다짐'을 출력해 볼까요?

```
select * from checkins
order by likes desc;
```

- 적게 받은 순서대로는 보기 위해서는, 맨 끝의 desc를 제거해 주면 돼요!

▼ 규칙을 살펴볼까요?

```
select * from 테이블명
order by 정렬의 기준이 될 필드명;
```

- 바로 위 예시에서는, like의 갯수가 정렬의 기준이 되는 필드명이겠죠!

▼ 7) Order by 제대로 알아보기: SQL 쿼리가 실행되는 순서



에러가 안 나는 쿼리를 작성하기 위해서는 SQL 쿼리가 실행되는 순서를 아는 것이 중요해요!

```
select name, count(*) from users
group by name
order by count(*);
```



위 쿼리가 실행되는 순서: from → group by → select → order by

1. from users: users 테이블 데이터 전체를 가져옵니다.
2. group by name: users 테이블 데이터에서 같은 name을 갖는 데이터를 합쳐줍니다.
3. select name, count(*): name에 따라 합쳐진 데이터가 각각 몇 개가 합쳐진 것인지 세어줍니다.
예) 이**, 이**, 김**, 김**, 박** 이렇게 데이터가 있었다면, 이**는 2개, 김**은 2개, 박**은 1개겠죠!
4. order by count(*): 합쳐진 데이터의 개수에 따라 오름차순으로 정렬해줍니다.



저랑 차근차근, 같이 해봐요!

06. Where와 함께 사용해보기

▼ 8) Where와 Group by, Order by 함께 사용해보기



원리는 간단해요. Where절로 조건이 하나 추가되고, 그 이후에 Group by, Order by가 실행되는 것!

▼ 웹개발 종합반의 결제수단별 주문건수 세어보기



자, 조금 감이 오나요? 차근차근 한단계씩 생각해봐요!



[순서]

1. orders 테이블에서 주문 데이터를 읽어오고
2. 웹개발 종합반 데이터만 남기고
3. 결제수단(범주) 별로 그룹화하고
4. 결제수단별 주문건수를 세어준다!

▼ 쿼리를 살펴볼까요?

```
select payment_method, count(*) from orders
where course_title = "웹개발 종합반"
group by payment_method;
```

- 위와 같이, group by와 select 사이에 where로 조건을 넣어주면 끝!

▼ 9) 더 알아보기: SQL 쿼리가 실행되는 순서



에러가 안 나는 쿼리를 작성하기 위해서는 SQL 쿼리가 실행되는 순서를 아는 것이 중요해요!

```
select payment_method, count(*) from orders
where course_title = "웹개발 종합반"
```

```
group by payment_method;
```



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

1. from orders: users 테이블 데이터 전체를 가져옵니다.
2. where course_title = "웹개발 종합반": 웹개발 종합반 데이터만 남겨줍니다.
3. group by payment_method: 같은 payment_method을 갖는 데이터를 합쳐줍니다.
4. select payment_method, count(*): payment_method에 따라 합쳐진 데이터가 각각 몇 개가 합쳐진 것인지 세어줍니다.

예) CARD, CARD, kakaopay 이렇게 데이터가 있었다면, CARD는 2개, kakaopay는 1개겠죠!



만약 order by가 추가된다면? order by는 맨 나중에 실행됩니다! (결과물을 정렬해주는 것이기 때문!)

07. 같이 삽질해보기

▼ 10) 혼자서도 문제를 해결하려면



다시 돌아온 같이 삽질해보기 시간!

SQL을 사용하다보면 예상하지 못했던 에러나 결과를 자주 마주하게 됩니다.
이에 대한 해결책을 모두 외우는 것은 현실적으로 불가능하겠죠?

- 자주 발생하는 문제, 저와 같이 삽질하면서 해결해봐요!

▼ 왜 이런 결과가 나왔을까요?


```
select * from orders
group by payment_method
```

		ABC course_title	ABC user_id	ABC payment_method	ABC email
1	2817	앱개발 종합반	49905ce9	kakaopay	python499@yahoo.com
2	2816	웹개발 종합반	3af5bc14	CARD	sparta3af@hanmail.com
3	2816	웹개발 종합반	b4bc2fc8	TAXBILL	spartab4b@naver.com
4	2816	웹개발 종합반	2dfbb19f	MONEY	hello2df@gmail.com

👉 범주에 따른 통계치를 구하고 싶어 group by를 사용해 보았습니다.
하지만, 통계치는 나오지 않고 4개의 데이터만 출력된 것을 알 수 있죠.

- 왜 그럴까요?

👉 범주별로 묶어달라는 명령어 (group by)는 작성해줬지만,
묶은 데이터를 '어떤 통계치'로 출력해달라는 명령어가 없어서 그렇습니
다!

- 그러면 어떻게 하면 좋을까요?

👉 '어떤 통계치로' 출력해달라는 명령어를 추가하면 됩니다. 범주별 갯수를
세어볼까요? 아래와 같이요!

```
select count(*) from orders
group by payment_method
```

	count(*)
	56
	218
	4
	8

- 앗, 뭔가 원하는 통계치가 나온 것 같지만 각각 어떤 범주에 대한 통계치인지는 나와있지 않아요.
- 그도 당연한 것이, select 문 안에 count(*)만 적혀있어서 그렇겠죠?



그렇다면 group by에 들어간 필드를 똑같이 적어주면 됩니다. 깔끔하게 원하는 데이터가 잘 나오는걸 확인할 수 있습니다!

```
select payment_method, count(*) from orders
group by payment_method
```

payment_method	count(*)
kakaopay	56
CARD	218
TAXBILL	4
MONEY	8



원하는 결과가 출력되지 않을 땐, SQL 쿼리의 실행순서에 따라 차근차근 생각해봐요.
어디서 쿼리를 잘못 짰는지 찾아내는 가장 빠르고 정확한 방법입니다!

08. Order by, Group by 같이 연습해보기

▼ 11) Order by 연습하기



쉬운 것부터 하나씩 이해하고 가볼까요? 자, Order by 연습 시작합니다!

▼ 문자열을 기준으로 정렬해보기

```
select * from users
order by email;
```

- 동일하게 알파벳으로도 잘 정렬이 됩니다.

```
select * from users
order by name;
```

- 그리고, 한글로도 잘 정렬이 됩니다. 참 쉽죠?

▼ 시간을 기준으로 정렬해보기

```
select * from users
order by created_at desc;
```

- 최근 데이터부터 보고싶을 때, 유용해요!

▼ 12) Group by 연습하기



자, Group by는 퀴즈를 풀어볼까요?

(1) 원하는 테이블과 (2) 범주로 사용할 필드 (3) 범주에 따라 통계를 계산하고 싶은 필드 (개수의 경우 제외)

이 세 가지만 기억하면 됩니다!

▼ [퀴즈] 앱개발 종합반의 결제수단별 주문건수 세어보기



[힌트!] (1) 원하는 테이블: orders (2) 범주로 사용할 필드: payment_method


	ABC payment_method	123 count(*)
1	kakaopay	34
2	CARD	111
3	MONEY	5
4	TAXBILL	3

- 이렇게 출력돼야 해요!

▼ 정답 쿼리 살펴보기!

```
select payment_method, count(*) from orders
where course_title = "앱개발 종합반"
group by payment_method;
```

▼ [퀴즈] Gmail 을 사용하는 성씨별 회원수 세어보기

 **[힌트!]** where와 like를 함께 쓰면 돼요!

	ABC name 🔼🔽	123 count(*) 🔼🔽
1	신**	7
2	이**	29
3	강**	5
4	김**	32
5	심**	3
6	남**	5
7	문**	1
8	손**	1
9	윤**	3
10	안**	1
11	권**	4
12	박**	7
13	유**	5
14	최**	8
15	황**	4
16	정**	6
17	송**	2
18	한**	2
19	염**	1
20	구**	1
21	임**	2
22	우**	1
23	천**	1
24	장**	2

- 요렇게 나와야 합니다~~!

▼ 정답 쿼리 살펴보기

```
select name, count(*) from users
where email like '%gmail.com'
group by name;
```

▼ [퀴즈] course_id별 '오늘의 다짐'에 달린 평균 like 개수 구해보기



요건 힌트 없이 혼자 해보세요!

	ABC course_id	123 avg(likes)
1	5f0ae408765dae0006002817	2.5
2	5f0ae408765dae0006002816	1.6129

- 요렇게 나와야 합니다~~!

▼ 정답 쿼리 살펴보기

```
select course_id, avg(likes) from checkins
group by course_id;
```

▼ 13) [꿀팁] 이렇게 쿼리를 작성하면 편해요!



- 1) show tables로 어떤 테이블이 있는지 살펴보기
- 2) 제일 원하는 정보가 있을 것 같은 테이블에 select * from 테이블명 limit 10 쿼리 날려보기
- 3) 원하는 정보가 없으면 다른 테이블에도 2)를 해보기
- 4) 테이블을 찾았다! 범주를 나눠서 보고싶은 필드를 찾기
- 5) 범주별로 통계를 보고싶은 필드를 찾기
- 6) SQL 쿼리 작성하기!



참 쉽죠? 여러분은 이제 범주에 따라 통계치를 계산할 수 있는 사람이 되었습니다! 🙌

09. 이외 유용한 문법 배워보기

▼ 14) 별칭 기능: Alias



쿼리가 점점 길어지면서 종종 헷갈리는 일이 생길 수 있습니다. 그래서 SQL은 Alias라는 별칭 기능을 지원합니다.

- 지금 당장은 안 쓰지만, 다음 주차부터 유용하게 쓸거예요!

▼ 백문이 불여일견! 직접 SQL 쿼리를 볼까요?

```
select * from orders o
where o.course_title = '앱개발 종합반'
```

- 요렇게 테이블명 뒤에 as를 붙여서 별칭을 추가하는 것도 가능하고,

```
select payment_method, count(*) as cnt from orders o
where o.course_title = '앱개발 종합반'
group by payment_method
```

- 출력될 필드에 별칭을 붙이는 것도 가능해요! 그럼 아래와 같이 출력됩니다.

	ABC payment_method	123 cnt
1	kakaopay	34
2	CARD	111
3	MONEY	5
4	TAXBILL	3

- count(*)가 아니라 cnt로 출력되었네요!



이처럼, 혼동을 최소화하고 원하는 이름으로 결과를 출력하기 위해 사용돼요. 이 기능은 다음 주차부터 굉장히 유용하게 사용되니, '이런게 있구나' 정도로만 기억해 주세요!

10. 끝 & 숙제 설명



모두 2주차 고생 많았습니다! 이제, 즐거운 숙제를 해볼까요?

숙제: 네이버 이메일을 사용하여 앱개발 종합반을 신청한 주문의 결제수단별 주문건수 세어보기

▼ 결과

	ABC payment_method	123 count(*)
1	CARD	38
2	kakaopay	10
3	TAXBILL	2

😎 이런 결과가 나오면 정답! 🙌

11. 2주차 숙제 답안 코드

▼ [코드스니펫] - 2주차 숙제 답안 코드

```
select payment_method, count(*) from orders
where email like '%naver.com' and course_title = '앱개발 종합반'
group by payment_method
```

Copyright © TeamSparta All rights reserved.