



[스파르타코딩클럽] 엑셀보다 쉬운 SQL - 1주차



매 주차 강의자료 시작에 PDF파일을 올려두었어요!

▼ PDF 파일

[수업 목표]

1. 데이터베이스(DB)와 SQL이 왜 필요한지 이해한다.
2. DB에서 데이터를 가져오는 기초적인 문법인 Select와 Where문을 이해한다.
3. 조금 더 복잡한 분석을 위해 자주 사용되는 유용한 문법을 익힌다.

[목차]

01. 오늘 배울 것
02. 필수 프로그램 설치
03. Select 쿼리문이란?
04. 4주동안 사용할 스파르타 데이터베이스 살펴보기
05. 특정 데이터만 가져오고 싶다면?
06. Where 절 같이 연습해보기
07. Where 절과 자주 같이 쓰는 문법 써보기
08. 이외 유용한 문법 배워보기
09. 같이 삼질해보기
10. 퀴즈 풀어보기
10. 끝 & 숙제 설명
11. 1주차 숙제 답안 코드



모든 토글을 열고 닫는 단축키

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

Mac : **⌘** + **⌥** + **t**

01. 오늘 배울 것

▼ 1) 환영합니다!



환영합니다 여러분!

저는 '엑셀보다 쉬운 SQL' 튜터를 맡은 이범규입니다.

여러분들과 4주동안 SQL을 함께 연습할 생각을 하니 가슴이 두근두근하네요.



개인적인 경험

맨 처음 SQL을 접했을 때, 너무 헷갈리고 어려웠던 개념들.

SQL 쿼리를 작성하기 전에 익혀야 할 개념이 너무나도 많았습니다.

예) 관계형 데이터베이스, ERD, 스키마, 엔티티 등..

- 아쉽게도, 이런 개념이 실무에서 그렇게 도움이 되지 않았습니다.



SQL을 잘 하기 위해서는,

복잡한 개념을 모두 이해하는 방식이 아니라, 일단 반복적으로 써보면서 감과 원리를 익혀가는 것이 중요하다고 느꼈습니다.

이 수업은

SQL의 쉬운 개념부터 어려운 개념까지,
SQL 쿼리를 저와 함께 작성해보며 '같이 삽질하며 익히는' 수업입니다.

저는, 이러한 학습법이 SQL을 잘하게 되기까지의 지름길이라 확신합니다.

- 자, 이제 본격적으로 시작해볼까요?

▼ 2) 데이터베이스(DB)와 SQL이 왜 필요할까?

▼ 1. 내가 배달음식점의 사장이라면?



자, 배달음식점 사장이 되었다고 상상해봅시다!
아직까지는 규모가 크지 않아서, 혼자서 이렇게 엑셀 시트에 주문내역을
정리하고 있어요.

| | A | B | C | D | E |
|---|------------|-----|-------|-------|------|
| 1 | 주문날짜 | 회원명 | 음식명 | 결제금액 | 상태 |
| 2 | 2020.09.01 | 이*석 | 제육볶음 | 7000 | 배송완료 |
| 3 | 2020.09.01 | 이*규 | 제육볶음 | 7000 | 배송완료 |
| 4 | 2020.09.01 | 남*관 | 김치찌개 | 6000 | 배송완료 |
| 5 | 2020.09.01 | 이*은 | 제육볶음 | 7000 | 배송완료 |
| 6 | 2020.09.01 | 이*석 | 오겹살구이 | 10000 | 준비중 |
| 7 | | | | | |



그런데 어느날, 장사가 엄청 잘 되기 시작했어요!
이제는 혼자서는 힘들어져서, 주문을 기록해주고 일을 도와줄 직원 여러
명을 채용했어요.

| | A | B | C | D | E |
|--------|------------|-----|-------|-------|------|
| 100983 | 2020.09.05 | 이*석 | 제육볶음 | 7000 | 배송완료 |
| 100984 | 2020.09.05 | 이*규 | 제육볶음 | 7000 | 배송완료 |
| 100985 | 2020.09.05 | 남*관 | 김치찌개 | 6000 | 배송완료 |
| 100986 | 2020.09.05 | 이*은 | 제육볶음 | 7000 | 배송완료 |
| 100987 | 2020.09.05 | 이*석 | 오겹살구이 | 10000 | 준비중 |
| 100988 | 2020.09.05 | 황*영 | 김치찌개 | 6000 | 배송완료 |
| 100989 | 2020.09.05 | 신*원 | 제육볶음 | 7000 | 배송완료 |
| 100990 | 2020.09.05 | 이*현 | 김치찌개 | 6000 | 배송완료 |
| 100991 | 2020.09.05 | 이*은 | 제육볶음 | 7000 | 배송완료 |
| 100992 | 2020.09.05 | 김*영 | 오겹살구이 | 10000 | 주문취소 |
| 100993 | 2020.09.05 | 이*은 | 제육볶음 | 7000 | 배송완료 |



그런데 문제가 생겼어요!

데이터가 많아지면서 엑셀 시트가 점점 느려지고, 여러사람이 작업하는 내용이 충돌하면서 취소된 주문에 배달이 나가는 등의 문제가 생겨나기 시작했어요.



고민을 하다가, 데이터베이스에 모든 주문데이터를 저장하기 시작했어요.

- 1) 필요한 데이터만 불러와서 작업할 수 있어서 속도는 빨라졌고
- 2) 충돌이 생겼을 경우 에러를 띄우는 기능을 추가해서 충돌로 생기는 문제도 사라졌어요.

- 아주 작은 회사에서는 불필요하겠지만, 많은 양의 데이터를 효과적으로 저장/수정/사용하기 위해서는 데이터베이스가 필요합니다. 그래서 거의 모든 회사의 데이터는 데이터베이스에 저장되어 있어요.
- 이 외에도, 데이터베이스가 사용자에게 주는 이점은 정말 많답니다! 오늘은 첫 날이니, 이 정도만 살펴보아요.

▼ 2. 데이터베이스 기초 개념



데이터베이스란? 여러 사람들이 같이 사용할 목적으로 데이터를 담은 통이라고 생각하면 돼요.



원하는대로 데이터를 저장하고 사용하기 위해서는 다양한 기능이 있어야겠죠? 그래서 통상적으로, 모든 데이터베이스는 CRUD에 해당하는 기능을 지원해요.

C (Create): 데이터의 생성을 의미합니다

R (Read): 저장된 데이터를 읽어오는 것을 의미해요

U (Update): 저장된 데이터를 변경!

D (Delete): 저장된 데이터를 삭제하는 것을 의미해요



점점 어려워진다고요? 괜찮아요!

이 강의에서는, 저장된 데이터를 읽어오는 "R (Read)"에 대해서만 배울 것입니다.

▼ 3. SQL이 왜 필요할까?



자, 그럼 SQL은 왜 필요할까요?

데이터를 읽어오는 과정인 "R (Read)"를 엄청나게 편하게 만들어줍니다. 그리고, 데이터를 손쉽게 깔끔하게 정리/분석하는 기능도 지원하는답니다!

열심히 저장된 데이터, 잘 가져다 써야겠죠?

👉 SQL은 Structured Query Language의 약자인데요, 결국 데이터베이스에 요청 (Query)을 날려서 원하는 데이터를 가져오는 것을 도와주는 언어라는 의미예요!

select * from orders

| | | ABC course_title | ABC user_id | ABC payment_method | ABC email |
|----|----|------------------|-------------|--------------------|-----------------------|
| 1 | 17 | 앱개발 종합반 | 49905ce9 | kakaopay | python499@yahoo.com |
| 2 | 16 | 웹개발 종합반 | 3af5bc14 | CARD | sparta3af@hanmail.com |
| 3 | 17 | 앱개발 종합반 | 856ff83f | kakaopay | hello856@hanmail.com |
| 4 | 17 | 앱개발 종합반 | 1078f508 | CARD | coolcode107@yahoo.com |
| 5 | 17 | 앱개발 종합반 | 996b5585 | kakaopay | abc996@gmail.com |
| 6 | 16 | 웹개발 종합반 | 65b2fd34 | kakaopay | abc65b@gmail.com |
| 7 | 16 | 웹개발 종합반 | 1f301966 | CARD | python1f3@hanmail.com |
| 8 | 17 | 앱개발 종합반 | efaf5439 | CARD | sqlgoodefa@naver.com |
| 9 | 17 | 앱개발 종합반 | 1f301966 | CARD | python1f3@hanmail.com |
| 10 | 17 | 앱개발 종합반 | 0fe500cc | CARD | coolcode0fe@naver.com |
| 11 | 16 | 웹개발 종합반 | 03667091 | CARD | qwerty036@naver.com |
| 12 | 17 | 앱개발 종합반 | f7138c43 | CARD | mysqlf71@gmail.com |
| 13 | 17 | 앱개발 종합반 | f6f3059e | CARD | spartaf6f@daum.net |



짧은 SQL 쿼리 한 줄이면 저장된 데이터를 이렇게 간단히 가져올 수 있어요!

이렇게, 데이터를 가져오는 명령어를 작성하는 것을 'SQL 쿼리를 작성한다' 라고 한답니다.

▼ 4. 1~4주차에 배울 순서!

▼ 1주차: Select, Where



데이터를 불러오고 (Select), 조건에 맞게 필터링 하는 것 (Where)!

▼ 2주차: Group by, Order by



데이터를 범주에 따라 묶어서 통계치를 구하고 (Group by), 정렬하는 것 (Order by)!

▼ 3주차: Join



여러 데이터를 합쳐서 분석하기 (Join), 더 쉽고 깔끔하게 원하는 데이터를 얻기 (Subquery)

▼ 4주차: Subquery, 그 외



실전의 데이터는 생각보다 잘 정리되어 있지 않습니다! 데이터 분석을 위해서 데이터를 원하는 형태로 정리하는 방법을 배워요.

02. 필수 프로그램 설치

▼ 3) DBeaver ([다운로드 링크](#))



DBeaver는 여러분이 SQL을 보다 손쉽게 사용할 수 있도록 도와주는 도구입니다! 🌟😊

▼ ([←눌러보기](#)) DBeaver 다운로드 하기

1. [다운로드 링크](#)를 클릭하세요.
2. Windows를 사용하고 계시면 (**installer**), Mac을 사용하고 계시면 (**dmg**) 라고 되어있는 링크를 클릭하여 DBeaver를 다운로드 받은 후 **설치까지 마무리**해주세요.

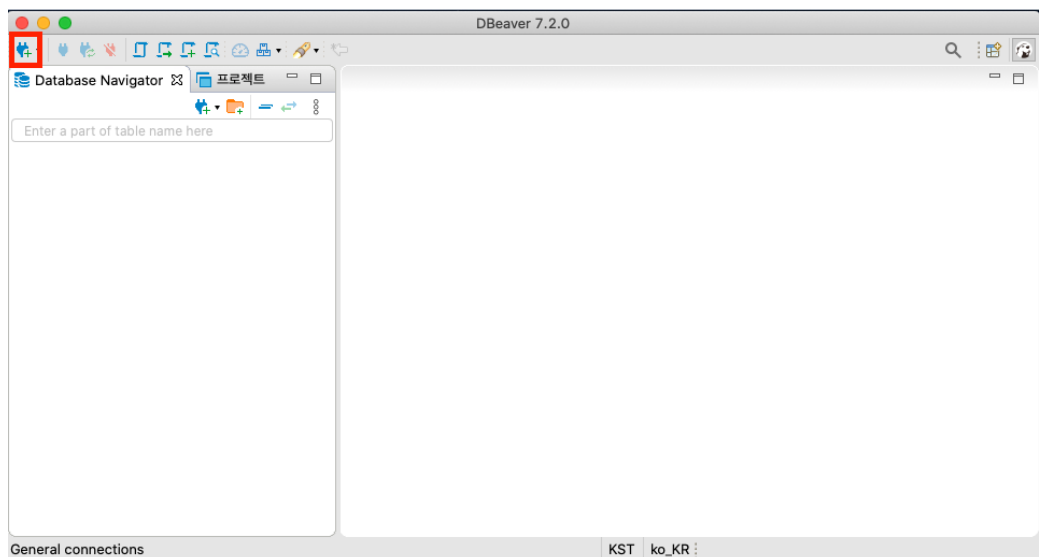
| Community Edition 7.3.4 | Enterprise Edition 7.3.0 |
|---|--|
| Released on February 1, 2021 (Milestones). It is free and open source (license). Also you can get it from the GitHub mirror . | Released on December 7, 2020 EE version web site: dbeaver.com Trial version is available. |
| <div> <div>Windows</div> <ul style="list-style-type: none"> Windows 64 bit (installer) Windows 64 bit (zip) Install from Microsoft Store Chocolatey package <div>Mac OS X</div> <ul style="list-style-type: none"> Mac OS X (dmg) Mac OS X (zip) </div> <ul style="list-style-type: none"> Brew Cask (brew install --cask dbeaver-community) MacPorts (sudo port install dbeaver-community) | <div>Enterprise Edition features:</div> <ul style="list-style-type: none"> Support of NoSQL databases: <ul style="list-style-type: none"> MongoDB Cassandra InfluxDB Redis Amazon DynamoDB Amazon DocumentDB Google Bigtable Couchbase CouchDB Advanced extensions for: <ul style="list-style-type: none"> Oracle SQL Server Netezza |



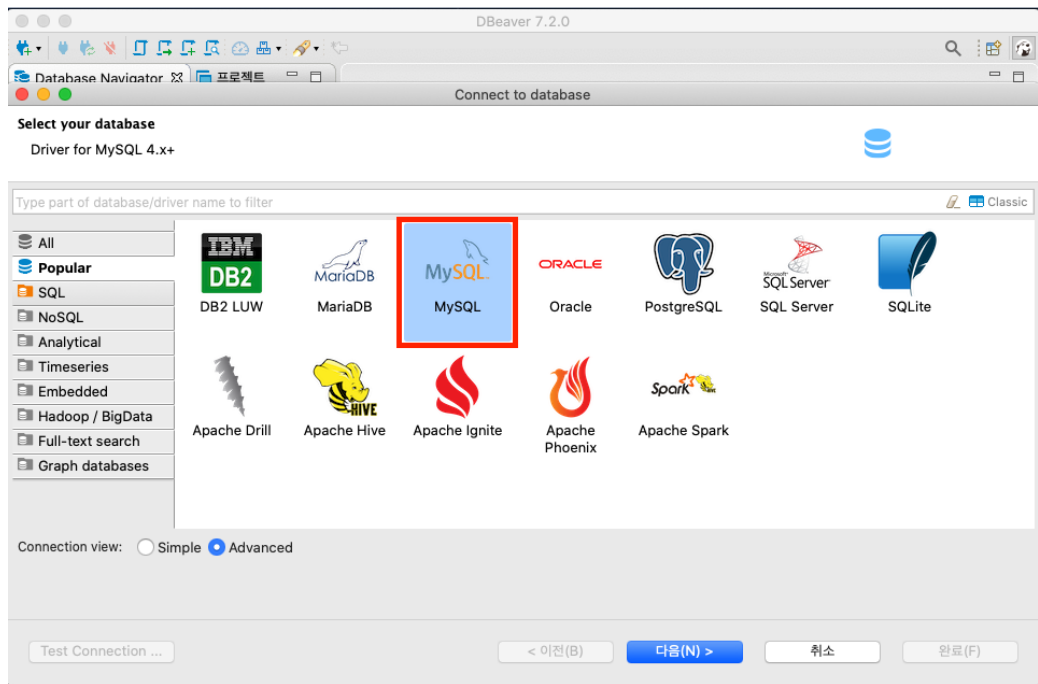
'계속' 버튼을 눌러서 진행해주시면 됩니다!

▼ (←눌러보기) 스파르타 데이터베이스 연결 방법 참고

1. 설치한 DBeaver를 실행합니다
2. 좌측 상단 플러그 모양의 아이콘을 클릭합니다



3. 새 창이 뜨면, MySQL을 선택하고 '다음' 버튼을 누릅니다



4. 아래와 같이 정보를 입력하고 좌측 하단 'Test Connection' 버튼을 클릭합니다.

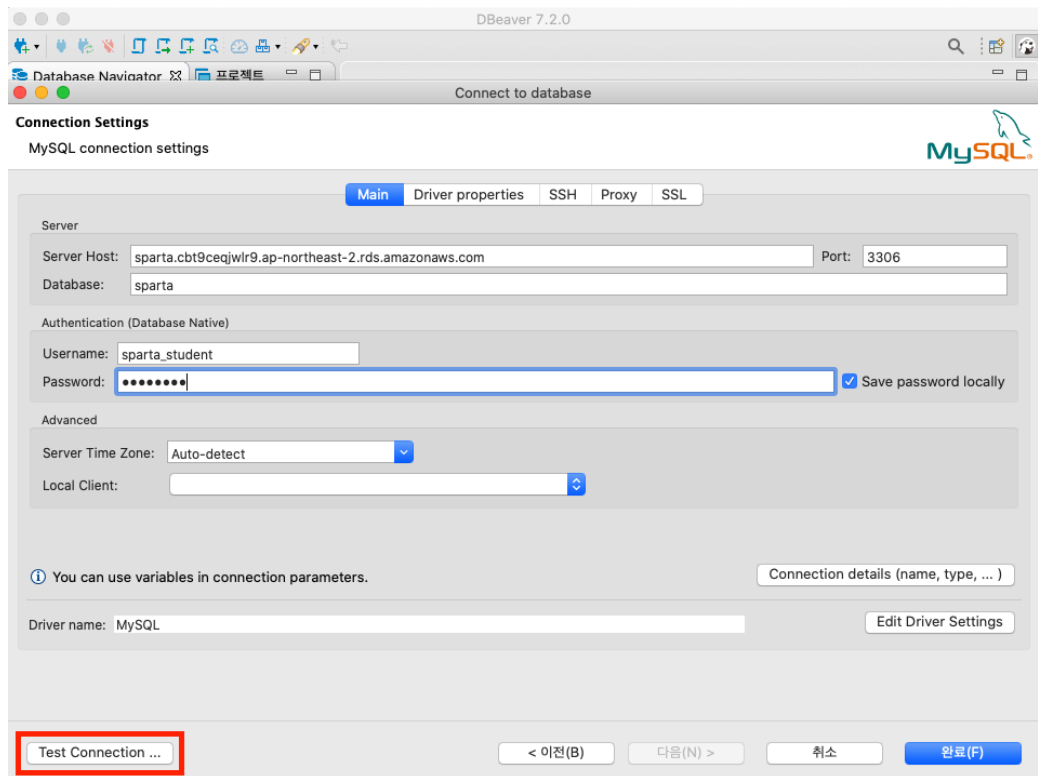


Server Host: sparta.cbt9ceqjwlr9.ap-northeast-2.rds.amazonaws.com

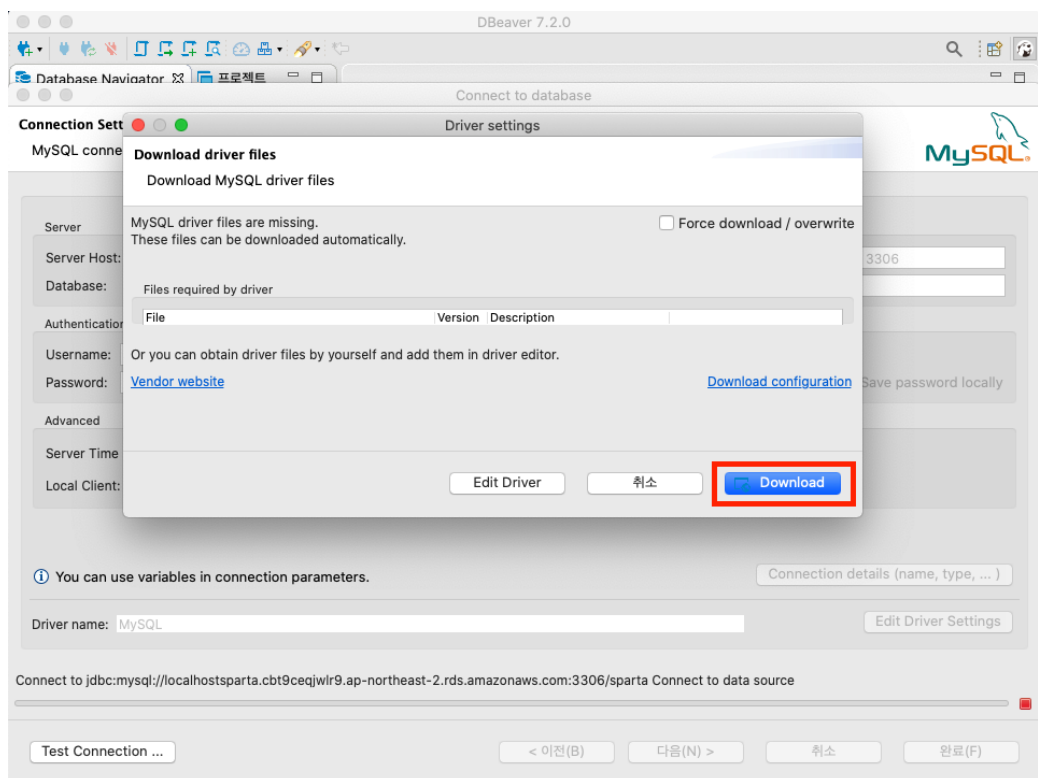
Database: sparta

Username: sparta_student

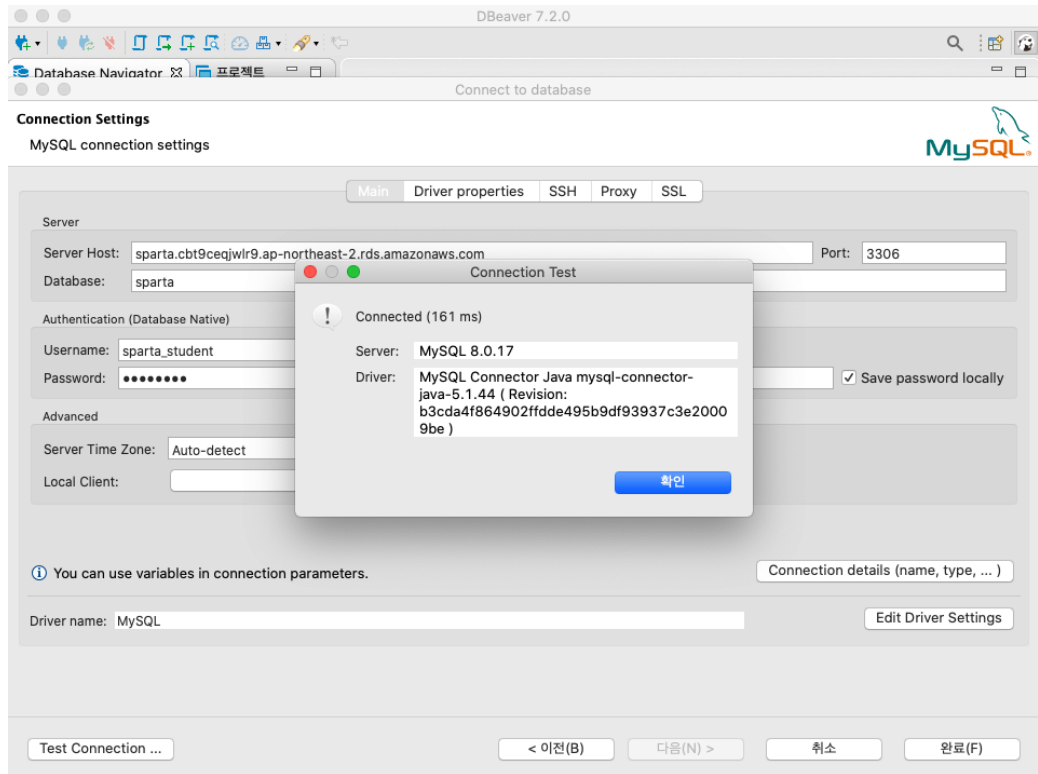
Password: sparta99



5. 만약 Driver가 설치되어있지 않으면 아래와 같은 창이 뜹니다. 다운로드 버튼을 눌러줍니다.



6. Connection Test 결과에 'Connected'라고 뜨면, 확인 버튼을 누르고 완료 버튼을 눌러줍니다.



03. Select 쿼리문이란?

👉 시작하기 전에, sparta_week1.sql 이라는 파일을 먼저 만들어봅시다!

▼ 4) Select 쿼리문의 개념

👉 쿼리(Query)문이란? 쿼리는 질의를 의미하죠. 데이터베이스에 명령을 내리는 것을 의미합니다.

여기서 Select 쿼리문은, 데이터베이스에서 '데이터를 선택해서 가져오겠다'는 의미입니다.

- 4주동안 배울 것은 데이터를 잘 뽑아내는 것이죠? 따라서, 수업의 모든 쿼리문에는 Select가 들어갑니다.
- Select 쿼리문은 1) 어떤 테이블에서 2) 어떤 필드의 데이터를 가져올지 로 구성됩니다.



잠깐! 테이블과 필드가 뭐냐고요? 아래 예시를 보고갈게요.

▼ 테이블과 필드가 뭘까요?

| | A | B | E | F | G | H |
|----|----------|---------------------|--------------|----------|----------------|-----------------------|
| 1 | order_no | created_at | course_title | user_id | payment_method | email |
| 2 | f4884401 | 2020-07-13 20:07:39 | 앱개발 종합반 | 49905ce9 | kakaopay | python499@yahoo.com |
| 3 | 3d3b0d3b | 2020-07-13 22:30:18 | 웹개발 종합반 | 3af5bc14 | CARD | sparta3af@hanmail.com |
| 4 | af831367 | 2020-07-13 22:40:13 | 앱개발 종합반 | 856ff83f | kakaopay | hello856@hanmail.com |
| 5 | 436860dc | 2020-07-14 11:52:56 | 앱개발 종합반 | 1078f508 | CARD | coolcode107@yahoo.com |
| 6 | abb920a1 | 2020-07-14 12:58:49 | 앱개발 종합반 | 996b5585 | kakaopay | abc996@gmail.com |
| 7 | 2920906e | 2020-07-14 13:21:18 | 웹개발 종합반 | 65b2fd34 | kakaopay | abc65b@gmail.com |
| 8 | 1e305953 | 2020-07-14 13:23:27 | 웹개발 종합반 | 1f301966 | CARD | python1f3@hanmail.com |
| 9 | b62e61ca | 2020-07-14 13:26:37 | 앱개발 종합반 | efaf5439 | CARD | sqlgoodefa@naver.com |
| 10 | 22b6558f | 2020-07-14 13:29:19 | 앱개발 종합반 | 1f301966 | CARD | python1f3@hanmail.com |
| 11 | 47b930a1 | 2020-07-14 13:29:41 | 앱개발 종합반 | 0fe500cc | CARD | coolcode0fe@naver.com |

- 테이블: orders라는 엑셀 시트명 보이시죠? 테이블은 데이터가 담긴 엑셀 시트와 동일합니다. 이런 형태의 값이 데이터베이스에 담기면, orders라는 이름의 테이블이 되겠죠.
- 필드: order_no, created_at, course_title, user_id, payment_method, email 각각이 필드입니다.

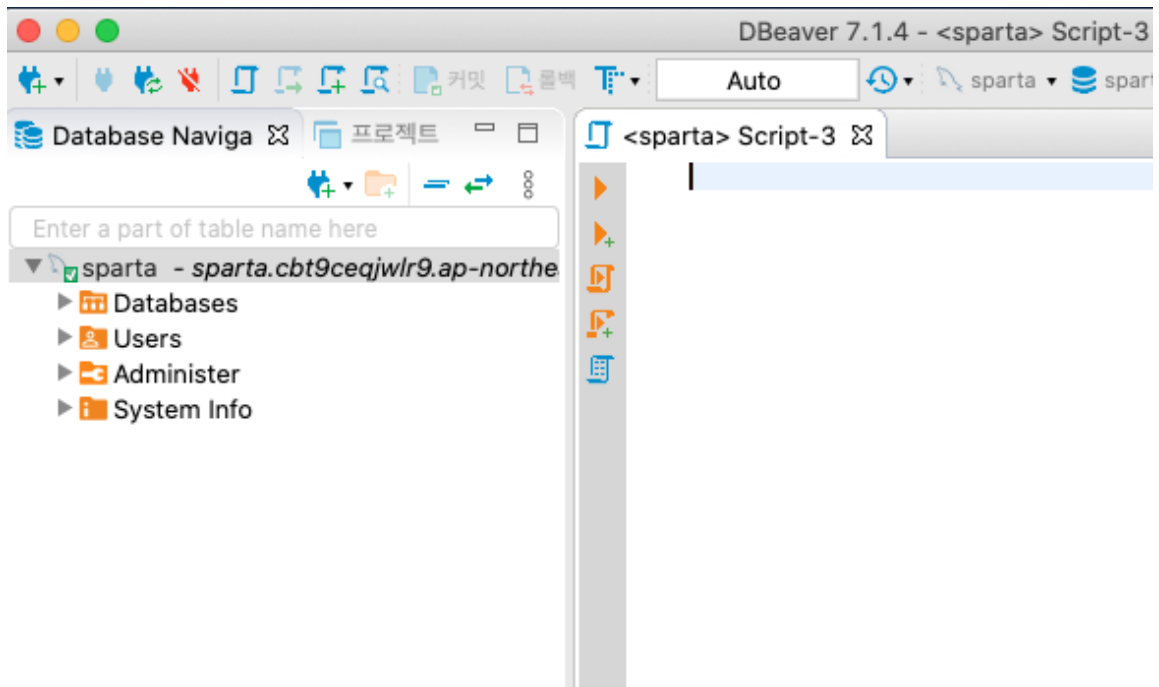


예시) Select 쿼리문을 통해 'orders 테이블의 created_at, course_title, payment_method, email 필드를 가져와줘!'라고 명령을 내릴 수 있습니다.

▼ 5) Select 쿼리문 연습



자, 미리 설치한 DBeaver를 켜볼까요?



- 이런 화면이 나오면 성공!

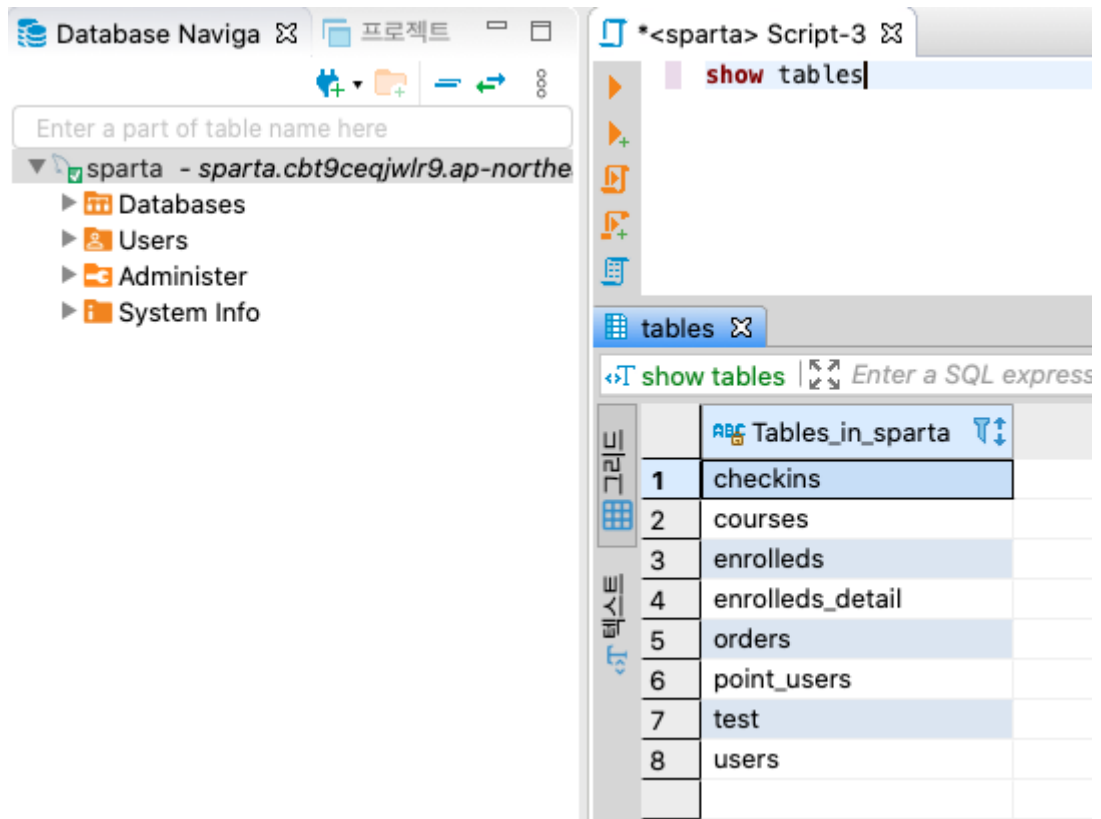


SQL 쿼리를 작성하는 입력창이 없다면, 좌측 상단 양피지 모양을 클릭해서 New Script를 해주세요!

▼ [코드스니펫] - 스파르타 데이터베이스의 테이블 보기

```
show tables;
```

- 아래와 같은 화면이 나오면 성공!



☞ checkins, courses, enrolleds ... 라는 이름의 테이블들이 있다는 이
야깁니다!

▼ [코드스니펫] - orders 테이블의 데이터 가져와보기

```
select * from orders;
```

The screenshot shows a database interface with a sidebar on the left containing 'Databases', 'Users', 'Administer', and 'System Info'. The main area displays the query 'select * from orders;' and a table of results with columns: order_no, created_at, and updated_at.

| | order_no | created_at | updated_at |
|----|----------|-----------------------|-----------------------|
| 1 | f4884401 | 2020-07-13 20:07:39.0 | 2020-07-13 20:10:00.0 |
| 2 | 3d3b0d3b | 2020-07-13 22:30:18.0 | 2020-07-13 22:31:24.0 |
| 3 | af831367 | 2020-07-13 22:40:13.0 | 2020-07-13 22:40:29.0 |
| 4 | 436860dc | 2020-07-14 11:52:56.0 | 2020-07-14 11:53:47.0 |
| 5 | abb920a1 | 2020-07-14 12:58:49.0 | 2020-07-14 12:59:36.0 |
| 6 | 2920906e | 2020-07-14 13:21:18.0 | 2020-07-14 13:22:27.0 |
| 7 | 1e305953 | 2020-07-14 13:23:27.0 | 2020-07-14 13:25:25.0 |
| 8 | b62e61ca | 2020-07-14 13:26:37.0 | 2020-07-14 13:27:24.0 |
| 9 | 22b6558f | 2020-07-14 13:29:19.0 | 2020-07-14 13:31:17.0 |
| 10 | 47b930a1 | 2020-07-14 13:29:41.0 | 2020-07-14 13:31:08.0 |
| 11 | a419e8a6 | 2020-07-14 13:30:27.0 | 2020-07-14 13:31:40.0 |

- 요런 데이터가 나오면 성공!

▼ [코드스니펫] - orders 테이블의 특정 필드만 가져와보기

```
select created_at, course_title, payment_method, email from orders;
```

The screenshot shows the same database interface with the query 'select created_at, course_title, payment_method, email from orders;'. The results table has columns: created_at, course_title, payment_method, and email.

| | created_at | course_title | payment_method | email |
|---|-----------------------|--------------|----------------|--------------|
| 1 | 2020-07-13 20:07:39.0 | 앱개발 종합반 | kakaopay | python499@y |
| 2 | 2020-07-13 22:30:18.0 | 웹개발 종합반 | CARD | sparta3af@ha |
| 3 | 2020-07-13 22:40:13.0 | 앱개발 종합반 | kakaopay | hello856@har |
| 4 | 2020-07-14 11:52:56.0 | 앱개발 종합반 | CARD | coolcode107@ |
| 5 | 2020-07-14 12:58:49.0 | 앱개발 종합반 | kakaopay | abc996@gma |
| 6 | 2020-07-14 13:21:18.0 | 웹개발 종합반 | kakaopay | abc65b@gma |
| 7 | 2020-07-14 13:23:27.0 | 웹개발 종합반 | CARD | python1f3@ha |
| 8 | 2020-07-14 13:26:37.0 | 앱개발 종합반 | CARD | sqlgoodefa@n |
| 9 | 2020-07-14 13:29:19.0 | 앱개발 종합반 | CARD | python1f3@ha |

- 마찬가지로, 이런 데이터가 나오면 성공!



참 쉽죠? 축하합니다! 여러분은 이제 데이터베이스 테이블에서 데이터를 꺼낼 수 있게 되었어요.

04. 4주동안 사용할 스파르타 데이터베이스 살펴보기

▼ 6) 스파르타 데이터베이스의 구조



스파르타 데이터베이스에는 아래와 같은 테이블이 있어요.

- checkins: 여러분이 강의실 들어오시며 남기는 '오늘의 다짐'이 들어있어요
- courses: 스파르타의 개설 강좌 정보가 들어있어요
- enrolleds: 유저별 강좌 등록정보가 들어있어요
- enrolleds_detail: 유저별 들을 수 있는 영상과, 들었는지 여부가 들어있어요
- orders: 주문 (수강등록) 정보가 들어있어요
- point_users: 유저별 포인트 점수가 들어있어요
- users: 유저 정보가 들어있어요



자, 저와 함께 살펴볼까요?

- (소곤소곤) 직접 `select * from 테이블명` 이렇게 쿼리를 날려보셔도 돼요!

05. 특정 데이터만 가져오고 싶다면?

▼ 7) Where 절의 개념



Where 절은, Select 쿼리문으로 가져올 데이터에 조건을 걸어주는 것을 의미해요.

예1) orders 테이블에서 결제수단이 카카오페이인 데이터만 가져와줘!

예2) point_users 테이블에서 포인트가 5000점 이상인 데이터만 가져와줘!

예3) orders 테이블에서 주문한 강의가 앱개발 종합반이면서, 결제수단이 카드인 데이터만 가져와줘!

▼ 8) Select 쿼리문에 Where 절 함께 써보기



쿼리는 열 번 듣는 것보다, 한 번 짜는게 빨리 습니다! 앞의 예시를 같이 해볼까요?

▼ 쿼리를 직접 써보면서 이해해볼까요? 일단 따라 써 봅니다!

```
select * from orders
where payment_method = "kakaopay";
```

- 아주 직관적이죠? "orders 테이블에서 payment_method가 kakaopay인 것만 가져와줘!" 라는 의미입니다.

The screenshot shows a database client interface. The SQL editor contains the query: `select * from orders where payment_method = "kakaopay";`. Below the editor, the results of the query are displayed in a table.

| | course_title | user_id | payment_method | email |
|---|--------------|----------|----------------|----------------------|
| 1 | 개발 종합반 | 49905ce9 | kakaopay | python499@yahoo.com |
| 2 | 개발 종합반 | 856ff83f | kakaopay | hello856@hanmail.com |
| 3 | 개발 종합반 | 996b5585 | kakaopay | abc996@gmail.com |
| 4 | 개발 종합반 | 65b2fd34 | kakaopay | abc65b@gmail.com |
| 5 | 개발 종합반 | 8367f602 | kakaopay | qwerty836@gmail.com |
| 6 | 개발 종합반 | 0b9ca920 | kakaopay | qwerty0b9@naver.com |
| 7 | 개발 종합반 | b2b9db3e | kakaopay | spartab2b@yahoo.com |
| 8 | 개발 종합반 | 2b912659 | kakaopay | abc2b9@daum.net |
| 9 | 개발 종합반 | ef860677 | kakaopay | hellnef8@gmail.com |



Q) 잠깐! 왜 kakaopay가 아니라 "kakaopay"라고 쓰나요?

A) kakaopay를 필드명이나 테이블명이 아닌 문자열로 인식시키려는거예요.

▼ 만약 "kakaopay"가 아니라 kakaopay라고 쓰면?

The screenshot shows a database client interface. The SQL editor contains the query: `select * from orders where payment_method = kakaopay`. Below the editor, an error message is displayed:

SQL Error [1054] [42S22]:
Unknown column 'kakaopay' in
'where clause'

- 당연히 에러가 납니다! 🤔
- **[Unknown column 'kakaopay' in 'where clause']** 에러문구를 천천히 읽어볼까요?

👉 Where 절에 있는 kakaopay라는 컬럼은 없다! 는 뜻이에요. 여기서 컬럼은 필드를 의미해요. 정말, kakaopay라는 글자를 컬럼(필드명)으로 인식해버렸죠?

▼ 여러 조건을 걸어주기 위해서는? 이것도 우선 따라 써 봅시다!

```
select * from orders
where course_title = "앱개발 종합반" and payment_method = "kakaopay";
```

- 마찬가지로 엄청 직관적이죠? and 를 넣어서 원하는 조건을 계속 추가해주면 됩니다!

The screenshot shows a SQL IDE window titled '<sparta> sparta_week1'. The query editor contains the following SQL query:

```
select * from orders
where course_title = "앱개발 종합반"
and payment_method = "kakaopay";
```

Below the query editor, the results of the query are displayed in a table. The table has 5 columns: **course_title**, **user_id**, **payment_method**, and **email**. The results show 11 rows of data, all of which have the course title '앱개발 종합반' and payment method 'kakaopay'.

| | course_title | user_id | payment_method | email |
|----|--------------|----------|----------------|-------------------------|
| 1 | 앱개발 종합반 | 49905ce9 | kakaopay | python499@yahoo.com |
| 2 | 앱개발 종합반 | 856ff83f | kakaopay | hello856@hanmail.com |
| 3 | 앱개발 종합반 | 996b5585 | kakaopay | abc996@gmail.com |
| 4 | 앱개발 종합반 | 8367f602 | kakaopay | qwerty836@gmail.com |
| 5 | 앱개발 종합반 | b2b9db3e | kakaopay | spartab2b@yahoo.com |
| 6 | 앱개발 종합반 | a80594a6 | kakaopay | pythona80@gmail.com |
| 7 | 앱개발 종합반 | 971b0a29 | kakaopay | coolcode971@naver.com |
| 8 | 앱개발 종합반 | 7861bfed | kakaopay | sqlgood786@hanmail.com |
| 9 | 앱개발 종합반 | 8336fc35 | kakaopay | coolcode833@teamsparta. |
| 10 | 앱개발 종합반 | c66c38d3 | kakaopay | mysqlc66@hanmail.com |
| 11 | 앱개발 종합반 | f6f6d06e | kakaopay | hellof6f@naver.com |

06. Where 절 같이 연습해보기

▼ 9) Select 쿼리문에 Where 절 함께 써보기 연습



자, 이제 연습을 해볼까요?

(1) 원하는 테이블과 (2) 조건, 이 두 가지만 기억하면 됩니다!

- SQL을 잘 하기 위해서는, 원하는 정보가 어느 테이블에 담겨있는지 잘 찾는 것이 매우 중요하답니다!

▼ [퀴즈] 포인트가 20000점보다 많은 유저만 뽑아보기!



[힌트!] (1) 원하는 테이블: point_users (2) 조건: point > 20000

| | ABC point_user_id | created_at | updated_at | ABC user_id | 123 point |
|---|--------------------------|-----------------------|-----------------------|-------------|-----------|
| 1 | 5f21264a9441d800091eae34 | 2020-07-29 16:33:31.0 | 2020-08-10 16:08:03.0 | c9ea3998 | 44,113 |
| 2 | 5f222d25bde57f00096c1b4c | 2020-07-30 11:15:01.0 | 2020-08-11 11:40:34.0 | c6752f58 | 23,962 |
| 3 | 5f225cdca4a6870009d8c96e | 2020-07-30 14:38:36.0 | 2020-08-10 20:01:29.0 | fac91699 | 20,330 |
| 4 | 5f23d7eb385fc70009fdff75 | 2020-07-31 17:35:55.0 | 2020-08-06 20:34:40.0 | 7a2c229e | 22,476 |

- 이렇게 출력돼야 해요!

▼ 정답 쿼리 살펴보기!

```
select * from point_users
where point > 20000;
```

▼ [퀴즈] 성이 황씨인 유저만 뽑아보기



이 퀴즈는 힌트 없이 혼자 해보세요!

| | ABC user_id | created_at | updated_at | ABC name | ABC email |
|----|-------------|-----------------------|-----------------------|----------|----------------------|
| 1 | bff6cc85 | 2020-07-15 09:51:48.0 | 2020-07-15 09:51:48.0 | 황** | hellobff@daum.net |
| 2 | 5f76e24a | 2020-07-16 16:24:31.0 | 2020-07-16 16:24:31.0 | 황** | hello5f7@gmail.com |
| 3 | f0fd46ce | 2020-07-17 18:27:44.0 | 2020-07-17 18:27:44.0 | 황** | hellof0f@naver.com |
| 4 | dbbb75bb | 2020-07-17 23:47:31.0 | 2020-07-17 23:47:31.0 | 황** | sqlgooddbb@naver.com |
| 5 | 49889eea | 2020-07-21 14:52:46.0 | 2020-07-21 14:52:46.0 | 황** | sparta498@naver.com |
| 6 | 8d960c10 | 2020-07-21 15:33:23.0 | 2020-07-21 15:33:23.0 | 황** | sparta8d9@gmail.com |
| 7 | 98f2bdbb | 2020-07-21 17:03:52.0 | 2020-08-05 21:31:23.0 | 황** | sqlgood98f@gmail.com |
| 8 | 9d69e140 | 2020-07-24 14:02:46.0 | 2020-08-06 18:48:10.0 | 황** | sqlgood9d6@naver.com |
| 9 | cc3151f4 | 2020-07-25 13:10:44.0 | 2020-07-25 13:10:44.0 | 황** | sqlgoodcc3@yahoo.com |
| 10 | dc65b373 | 2020-07-28 01:10:34.0 | 2020-07-29 19:07:17.0 | 황** | happydc6@gmail.com |
| 11 | b73f2136 | 2020-07-29 18:10:33.0 | 2020-07-29 18:10:33.0 | 황** | happyb73@naver.com |
| 12 | 8dba25ae | 2020-07-30 08:05:59.0 | 2020-07-30 08:05:59.0 | 황** | happy8db@hanmail.com |

- 요렇게 나와야 합니다~~!

▼ 정답 쿼리 살펴보기

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

▼ [퀴즈] 웹개발 종합반이면서 결제수단이 CARD인 주문건만 뽑아보기!



[힌트!] 여러 조건은 and로 함께 넣을 수 있다고 했었죠?

| | asc order_no | cr | update | asc course | asc course_title | asc user_id | asc payment | asc email |
|----|--------------|--------|---------|------------|------------------|-------------|-------------|-------------------------|
| 1 | 3d3b0d3b | -07-13 | 07-13 2 | 5f0ae4087 | 웹개발 종합반 | 3af5bc14 | CARD | sparta3af@hanmail.com |
| 2 | 1e305953 | -07-14 | 07-14 1 | 5f0ae4087 | 웹개발 종합반 | 1f301966 | CARD | python1f3@hanmail.com |
| 3 | a419e8a6 | -07-14 | 07-14 1 | 5f0ae4087 | 웹개발 종합반 | 03667091 | CARD | qwerty036@naver.com |
| 4 | fe87bff0 | -07-14 | 07-14 2 | 5f0ae4087 | 웹개발 종합반 | 14e6200d | CARD | sparta14e@gmail.com |
| 5 | 4476be09 | -07-14 | 07-14 2 | 5f0ae4087 | 웹개발 종합반 | 5aafaccc | CARD | python5aa@naver.com |
| 6 | 38aaee8a | -07-15 | 07-15 0 | 5f0ae4087 | 웹개발 종합반 | d9efc5c8 | CARD | hellod9e@daum.net |
| 7 | 7606af67 | -07-15 | 07-15 0 | 5f0ae4087 | 웹개발 종합반 | 7e9b4584 | CARD | mysql7e9@teamsparta.co |
| 8 | 1cddcf6f | -07-16 | 07-16 0 | 5f0ae4087 | 웹개발 종합반 | ed12149c | CARD | spartaed1@gmail.com |
| 9 | 088e5dd5 | -07-16 | 07-16 1 | 5f0ae4087 | 웹개발 종합반 | 38f1dad8 | CARD | sqlgood38f@naver.com |
| 10 | 956501d4 | -07-16 | 07-16 2 | 5f0ae4087 | 웹개발 종합반 | bd424673 | CARD | coolcodebd4@hanmail.com |
| 11 | 1129adc5 | -07-16 | 07-16 2 | 5f0ae4087 | 웹개발 종합반 | f99d9f84 | CARD | coolcodef99@naver.com |
| 12 | 27996e22 | -07-16 | 07-16 2 | 5f0ae4087 | 웹개발 종합반 | 14e53de1 | CARD | sparta14e@naver.com |
| 13 | 92a79117 | -07-17 | 07-17 0 | 5f0ae4087 | 웹개발 종합반 | e40d555f | CARD | abce40@naver.com |
| 14 | 139a1a7b | -07-17 | 07-17 1 | 5f0ae4087 | 웹개발 종합반 | ebef6f9b | CARD | sqlgoodebe@hanmail.com |
| 15 | f37ecd37 | -07-17 | 07-17 1 | 5f0ae4087 | 웹개발 종합반 | f0fd46ce | CARD | hellof0f@naver.com |
| 16 | a782c8d2 | -07-17 | 07-17 2 | 5f0ae4087 | 웹개발 종합반 | 8882e93c | CARD | mvsnl888@naver.com |

- 요렇게 나와야 합니다~~!

▼ 정답 쿼리 살펴보기

```
select * from orders
where course_title = "웹개발 종합반" and payment_method = "CARD";
```

▼ 10) [꿀팁 🍯] 이렇게 쿼리를 작성하면 편해요!



- 1) show tables로 어떤 테이블이 있는지 살펴보기
- 2) 제일 원하는 정보가 있을 것 같은 테이블에 select * from 테이블명 쿼리 날려보기
- 3) 원하는 정보가 없으면 다른 테이블에도 2)를 해보기
- 4) 테이블을 찾았다! 조건을 걸 필드를 찾기
- 5) select * from 테이블명 where 조건 이렇게 쿼리 완성!



참 쉽죠? 여러분은 이제 조건에 맞는 데이터를 가져올 수 있는 사람이 되었답니다! 🙌

07. Where 절과 자주 같이 쓰는 문법 써보기

▼ 11) Where 절과 자주 같이 쓰는 문법 배우기

▼ '같지 않음' 조건 걸어보기



자, 쉬운 것부터 시작해 볼까요?

- '같지 않음' 조건은 != 로 걸 수 있습니다.



'웹개발 종합반'을 제외하고 주문데이터를 보고 싶어졌어요. 어떻게 하면 좋을까요?

▼ [코드스니펫] '같지 않음' 조건 걸어보기

```
select * from orders
where course_title != "웹개발 종합반";
```

- "웹개발 종합반" 데이터 없이 잘 나오죠?



잠깐 상식!

'!=' 에서 ! (느낌표)는 부정 (not)을 의미합니다. '='는 같음을 의미하니, '!='는 같지 않음이겠죠!

▼ '범위' 조건 걸어보기



7월 13일, 7월 14일의 주문데이터만 보고 싶어졌어요. 어떻게 해야 할까요?


- '범위' 조건은 between 으로 걸 수 있어요.

▼ [코드스니펫] '범위' 조건 걸어보기

```
select * from orders
where created_at between "2020-07-13" and "2020-07-15";
```

- 7월 13일, 7월 14일 주문데이터만 잘 나오죠?

▼ '포함' 조건 걸어보기

 1, 3주차 사람들의 '오늘의 다짐' 데이터만 보고 싶어졌어요.

- '포함' 조건은 in 으로 걸 수 있어요.

▼ [코드스니펫] '포함' 조건 걸어보기

```
select * from checkins
where week in (1, 3);
```

- 1, 3 주차 사람들의 '오늘의 다짐'만 잘 나오죠?

▼ '패턴' (문자열 규칙) 조건 걸어보기

 다음 (daum) 이메일을 사용하는 유저만 보고 싶어졌어요. 어떻게 하죠?

- '패턴' 조건은 like 으로 걸 수 있어요.

▼ [코드스니펫] '패턴' (문자열 규칙) 조건 걸어보기

```
select * from users
where email like '%daum.net';
```

- 오! 엄청 신기하죠? 많이 사용되는 유용한 기능입니다!

▼ [꿀팁 🍯] Like의 다양한 사용법



Like는 패턴으로 조건을 거는 문법으로, 사용법이 아주 다양합니다!

- where email like 'a%': email 필드값이 a로 시작하는 모든 데이터
- where email like '%a' email 필드값이 a로 끝나는 모든 데이터
- where email like '%co%' email 필드값에 co를 포함하는 모든 데이터
- where email like 'a%o' email 필드값이 a로 시작하고 o로 끝나는 모든 데이터



이외에도 여러 문법이 있는데, 그때그때 필요한 것을 찾아서 쓰면 됩니다!

같이 한번 찾아볼까요? ('how to use like in sql' 구글링!)

▼ 12) [퀴즈] 자, 이제 직접 쿼리를 짜볼까요?



SQL 잘하는 가장 쉬운 방법: 직접 SQL 쿼리를 많이 짜보기!

▼ [퀴즈] 결제수단이 CARD가 아닌 주문데이터만 추출해보기



[힌트!] '같지 않음' 조건은 != 를 사용한다고 했었죠?

| | abc_order_no | cr | upda | abc_course | abc_course_title | abc_user_id | abc_paymen | abc_email |
|----|--------------|--------|---------|------------|------------------|-------------|------------|-----------------------|
| 1 | f4884401 | -07-13 | 07-13 2 | 5f0ae4087 | 앱개발 종합반 | 49905ce9 | kakaopay | python499@yahoo.com |
| 2 | af831367 | -07-13 | 07-13 2 | 5f0ae4087 | 앱개발 종합반 | 856ff83f | kakaopay | hello856@hanmail.com |
| 3 | abb920a1 | -07-14 | 07-14 1 | 5f0ae4087 | 앱개발 종합반 | 996b5585 | kakaopay | abc996@gmail.com |
| 4 | 2920906e | -07-14 | 07-14 1 | 5f0ae4087 | 웹개발 종합반 | 65b2fd34 | kakaopay | abc65b@gmail.com |
| 5 | 47823d05 | -07-15 | 07-15 1 | 5f0ae4087 | 앱개발 종합반 | 8367f602 | kakaopay | qwerty836@gmail.com |
| 6 | 7fe2b6e3 | -07-16 | 07-16 2 | 5f0ae4087 | 웹개발 종합반 | 0b9ca920 | kakaopay | qwerty0b9@naver.com |
| 7 | 065585c7 | -07-17 | 07-17 1 | 5f0ae4087 | 웹개발 종합반 | b4bc2fc8 | TAXBILL | spartab4b@naver.com |
| 8 | 459505cb | -07-18 | 07-18 0 | 5f0ae4087 | 앱개발 종합반 | b2b9db3e | kakaopay | spartab2b@yahoo.com |
| 9 | 5b05be16 | -07-18 | 07-18 1 | 5f0ae4087 | 웹개발 종합반 | 2b912659 | kakaopay | abc2b9@daum.net |
| 10 | 8fb38fd3 | -07-19 | 07-19 2 | 5f0ae4087 | 웹개발 종합반 | ef860677 | kakaopay | helloef8@gmail.com |
| 11 | 9f3f0e90 | -07-20 | 07-20 1 | 5f0ae4087 | 웹개발 종합반 | 2dfbb19f | MONEY | hello2df@gmail.com |
| 12 | a2cbe0a8 | -07-20 | 07-20 1 | 5f0ae4087 | 앱개발 종합반 | 0b47bcb | MONEY | hello0b4@yahoo.com |
| 13 | 6cc6ea9c | -07-20 | 07-20 1 | 5f0ae4087 | 앱개발 종합반 | a80594a6 | kakaopay | pythona80@gmail.com |
| 14 | b2d7f72c | -07-20 | 07-20 2 | 5f0ae4087 | 앱개발 종합반 | 971b0a29 | kakaopay | coolcode971@naver.com |
| 15 | b5a3174c | -07-21 | 07-21 1 | 5f0ae4087 | 앱개발 종합반 | c838274b | TAXBILL | spartac83@naver.com |

- 요런 식으로 나와야 합니다! (일부 데이터만 나온 이미지)

▼ 정답 쿼리 살펴보기

```
select * from orders
where payment_method != 'CARD';
```

▼ [퀴즈] 20000~30000 포인트 보유하고 있는 유저만 추출해보기



[힌트!] '범위' 조건은 between을 사용한다고 했었죠?

| | ABC point_user_id | created_at | updated_at | ABC user_id | 123 point |
|---|--------------------------|-----------------------|-----------------------|-------------|-----------|
| 1 | 5f222d25bde57f00096c1b4c | 2020-07-30 11:15:01.0 | 2020-08-11 11:40:34.0 | c6752f58 | 23,962 |
| 2 | 5f225cdca4a6870009d8c96e | 2020-07-30 14:38:36.0 | 2020-08-10 20:01:29.0 | fac91699 | 20,330 |
| 3 | 5f23d7eb385fc70009fdff75 | 2020-07-31 17:35:55.0 | 2020-08-06 20:34:40.0 | 7a2c229e | 22,476 |

- 요런 식으로 나와야 합니다!

▼ 정답 쿼리 살펴보기

```
select * from point_users
where point between 20000 and 30000
```

▼ [퀴즈] 이메일이 s로 시작하고 com로 끝나는 유저만 추출해보기



[힌트!] Like에서 'a%o'를 사용하면 a로 시작하고 o로 끝나는 데이터를 얻을 수 있어요!

| | ABC user_id | created_at | updated_at | ABC name | ABC email |
|----|-------------|-----------------|-----------------|----------|-----------------------|
| 1 | 3af5bc14 | 7-13 22:30:00.0 | 7-13 22:30:00.0 | 고** | sparta3af@hanmail.com |
| 2 | efaf5439 | 7-14 12:43:08.0 | 7-14 12:43:08.0 | 임** | sqlgoodefa@naver.com |
| 3 | deb54ce4 | 7-14 12:46:42.0 | 7-14 12:46:42.0 | 임** | sqlgooddeb@naver.com |
| 4 | 24b83a8e | 7-14 12:49:18.0 | 7-14 12:49:18.0 | 김** | sparta24b@yahoo.com |
| 5 | 7026cc54 | 7-14 12:56:51.0 | 7-14 12:56:51.0 | 남** | sparta702@gmail.com |
| 6 | 38fa9ab8 | 7-14 13:13:43.0 | 7-14 13:13:43.0 | 손** | sparta38f@gmail.com |
| 7 | 3c6033d6 | 7-14 15:21:10.0 | 7-14 15:21:10.0 | 김** | sparta3c6@yahoo.com |
| 8 | 29b9b925 | 7-14 15:22:23.0 | 7-14 15:22:23.0 | 이** | sparta29b@gmail.com |
| 9 | 14e6200d | 7-14 16:34:10.0 | 7-14 16:34:10.0 | 윤** | sparta14e@gmail.com |
| 10 | a96f75d7 | 7-14 16:54:44.0 | 7-21 17:42:19.0 | 안** | sqlgooda96@gmail.com |
| 11 | 21c6ed67 | 7-14 17:34:53.0 | 7-14 17:34:53.0 | 이** | sqlgood21c@gmail.com |
| 12 | 665da369 | 7-15 08:18:52.0 | 7-15 08:18:52.0 | 김** | sqlgood665@gmail.com |
| 13 | 4bde25be | 7-15 21:35:12.0 | 7-15 21:35:12.0 | 김** | sqlgood4bd@gmail.com |
| 14 | ed12149c | 7-16 00:48:07.0 | 7-16 00:48:07.0 | 시** | spartaed1@gmail.com |

- 요런 식으로 나와야 합니다!

▼ 정답 쿼리 살펴보기

```
select * from users
where email like 's%com';
```

▼ [퀴즈] 이메일이 s로 시작하고 com로 끝나면서 성이 이씨인 유저만 추출해보기



[힌트!] Like에서 'a%o'를 사용하면 a로 시작하고 o로 끝나는 데이터를 얻을 수 있어요!

| | ABC user_id | created_at | updated_at | ABC name | ABC email |
|----|-------------|-----------------|-----------------|----------|------------------------|
| 1 | 29b9b925 | 7-14 15:22:23.0 | 7-14 15:22:23.0 | 이** | sparta29b@gmail.com |
| 2 | 21c6ed67 | 7-14 17:34:53.0 | 7-14 17:34:53.0 | 이** | sqlgood21c@gmail.com |
| 3 | 17638b60 | 7-16 19:24:24.0 | 7-16 19:24:24.0 | 이** | sqlgood176@gmail.com |
| 4 | 35980035 | 7-17 13:33:41.0 | 7-17 13:33:41.0 | 이** | sqlgood359@yahoo.com |
| 5 | 762fb992 | 7-20 13:37:33.0 | 7-20 13:39:27.0 | 이** | sparta762@naver.com |
| 6 | c838274b | 7-21 14:10:31.0 | 8-06 20:38:25.0 | 이** | spartac83@naver.com |
| 7 | bcd441f6 | 7-21 15:32:52.0 | 7-21 15:55:26.0 | 이** | sqlgoodbcd@hanmail.com |
| 8 | 081e2577 | 7-21 16:03:18.0 | 8-03 12:18:42.0 | 이** | sparta081@gmail.com |
| 9 | f4c949c8 | 7-27 17:48:57.0 | 7-27 17:48:57.0 | 이** | spartaf4c@naver.com |
| 10 | 5e65575d | 7-29 09:43:13.0 | 7-29 09:43:13.0 | 이** | sparta5e6@naver.com |
| 11 | d848f82d | 8-01 16:28:27.0 | 8-01 16:41:14.0 | 이** | sqlgoodd84@gmail.com |

- 요런 식으로 나와야 합니다!

▼ 정답 쿼리 살펴보기

```
select * from users
where email like 's%com' and name = "이**";
```

08. 이외 유용한 문법 배워보기

▼ 13) 일부 데이터만 가져오기: Limit



테이블에 어떤 데이터가 들어있나 잠깐 보려고 들어왔는데, 데이터를 다 불러오느라 시간이 오래 걸리면 힘들겠죠?

- 그런 경우를 대비해, 일부 데이터만 가져오는 Limit이라는 기능이 있습니다.

▼ 백문이 불여일견! 직접 SQL 쿼리를 볼까요?

```
select * from orders
where payment_method = "kakaopay"
limit 5;
```

- 요렇게, 맨 뒤에 limit을 써주고 몇 개 출력할지 숫자를 적어주면 끝!

| | | ABC course_title | ABC user_id | ABC payment_method | ABC email |
|---|---|------------------|-------------|--------------------|----------------------|
| 1 | 7 | 앱개발 종합반 | 49905ce9 | kakaopay | python499@yahoo.com |
| 2 | 7 | 앱개발 종합반 | 856ff83f | kakaopay | hello856@hanmail.com |
| 3 | 7 | 앱개발 종합반 | 996b5585 | kakaopay | abc996@gmail.com |
| 4 | 6 | 웹개발 종합반 | 65b2fd34 | kakaopay | abc65b@gmail.com |
| 5 | 7 | 앱개발 종합반 | 8367f602 | kakaopay | qwerty836@gmail.com |

- 딱 5개만 출력된 것을 알 수 있죠?

▼ 14) 중복 데이터는 제외하고 가져오기: Distinct

- 정답은 2번! 너무 쉽죠?

▼ 마찬가지로, 직접 바로 SQL 쿼리로 가시죠!

```
select distinct(payment_method) from orders;
```

| | ABC payment_method |
|---|--------------------|
| 1 | kakaopay |
| 2 | CARD |
| 3 | TAXBILL |
| 4 | MONEY |

- 요렇게, 네 개의 결제수단이 있다는 사실을 한 줄로 알아낼 수 있습니다!

▼ 15) 몇 개인지 숫자 세보기: Count

☞ 이번에는 orders 테이블에 데이터가 몇 개 들어있는지 궁금해요!
(결제가 몇 건이나 들어왔죠? 두근두근)

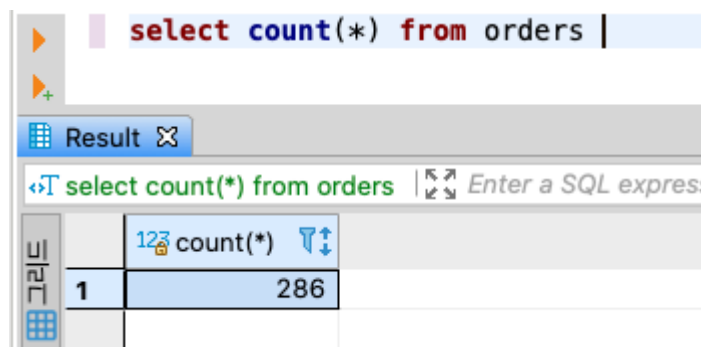
- 1) orders 테이블의 데이터를 모두 불러와서, 한줄한줄씩 센다
- 2) count 기능을 사용해서 한방에 본다

- 당연히 2번이죠?

▼ 바로 SQL 쿼리부터 보시죠!

```
select count(*) from orders
```

- 한 줄로 끝! select * from orders에서 *을, count로 감싸주면 됩니다.



The screenshot shows a SQL query editor with the query `select count(*) from orders`. Below the editor, a 'Result' window displays the output. The result is a single row with the value 286, representing the count of rows in the orders table.

| | count(*) |
|---|----------|
| 1 | 286 |

- 이렇게 한 줄로 끝!

▼ 16) [응용] Distinct와 Count를 같이 써보기

☞ 스파르타 회원 분들의 성(family name)씨가 몇개인지 궁금하다면?
이번에도 마찬가지로 쉬운 방법이 있습니다!

▼ SQL 쿼리 보러가기

```
select distinct(name) from users;
```

| users | |
|----------------------------|-----|
| SELECT distinct(name) from | |
| ABC name | |
| 1 | 이** |
| 2 | 신** |
| 3 | 강** |
| 4 | 정** |
| 5 | 남** |
| 6 | 고** |
| 7 | 문** |
| 8 | 방** |
| 9 | 노** |
| 10 | 김** |
| 11 | 임** |
| 12 | 심** |
| 13 | 최** |
| 14 | 손** |
| 15 | 배** |

- 이렇게 하면 성씨가 짝 나오죠?

```
SELECT count(distinct(name)) from users;
```

| | |
|---|----|
| SELECT count(distinct(name)) from users | |
| Result | |
| SELECT count(distinct(name)) | |
| 123 count(distinct(name)) | |
| 1 | 54 |

- 그리고, 여기에 count를 입혀주면 몇 개의 성씨가 있는지 알 수 있습니다. (참 쉽죠?)

09. 같이 삼질해보기

- ▼ 17) 혼자서도 문제를 해결하려면



SQL을 사용하다보면 예상하지 못했던 에러를 자주 마주하게 됩니다.
모든 에러에 대한 해결책을 외워서 해결하는 건 비효율적이겠죠?

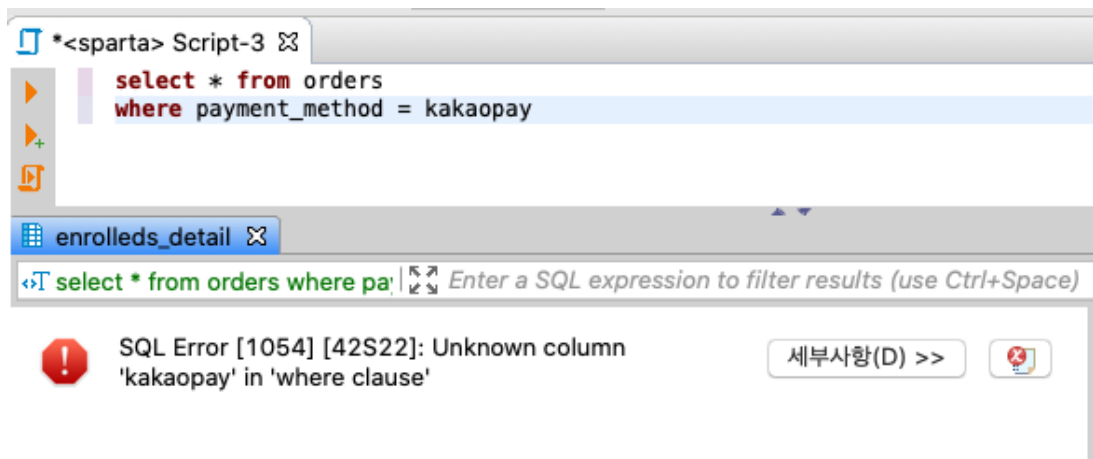
- 그럼 어떻게 하는게 좋을까요?



기초적인 SQL의 경우, 에러 메시지를 보고 원인을 파악하여 문제를 해결해 나가는 것!

아직은 감이 안 잡힌다고요? 괜찮아요! 저랑 같이 문제를 해결해봐요.

▼ 에러 메시지 해석해보기 (1)



- 자, 뭔가 에러가 났습니다! 어떤 에러메시지일까요?
- Unknown column 'kakaopay' in 'where clause'라고 나와있죠?



Where 절에 있는 kakaopay라는 컬럼(필드)가 알려지지 않았다 (컴퓨터는 알지 못한다) 라는 것입니다.

즉, kakaopay라는 글자가 '컬럼(필드)' 이름으로 인식되어서 발생한 에러예요.

- 그러면 어떻게 하면 좋을까요?



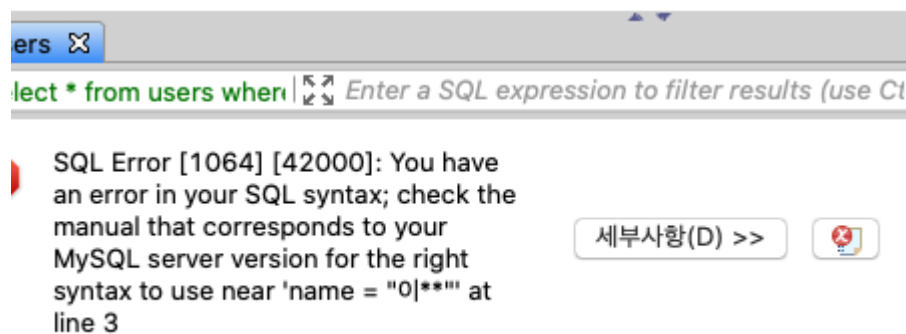
kakao-pay를 'kakaopay'로 바꿔주면 되겠죠?

글자를 "로 감싸면 변수가 아닌, 문자열(값)로 인식하겠다는 의미입니다.

결제수단이 kakao-pay라는 값을 가진 데이터만 불러오기 위해서는, kakao-pay를 문자열(값)으로 지정해 줘야겠죠!

▼ 에러 메시지 해석해보기 (2)

```
select * from users
where email like 's%net'
name = "0|**"|
```



에러가 생겼을 땐, 당황하지 않고 에러 메시지를 차근차근 읽어보세요!
지금쯤은 어려울 수 있어요, 저랑 4주동안 연습하면 어느샌가 익숙해질거예요.

10. 퀴즈 풀어보기

▼ 18) [퀴즈] Select 쿼리문, Where 절 연습하기



성이 남씨인 유저의 이메일만 추출하기

▼ 정답 쿼리 살펴보기!

```
select email from users
where name = "남**";
```



이해가 안 되신다고요? 저랑 같이 해봐요!

▼ 19) [퀴즈] Where 절과 자주 같이쓰는 문법 연습하기



Gmail을 사용하는 2020/07/12~13에 가입한 유저를 추출하기

▼ 정답 쿼리 살펴보기!

```
select * from users
where created_at between "2020-07-12" and "2020-07-14"
and email like "%gmail.com";
```



이해가 안 되신다고요? 저랑 같이 해봐요!

▼ 20) [퀴즈] 이외 유용한 문법 연습하기



Gmail을 사용하는 2020/07/12~13에 가입한 유저의 수를 세기

▼ 정답 쿼리 살펴보기!

```
select count(*) from users
where created_at between "2020-07-12" and "2020-07-14"
and email like "%gmail.com";
```



이해가 안 되신다고요? 저랑 같이 해봐요!

10. 끝 & 숙제 설명



자, 1주차 고생 많았습니다! 고생의 끝은 숙제죠?

숙제: naver 이메일을 사용하면서, 웹개발 종합반을 신청했고 결제는 kakaopay로 이뤄진 주문데이터 추출하기

▼ 결과

| | asc order_no | created_at | updated_at | asc course_id | asc course_title | asc user_id | asc payment_method | asc email |
|---|--------------|-----------------------|-----------------------|--------------------------|------------------|-------------|--------------------|---------------------|
| 1 | 7fe2b6e3 | 2020-07-16 23:15:03.0 | 2020-07-16 23:17:04.0 | 5f0ae408765dae0006002816 | 웹개발 종합반 | 0b9ca920 | kakaopay | qwerty0b9@naver.com |
| 2 | 4d20a6a5 | 2020-07-23 12:21:46.0 | 2020-07-23 12:22:31.0 | 5f0ae408765dae0006002816 | 웹개발 종합반 | 20ef8ca2 | kakaopay | happy20e@naver.com |
| 3 | 74680cca | 2020-07-28 00:06:48.0 | 2020-07-28 00:14:44.0 | 5f0ae408765dae0006002816 | 웹개발 종합반 | c5db1598 | kakaopay | qwertyc5d@naver.com |
| 4 | 5b5fd32c | 2020-07-28 00:57:43.0 | 2020-07-28 01:02:29.0 | 5f0ae408765dae0006002816 | 웹개발 종합반 | 6e95e7c2 | kakaopay | hello6e9@naver.com |
| 5 | cccafcfb | 2020-08-04 16:08:25.0 | 2020-08-04 16:11:57.0 | 5f0ae408765dae0006002816 | 웹개발 종합반 | b42aeb18 | kakaopay | qwertyb42@naver.com |
| 6 | 0e1ec034 | 2020-08-04 22:15:14.0 | 2020-08-04 22:17:13.0 | 5f0ae408765dae0006002816 | 웹개발 종합반 | 12316c05 | kakaopay | qwerty123@naver.com |
| 7 | f9710570 | 2020-08-05 19:35:56.0 | 2020-08-05 19:44:28.0 | 5f0ae408765dae0006002816 | 웹개발 종합반 | e3b2fcde | kakaopay | happye3b@naver.com |
| 8 | 5a816460 | 2020-08-11 17:50:54.0 | 2020-08-11 17:51:51.0 | 5f0ae408765dae0006002816 | 웹개발 종합반 | 301c8b25 | kakaopay | python301@naver.com |



이런 결과가 나오면 정답! 🙌

11. 1주차 숙제 답안 코드

▼ [코드스니펫] - 1주차 숙제 답안 코드

```
select * from orders
where email like '%naver.com'
and course_title = '웹개발 종합반'
and payment_method = 'kakaopay'
```

Copyright © TeamSparta All rights reserved.