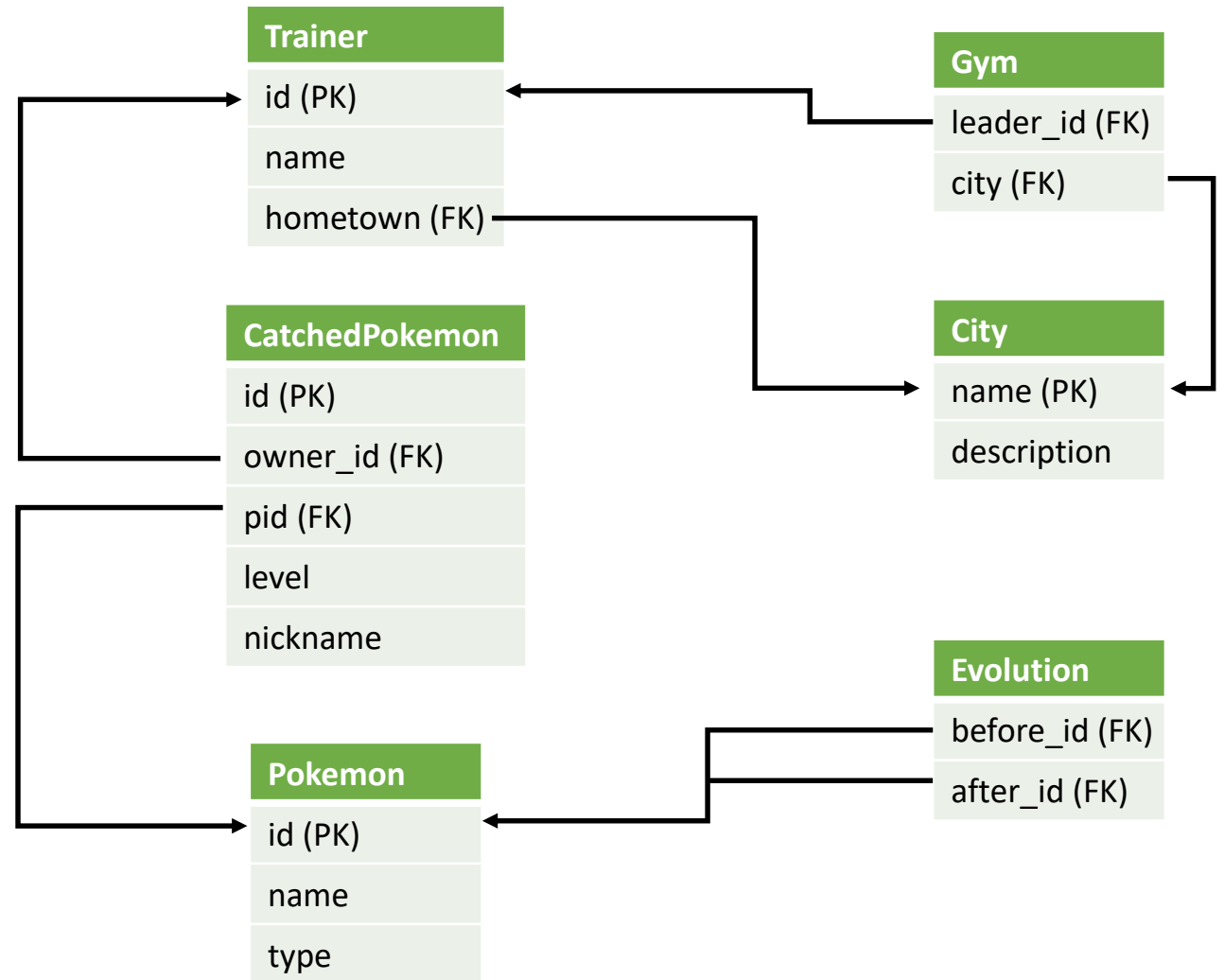


Project1

SQL Practice

Pokemon Database Schema

- Pokemon Database 를 주어진 Schema 를 이용해서 Table 을 만들어보고, insert.sql 파일을 이용해서 데이터를 추가하세요. (insert.sql 파일은 piazza 에서 다운받을 수 있음)



Example

Trainer

id	name	hometown
1	Red	Sangnok City
2	Green	Sangnok City
3	Blue	Sangnok City

City

name	description
Blue City	Waterpark
Brown City	Famous Harbor
Sangnok City	Amazon

CatchedPokemon

id	pid	owner_id	level	nickname
1	6	2	70	After eating hot peppers
2	25	1	10	AAA
3	25	4	50	Mouse

Gym

leader_id	city
2	Sangnok City
7	Brown City
9	Blue City

Pokemon

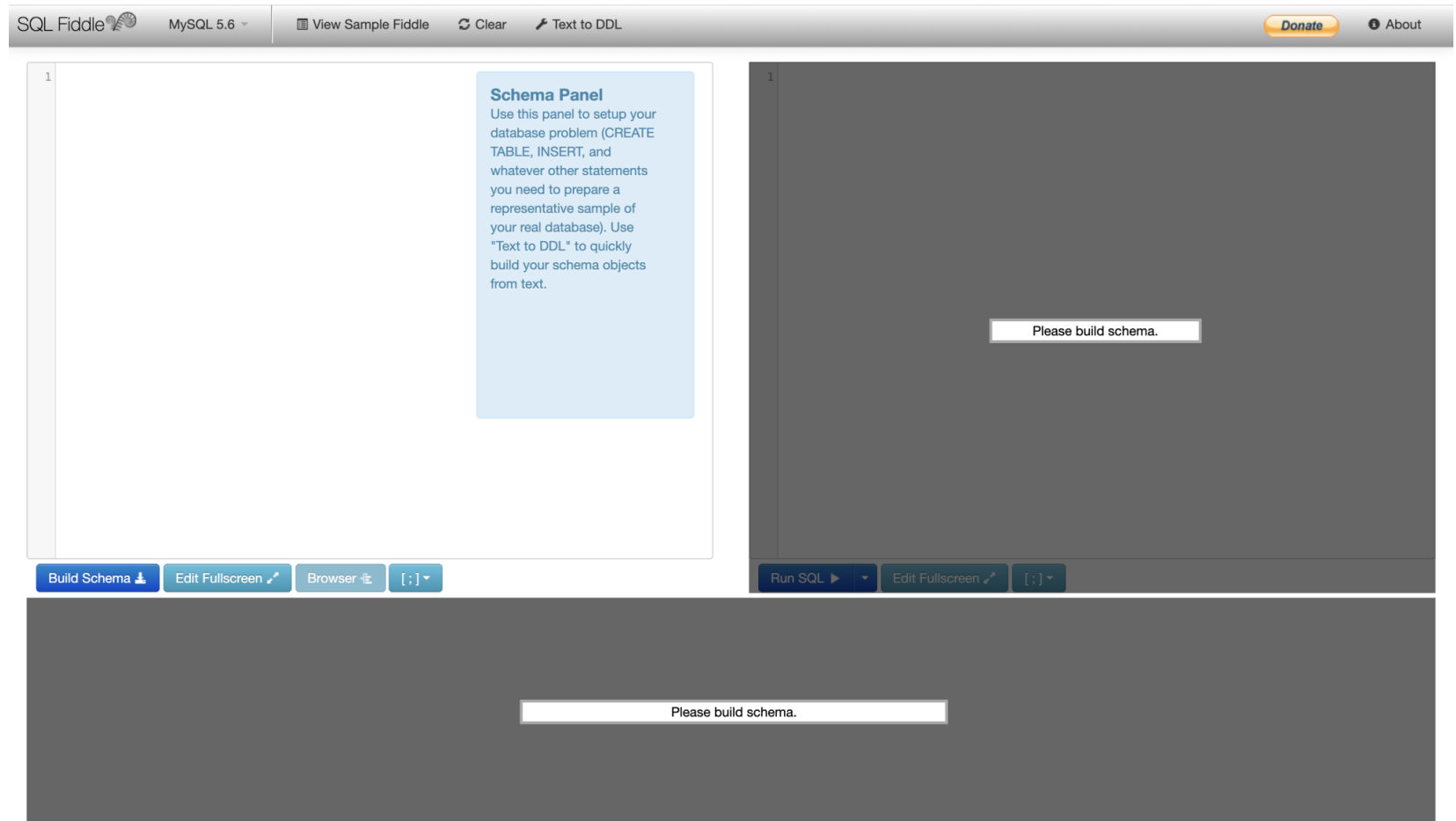
id	name	type
6	Charizard	Fire
25	Pikachu	Electric
26	Raichu	Electric

Evolution

before_id	after_id
1	2
2	3
25	26

How to use sqlfiddle

- <http://sqlfiddle.com/>



How to use sqlfiddle

The screenshot displays the SQL Fiddle web application interface. At the top, the header includes the 'SQL Fiddle' logo, a dropdown menu currently set to 'MySQL 5.6' (highlighted with a red box), and navigation links for 'View Sample Fiddle', 'Clear', and 'Text to DDL'. On the right side of the header are 'Donate' and 'About' buttons.

The main workspace is divided into two panels. The left panel, titled 'Schema Panel' with the instruction 'Use this panel to setup your schema', contains a text area with the number '1' and a 'Build Schema' button. An orange callout box with the text 'Select MySQL' points to the 'MySQL 5.6' dropdown menu. The right panel is a large dark gray area with a message box that says 'Please build schema.' and a 'Run SQL' button.

At the bottom of the interface, there is a dark gray bar containing a message box that says 'Please build schema.' and a 'Run SQL' button.

How to use sqlfiddle

The screenshot shows the SQL Fiddle web application interface. The top bar includes the "SQL Fiddle" logo, a "MySQL 5.6" dropdown, and links for "View Sample Fiddle", "Clear", and "Text to DDL". On the right, there are "Donate" and "About" buttons. The main area consists of two text editors. The left editor contains a SQL statement to create a table named "Students":

```
1 CREATE TABLE Students (  
2   sid INT UNSIGNED NOT NULL AUTO_INCREMENT,  
3   name CHAR(10) NOT NULL DEFAULT '',  
4   age INT UNSIGNED,  
5   gpa DOUBLE NOT NULL DEFAULT '0.0',  
6   PRIMARY KEY (sid)  
7 );  
8
```

The right editor is empty. Below the editors, there are two rows of buttons. The left row includes a "Build Schema" button (highlighted with a red box), an "Edit Fullscreen" button, a "Browser" button, and a syntax highlighting dropdown menu. The right row includes a "Run SQL" button, an "Edit Fullscreen" button, and a syntax highlighting dropdown menu. A yellow callout box with the text "Create table" points to the "Build Schema" button. At the bottom, a green status bar displays "Schema Ready".

How to use sqlfiddle

SQL Fiddle MySQL 5.6 View Sample Fiddle Clear Text to DDL Donate About

```
1 CREATE TABLE Students (  
2   sid INT UNSIGNED NOT NULL AUTO_INCREMENT,  
3   name CHAR(10) NOT NULL DEFAULT '',  
4   age INT UNSIGNED,  
5   gpa DOUBLE NOT NULL DEFAULT '0.0',  
6   PRIMARY KEY (sid)  
7 );  
8  
9 INSERT INTO Students  
10 VALUES (100, 'Fred', 23, 3.5);  
11
```

Insert tuples

Build Schema Edit Fullscreen Browser [;]

Run SQL Edit Fullscreen [;]

Query Panel
Use this panel to try to solve the problem with other SQL statements (SELECTs, etc...). Results will be displayed below. Share your queries by copying and pasting the URL that is generated after each run.

Schema Ready

How to use sqlfiddle

SQL Fiddle MySQL 5.6 View Sample Fiddle Clear Text to DDL Donate About

```
1 CREATE TABLE Students (  
2   sid INT UNSIGNED NOT NULL AUTO_INCREMENT,  
3   name CHAR(10) NOT NULL DEFAULT '',  
4   age INT UNSIGNED,  
5   gpa DOUBLE NOT NULL DEFAULT '0.0',  
6   PRIMARY KEY (sid)  
7 );  
8  
9 INSERT INTO Students  
10 VALUES (100, 'Fred', 23, 3.5);  
11
```

```
1 SELECT * FROM Students;
```

Run SQL query

Build Schema Edit Fullscreen Browser [;]

Run SQL Edit Fullscreen [;]

sid	name	age	gpa
100	Fred	23	3.5

✓ Record Count: 1; Execution Time: 5ms + View Execution Plan link

Did this query solve the problem? If so, consider donating \$5 to help make sure SQL Fiddle will be here next time you need help with a database problem. Thanks!


Project 1

The screenshot shows the SQL Fiddle MySQL 5.6 interface. The left pane contains the following SQL code:

```
1 # create table
2 CREATE TABLE City (
3   name varchar(32),
4   description varchar(32),
5   primary key (name)
6 );
7
8 CREATE TABLE Trainer (
9   id INT(11) unsigned AUTO_INCREMENT,
10  name varchar(32),
11  hometown varchar(32),
```

The 'Build Schema' button is highlighted with a red box. A yellow callout bubble points to the right pane with the text 'Copy & Paste insert.sql'. The right pane is currently empty. At the bottom, a green status bar indicates '✓ Schema Ready'.


Project 1


SQL Fiddle  MySQL 5.6


```
1 # crea
2 CREATE
3   name
4   desc
5   prim
6 );
7
8 CREATE TABLE Trainer (
9   id INT(11) unsigned AUTO_INCREMENT,
10  name varchar(32),
11  hometown varchar(32),
```


Solve SQL Practice Here!

1

Build Schema 

Edit Fullscreen 

Browser  [;] ▾

Run SQL ▶ ▾ Edit Fullscreen  [;] ▾

✓ Schema Ready

SQL Practice

- 아래 내용을 출력하는 SQL 쿼리를 작성해 보세요.

01. 잡은 포켓몬이 3마리 이상인 트레이너의 이름을 잡은 포켓몬의 수가 많은 순서대로 출력하세요.
02. 전체 포켓몬들의 타입 중 가장 많은 2개의 타입 중 하나를 갖고 있는 포켓몬들의 이름을 사전 순으로 출력하세요. (전체 포켓몬 타입의 개수가 중복이 되어서, 2번째로 가장 많은 타입들까지 중복된 타입이 생긴다면, 전체 타입을 출력)
03. 상록시티 출신 트레이너들이 가지고 있는 Electric 타입의 포켓몬 레벨의 평균을 출력하세요.
04. 잡은 포켓몬 중 레벨이 50 이상인 포켓몬의 닉네임을 사전순으로 출력하세요.
05. 체육관 관장을 맡고 있지 않은 트레이너의 이름을 사전 순으로 출력하세요.
06. 블루시티 출신의 트레이너의 이름을 사전순으로 출력하세요.
07. 트레이너의 이름을 트레이너의 고향의 사전 순으로 출력하세요.
08. Red가 잡은 포켓몬의 평균 레벨을 출력하세요.
09. 체육관 관장 이름과, 관장이 가진 포켓몬 레벨의 평균을 관장 이름의 사전순으로 출력하세요.
10. 잡은 포켓몬 중 레벨이 50 이상 이고 owner_id가 6 이상인 포켓몬의 닉네임을 사전순으로 출력하세요.
11. 포켓몬 도감에 있는 모든 포켓몬의 ID와 이름을 ID의 오름차순으로 정렬해 출력하세요.
12. 잡힌 포켓몬 중 레벨이 30 이상인 포켓몬의 이름과 타입을 포켓몬 이름의 사전 순으로 중복 없이 출력하세요.

SQL Practice

- 아래 내용을 출력하는 SQL 쿼리를 작성해 보세요.

13. 상록시티 출신 트레이너가 가진 포켓몬들의 이름과 포켓몬 ID를 포켓몬 ID의 오름차순으로 정렬해 출력하세요.
14. Grass 타입 포켓몬 중 진화가능한 포켓몬의 이름을 출력하세요.
15. 포켓몬을 가장 많이 잡은 트레이너의 ID와 그 트레이너가 잡은 포켓몬의 수를 트레이너 ID의 오름차순으로 출력하세요.
16. 포켓몬 도감에 있는 포켓몬 중 Water 타입, Electric 타입, Psychic 타입 포켓몬의 총합을 출력하세요.
17. 상록시티 출신 트레이너들이 가지고 있는 포켓몬 종류의 개수를 출력하세요.
18. 각 체육관의 관장에게 잡힌 전체 포켓몬의 평균 레벨을 출력하세요.
19. 상록시티 체육관 관장이 가지고 있는 포켓몬 타입의 개수를 출력하세요.
20. 상록시티 출신 트레이너의 이름과 각 트레이너가 잡은 포켓몬 수를 잡은 포켓몬 수에 대해 오름차순 순으로 출력하세요.
21. 체육관 관장들의 이름과 각 관장이 잡은 포켓몬의 수를 관장 이름의 사전순으로 출력하세요.
22. 포켓몬의 타입과 해당 타입을 가지는 포켓몬의 수가 몇 마리인지를 출력하세요. 수에 대해 오름차순으로 출력하세요. (같은 수를 가진 경우 타입명의 사전순으로 정렬)
23. 잡힌 포켓몬 중 레벨이 10이하인 포켓몬을 잡은 트레이너의 이름을 중복없이 사전순으로 출력하세요.

SQL Practice

- 아래 내용을 출력하는 SQL 쿼리를 작성해 보세요.

24. 각 시티의 이름과 해당 시티 출신 트레이너들이 잡은 포켓몬들의 평균 레벨을 평균 레벨의 오름차순으로 정렬해서 출력하세요.
25. 상록시티 출신 트레이너와 브라운 시티 출신 트레이너가 공통으로 잡은 포켓몬의 이름을 사전순으로 출력하세요. (중복은 제거할 것)
26. 잡힌 포켓몬 중 닉네임에 공백이 들어가는 포켓몬의 이름을 사전 역순으로 출력하세요.
27. 포켓몬을 4마리 이상 잡은 트레이너의 이름과 해당 트레이너가 잡은 포켓몬 중 가장 레벨이 높은 포켓몬의 레벨을 트레이너 이름의 사전순으로 정렬해서 출력하세요.
28. Normal 타입 혹은 Electric 타입의 포켓몬을 잡은 트레이너의 이름과 해당 포켓몬의 평균 레벨을 구한 평균 레벨의 오름차순으로 정렬해서 출력하세요.
29. 각 타입별로 잡힌 포켓몬의 수를 타입 이름의 사전순으로 출력하세요.
30. 포켓몬 중, 3단 진화가 가능한 포켓몬의 ID와 1단 진화 형태 포켓몬의 이름, 2단 진화 형태 포켓몬의 이름, 3단 진화 형태 포켓몬의 이름을 ID의 오름차순으로 출력하세요.
31. 진화 가능한 전체 포켓몬을 타입으로 분류한 후 포켓몬의 수가 3개 이상인 타입을 사전 역순으로 출력하세요.

SQL Practice

- 아래 내용을 출력하는 SQL 쿼리를 작성해 보세요.


32. 어느 트레이너에게도 잡히지 않은 포켓몬의 이름을 사전순으로 출력하세요.
33. 트레이너 Matis가 잡은 포켓몬들의 레벨의 총합을 출력하세요.
34. 체육관 관장이 잡은 포켓몬 중, 닉네임이 A로 시작하는 포켓몬의 이름과 레벨, 닉네임을 포켓몬의 이름의 사전 역순으로 출력하세요.
35. 진화 후의 ID가 더 작아지는 포켓몬의 진화 전 이름을 사전 순으로 출력하세요.
36. 진화가 가능한 포켓몬 중 최종 진화 형태의 포켓몬을 잡은 트레이너의 이름을 사전순으로 출력하세요.
37. 잡은 포켓몬 레벨의 총합이 가장 높은 트레이너의 이름과 그 총합을 출력하세요. 결과가 여러 명일 경우, 트레이너 이름의 사전순으로 출력하세요.
38. 진화 가능한 포켓몬 중 최종 진화 형태의 포켓몬의 이름을 사전순으로 출력하세요.
39. 동일한 포켓몬을 두 마리 이상 잡은 트레이너의 이름을 사전순으로 출력하세요.
40. 출신 도시명과, 각 출신 도시별로 트레이너가 잡은 가장 레벨이 높은 포켓몬의 닉네임을 출신 도시명의 사전순으로 출력하세요.


Submit


❖ 주의사항



- 과제 디렉토리 내에 project1 디렉토리를 새로 만들어서, gitlab에 푸시할 것.
- 과제 제출 시, 매 문제 마다, 문제 번호.sql 이라는 파일이름으로 제출할 것.
- 예시 사진

master ▾ dbprojecttest / + ▾

History Find file Web IDE  ▾ Clone ▾

 project1 example
이기택 authored 15 seconds ago

68ae0526 

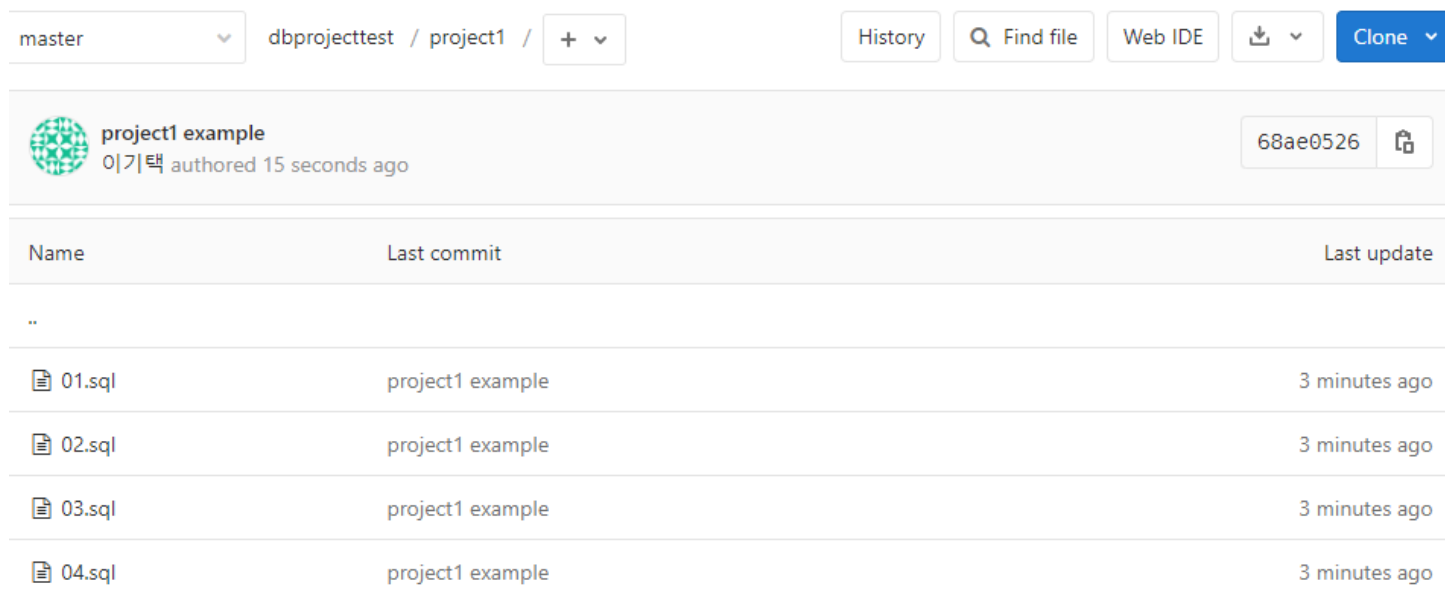
Name	Last commit	Last update
 project1	project1 example	4 minutes ago
 README.md	first commit	5 minutes ago

**** 주의사항을 지키지 않을 경우 올바르게 채점되지 않을 수 있습니다. ****

Submit

❖ 주의사항

- 과제 디렉토리 내에 project1 디렉토리를 새로 만들어서, gitlab에 푸시할 것.
- 과제 제출 시, 매 문제 마다, 문제 번호.sql 이라는 파일이름으로 제출할 것.
- 예시 사진



The screenshot shows a GitLab repository interface. At the top, there's a breadcrumb navigation: 'master' (with a dropdown arrow), 'dbprojecttest / project1 /' (with a '+' dropdown arrow), and buttons for 'History', 'Find file', 'Web IDE', a download icon, and 'Clone'. Below this is a header for the repository 'project1 example' by '이기택', 'authored 15 seconds ago', with a commit hash '68ae0526' and a copy icon. The main content is a table listing files in the repository.

Name	Last commit	Last update
..		
01.sql	project1 example	3 minutes ago
02.sql	project1 example	3 minutes ago
03.sql	project1 example	3 minutes ago
04.sql	project1 example	3 minutes ago

**** 주의사항을 지키지 않을 경우 올바르게 채점되지 않을 수 있습니다. ****

Submit

❖ 주의사항

- insert.sql 은 piazza 에서 다운로드하며 **절대 임의로 수정하지 말 것.**
- 반드시 데이터가 오류없이 모두 들어감을 확인한 후 다음 쿼리 문제를 해결할 것.
- UPDATE, DELETE 등 테이블의 data를 변경하는 쿼리는 수행하지 말 것.

****** 주의사항을 지키지 않을 경우 올바르게 채점되지 않을 수 있습니다. ******

Thank You
