HW2 --- Simple DBMS feature extend (Basic)

1 Overview

In this homework, you are asked to implement new features of the simpleDBMS, this project is under Apache 2.0 License, the detail of this project refer to its GitHub Repo**.** The expected result of this homework is to support the below query format

select { field } from table [offset <offset_num>] [limit <limit_num>]

1.1 Format description

field means the attributes in the table, i.e.: id, name, email, age
offset_num & limit_num are numeric input, they should always be unsigned int
[ ] means optional argument

2 Prerequisite

- Installation of simpleDBMS
- Git
- C programming language
  - You can also use C++ to implement this project, but it needs to pass all the system tests and unit test without any modification on test code, as well
- Understanding of simpleDBMS

3 Tasks

- Primary Key
- Limit argument
- Offset argument
- Projection (field select)

3.1 Primary Key

Implement ***user.id*** as the primary key in the Table

Example:

Table content:

ID	Name	email	age
1	user1	user1@example.com	20
2	user2	user2@example.com	21

If we execute the following command, the table will remain the same, because of the primary key restriction

```
db > insert 2 user3 user3@example.com 22
```

3.2 Limit argument

Implement limit argument to restrict the number of output

Example:

Table content:

ID	Name	email	age
1	user1	user1@example.com	20
2	user2	user2@example.com	21
3	user3	user3@example.com	22
4	user4	user4@example.com	23

If we execute the following command

```
db > select limit 2
```

The result will be as below shows because we restrict it only output 2 records

```
(1, user1, user1@example.com, 20)
(2, user2, user2@example.com, 21)
```

3.3 Offset argument

Implement offset argument to add offset for the query result

Example:

Table content:

ID	Name	email	age
1	user1	user1@example.com	20
2	user2	user2@example.com	21
3	user3	user3@example.com	22
4	user4	user4@example.com	23

If we execute the following command

```
db > select offset 2
```

The result will be as below shows because we shift the output with offset 2

```
(3, user3, user3@example.com, 22)
(4, user4, user2@example.com, 24)
```

3.4 Projection (field selection)

Implement projection(field selection) in select query

Example:

Table content:

ID	Name	email	age
1	user1	user1@example.com	20
2	user2	user2@example.com	21
3	user3	user3@example.com	22
4	user4	user4@example.com	23

If we execute the following command

```
db > select id, name from table
```

The result will be as below shows

```
(1, user1)
(2, user2)
(3, user3)
(4, user4)
```

4 Testing

This project is implemented with two kinds of tests, the different type of tests focuses on a different aspect.

4.1 Unit tests

The unit tests are focus on the function level specification, it will check the input and output of the function.

Use the following command to execute unit tests

```
$ make check
```

4.2 System tests (For above requirement)

The system tests focus on system level behavior, and these tests are also used to test the new features you need to implement

Use the following command to execute system tests

- `$ python test/system/system_test.py ./shell [test_case [test_case ...]]`
  - Execute this command in the root folder of this project
  - By default, if no test_cases are given, it will run all test cases, otherwise, it will execute all specified test cases.
  - To see the list of test cases, please check `test/system/testcases` folder
- `$ python test/system/system_test.py [-h]`
  - Show the usage of system test
- You can check the difference of your test output with the expected answer by comparing two files in these two paths
  "test/system/output/<test_case>/<output file>"
  "test/system/answer/<test_case>/<answer_file>"
  - Hint: 1) `diff` command can show the difference of two files
  - 2) Install `vim` and using `vimdiff` provide a better looking

5 Requirements:

- Pass unit tests
- Pass provided system tests
- You should at least 5 commits to your own repository
- Upload a zip file to New E3
- Makefile
    - The original project already contains a Makefile, your final submission should also be able to compile the shell using Makefile, and the compiled binary should be **shell** as we provided, **otherwise you will get 0 score**

Passing unit tests and system tests only means that your program reaches the basic requirements. There are some hidden tests we do not provide and we will test them on your submission. Therefore, make sure your program works well on query with different combinations of features(field selection + offset + limit).

6 Submission

6.1 E3

- Compress your source code into one single zip file named \`<student_ID>.zip\` (like 0756000.zip), and upload to New E3

6.2 GitLab

- Create an project named <student_id>_hw2 in the Group DBMS_2019/<student_id>
    - The <student_id> should be replaced with your own student_id, such as 0756000_hw2 in group DBMS_2019/0756000

6.3 Deadline

- You are required to hand in your homework before 2019/04/21 23:59
- Late submission is not allowed

7 Disscussion Forum

HW2 discussion <HackMD>

Plagiarism is not allowed, you will get 0 points when we found that happened.