Assignment-4 CS313

Shashank P
200010048

September 13, 2022

1 Problem 1

Create a user called universityDB0048.

```
create user 'universityDB0048'@'localhost' identified by 'MyPassword@123';
grant all privileges on university.* to 'universityDB0048'@'localhost';
/* Logout and login as user universityDB0048 */
```

```
mysql> create user 'universityDB0048'@'localhost' identified by 'MyPassword@123';
Query OK, 0 rows affected (0.15 sec)

mysql> \q
Bye
shashankp@ubuntu:~$ mysql -u univeristyDB0048 -p'MyPassword@123'
mysql: [Warning] Using a password on the command line interface can be insecure.
ERROR 1045 (28000): Access denied for user 'univeristyDB0048'@'localhost' (using password: YES)
shashankp@ubuntu:~$ mysql -u universityDB0048 -p'MyPassword@123'
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.30-0ubuntu0.20.04.2 (Ubuntu)

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Figure 1: Creation of User and adding previleges

2 Problem 2

Create a database called university.

```
1  create database university;
```

```
mysql> create database university;
Query OK, 1 row affected (0.01 sec)

mysql>
```

Figure 2: Create Database

3 Problem 3

Use the database named university.

```
1  use university;
```

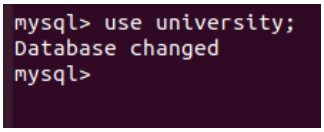


Figure 3: Use Database

4 Problem 4

Create the tables in the university database using DDL.sql file.

```
source DDL.sql;
```

```
mysql> source DDL.sql;
Query OK, 0 rows affected (0.03 sec)

Query OK, 0 rows affected (0.02 sec)

Query OK, 0 rows affected (0.02 sec)

Query OK, 0 rows affected (0.01 sec)

Query OK, 0 rows affected (0.02 sec)

Query OK, 0 rows affected (0.02 sec)

Query OK, 0 rows affected (0.02 sec)

Query OK, 0 rows affected (0.02 sec)

Query OK, 0 rows affected (0.01 sec)

Query OK, 0 rows affected (0.02 sec)

Query OK, 0 rows affected (0.01 sec)
```

Figure 4: Create Output

```
mysql> show tables;
+----------------------+
| Tables_in_university |
+----------------------+
| advisor              |
| classroom            |
| course               |
| department           |
| instructor           |
| prereq               |
| section              |
| student              |
| takes                |
| teaches              |
| time_slot            |
+----------------------+
11 rows in set (0.00 sec)
```

Figure 5: Create Tables

5 Problem 5

Load the data into tables using InsertValues.sql.

```
source InsertValues.sql;
```

```
Query OK, 1 row affected (0.01 sec)

Query OK, 1 row affected (0.01 sec)

Query OK, 1 row affected (0.00 sec)

Query OK, 1 row affected (0.00 sec)

Query OK, 1 row affected (0.00 sec)

Query OK, 1 row affected (0.01 sec)

Query OK, 1 row affected (0.00 sec)

Query OK, 1 row affected (0.01 sec)

Query OK, 1 row affected (0.00 sec)

Query OK, 1 row affected (0.00 sec)

Query OK, 1 row affected (0.00 sec)

Query OK, 1 row affected (0.00 sec)

Query OK, 1 row affected (0.01 sec)

Query OK, 1 row affected (0.00 sec)

Query OK, 1 row affected (0.00 sec)
```

Figure 6: Insert QUery Output

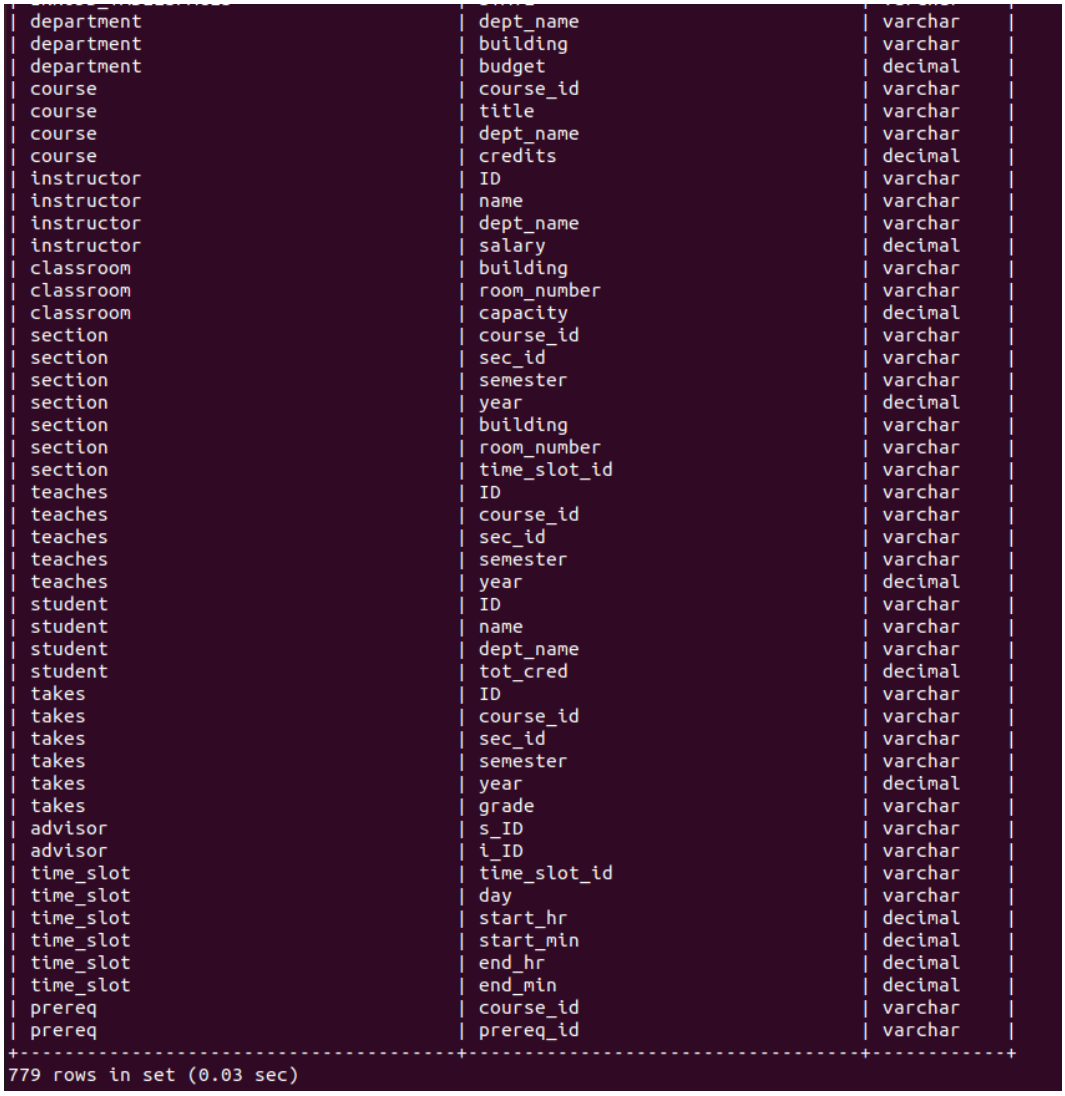
```
mysql> select * from student;
+-------+----------+------------+----------+
| ID    | name     | dept_name  | tot_cred |
+-------+----------+------------+----------+
| 00128 | Zhang    | Comp. Sci. |      102 |
| 12345 | Shankar  | Comp. Sci. |       32 |
| 19991 | Brandt   | History    |       80 |
| 23121 | Chavez   | Finance    |      110 |
| 44553 | Peltier  | Physics    |       56 |
| 45678 | Levy     | Physics    |       46 |
| 54321 | Williams | Comp. Sci. |       54 |
| 55739 | Sanchez  | Music      |       38 |
| 70557 | Snow     | Physics    |        0 |
| 76543 | Brown    | Comp. Sci. |       58 |
| 76653 | Aoi      | Elec. Eng. |       60 |
| 98765 | Bourikas | Elec. Eng. |       98 |
| 98988 | Tanaka   | Biology    |      120 |
+-------+----------+------------+----------+
13 rows in set (0.00 sec)
```

Figure 7: Insert into Tables (Example: student table)

6 Problem 6

Get the details of all the tables using informatio_schema.

```
1   select table_name, column_name, data_type from information_schema.columns;
```



department	dept_name	varchar
department	building	varchar
department	budget	decimal
course	course_id	varchar
course	title	varchar
course	dept_name	varchar
course	credits	decimal
instructor	ID	varchar
instructor	name	varchar
instructor	dept_name	varchar
instructor	salary	decimal
classroom	building	varchar
classroom	room_number	varchar
classroom	capacity	decimal
section	course_id	varchar
section	sec_id	varchar
section	semester	varchar
section	year	decimal
section	building	varchar
section	room_number	varchar
section	time_slot_id	varchar
teaches	ID	varchar
teaches	course_id	varchar
teaches	sec_id	varchar
teaches	semester	varchar
teaches	year	decimal
student	ID	varchar
student	name	varchar
student	dept_name	varchar
student	tot_cred	decimal
takes	ID	varchar
takes	course_id	varchar
takes	sec_id	varchar
takes	semester	varchar
takes	year	decimal
takes	grade	varchar
advisor	s_ID	varchar
advisor	i_ID	varchar
time_slot	time_slot_id	varchar
time_slot	day	varchar
time_slot	start_hr	decimal
time_slot	start_min	decimal
time_slot	end_hr	decimal
time_slot	end_min	decimal
prereq	course_id	varchar
prereq	prereq_id	varchar

779 rows in set (0.03 sec)

Figure 8: Schema Information for each table

7 Problem 7

7.1 Login

Login to the user which you have created in Question number 1.

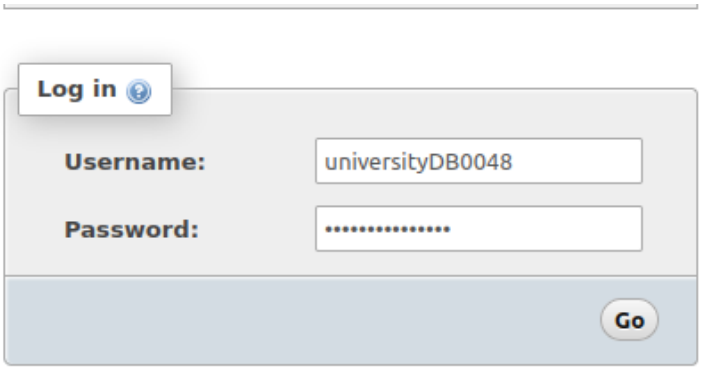


Figure 9: Logging into phpmyadmin

7.2 University Database

Use database University.

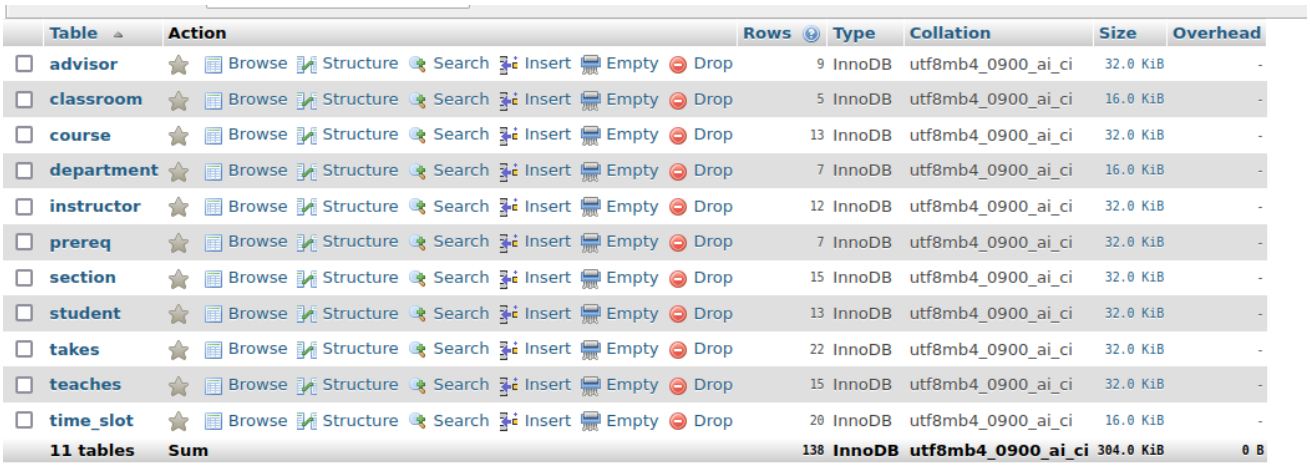


	Table	Action	Rows	Type	Collation	Size	Overhead
☐	advisor	Browse Structure Search Insert Empty Drop	9	InnoDB	utf8mb4_0900_ai_ci	32.0 KiB	-
☐	classroom	Browse Structure Search Insert Empty Drop	5	InnoDB	utf8mb4_0900_ai_ci	16.0 KiB	-
☐	course	Browse Structure Search Insert Empty Drop	13	InnoDB	utf8mb4_0900_ai_ci	32.0 KiB	-
☐	department	Browse Structure Search Insert Empty Drop	7	InnoDB	utf8mb4_0900_ai_ci	16.0 KiB	-
☐	instructor	Browse Structure Search Insert Empty Drop	12	InnoDB	utf8mb4_0900_ai_ci	32.0 KiB	-
☐	prereq	Browse Structure Search Insert Empty Drop	7	InnoDB	utf8mb4_0900_ai_ci	32.0 KiB	-
☐	section	Browse Structure Search Insert Empty Drop	15	InnoDB	utf8mb4_0900_ai_ci	32.0 KiB	-
☐	student	Browse Structure Search Insert Empty Drop	13	InnoDB	utf8mb4_0900_ai_ci	32.0 KiB	-
☐	takes	Browse Structure Search Insert Empty Drop	22	InnoDB	utf8mb4_0900_ai_ci	32.0 KiB	-
☐	teaches	Browse Structure Search Insert Empty Drop	15	InnoDB	utf8mb4_0900_ai_ci	32.0 KiB	-
☐	time_slot	Browse Structure Search Insert Empty Drop	20	InnoDB	utf8mb4_0900_ai_ci	16.0 KiB	-
	11 tables	**Sum**	**138**	**InnoDB**	**utf8mb4_0900_ai_ci**	**304.0 KiB**	**0 B**

Figure 10: Using university database

7.3 Select and Insert Queries

Performed select and insert queries on each table.

```
insert into department values('My New Department', 'Watson', 200000);
select * from department where budget between 50000 and 210000;

insert into course values('NN-101', 'A New Course', 'History', 5);
select max(credits) from course;

insert into instructor values('12345', 'A New Instructor', 'Comp. Sci.', 50000);
select ID, name from instructor where dept_name in ('Comp. Sci.', "Physics");

insert into section values('CS-101', '3', 'Summer', 2019, 'Watson', '100', 'A');
select * from section where course_id='CS-101' order by time_slot_id asc;

insert into teaches values('45565', 'CS-101', '1', 'Fall', 2009);
select max(year), min(year) from teaches where ID='45565';

insert into student values('99999', 'New Name', 'Comp. Sci.', 250);
select ID, name from student where tot_cred in (select max(tot_cred) from student);

insert into takes values('44553', 'CS-347', '1', 'Fall', 2009, 'A-');
select * from takes where ID=='44553';

insert into advisor values('19991', '22222');
select * from advisor where i_ID='22222';

insert into time_slot values('NEW', 'R', 13, 31, 14, 45);
select * from time_slot where day='R';

insert into prereq values('PHY-101', 'BIO-101');
select * from prereq where prereq_id='BIO-101';
```

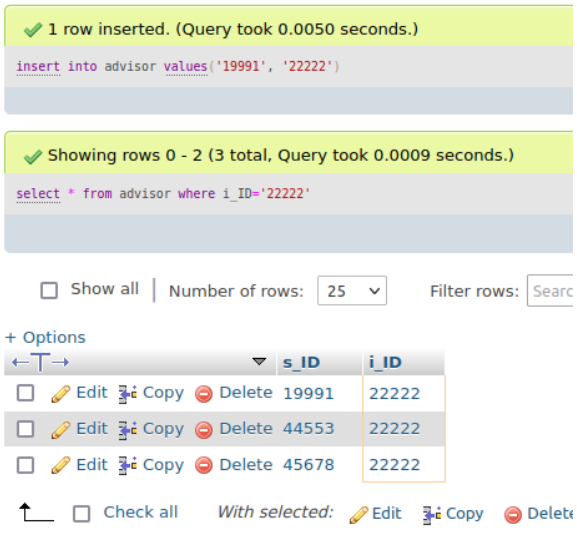


Figure 11: Select and Insert (Ex: advisor table)

8 Problem 8

8.1 Question 1

Instructors from CSE department teaching Civil courses. Since there was no such data I have inserted an instructor and a course that satisfy these conditions.

```
1  select distinct(course.course_id), course.title, instructor.ID, instructor.name
2  from (instructor natural join teaches), course
3  where course.course_id=teaches.course_id
4    and instructor.dept_name='Comp. Sci.'
5    and teaches.year=2009
6      and course.dept_name='Civil'
7  order by instructor.name ASC;
```

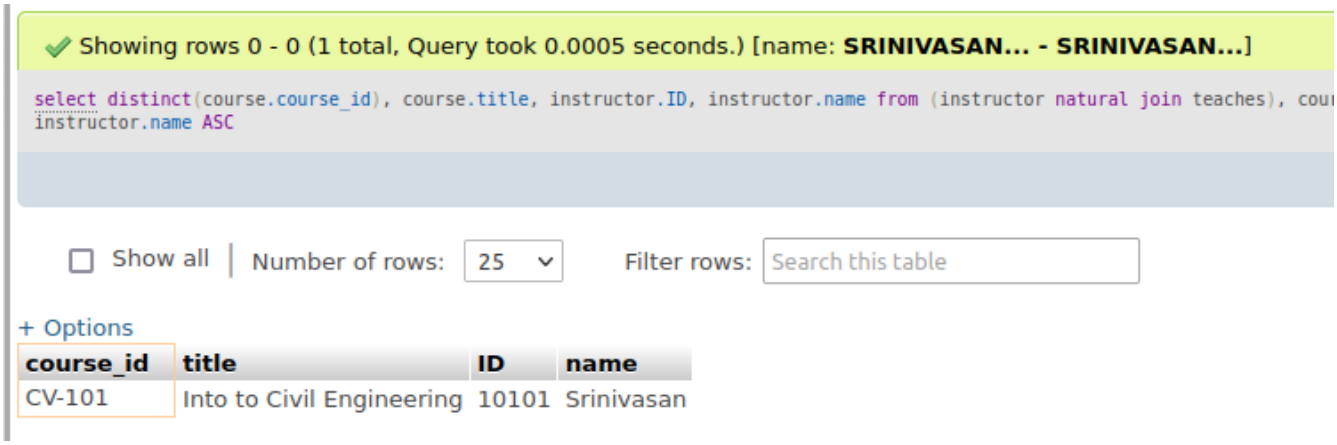


Figure 12: Question 1

8.2 Question 2

Insert a new course and its pre-requisite.

```
insert into course values('CS-303', 'DBIS', 'Comp. Sci.', 6),
('CS-333', 'New Course', 'Comp. Sci.', 3);
insert into prereq values('CS-333', 'CS-303');
select * from prereq;
```

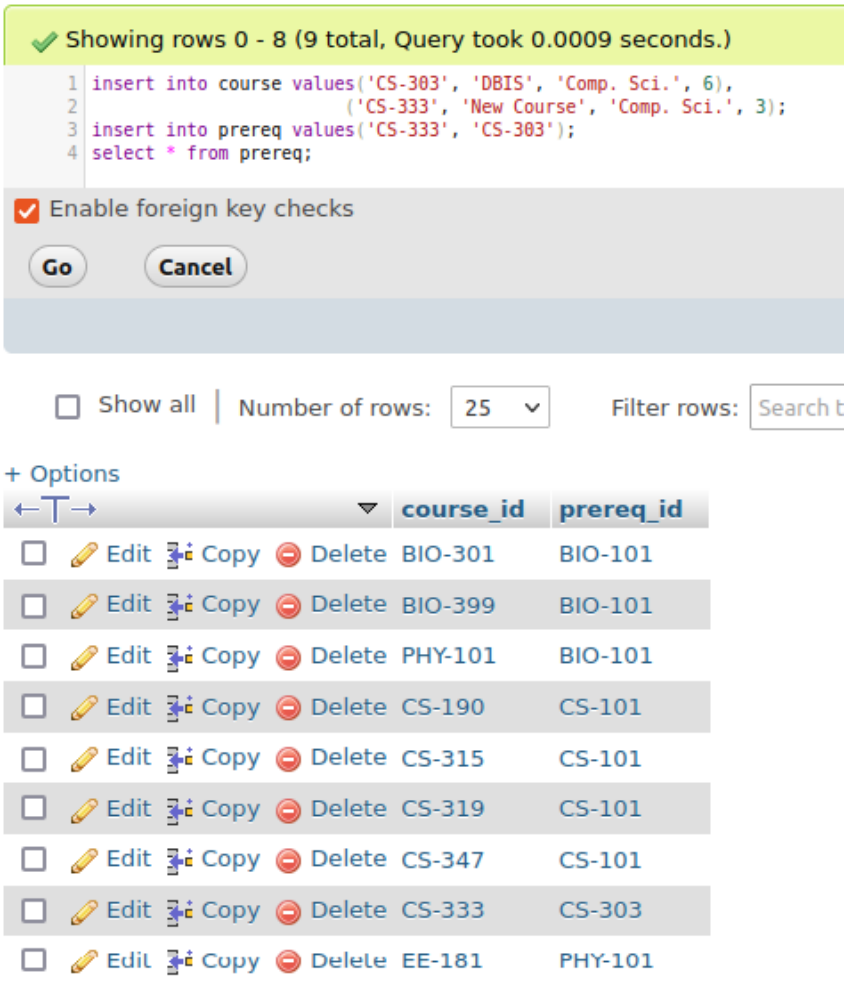


Figure 13: Question 2

8.3 Question 3

Increment salaries of instructors whose department budget is greater than 90000.

```
update instructor
set salary = salary*1.1
where dept_name in (select department.dept_name
                    from department
                    where departmenet.budget>90000);
```

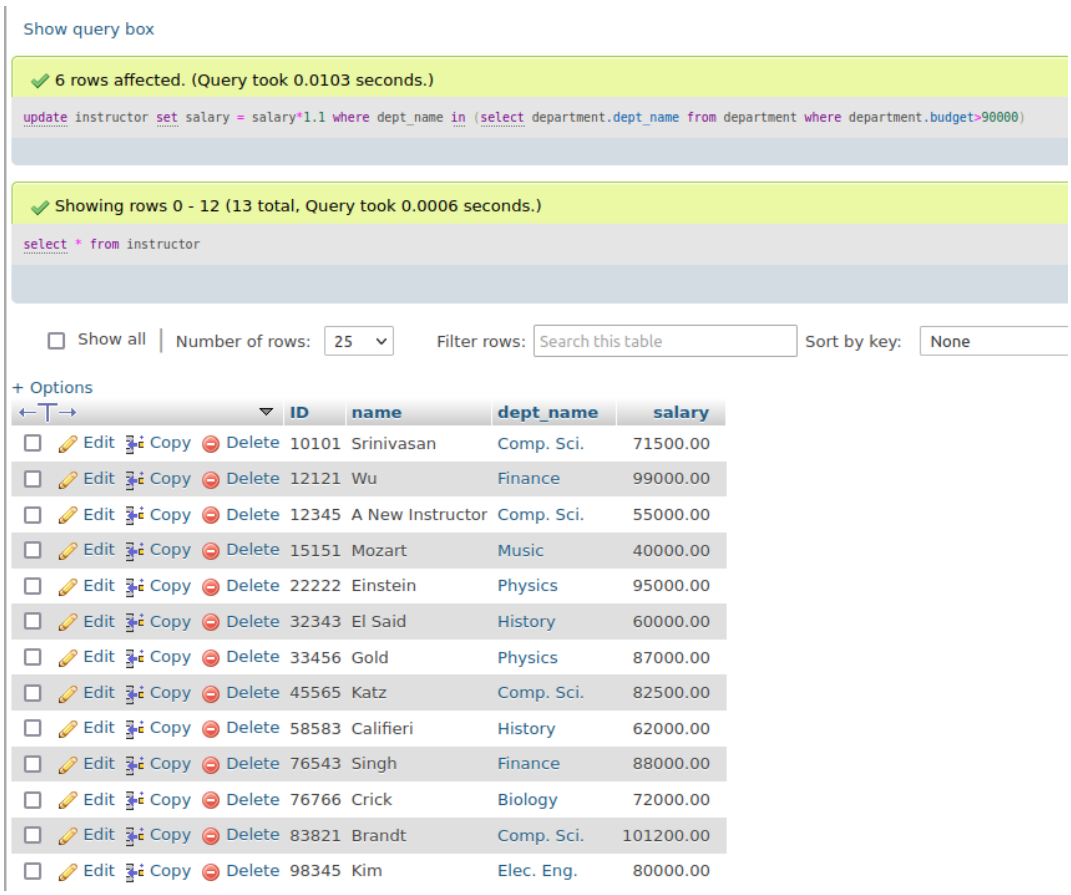


Figure 14: Question 3

8.4 Question 4

Get Courses taken by some minimum number of students based on semester and year. Since no such data existed I have modified the values of the query keeping the essence of the question same.

```
select count(ID), course_id
from takes natural join course
where course.dept_name="Comp. Sci."
    and takes.year=2009
    and takes.semester="Fall"
group by course_id
having count(ID)>2
order by course_id ASC;
```

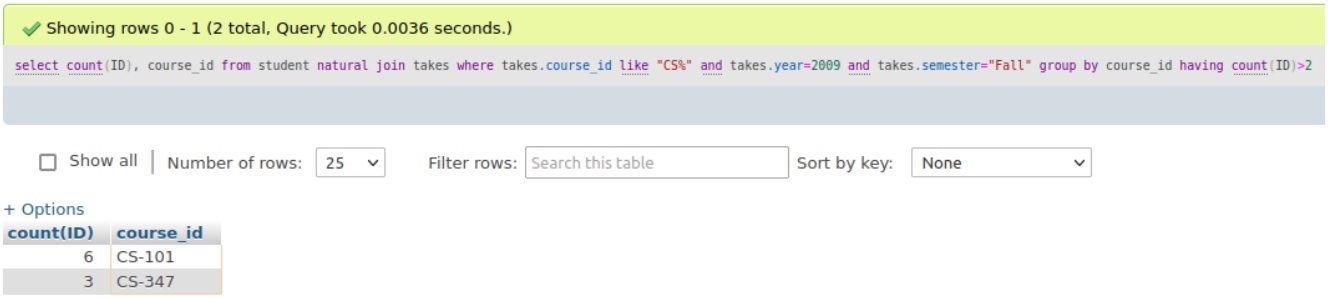


Figure 15: Question 4