

Assignment-4 CS313

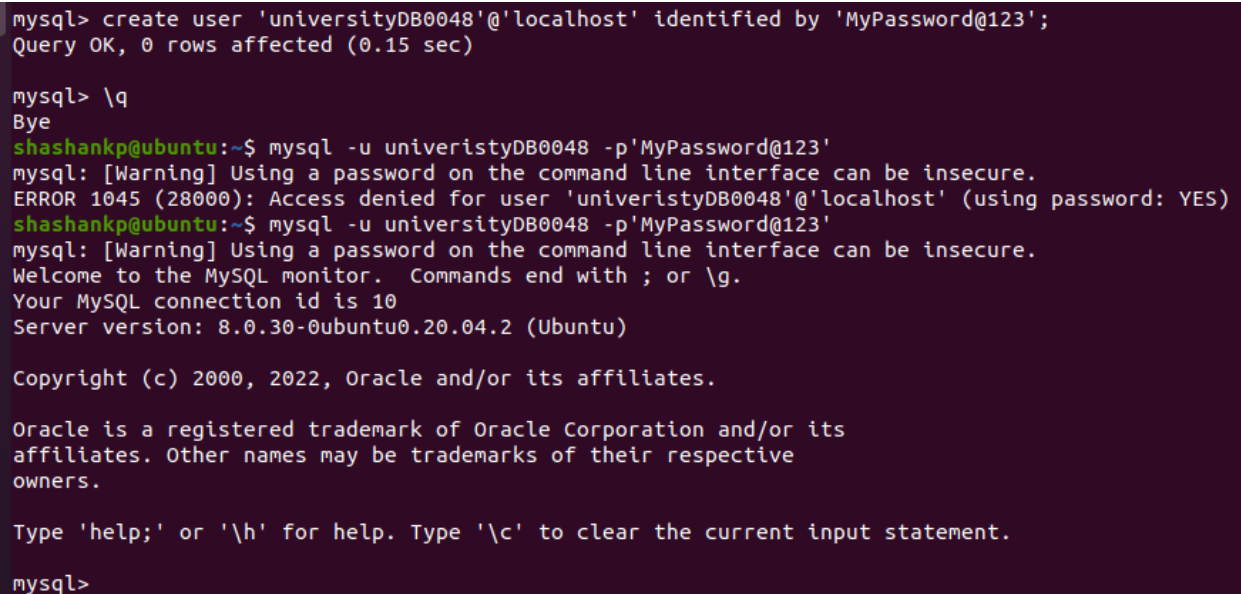
Shashank P
200010048

September 13, 2022

1 Problem 1

Create a user called universityDB0048.

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



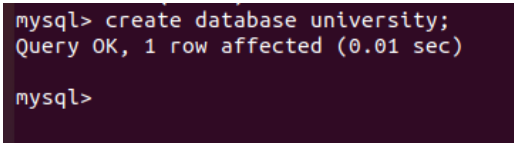
```
mysql> create user 'universityDB0048'@'localhost' identified by 'MyPassword@123';  
Query OK, 0 rows affected (0.15 sec)  
  
mysql> \q  
Bye  
shashankp@ubuntu:~$ mysql -u univeristyDB0048 -p'MyPassword@123'  
mysql: [Warning] Using a password on the command line interface can be insecure.  
ERROR 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 privileges

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>
```

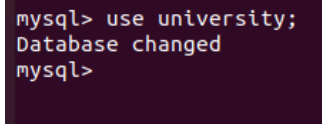
Figure 2: Create Database

3 Problem 3

Use the database named university.

1

```
use university;
```



```
mysql> use university;  
Database changed  
mysql>
```

Figure 3: Use Database

4 Problem 4

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

```
1 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.

```
1 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 (0.00 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 `information_schema`.

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

```

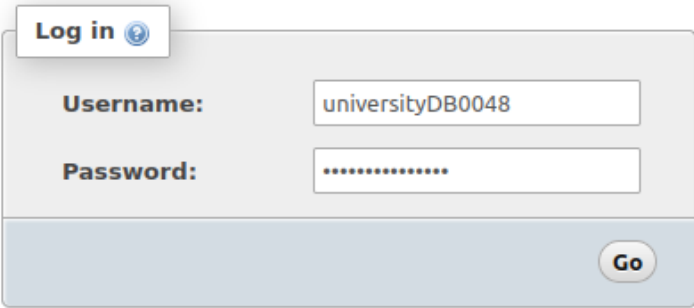
| department | dept_name | varchar |
| department | building  | varchar |
| department | budget    | decimal |
| course      | course_id | varchar |
| course      | title     | varchar |
| course      | dept_name | varchar |
| course      | credits    | decimal |
| instructor  | ID        | varchar |
| 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.



The screenshot shows the phpmyadmin login interface. At the top left, there is a 'Log in' button with a question mark icon. Below this, the form has two input fields: 'Username:' with the value 'universityDB0048' and 'Password:' with a masked password represented by dots. A 'Go' button is located at the bottom right of the form.

Figure 9: Logging into phpmyadmin

7.2 University Database

Use database University.

Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> advisor	★ Browse Structure Search Insert Empty Drop	9	InnoDB	utf8mb4_0900_ai_ci	32.0 KiB	-
<input type="checkbox"/> classroom	★ Browse Structure Search Insert Empty Drop	5	InnoDB	utf8mb4_0900_ai_ci	16.0 KiB	-
<input type="checkbox"/> course	★ Browse Structure Search Insert Empty Drop	13	InnoDB	utf8mb4_0900_ai_ci	32.0 KiB	-
<input type="checkbox"/> department	★ Browse Structure Search Insert Empty Drop	7	InnoDB	utf8mb4_0900_ai_ci	16.0 KiB	-
<input type="checkbox"/> instructor	★ Browse Structure Search Insert Empty Drop	12	InnoDB	utf8mb4_0900_ai_ci	32.0 KiB	-
<input type="checkbox"/> prereq	★ Browse Structure Search Insert Empty Drop	7	InnoDB	utf8mb4_0900_ai_ci	32.0 KiB	-
<input type="checkbox"/> section	★ Browse Structure Search Insert Empty Drop	15	InnoDB	utf8mb4_0900_ai_ci	32.0 KiB	-
<input type="checkbox"/> student	★ Browse Structure Search Insert Empty Drop	13	InnoDB	utf8mb4_0900_ai_ci	32.0 KiB	-
<input type="checkbox"/> takes	★ Browse Structure Search Insert Empty Drop	22	InnoDB	utf8mb4_0900_ai_ci	32.0 KiB	-
<input type="checkbox"/> teaches	★ Browse Structure Search Insert Empty Drop	15	InnoDB	utf8mb4_0900_ai_ci	32.0 KiB	-
<input type="checkbox"/> 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.

```
1  insert into department values('My New Department', 'Watson', 200000);
2  select * from department where budget between 50000 and 210000;
3
4  insert into course values('NN-101', 'A New Course', 'History', 5);
5  select max(credits) from course;
6
7  insert into instructor values('12345', 'A New Instructor', 'Comp. Sci.', 50000);
8  select ID, name from instructor where dept_name in ('Comp. Sci.', 'Physics');
9
10 insert into section values('CS-101', '3', 'Summer', 2019, 'Watson', '100', 'A');
11 select * from section where course_id='CS-101' order by time_slot_id asc;
12
13 insert into teaches values('45565', 'CS-101', '1', 'Fall', 2009);
14 select max(year), min(year) from teaches where ID='45565';
15
16 insert into student values('99999', 'New Name', 'Comp. Sci.', 250);
17 select ID, name from student where tot_cred in (select max(tot_cred) from student);
18
19 insert into takes values('44553', 'CS-347', '1', 'Fall', 2009, 'A-');
20 select * from takes where ID='44553';
21
22 insert into advisor values('19991', '22222');
23 select * from advisor where i_ID='22222';
24
25 insert into time_slot values('NEW', 'R', 13, 31, 14, 45);
26 select * from time_slot where day='R';
27
28 insert into prereq values('PHY-101', 'BIO-101');
29 select * from prereq where prereq_id='BIO-101';
```

✓ 1 row inserted. (Query took 0.0050 seconds.)

insert into advisor values('19991', '22222')

✓ Showing rows 0 - 2 (3 total, Query took 0.0009 seconds.)

select * from advisor where i_ID='22222'

☐ Show all | Number of rows: 25 Filter rows: Search

+ Options

	s_ID	i_ID
<input type="checkbox"/> Edit Copy Delete	19991	22222
<input type="checkbox"/> Edit Copy Delete	44553	22222
<input type="checkbox"/> Edit Copy Delete	45678	22222

☐ Check all With selected: Edit Copy Delete


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;
```

 Showing rows 0 - 0 (1 total, Query took 0.0005 seconds.) [name: **SRINIVASAN... - SRINIVASAN...**]

```
select distinct(course.course_id), course.title, instructor.ID, instructor.name from (instructor natural join teaches), cou
instructor.name ASC
```

☐ Show all | Number of rows: Filter rows:

+ Options

course_id	title	ID	name
CV-101	Into to Civil Engineering	10101	Srinivasan

Figure 12: Question 1

8.2 Question 2

Insert a new course and its pre-requisite.

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

✓ Showing rows 0 - 8 (9 total, Query took 0.0009 seconds.)

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

☒ Enable foreign key checks

Go

Cancel

☐ Show all | Number of rows: 25 ▾ Filter rows:

+ Options

← T → ▾

	course_id	prereq_id
<input type="checkbox"/> Edit Copy Delete	BIO-301	BIO-101
<input type="checkbox"/> Edit Copy Delete	BIO-399	BIO-101
<input type="checkbox"/> Edit Copy Delete	PHY-101	BIO-101
<input type="checkbox"/> Edit Copy Delete	CS-190	CS-101
<input type="checkbox"/> Edit Copy Delete	CS-315	CS-101
<input type="checkbox"/> Edit Copy Delete	CS-319	CS-101
<input type="checkbox"/> Edit Copy Delete	CS-347	CS-101
<input type="checkbox"/> Edit Copy Delete	CS-333	CS-303
<input type="checkbox"/> Edit Copy Delete	EE-181	PHY-101

Figure 13: Question 2

8.3 Question 3

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

```
1 update instructor
2 set salary = salary*1.1
3 where dept_name in (select department.dept_name
4                     from department
5                     where department.budget>90000);
```

Show query box

✓ 6 rows affected. (Query took 0.0103 seconds.)

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

✓ Showing rows 0 - 12 (13 total, Query took 0.0006 seconds.)

```
select * from instructor
```

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

+ Options

	ID	name	dept_name	salary
<input type="checkbox"/> Edit Copy Delete	10101	Srinivasan	Comp. Sci.	71500.00
<input type="checkbox"/> Edit Copy Delete	12121	Wu	Finance	99000.00
<input type="checkbox"/> Edit Copy Delete	12345	A New Instructor	Comp. Sci.	55000.00
<input type="checkbox"/> Edit Copy Delete	15151	Mozart	Music	40000.00
<input type="checkbox"/> Edit Copy Delete	22222	Einstein	Physics	95000.00
<input type="checkbox"/> Edit Copy Delete	32343	El Said	History	60000.00
<input type="checkbox"/> Edit Copy Delete	33456	Gold	Physics	87000.00
<input type="checkbox"/> Edit Copy Delete	45565	Katz	Comp. Sci.	82500.00
<input type="checkbox"/> Edit Copy Delete	58583	Califieri	History	62000.00
<input type="checkbox"/> Edit Copy Delete	76543	Singh	Finance	88000.00
<input type="checkbox"/> Edit Copy Delete	76766	Crick	Biology	72000.00
<input type="checkbox"/> Edit Copy Delete	83821	Brandt	Comp. Sci.	101200.00
<input type="checkbox"/> Edit Copy Delete	98345	Kim	Elec. Eng.	80000.00

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.

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

✓ Showing rows 0 - 1 (2 total, Query took 0.0036 seconds.)

select count(ID), course_id from student natural join takes where takes.course_id like "CS%" and takes.year=2009 and takes.semester="Fall" group by course_id having count(ID)>2

☐ Show all | Number of rows: 25 ▾ | Filter rows: | Sort by key: None ▾

+ Options

count(ID)	course_id
6	CS-101
3	CS-347

Figure 15: Question 4