1. Submit the DDL queries for the followings,
   1. You need to create a movie database. Then create three tables, one for actors(AID, name), one for movies(MID, title) and one for actor\_role(MID, AID, rolename). Use appropriate data types for each of the attributes, and add appropriate primary. **Do not add any foreign key constraints when you are creating the table.** **10 points**

create database movie;

create table actors(

AID char(8),

name varchar(20),

primary key (AID));

create table movies(

MID char(8),

title varchar(20),

primary key (MID));

create table actor\_role(

MID char(8),

AID char(8),

rolename varchar(20),

primary key (MID, AID));

* 1. Now modify the schemas as follow, **10 points**
     1. Add a new column called “rating” in the movies table. The rating is a numerical value.

alter table movies add rating numeric(4,2);

* + 1. Add the necessary foreign key references. [examples queries can be found in S2]

alter table actor\_role add foreign key (AID) references actors(AID);

alter table actor\_role add foreign key (MID) references movies(MID);

1. Write the DQL queries on the university database that you have created on your computer for the following sentences,
   1. Find the names of instructors with the salary between 70000 and 100000. **5 points**

select name from instructor where salary between 70000 and 100000;

* 1. Find All departments with the budget greater than the budget of the Physics *department*. **5 points**

select dept\_name from department where budget > all (select budget from department where dept\_name = ‘Physics’);

* 1. Find the instructors with exactly four characters in their names. **5 Points**

select name from instructor where name LIKE ‘\_\_\_\_’;

* 1. Find the ID and name of the instructor with the highest salary **5 Points**

select ID, name from instructor where salary = (select max(salary) from instructor);

* 1. Find the ID and name of the instructor who has the highest salary in the Department of Computer Science. **5 Points**

select ID, name from instructor where salary = (select max(salary) from instructor where dept\_name= ‘ Comp. Sci.’);

* 1. In the university database, each course section is associated with a classroom number. So, each classroom is assigned multiple times for multiple courses over the years. Find the number of courses scheduled in a room each year. **5 Points**

select count (course\_id) from section where room\_number = ‘ a given room number’;

\* ‘a given room number’ is to take the place of any arbitrary room number as the question does not specify a specific room number

* 1. From the answer to the previous question, can you find the classroom with the highest schedules over the years? Also, show the year. **5 Points**

select year from section( select max( select count (course\_id) from section where room\_number = ‘ a given room number’));

* 1. Find the ***ID*** of the instructors who have taught at least two courses. **5 Points**

select distinct ID from teaches where 1 < (select count(ID) from teaches) ;

* 1. Find the ***ID and name*** of the instructors who have taught at least two courses. **5 Points**

[select](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) ID, name from instructor where ID =some([select](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) distinct ID from teaches where 1 < ([select](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) [count](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/group-by-functions.html#function_count)(ID) from teaches) );

1. Write the DML queries for the following questions
   1. Insert at least two records in the *takes* table. Display the *takes* table before and after the insertions.  **5 points**

select \* from takes;

insert into takes values (‘05142’, ‘CS33007’, ‘01’, ‘Fall’, ‘2020’, ‘Jr’);

insert into takes values (‘82404’, ‘CS23001’, ‘02’, ‘Spring’, ‘2019’, ‘Jr’);

select \* from takes;

* 1. Update the budget of the Music department by 10%. Show the department table before and after the update.  **5 points**

select \* from department;

update department set budget = budget \*1.1 where dept\_name = ‘Music’;

select \* from department;

* 1. Delete the classrooms with the capacity less than 15. Show the classroom table before and after the deletion. **5 points**

select \* from classroom;

delete from classroom where capacity < 15;

select \* from classroom;