Q1) Write SQL queries to create the following tables. Make sure that you specify the **primary** and **foreign** keys in your queries. Also, the agency column cannot contain null values. (10)

films (film_id, title, author_name) (2)

roles (actor_id, film_id, character_name) (2,2,1)

actors (actor_id, name, agency) (3)

create table films

 (film_id varchar (20),

title varchar (50),

author_name varchar (30),

primary key (film_id));

create table roles

 (actor_id varchar (20),

film_id varchar (20),

character_name varchar (30),

primary key (actor_id, film_id, character_name).

foreign key (actor_id) references actors,

foreign key(film_id) references films);

create table actors

 (actor_id varchar (20),

name varchar (30),

agency varchar (30) not null,

primary key (actor_id));

Q2) Given the following schema, find

    a)   Write an SQL query that counts the number of students who have scored an A in PHY-101 in 2018. (5)

        SELECT count(ID)
        FROM Takes
        WHERE grade = 'A' and year = 2018 and course_id = PHY-101

b) Write an SQL query to display the department name and the highest salary that they offer to their instructors. (5)

```
        select dept_name, max (salary) as max_salary
from instructor
group by dept_name
```

c) Write an SQL query to display the student IDs of students who took CS 225 in Fall 2018 but did not take CS 300 in Spring 2018. (7)

```
(select ID from Takes where semester = 'Fall' and year= 2018) except (select ID from Takes where semester = 'Spring' and year= 2018);
```

d) Write an SQL query to display the names of instructors from Computer Science who have taught a course in 2019. (7)

```
        select name
from instructor, teaches
where instructor.ID = teaches.ID  and  instructor. dept_name = 'Computer Science' and teaches.year = 2019
```

e) Give a $500 rise to instructors whose salary is equal to the minimum salary offered overall (6).
```
update instructor
set salary = salary + 500
where salary = (select min (salary)
from instructor);
```