Name:

CS 33007 Introduction to Database System Design, Summer 2019

Midterm Solutions

Instructions:

- *This examination is closed book (no access to the book, lecture notes, phone, laptop, tablet, etc.).*
- *Please write your answer in the given blank space for each question. If your answer doesn't fit in the given space, you can use the back side of the papers but write question number.*
- ***A separate sheet will be provided for the university database relational schema.***

Total Points:100 **Time:** 3:00PM – 4:10PM

1.

a. Identify candidate keys for following relation schemas, [***5 points***]

Doctors (DoctorID, email, Name, MailingAddress, Salary)
Candidate Keys: DoctorID; email
Patient (PatientID, SSN, name, address)
Candidate Keys: PatientID; SSN

b. Explain referential integrity constraints with examples. [***15 points***]

Answer: value that appears in one relation for a given set of attributes must appears for a certain set of attributes in another relation. [Details in chapter 4-slide 30]

c. Write query for creating relation *account=(account_number, customer_name, account_type, branch_name, balance).* The primary key of the table is account_number, and the branch_name is a foreign key from the branch table. For account_type you must use integrity constraints check so that only "checking" or "savings" are allowed to insert. [***10 points***]

Answer:

```
Create table account(
   account_number char (20),
   customer_name varchar(20),
   account_type char(10) check(account_type in('checking','savings')),
   branch_name varchar(20),
   balance int,
   primary key (account_number),
   foreign key (branch_name) references branch(branch_name)
);
```

2.

a. Write the following queries in relational algebra, using the university database schema. [***10 points]***

i. Find the name of the instructors whose salary is between 70000 and 85000.

Answer: $\prod_{name}(\sigma_{salary>=70000 \wedge salary<=85000}(instructor)))$

ii. Find the courses that have prerequisite CS12401.

Answer: $\prod_{course\_id}(\sigma_{prereq\_id = 'CS12401'}(prereq))$

b. Write SQL queries for the following statements (Considering the university database) **[15 points]**

i. Find all students of the History department by the descending order of their total earned credits.

Select * from student where dept_name ="History" order by tot_cred desc

ii. Find the instructors with exactly three characters in their names.

select name from instructor where name like "___";

iii. Update the salary of instructors by 5% whose current salary is between 70000 and 85000

update instructor set salary = salary*1.05 where salary between 70000 and 85000;

3.

a. Find the names of the instructors who have taught one or more courses in his/her department. You must also show the title of the course they have taught. ***[10 points]***

Select name, title **from** instructor **natural join** teaches **natural join** course ;

b. In the university database, create a view *MSBinstructors*, showing all information about instructors from the Comp. Sci. and Math department. ***[8 points]***

create view MSBinstructors as SELECT * from instructor where dept_name = "Comp. Sci." or dept_name = "Math";

c. Grant permission to one of your friends to view all data in your student relation of university database. Also, make sure that you are granting your friend to pass the permission to others. [***7 points]***

grant select on university.student to 'friends'@'localhost' with grant option;

4.

a. Write a SQL procedure that takes the department name as input and increase the salary 10% only for the instructors whose salary is less than the average salary of the instructors of the department. ***[10 points]***

```
DELIMITER //
CREATE procedure increment_on_salary(in dept_name varchar(30))
BEGIN
        DECLARE avg_salary float;
  SELECT AVG(salary) into avg_salary from instructor where instructor.dept_name = dept_name;
  update instructor
        set salary = 1.1*salary where salary<avg_salary and instructor.dept_name = dept_name;
END //
DELIMITER ;
```

b. What is the trigger? Explain the purpose of using a trigger in the database with an example. **[10 points]**

The purpose of using trigger is to enable automatic operations on database based on the specific events. This helps to improve data integrity. Example can be anything supporting the definition above. It can be descriptive or using SQL

Ex1: When a new grade is assinged to a student for a course, a trigger can detect this event, and automatically calculate and update the total earned credit of the student in another table.

More examples with SQL query can be found in chapter 5