**Task 1. [50 points] Answer each of the following questions briefly. The questions are based on the following relational schema:**

Emp(eid: integer, ename: string, age: integer, salary: real)

Works(eid: integer, did: integer, pcttime: integer)

Dept(did: integer, dname: string, budget: real, managerid: integer)

1. Give an example of a foreign key constraint that involves the Emp relation. What are the options for enforcing this constraint when a user attempts to delete an Emp tuple?

*A foreign key constraint could be that the employees ID is used in the “works” relation. This means to delete an Emp tuple would mean an employee who does not exist could be in the Works table, which would be an error. The options would be to not allow an Emp tuple to be deleted if it is used elsewhere, or to delete an Emp tuple everywhere if it is deleted in the Emp table.*

1. Define the Works relation in SQL so that an employee can work only in one department

*CREATE TABLE (Employee) (*

*Eid INTEGER*

*Ename CHAR(20)*

*Age INTEGER*

*Salary REAL*

*PRIMARY KEY (eid)*

*FOREIGN KEY (did) REFERENCES Dept*

1. Define the Dept relation in SQL so that every department is guaranteed to have a manager.

*CREATE TABLE (Dept)*

*Did INTEGER*

*Dname CHAR(20)*

*Budget REAL*

*Managerid INTEGER*

*PRIMARY KEY (managerid)*

1. Write an SQL statement to add John Smith as an employee with eid = 301, age = 20 and salary = 90, 000.

*INSERT INTO employee (eid, age, salary)*

*VALUES (301, 20, 90,000)*

1. Write an SQL statement to give every department a 20 percent raise of budget.

*UPDATE dept*

*SET budget = budget + (budget \* 0.20)*

**Task 2 [30 points] Given an ER diagram (as shown below), write SQL statements to create the corresponding relations and capture as many of the constraints as possible. If you cannot capture some constraints, explain why.**

*CREATE TABLE (professor)*

*Pid INTEGER*

*Title CHAR(20)*

*Salary INTEGER*

*Phone INTEGER*

*Building INTEGER*

*PRIMARY KEY (pid)*

*FOREIGN KEY (did) REFERENCES department*

*CREATE TABLE (department)*

*Did INTEGER*

*Dname CHAR(20)*

*Budget INTEGER*

*PRIMARY KEY did*

*FOREIGN KEY pid REFERENCES professor*

*CREATE TABLE (child)*

*Name CHAR(20)*

*Age INTEGER*

*Pid INTEGER*

*PRIMARY KEY (pname, pid)*

*FOREIGN KEY (pid) REFERENCES professor, ON DELETE CASCADE*

**Task 3 [20 points]. Suppose that we have a ternary relationship R between entity sets A, B, and C such that (i) A has a key constraint; and (ii) B has a key constraint and total participation. These are the only constraints. A has attributes a1 and a2, with a1 being the key; B has attributes b1 and b2, with b1 being the key; and C has attributes c1 and c2, with c1 being the key. R has no descriptive attributes. All of the above-mentioned attributes are to be stored as Strings of 10 characters. Write SQL statements that create tables corresponding to this information so as to capture as many of the constraints as possible. If you cannot capture some constraint, explain why.**

*CREATE TABLE (A)*

*A1 CHAR(10)*

*A2 CHAR(10)*

*PRIMARY KEY A1*

*CREATE TABLE (B)*

*B1 CHAR(10)*

*B2 CHAR(10)*

*PRIMARY KEY B1*

*CREATE TABLE (C)*

*C1 CHAR(10)*

*C2 CHAR(10)*

*PRIMARY KEY C1*

There are a few constraints I was unable to capture. I was unable to capture B’s total participation, because I do not know what table it is relying on, so I could not list a foreign key, or the attributes it would draw from its total participation relationship. I was also unable to demonstrate the connection between these three tables, as it was not directly stated.