Final Assignment #11

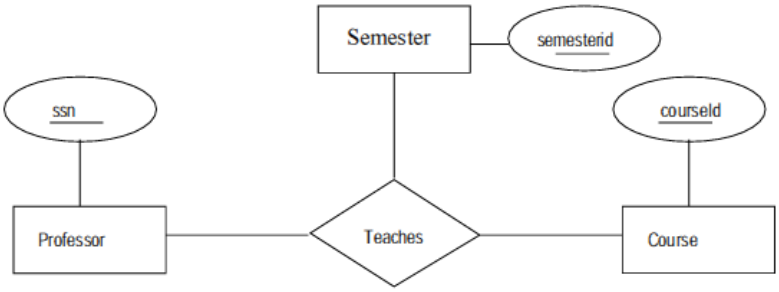
Instructor: Richard Frank (rfrank@sfu.ca)
TA: Ankit Gupta (aga53@sfu.ca)

Total Marks: 50 (5% of the Individual Assignments)
Due Date: Dec 2, 14:30 (paper and/or via the Online Submission Server)

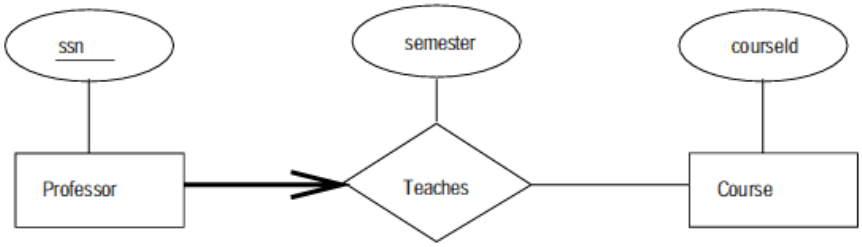
Question 1) Please complete Exercise 2.2 from the book. It has 6 sub-questions [3 marks each]. The question is also below:

> *Exercise 2.2* A university database contains information about professors (identified by social security number, or SSN) and courses (identified by courseid). Professors teach courses; each of the following situations concerns the Teaches relationship set. For each situation, draw an ER diagram that describes it (assuming no further constraints hold).

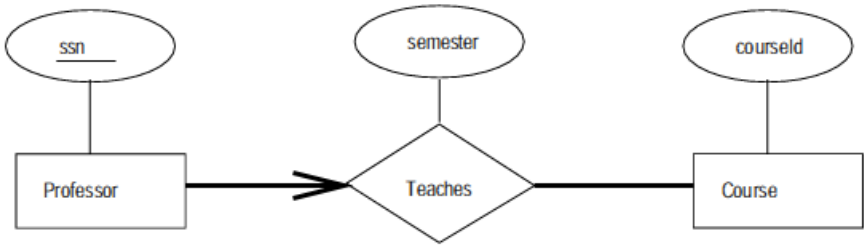
1. Professors can teach the same course in several semesters, and each offering must be recorded.



2. Every professor teaches exactly one course (no more, no less).



3. Every professor teaches exactly one course (no more, no less), and every course must be taught by some professor.



Question 2) Create the SQL statements to create the tables required for Question 1.2 above ("*Every professor teaches exactly one course (no more, no less)*". [5 marks]

```
CREATE TABLE Professorteaches ( ssn CHAR(10),
        courseId INTEGER,
```

semester CHAR(10),
PRIMARY KEY (ssn),
FOREIGN KEY (courseId)
REFERENCES Course)

CREATE TABLE Course (courseId INTEGER,
PRIMARY KEY (courseId))

Observe that we do not need a separate table for Professor.

Question 3) Assume your database is made up of the Customer, CustomerAddress, Address tables from AdventureWorks. These 3 tables are represented below as C, CA and A, respectively. Given the Given the Relational Algebra (RA):

$$\pi_{FirstName,AddressType,Address.*}\sigma_{A.City='Burnaby'AND\ LastName='Doe'}(C \rhd\lhd CA \rhd\lhd A)$$

a) What does it mean in English? [1 mark]

Select all customers with last name Doe who have an address in Burnaby. Their address could be their home or work. Display the resulting first-name, address type and all their address information.

b) Write the corresponding SQL code [2 marks].

```
SELECT C.Firstname, CA.AddressType, A.*
FROM C, CA, A
WHERE C.CustomerID = CA.CustomerID AND CA.AddressID = A.AddressID
      AND A.City = 'Burnaby' AND C.LastName = 'Doe'
```

c) Write the RA that is more efficient (no need to include calculations)? [3 marks]

$$\pi_{FirstName,AddressType,Address.*}(\pi_{FirstName,CustomerID}\sigma_{LastName='Doe'}C \rhd\lhd CA \rhd \lhd \sigma_{A.City='Burnaby'}A)$$

d) Write the SQL code corresponding to the *efficient* RA? [3 marks]

```
SELECT C.Firstname, CA.AddressType, A.*
FROM
      (Select FirstName, CustomerID FROM C WHERE LastName ='Doe') AS C,
      CA,
      (SELECT * FROM A WHERE City ='Burnaby') AS A
WHERE C.CustomerID = CA.CustomerID AND CA.AddressID = A.AddressID
```

Question 4) Consider a relation R with five attributes ABCDE. You are given the following dependencies: A → B, BC → E, and ED → A.

1. List all keys for R. [3 marks]

CDE, ACD, BCD

2. Is R in 3NF? [2 marks]

   R is in 3NF because B, E and A are all parts of keys.

3. Is R in BCNF? [2 marks]

   R is not in BCNF because none of A, BC and ED contain a key.

Question 5) Decompose R1(A,C,B,D,E) with functional dependency of A → B and C → D into BCNF. [2 marks]

BCNF decomposition: AB, CD, ACE.

Question 6) Suppose you are given a relation R(A,B,C,D). The dependencies B → C and D → A hold.

a) What is the candidate key(s) for R? [2 marks]

   Candidate key(s): BD.

b) Is decomposing them into BC and AD a good decomposition? Why? [4 marks]

   The decomposition into BC and AD is unsatisfactory because it is lossy (the join of BC and AD is the cartesian product which could be much bigger than ABCD)

Question 7) Suppose you are given a relation R(A,B,C,D). The dependencies A → B, B → C and C → D hold.

a) What is the candidate key(s) for R? [2 marks]

   Candidate key(s): A.

a) Is decomposing them into AB, AD and CD a good decomposition? Why? [4 marks]

   This is a lossless BCNF decomposition however, not dependency preserving (B → C cannot be enforced). This is not the best decomposition (the decomposition AB, BC, CD is better.)

Question 7) Assume that you are given a relation with attributes ABCD and that no record has NULL values. Write an SQL query that checks whether the functional dependency A → B holds. HINT: A → B means that for each value of A, there is a single value of B (the converse is not true: for a value of B, there could be multiple values of A). [6 marks]

```
SELECT COUNT (*)
FROM R AS R1, R AS R2
WHERE (R1.B != R2.B) AND (R1.A = R2.A)
```