CSCD 327 – Take-Home Midterm Exam II

Due: August 11, 2014, 11:59pm
Please submit online via Canvas system.
No late submission will be accepted.
No discussion, no collaboration please!!!

Your name: Dan Hervé

1. (20 points) SQL Queries

Consider the following relational schema (primary keys are underlined):

Product(<u>pid</u>, name, price, mfgr)
Buys(<u>cid, pid</u>)
Customer(<u>cid</u>, cname, age)

a) Consider the following SQL query:

```
SELECT      C.cid, C.cname
FROM        Customer C, Buys B
WHERE       C.cid = B.cid
GROUP BY    C.cid
HAVING      count(pid) > 100
```

Rewrite the above SQL query **without** using the "HAVING" clause so that the resulting query still produces the same query result.

```
SELECT cid, cname
FROM (SELECT count(pid) cnt, cid, cname FROM Customer join Buys using(pid)
        GROUP BY cid) temp
WHERE cnt > 100
GROUP BY cid
```

b) Write the following query **in SQL**: Find the *cids* of customers who buy **only** the products made by manufactory "D" (i.e., mfgr = 'D').

```
SELECT cid
FROM Product join Buys using(pid)
WHERE pid in (SELECT pid FROM Products WHERE mfgr = 'D') and pid not in
(SELECT pid FROM Products WHERE mfgr != 'D')
GROUP BY cid
```

c) Write the following query **in SQL**: "Find the *cids* and *cnames* of all customers who have purchased the **second most expensive** product." You can assume that no two products have the same price. Please don't use "ORDER BY … LIMIT …" in your answer.

SELECT cid, cname
FROM (Customer join Buys using(cid)) join Products using(pid)
WHERE price = (SELECT max(price) FROM Products WHERE salary < (SELECT max(price) FROM Products))

Consider the following relational schema, where the primary keys are underlined,

Students(<u>sid</u>, sname, address, department_id)
Courses(<u>cid</u>, cname, year, semester, department_id)
Departments(<u>department_id</u>, dname, college)
Grades(<u>sid, cid</u>, grade)

d) Write the following queries in **SQL**: Give all students who have selected Database course (i.e., *cname* = 'Database') in Fall 2012 a 10 percent grade raise.

UPDATE Grades
SET grade = grade * 1.1
WHERE exists (SELECT cid FROM Courses WHERE year = 2012 and semester = 'Fall' and cname = 'Database')

2. (5 points) Stored Procedure

Consider the following stored procedure defined in MySQL.

```
DELIMITER $$
CREATE PROCEDURE MyProcedure(IN String VARCHAR(80), OUT newString VARCHAR(80), OUT Length
INTEGER)
BEGIN
SET newString = UPPER(String);
SET Length = LENGTH(newString);
SELECT String, newString, Length;
END$$
DELIMITER ;
```

What are the outputs after executing the following statements?

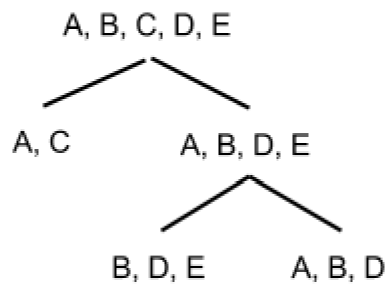
```
CALL MyProcedure('My Stored Procedure', @newString, @Length);
SELECT @newString, @Length;
```

@newString	@Length
MY STORED PROCEDURE	19

3. (13 points) Design Theory

(1) Given a relation schema R = (A, B, C, D, E), and a set of functional dependencies F = {A → C, BD → E, AD → E} held on R. **AB** is a candidate key.

a. (4 points) Is R in BCNF? If not, decompose it into smaller relations such that these smaller relations are in BCNF.



b. (4 points) Is the decomposition a lossless-join decomposition? Is the decomposition dependency preserving? **Justify your answer**.

The decomposition is a lossless-join decomposition. This can be demonstrated with the following formula: {B, D, E} ∩ {A, B, D} = {B, D}, $\{B, D\}^+$ = {B, D, E}, so {B, D} is a superkey of {B, D, E} and {A, C} ∩ {A, B, D, E} = {A}, $\{A\}^+$ = {A, C}, so it is a superkey of {A, C}

It is not dependency preserving since {AD → E} is not preserved in the final relations.

(2) For relation R(A, B, C, D, E) and given set of functional dependencies F={AB→C, C→D, D→B, D→E}

a. (3 points) List all the candidate keys of relation R.

$\{A\}^+$ = {A} is not a candidate key
$\{A, B\}^+$ = {A, B, C, D, E} is a candidate key
$\{A, C\}^+$ = {A, C, D, B, E} is a candidate key
$\{A, D\}^+$ = {A, D, B, C, E} is a candidate key
$\{A, E\}^+$ = {A, E} is not a candidate key

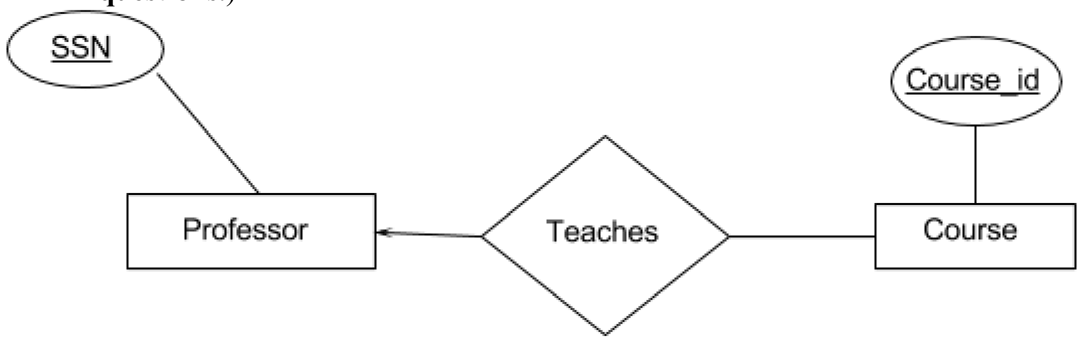
b. (2 points) Indicate all the 3NF violations.

{C}, the α of {C→D}, is not a superkey
{D}, the α of {D→B} and {D→E}, is not a superkey
{E}, the β of {D→E}, is not contained in a candidate key

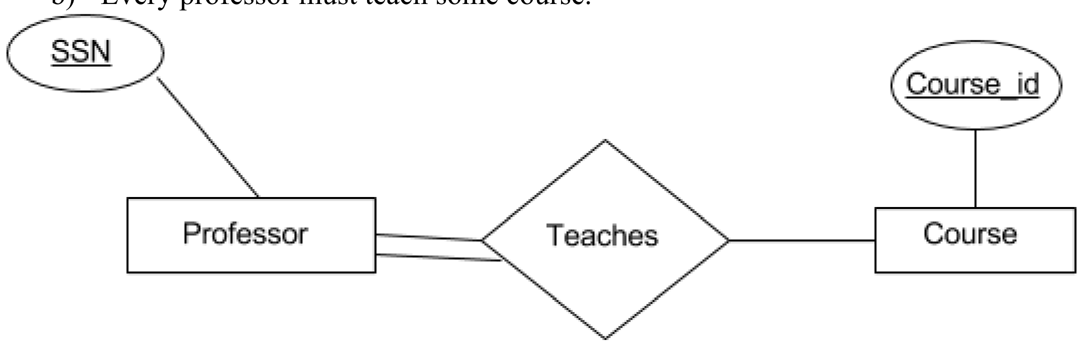
4. (12 points) ERD

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

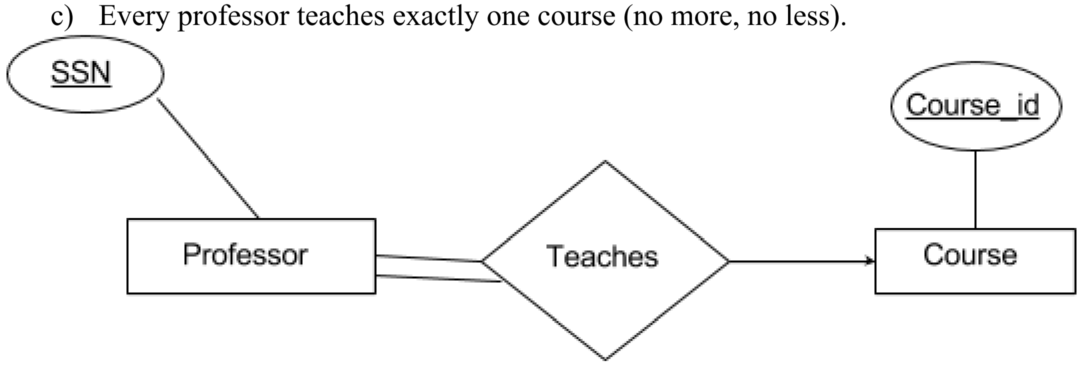
a) Professors can teach the same course in several semesters, and only the most recent such offering needs to be recorded. (**Assume this condition applies in all subsequent questions.**)



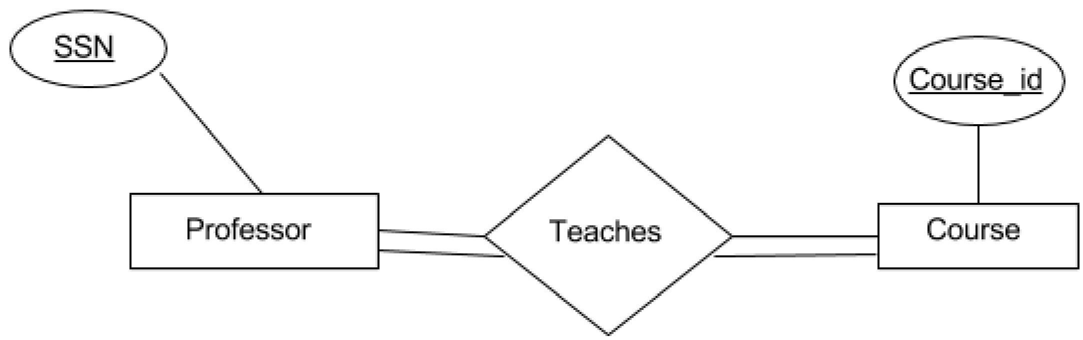
b) Every professor must teach some course.



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



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


SSN
Course_id
Professor
Teaches
Course