CIS 560, Fall 2013 **Name:_______________________**

Exam 1 – 50 minutes

This test consists of questions in four categories. The number of points for each question is shown below.

- Read all questions carefully before starting to answer them.
- Write all your answers on the space provided in the exam paper.
- The order of the questions is arbitrary, so the difficulty may vary from question to question. Don’t get stuck by insisting on doing them in order.
- Show your work. Correct answers without justification will not receive full credit. However, also be concise. Excessively verbose answers may be penalized.
- Clearly state any simplifying assumptions you may make when answering a questions.
- **Be sure to write your name on the test paper.**

Question	I	II	III	IV	total
Points	20	35	25	20	100
Your points					

I. True/False Questions [10 questions: you get 2 points for each correct answer; you lose 1point for each incorrect answer; you get 0 points for questions that you don't answer]:

True [✓ **False**]
If entity sets A and B have a many-to-many relationship, each entity in A must be related to more than one entity in B.

[✓ **True**] **False**
When translating an E/R diagram with subclass entity sets into a relational schema, the "Object oriented (OO) approach" will produce at least as many tables as the "E/R approach".

[✓ **True**] **False**
A table in BCNF must be in 3NF.

True [✓ **False**]
If we define a foreign key in relation R, the DBMS checks the foreign-key constraint whenever a tuple in R is deleted.

True [✓ **False**]
Given the snapshot of a table as below, we are *certain* that AB → D is a functional dependency of the table.

A	B	C	D
1	3	2	2
2	3	2	4
3	1	3	6
3	1	1	6

[✓ **True**] **False**
Under the 3-valued logic, TRUE AND (TRUE OR NOT(UNKNOWN)) = TRUE.

[✓ **True**] **False**
If A→B and AB→C, then A→C.

[✓ **True**] **False**
Every ISA relation in the E/R model is one-one.

True [✓ **False**]
A weak entity set does not have any attribute that belongs to its key.

True [✓ **False**]
There always exists a *unique* BCNF decomposition for any given table.

II. SQL Queries [35 points]

Consider a database with the following schema, consisting of three tables. The Employee table stores information about employees. Every employee is identified by an employeeID. The Department table stores information about departments. The Vacations table stores information about the total number of days each employee has taken for vacation (employees who have not taken any vacation days will not appear in the Vacations table).

Employee(EmployeeID, FirstName, LastName, Office, Email, DepartmentID)
Department(DepartmentID, DepartmentName)
Vacations(EmployeeID, Days)

Answer the following queries using SQL.

a) [10 points] List the number of vacation Days taken by each employee with the name "John Smith".

```
SELECT Days
FROM Employee, Vacations
WHERE Employee.EmployeeID=Vacations.EmployeeID AND Employee.FirstName=‘John’
AND Employee.LastName=’Smith’ ;
```

If we want to also account for employees with name "John Smith" who don't have any vacation days, we can use Left outer join.

b) [10 points] List the FirstName and LastName of all employees who never took a vacation day.

```
SELECT FirstName, LastName
FROM Employee
WHERE EmployeeID NOT IN(SELECT DISTINCT EmployeeID FROM Vacations)
```

c) [15 points] List the DepartmentName of every department whose total number of vacation days taken by its employees is the largest among those of all the departments.

Many possible solutions (some of the solutions below are from students' exams):

```
SELECT DepartmentName
FROM Department, Employee, Vacations
WHERE Department.DepartmentID=Employee.DepartmentID AND
Employee.EmployeeID=Vacations.EmployeeID
GROUP BY DepartmentName
HAVING SUM(Days) = (SELECT max(t1.DAYS) FROM (SELECT Employee.DepartmentID,
SUM(Days) as DAYS FROM Department, Employee, Vacations
WHERE Department.DepartmentID=Employee.DepartmentID AND
Employee.EmployeeID=Vacations.EmployeeID
GROUP BY Employee.DepartmentID) as t1)
```

```
SELECT DepartmentName
FROM Department, Employee, Vacations
WHERE Department.DepartmentID=Employee.DepartmentID AND
Employee.EmployeeID=Vacations.EmployeeID
GROUP BY DepartmentName
HAVING SUM(Days) >= ALL (
SELECT SUM(Days) as DAYS FROM Department, Employee, Vacations
WHERE Department.DepartmentID=Employee.DepartmentID AND
Employee.EmployeeID=Vacations.EmployeeID
GROUP BY Employee.DepartmentID)
```

```
SELECT DepartmentName
FROM Department
WHERE (SELECT sum(Days) from Employee, Vacations
              WHERE Department.DepartmentID = Employee.DepartmentID and
              Employee.EmployeeID=Vacations.EmployeeID
             GROUP BY Employee. DepartmentID) >= ALL
(SELECT sum(Days) from Employee, Department, Vacations
              WHERE Department.DepartmentID = Employee.DepartmentID and
              Employee.EmployeeID=Vacations.EmployeeID
             GROUP BY Department.DepartmentID)
```

```
SELECT d1.DepartmentName
from (select d.DepartmentName, sum(v.Days) as days
from Department d, Employee e, Vacations v
where d.DepartmentID = e.DepartmentID and
e.EmployeeID=v.EmployeeID
group by d.DepartmentName) as d1
where d1.days >= ALL (
select sum(v.Days)
from Department d, Employee e, Vacations v
where d.DepartmentID = e.DepartmentID and
              e.EmployeeID=v.EmployeeID
group by d.DepartmentName)
```

III. E/R Diagram, Relational Schema, Basic SQL Statements [25 points]

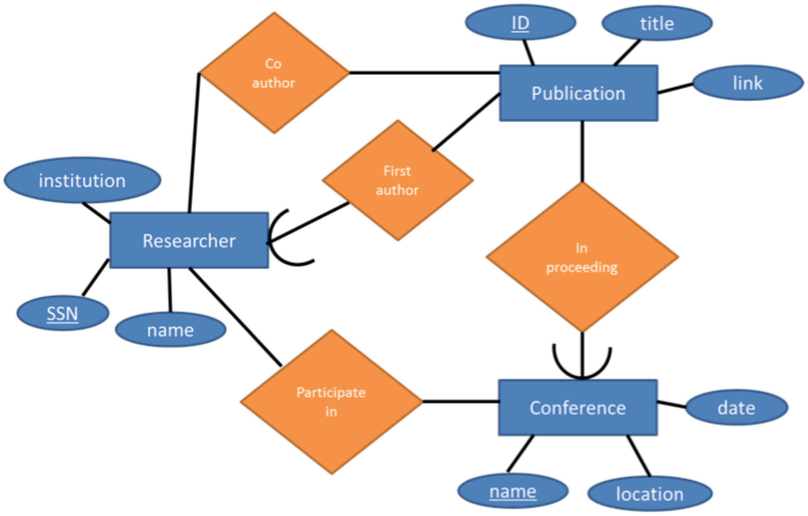
a) [10 points] You have been hired by a digital library to design a database for conference proceedings. The conference proceedings domain is described as follows:

1. Researchers have a unique SSN, as well as a name and an institution.
2. Conferences have a unique name, as well as a location and a date.
3. Publications have a unique id, as well as a title and a link to the digital article.
4. Each publication can have many co-authors (all of whom must be researchers), but it has exactly one “first author”.
5. Each publication is in the “proceedings” of exactly one conference.

6. Researchers can participate in any number of conferences.
7. A conference can have many publications and can draw many researchers to participate in it.

Your task is to design an E/R diagram for this domain. Your E/R diagram should include entity sets, attributes and relationships. If you feel that you must make any additional assumptions, please state them clearly in your solution. Remember to indicate the key for each entity set, as well as the multiplicity of each relationship using the appropriate notation.

Solution



b) [10 points] Translate your E/R diagram to a relational schema. Minimize the number of relations your solution has; merge relations where appropriate. Specify the keys and foreign keys of each relation in your schema. In translating a subclass hierarchy, use the E/R style translation.

If we do not merge relations at all, we get:

Researcher (<u>SSN</u>, name, institution)
Publication (<u>ID</u>, title, link)
Conference (<u>name</u>, location, date)
In_proceeding (<u>publicationID</u>, <u>conferenceName</u>)
Co-author (<u>researcherSSN</u>, <u>publicationID</u>)
First_author (<u>researcherSSN</u>, <u>publicationID</u>)
Participate_in (<u>researcherSSN</u>, <u>conferenceName</u>)

But, we should combine the In_proceeding relation and the First-author relation with the Publication relation to obtain the following schema:

Researcher (<u>SSN</u>, name, institution)
Publication (<u>ID</u>, title, link, in-proceeding-name, first-authorSSN)

Conference (name, location, date)
Co-author (researcherSSN, publicationID)
Participate_in (researcherSSN, conferenceName)

c) [5 points] Choose ONE relation from your relational schema and write the SQL statement to create the corresponding table. (You may make any reasonable choice of data types. Remember to include any constraints that follow from the description of the schema or your E/R diagram, including primary key and foreign key constraints.)

Answers depend on the relation you chose. An example is provided below:

```
CREATE TABLE Researcher (
SSN CHAR(9) PRIMARY KEY,
name VARCHAR(50) NOT NULL,
institution VARCHAR(50) NOT NULL);
```

IV. Functional Dependencies, Keys and Normalization [20 points]

a) [5 points] Consider a relation R(A,B,C,D,E,F) with functional dependencies:

CDE→B, ACD→F, BEF→C, B→D

Determine one key of the relation R. Do not list superkeys that are not (minimal) keys.

Possible keys: ABEF, ACDE, ABCE

BEF->C
B->D
ACD->F
ABEF+ = ABCDEF

CDE->B
ACD->F
ACDE+ = ABCDEF

B->D
ACD->F
ABCE+ = ABCDEF

b) [10 points] Consider a relation R(A,B,C,D), with the following functional dependencies:

A→B, C→D, AD→C, BC→A

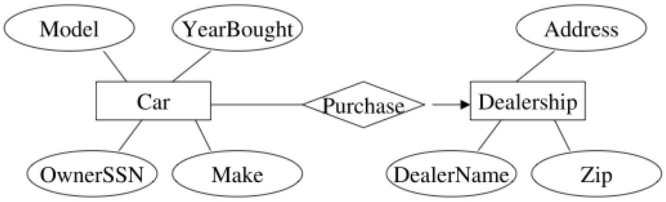
Decompose the relation R into a collection of Boyce-Codd Normal Form (BCNF) relations. Show your intermediate steps; for each step show the relation that you are decomposing and the violation of BCNF that you are using during that decomposition step. Indicate clearly your final result: the relations, their attributes and their keys. Does your decomposition preserve functional dependencies?

There is not a unique BNCF decomposition; any decomposition will get full credit.

We can see that A^+={A,B}, which is different from the set of all attributes {A,B,C,D}. That means we can decompose R(A,B,C,D) into two relations R_1(A,B) and R_2(A,C,D). The first relation R_1 is in BCNF. However, the second relation R_2 is not in BCNF, as C^+={C,D}≠{A,C,D}. So, we need to decompose R_2 into R_{21}(A,C) and R_{22}(C,D). Both R_{21} and R_{22} are in BCNF.

The key for R_1 is A. The key for R_{21} is A,C and the key for R_{22} is C.
The decomposition does not preserve functional dependencies.

c) [5 points] Consider the following E/R diagram:



For each of the following statements, write a functional dependency that best captures the statement.

(FD1) No two makes have cars with the same model name.

Model -> Make

(FD2) No owner has two cars of the same model.
OwnerSSN, Model -> Make, YearBought

(FD3) No two dealerships have the same name.

DealerName -> Address, Zip