Name _____

B561 Advanced Database Concepts
Midterm Exam - Fall 2011

Read the honor pledge and grading policy below, and sign your name in the space provided

I have neither given nor received aid on this exam, nor have I concealed any violations of the Honor Code. I have read and understood the requirements in the grading policy.

Signature : _____

Printed name: _____

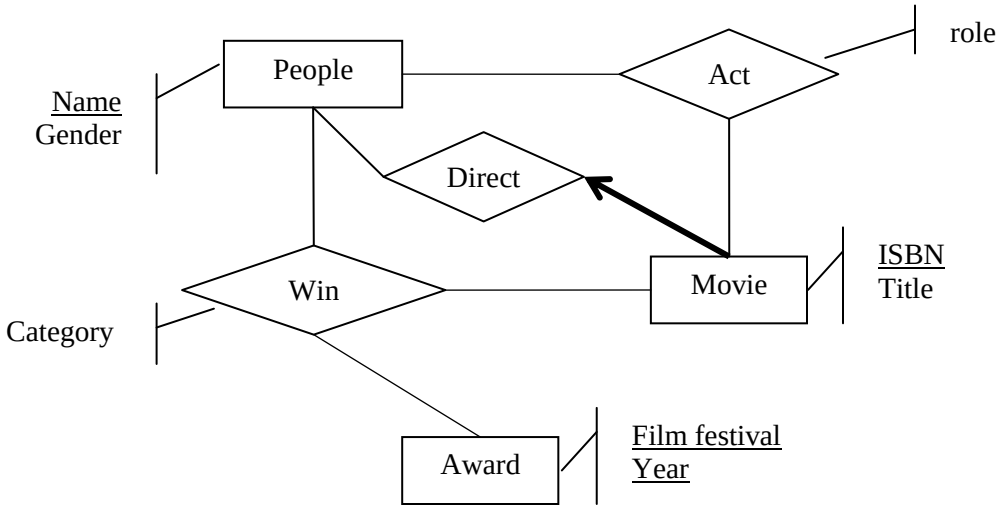
Grading Policy:

- The students have 75 minutes to finish the exam.
- The full score of this exam is 200 points (contributes 20% to your final score). For your reference, the point value of each question is provided.
- If you need more space for a specific question, please as the instructor/AI for white paper – the question # should be clearly marked. Answers written on the back of the exam paper will **NOT** be graded.
- Students are required to write their name on **EVERY** page of the exam paper. Pages without name bear the risk of getting lost.

Part1		Out of 60 points
Part2		Out of 40 points
Part2 bonus		Out of 20 bonus points
Part3		Out of 100 points
Total		Out of 200 points + 20 bonus points

Name _____

Part 1 (60 points)
Consider an ER diagram as shown below:



Question 1.1 (30 points) Consider the following scenarios:

(1) Tom Hanks won Oscar twice in 1993 and 1994.
(2) Documentary movie, such as "the Cove", involves no actors.
(3) Clint Eastwood win Oscar in 2004 as both director and producer for "Million Dollar Baby"
(4) Lindsay Lohan played Hallie Parker and Annie James in "Parent Trap".

Please identify which scenario can be expressed by the ER model above, and which not. Explain why.

(1) The primary key for the "win" relationship is (Name, ISBN, Film Festival, Year). Thus Tom Hanks winning 2 Oscars can be represented in 2 different instances with different primary keys.

(2) There is no participation constraint for people in "act" relationship. Thus it is OK that we have a movie with no actor.

(3) Category is not an entity but an descriptive attribute of the "win" relationship, thus actor, a movie and an award (identified by film festival and year) uniquely identify each instance of the "win" relationship. Thus, the fact that "Clint Eastwood win Oscar in 2004 as both director and producer for 'Million Dollar Baby'" can not be represented using the given ER model.

(4) this is a situlation similar to that explained in (3). Role is only an attribute and therefore the primary key for "act" relationship is (Name, ISBN). Thus there is at most one role a person can play in the movie.

Name _____

Question 1.2 (30 points) Consider the ER diagram with your annotation, please transform it into relational schema and discuss the rationale behind the optimization ideas in your design.

People (PID, Name, Gender)
UNIQUE: name

Movie (ISBN, Title, DirectorID)
FKs: DirectorID references People (PID)

Award (AID, Film Festival, Year)
UNIQUE: (filmFestival, year)

Act (ActorID, ISBN, Role)
FKs: ActorID references People (PID)
MID references Movie (MID)

Win (PID, SIBN, AID, Category)
FKs: PID references People (PID)
ISBN references Movie (ISBN)
AID references Award (AID)

Due to the key constraint between movie entity and direct relationship, we put the director into the "movie" table, instead of creating a separate table for the "direct" relation.

The introduction of artificial key to the movie table is not necessary, as ISBN itself is indeed an artificial key introduced by the industry.

The introduction of the AID in award table is an optimization, as it simplified the foreign key in the Win table.

The introductions of artificial keys PID is optional. It only serves the purpose of shortening the original primary keys and to simplify comparison operation as it is used as foreign key in the Movie, Act and Win tables.

Name _____

Part 2 (40 points) Consider a relational schema R = <U, *F*>, where U= (A, B, C, D, E), and *F* = {A→BC, B→D, E→A, AD→E}. (40 points + 20 bonus points)

Question 2.1 Please compute F_c**.** (20 points)

From A→BC we have A→B and A→C.
Since we have A→B and B→D, we have A→D. From A→D and AD→E we can now have A→E.
So one set of possible minimal cover is {A→B, A→C, B→D, E→A, A→E}

Question 2.2 Please determine whether this relation satisfies the 3rd normal form. Explain why (20 points)

This is not in 3NF. From the minimal cover (or from the original set of FDs), we know that A is a candidate key and E is also a candidate key, while B, C and D are not keys. In the minimal cover we have B→D, but B is not a super key while D is not a key attribute. So this violates 3NF.

Question 2.3 If your answer to Question2.2 is yes, please determine whether this relation satisfies BCNF and why; if your answer to Question 2.2 is no, please decompose the relation into relations that satisfies 3NF. (20 bonus points)

We can decompose the relation into R1 = (A, B, C, E), R2 = (B, D).

Note that all the FDs in the minimal cover are already preserved, and we have a lossless-join. Thus R1 and R2 is a decomposition of R.

Name _____

Part 3 (100 points)
Given the following relational schema:

Customers (<u>cid,</u> name, gender, city)
Accounts (<u>aid,</u> type, balance)
Type is of string data type, and can taken only values from the set {checking, saving}
Own (<u>cid, aid</u>)
aid is a foreign key, refers to Accounts (aid)
cid is a foreign key, refers to Customers (cid)
Transactions (<u>tid,</u> aid, amount)
aid is a foreign key, refers to Accounts (aid)
Value in "amount" can be positive (deposit) or negative (withdraw)

Please write queries in the query languages specified to satisfy the following query requests. (10 points per query)

Question 3.1 Find the balance of the accounts owned by male customer from Bloomington, with balance over $5000. (30 points)

Relational Algebra:

$\pi_{balance,aid}((\sigma_{balance>5000}\text{ Account}) \bowtie \text{Owner} \bowtie (\sigma_{gender=\text{"male"} \wedge city=\text{"Bloomington"}}\text{ Customer}))$

TRC:

$\{P|\exists A \in \text{Account}\ (A.balance>5000 \wedge P.balance = A.balance \wedge P.aid = A.aid \wedge \exists O \in \text{Owner}\ (A.aid = O.aid \wedge \exists C \in \text{Customer}\ (C.cid = O.cid \wedge C.gender=\text{"male"} \wedge C.city = \text{"Bloomington"})))$

SQL:

```
SELECT A.balance, A.aid
FROM Customer C, Account A, Owner O
WHERE C.cid = O.cid AND O.aid = A.aid
AND C.gender = "male" AND C.city = "Bloomington" AND A.balance > 5000
```

Name _____

Question 3.2 Find all the transactions that made deposits to accounts jointly owned by one male and one female. (10 points)

SQL:

```
SELECT T.*
FROM Transaction T, Owner O
WHERE T.aid = O.aid AND T.amount > 0
AND (SELECT COUNT(*)
        FROM Customer C
        WHERE C.cid = O.cid AND C.gender = "male") = 1
AND (SELECT COUNT(*)
        FROM Customer C
        WHERE C.cid = O.cid AND C.gender = "female") = 1
```

Question 3.3 Find the customers who own more than three accounts (including joint accounts), compute the total amount of money in the accounts owned by each such customer. (10 points)

SQL:

```
SELECT C.cid, SUM(A.balance)
FROM Customer C, Account A, Owner O
WHERE C.cid = O.cid AND O.aid = A.aid
GROUP BY C.cid
HAVING COUNT(A.aid) > 3
```

Question 3.4 Find the accounts solely owned by a female, and list all transactions on these accounts, if any. (10 points)

SQL:

```
SELECT T.*
FROM Transaction T, Owner O, Customer C
WHERE T.aid = O.aid AND O.cid = C.cid AND C.gender = "female"
AND NOT EXISTS (SELECT *
                FROM Customer C1
                WHERE C1.cid <> C.cid
                AND C1.cid = O.cid AND O.aid = T.aid)
```

Name _____

Question 3.5 Find the cities in which more males than females own a bank account (20 points)

SQL:

```
SELECT DISTINCT C.city
FROM Customer C, Owner O
WHERE C.cid = O.cid
AND (SELECT COUNT (DISTINCT C1.cid)
        FROM Customer C1, Owner O1
        WHERE C1.city = C.city AND C1.cid = O1.cid AND C1.gender = "male") >
        (SELECT COUNT (DISTINCT C2.cid)
        FROM Customer C2, Owner O2
        WHERE C2.city = C.city AND C2.cid = O2.cid AND C2.gender = "female")
```

Question 3.6 Find the average balance in shared accounts. (20 points)

SQL:

```
SELECT AVG(A.balance)
FROM Account A
WHERE A.aid IN (SELECT O.aid
                FROM Owner O
                GROUP BY O.aid
                HAVING (COUNT (DISTINCT O.cid)) > 1)
```