Name:_____ **Student ID:**_____

MIDTERM EXAM, March 10, 2005

CS 411 Introduction to Database Management Systems
Department of Computer Science
University of Illinois at Urbana-Champaign

Exam Rules:

1) Close book and notes, 75 minutes

2) Please write down your name and student ID number NOW.

3) Please wait until being told to start reading and working on the exam.

4) If you think a problem is ambiguous, write down your assumptions, argue that they are reasonable, then work on the problem using those assumptions.

Scores:

Problem 1:	out of	25 points
Problem 2:	out of	10
Problem 3:	out of	10
Problem 4:	out of	35
Problem 5:	out of	10
Problem 6:	out of	10
Total:	out of	100 points

Name: _____

1. (25 points) Suppose you are asked to write a database application that captures the following information:

Students take courses. Each student has a student ID number and a name. Each course has a course ID number, name, and room number. Each student is either an undergraduate or graduate student. For each undergraduate student, we want to record also his/her year, GPA, and (possibly multiple) email addresses. For each graduate student, we want to record his/her advisor name, and also his/her multiple email addresses. In addition, each course is assigned to exactly one graduate student, who acts as the head TA of the course. For this head TA role, we want to record the start and end times (e.g., starts in 01/05 and ends in 12/05).

a) [10 points] Draw an ER diagram for this application. Decide the key attributes and identify those.

b) [10 points] Convert the above ER diagram into a relational schema (using the OO style translation whenever appropriate). Specify the key of each relation in your schema.

c) [5 points] Why should we start modeling an application with ER diagram, why not starting the modeling process with relational schemas?

2. (10 points)

a) [5 points] Use the definition of functional dependencies, prove that if X -> Y and Z is a subset of Y, then X -> Z.

b) [5 points] Use the definition of functional dependencies, prove that any two-attribute relation is in BCNF.

3. (10 points)
Consider relation R(A,B,C,D,E) with the following functional dependencies: AB -> C, D -> E, DE -> B.

Is R in BCNF? If not, decompose R into a collection of BCNF relations. Show each step of the decomposition process.

4. (35 points; 7 points each)
Consider a database schema with the following relations:

Person(ssn, name, phone, city)
Purchase(buyer-ssn, seller-ssn, store, pid)
Product(pid, name, price, category, cid)
Company(cid, name, stock-price, country)

where Purchase.pid and Product.pid refer to product id, and Product.cid
and Company.cid refer to company id.

a) Write a relational algebra (NOT SQL) query that finds the names and stock prices of companies that sold products of category "TV" to people living in Champaign. (Returning duplicate names/stock prices is ok.)

b) Write a SQL QUERY that finds the names, stock prices, and countries of companies that have sold products to people living in Champaign but not to people living in Urbana. List each company only once. Do not use views for this question.

c) Write a SQL query that lists the names of all companies that have sold at least two different products (that is, products with different pids).

d) Write a SQL query that lists the names of all people who (a) are living in Champaign, and (b) buy at least five different products. For each person name, list also the total number of different products that that person has bought.

e) Write a SQL query that lists the names of all people who (a) are living in Champaign, and (b) buy more different products than the average number of products that each person living in Urbana has bought.

5. (10 points)

a) Explain when using a view can be useful.

b) Compare and contrast materialized views and virtual views.

6. (10 points)
Consider again the relational schema that has four relations in Problem 4.

Person(ssn, name, phone, city)
Purchase(buyer-ssn, seller-ssn, store, pid)
Product(pid, name, price, category, cid)
Company(cid, name, stock-price, country)

a) Suppose you want to enforce the constraint that no product price is greater than $6. What would be the best implementation method? Trigger, assertion, or check? Briefly explain your answer.

b) Suppose you want to enforce the constraints that (a) seller-ssn in table Purchase must exist in the ssn column of table Person, and (b) when inserting a tuple into the Purchase table, if seller-ssn is not found, then it must be automatically inserted into the table Person. Would you implement that using trigger, assertion, or something else? Briefly explain your answer.