EECS 484, Fall 2012
Midterm Exam

Please write your uniqname at the top of every page of the exam. This helps us keep exams organized in case the sheets become detached.

This exam is closed book, except for a single 8.5x11 sheet of notes.

This exam consists of 4 multi-part questions. The maximum possible score is 100. You are permitted 80 minutes to finish.

You may not use any electronic devices of any kind during this exam. A calculator will not be necessary. Please check now to ensure that all devices you may have with you are turned off, so as not to disturb others.

Do not start work on this exam until you are instructed to do so.

Honor code pledge: "I have neither given nor received aid on this exam."

Your signature:

Your name (printed):

Your uniqname (printed):

Question 1. ER Models (16 points)

You are called upon to design a database that tracks students and their extracurricular activities. Construct an ER diagram that reflects the following requirements. Be sure your diagram reflects all appropriate keys and constraints.

A student has a name, id, and year of graduation. Every activity is either a sport or a club. A student can be a member of any number of activities. Each activity has a name, a website, and a president (who must be a student). Each activity has exactly one president, and a student cannot be president of more than one activity. A sport has a practice-time, a practice-location and one or more managers (who must be students). A student can be a manager of any number of sports. Each club has an office-location.

Question 2. Relational Queries (40 points)

Answer the following questions based on this relational schema:

Car(carid, *mfrid*, color, year)
Manufacturer(mfrid, companyname, loc)
Dealer(dealerid, loc)
Salesperson(salespersonid, *dealerid*, name, salary)
Sale(saleid, *carid*, *ownerid*, *salespersonid*, price, date)
Owner(ownerid, name, loc, salary)

a) (4 points) Provide a relational algebra query to answer the question: "Find all the manufacturer ids for manufacturers of black cars sold to an owner whose salary is more than 5000".

b) (4 points) Provide a tuple relational calculus query to answer the question in (a).

c) (4 points) Provide a SQL query to answer the question in (a).

(The schema is reprinted here for convenience)

Car(carid, *mfrid*, color, year)

Manufacturer(mfrid, companyname, loc)

Dealer(dealerid, loc)

Salesperson(salespersonid, *dealerid*, name, salary)

Sale(saleid, *carid*, *ownerid*, *salespersonid*, price, date)

Owner(ownerid, name, loc, salary)

d) (4 points) In plain English, describe what the following relational algebra query does:

$\pi_{\text{salespersonid,price}}\ (\sigma_{\text{date = "October 22"}}\ (\sigma_{\text{loc="Ann Arbor"}}\ (\text{Sale} \bowtie (\text{Salesperson} \bowtie \text{Dealer}))))$

e) (5 points) Write a relational algebra query that yields the same result as (d), but is likely to be more efficient.

f) (5 points) Write a tuple relational calculus query to obtain the same result as (d).

g) (6 points) Provide a SQL query to answer the question: “Report the ids of all salespersons whose average selling price of cars made by Ford is more than 20,000 and who work at a dealership in Ann Arbor.”

h) (8 points) Provide a SQL query to answer the question: “Report the names and owner ids of all owners who have purchased a car with the color that is most popular among buyers who have a salary more than 25,000.”

Question 3. Normalization (28 points)

Imagine that you have a schema for a database operated by a busy restaurant. The database is meant to track tables and diners' reservations. It has attributes for (reservation-Id, Date, Table-id, diner-Name, diner-Phone-number, number-in-Group, Vegetarian-or-not). Shorten these attributes to (IDTNPGV). For relation R1(IDTNPGV), we have FDs I=>DTPG, P=>NV, DT=>I.

a) (6 points) What are the keys for R1?

b) (8 points) Is relation R1 in BCNF? Explain why or why not. If not, provide a decomposition that places the relation in BCNF.

c) (6 points) Imagine we have the following decomposition: {(IDTNG), (NPV)}. Is it a lossless join decomposition? Explain why or why not.

d) (8 points) Imagine we learn that the restaurant owner repeatedly runs a query that retrieves the reservation-Ids involved with a given Table-id. Give a decomposition that would enable this query to run as efficiently as possible (and make sure the decomposition satisfies the lossless join property).

Question 4. Grab Bag! (16 points)

a) (5 points) If a decomposition is not dependency-preserving, which relational operator is likely to be executed more frequently?

b) (5 points) What is the difference between UNION and UNION ALL?

c) (6 points) Consider two SQL queries over a relation Salaries(empid, name, salary):

```
SELECT empid, salary FROM Salaries;
SELECT empid, salary FROM Salaries WHERE salary >= 0;
```

Imagine we know that the salary column never contains a negative number. Will these queries always return identical results? If not, why not?