CSE 444 Midterm Exam

July 29, 2009

Name _____

Question 1	/ 24
Question 2	/ 20
Question 3	/ 20
Question 4	/ 18
Question 5	/ 18
Total	/ 100

Question 1. SQL (24 points, 8 each part) The Wrecks 'R Us Auto Repair Shoppe has a database to keep track of cars, repair orders, and repair technicians. The database has the following tables:

CAR(vin, make, model, year) -- car vehicle ID #, manufacturer, model, and year; for example: (17234, Toyota, Prius, 2008) or (38953, Chevy, Nova, 1968)
TECHNICIAN(techid, name, yearhired) -- technician ID number, name, and year hired
CANFIX(techid, make, model) -- table listing the car makes and models that a technician can fix
ORDER(num, vin, techid, year, month, day) -- repair order #, car ID, technician ID, and date

In these relations, vin is a foreign key in ORDER referring to the CAR relation, and techid is a foreign key in the CANFIX and ORDER relations referring to the TECHNICIAN relation. The keys in each relation are underlined.

(a) Write a SQL query that produces a list of the names of all technicians who repaired a Toyota car (i.e., make = 'Toyota') in the year 2004. The list should be sorted by technician name and should not contain duplicates.

(continued next page)

Question 1. (cont) Schemas repeated for convenience:

CAR(vin, make, model, year) -- car vehicle ID #, manufacturer, model, and year
TECHNICIAN(techid, name, yearhired) -- technician ID number, name, and year hired
CANFIX(techid, make, model) -- table listing the car makes and models that a technician can fix
ORDER(num, vin, techid, year, month, day) -- repair order #, car ID, technician ID, and date

(b) Write a SQL query that reports the car make and model that accounts for the most repair orders in the database. If two or more car make/model pairs are tied for the maximum, your query can list any one of them or all of them, at your discretion.

(c) Write a SQL query that lists the names of all technicians who were hired before 2005 and can fix a Ford Mustang (i.e., make is 'Ford' and model is 'Mustang').

Question 2. Conceptual design (20 points) We would like to design a database to keep track of students who participate in intramural sports.

(a) Give an E/R diagram that captures the following entities and relationships:

Entities:

- Student(name, id#)
- Sport (name)
- Team(name, sport)

Relationships:

- Each Team has team members, who are Students, and a single manager, who is also a Student
- Each Sport has one or more Teams
- Each Team belongs to exactly one Sport
- A Student can be a member of any number of teams and can manage any number of teams, including teams that he/she is a member of.

(continued next page)

Question 2 (cont.) (b) Give a relational schema that captures your E/R diagram from part (a). You should give the table and attribute names and clearly indicate which attributes are keys and foreign keys in the various tables. You do **not** need to give SQL CREATE TABLE statements for your tables.

Question 3. BCNF (20 points) As they do every four years, this summer Seattle Opera is presenting Wagner's *Der Ring des Nibelungen*. This 4-opera mini-series has a huge cast, and one of the major donors to the company would like a database to keep track of the characters. One of the summer interns was given the job and produced a table named RING. It lists the characters and information about them, such as their voice part, what they have power over (fire, love, the magic sword, etc.), the name of their residence, and its address.

RING

Character	Voice	Power	Residence	Address
Wotan	Baritone	Light	Valhalla	Rainbow Bridge
Wotan	Baritone	Air	Valhalla	Rainbow Bridge
Erda	Alto	Wisdom	Middle of Universe	Fremont
Erda	Alto	Fate	Middle of Universe	Fremont
Siegfried	Tenor	Sword	Forest	Hurricane Ridge
Brunnhilde	Soprano	Horse	Rock	Enchanted Fire Ring
Freia	Soprano	Love	Valhalla	Rainbow Bridge
Loge	Tenor	Fire	Valhalla	Rainbow Bridge
Hagen	Bass	Drink	Hut	Seward Park
Getrune	Soprano	Drink	House	Wallingford

(a) This table is not in BCNF. Based only on the data given above, identify the functional dependencies that violate BCNF.

(continued next page)

Question 3 (cont.) (b) Using the functional dependencies you identified in part (a), decompose the table into BCNF. Be sure to indicate which attributes are the key(s) of the various relations, and be sure to indicate any foreign key relationships between the tables.

Be sure to **show the steps** in your decomposition – don't just show the final result. You should also clearly show which functional dependencies you are using at each step in your decomposition, and you should underline the keys in each table so that we can follow your work and evaluate it carefully.

Hint: there may be more than one correct solution to the problem.

The schema of the original table is repeated here to get you started:

RING(Character, Voice, Power, Residence, Address)

(c) For 1 point of extra credit, and ***only*** if you have time left at the end of the exam, what is the name of Brunnhilde's horse?

Question 4. Serialization (18 points). For each of the following schedules,

i. Draw the precedence graph for the schedule.

ii. If the schedule is confict-serializable, give one equivalent serial schedule. If the schedule is not conflict-serializable, explain why not.

(a) r1(A) r3(A) w1(A) r2(A) w2(B) w3(B)

(b) r1(A) w1(A) r3(A) w1(B) r2(A) w3(B) r2(B)

Question 5. Two short questions on transactions and logging (18 points)

(a) Two of the choices we have when designing a transaction and buffer manager are whether to use a steal or no-steal policy and whether to use a force or no-force policy. Which combination of these two choices maximizes overall throughput and concurrency in the system? Give a brief explanation for your answer.

(b) In our presentation of the different logging algorithms (undo, redo, and undo-redo), we generally ignored the possibility that a single element (page) in the buffer pool might contain tuples that were being processed concurrently by different transactions. The fact that tuples might share physical pages causes problems for simple undo and redo logging, but not for an undo-redo protocol like ARIES. Briefly, what is (are) the problem(s) that can occur if tuples being processed by different transactions occupy the same page when a simple undo or redo log is used? How does the ARIES protocol solve this (these) problem(s)?