CSE 444 Final Exam

December 10, 2007

Name ________________________________

Question 1	/ 30
Question 2	/ 25
Question 3	/ 15
Question 4	/ 20
Question 5	/ 30
Total	/ 120

Question 1. SQL (30 points) Suppose we have the following schema to describe the inventory of snacks in the ACM lounge.

SNACK (name, price, qtyOnHand, reorderQty, vendor)
VENDOR (name, address, city, phone)

Every snack has a unique name, an integer price, a string with a vendor name, and two quantities: the number currently in stock, and a reorder quantity (the idea is that the when the quantity on hand falls below the reorder quantity we would get more of that item).

(a) Write a SQL query that returns a list of all snacks names and vendors where the qtyOnHand attribute for that snack is less than the reorderQty attribute. The result should be sorted by vendor name and, if a vendor supplies two or more snacks in the resulting list, the output should be further sorted by snack name.

(b) Write a SQL query that returns the names of all vendors located in Seattle and the number of different snack items (i.e., different snack names) supplied by that vendor.

(c) Write a SQL statement or statements to increase the qtyOnHand of the Snack with the name 'Skittles" by 25.

(d) One of your colleagues suggests that it would be helpful to add some nutrition information to the database so people can be better informed about what they are eating. The proposal is to add a "food group" attribute describing the major ingredients of each snack item. But when this is added as an additional simple attribute, the resulting table doesn't look quite right. Among other things, there's a lot of duplicated information:

<u>name</u>	foodGroup	price	qtyOnHand	reorderQty	Vendor
Cheetos	salt	50	10	5	Costco
Cheetos	fat	50	10	5	Costco
M&Ms	sugar	50	22	10	Safeway
M&Ms	chocolate	50	22	10	Safeway
M&Ms	fat	50	22	10	Safeway
Carrots	veggie	35	30	5	Safeway

(continued next page)

(d) (cont.) Describe the “bad” functional dependencies in the above table. Why are they “bad”? (Be reasonably precise here – closures, etc.) What else is wrong with the table, if anything?

(e) Using the functional dependency information from part (d) above, decompose the SNACK relation into relations that are in Boyce-Code Normal Form (BCNF), and that retain all of the information in the original table from part (d).

Question 2. XML (25 points) Suppose we have the following XML DTD defining the structure of XML data containing information about winners of the Nobel prizes.

```
<!DOCTYPE prizelist [
<!ELEMENT prizelist (prize)* >
<!ELEMENT prize (category, year, winners) >
<!ELEMENT category (#PCDATA) >
<!ELEMENT year (#PCDATA) >
<!ELEMENT winners (winner)+ >
<!ELEMENT winner ((name | organization), country?) >
<!ELEMENT name (first?, last) >
<!ELEMENT first (#PDCATA) >
<!ELEMENT last (#PCDATA) >
<!ELEMENT organization (#PCDATA) >
<!ELEMENT country (#PCDATA) >
] >
```

(a) Give a valid XML document using the above DTD that describes the following 2007 Nobel prizes:

Literature: Doris Lessing, Great Britain

Peace (two winners): Intergovernmental Panel on Climate Change, no country; and Al Gore, United States

(additional space is provided on the next page if needed) (additional space for question 2(a), if needed)

(b) Assuming that we have an XML document prize.xml that is a valid instance of the prizelist DTD given above, write XQuery expressions to return the following information as well-formed XML data. If there is more than one reasonable way to represent the result as well-formed XML, you should pick one.

(i) A list of all of the winners of any prize in the year 1990, showing the prize category and the winner's name(s) or organization(s) only.

(ii) A list of prize categories and years for all prizes won in the years 1920-1940 where at least one of the prize winner's country was Germany.

Question 3. Recovery (15 points) Suppose after a system crash we have the following ***redo***-log with non-quiescent checkpointing containing the following data:

<START T1>

<T1 A 17>

<START T3>

<T1 B 42>

<COMMIT T1>

<T3 A 100>

<START T2>

<START CKPT ____ >

<T2 B 100>

<T3 C 10>

<END CKPT>

<COMMIT T3>

<T2 B 20>

(a) Fill in the correct values in the START CKPT record, above. You should assume that this log is complete and that there are no other transactions involved.

(b) Describe the actions performed by the recovery manager when processing this redo log to recover the database to a consistent state.

Question 4. Serializibility (20 points) For each of the following schedules,

(i) Draw the precedence graph for the schedule.

(ii) If the schedule is conflict-serializable, give an equivalent serial schedule; if it is not conflict-serializable, explain why not.

(a) w3(A), r1(A); w1(B); r2(B); w2(C); r3(C)

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

Question 5. Query plans (30 points)

(a) Recall from the midterm exam that we had a question involving the following relations:

> BOOK (<u>isbn</u>, title, publisher, year)
> AUTHOR (<u>ssn</u>, name, city)
> WROTE (<u>isbn</u>, <u>ssn</u>)

and a query that returned the titles and years of all books written by Rowling:

```
SELECT b.title, b.year
FROM book b, author a, wrote w
WHERE b.isbn = w.isbn AND w.ssn = a.ssn AND a.name = 'Rowling';
```

Draw a logical query plan (tree) for this query. You do not need to worry about efficiency; you may choose any plan so long as it is correct.

(b) (A little dynamic programming) Assume that we have four relations W, X, Y, and Z, and we wish to find the lowest cost plan for computing the join of the relations W |x| X |x| Y |x| Z using *left-deep* query trees **only**. Assume that the sizes are as follows:

T(W) = 1500, T(X) = 2000, T(Y) = 1000, T(Z) = 3000

Also assume that the size of a join can be estimated to be 2% of the size of the cross product, i.e., the size of T(A |x| B) = 0.02 * T(A) * T(B).

Fill in the following table with the best plans and the associated costs of the subqueries and final query. Some of the table is filled in for you. Remember that you should only consider left-deep, not right-deep or bushy, join trees.

Subquery	Size	Cost	(Best) Plan
WX		0	WX
WY		0	WY
WZ		0	WZ
XY		0	XY
XZ		0	XZ
YZ		0	YZ
WXY			
WXZ			
WYZ			
XYZ			
WXYZ			