Name:_____

CSEP 544, Spring 2009, Final Examination
Take-home Exam
June 2-4, 2009

Rules

- Open books, open notes, access to the computer.
- No communication/collaborations allowed with your colleagues.
- Questions ? Send email to Dan Suciu and cc' Bhushan Mandhani.
- Posted: Tuesday, June 2nd, 9:30pm.
- Return by: Thursday, June 4th, 11:59pm.
- Dropbox:`https://catalysttools.washington.edu/collectit/dropbox/bhushan/5598`
- What to turn in: text file, or Word file.
- WRITE YOUR NAME !

Question	Max	Grade
1	25	
2	15	
3	20	
4	10	
5	20	
6	10	
Total	100	

Name:_____

1. (**25** points) **Relational Model** A large IT corporation stores their access control policy to their objects in a database with the following schema:

```
User(uid, uname)            /* all the users */
Group(gid, gname)           /* all the groups */
Member(uid, gid)            /* users belong to groups (many-many) */
Object(oid, oname)          /* all the objects */
AccessGranted(gid, oid)     /* access to an object is granted on a per group basis */
AccessDenied(uid, oid)      /* access is denied on a per user basis; overrides AccessGranted */
AccessLog(uid, oid, t)      /* logs whenever a user accesses an objectq; t=timestamp */
```

These policies are enforced in the applications, which have to read the database to determine if a request to an object should be granted or denied, then proceed accordingly. Grant policies are group-based: that is, an entire group is granted access to an object. Deny policies are user based: a particular user may be denied access to an object, even if the user belongs to a group that has access to the object. For auditing purposes, every time an object is accessed by a user, the system logs an entry into the `AccessLog` table.

(a) Describe the SQL schema and the constraints for this data. You have to turn in seven CREATE TABLE statements that contain the primary key and the foreign key constraints. All user ids, group ids, object ids, and the time stamps are integers; all other attributes are `VARCHAR(n)`, where you can choose a suitable value for n.

(b) Write a SQL query that computes the total number of distinct objects accessed by the system, in a sliding window of width 10. That is, your query should report pairs `(t, count)` where `count` represents the number of distinct objects accessed during the timestamps `t...t+9`.

(c) Write a SQL query that checks if any of the accesses recorded in AccessLog were illegal. Your query should return all illegal accesses, i.e. accesses where the user uid was not permitted to access the object `oid`.

(d) A group is called *useless* if all the access grant permissions given through that group are overridden by a access deny entry. That is, the group is useless if for every member U of that group and for every object O to whom that group is being granted access, there is an entry (U,O) in `AccessDenied`: it means that the group is not useful in granting any access at all, since all the accesses are overridden in the `AccessDenied` table. Write a SQL query to find all useless groups.

2. (**15** points) **Functional Dependencies and Database Design**

(a) Consider the following three relations, their attributes, and their keys:

$$R(\underline{A}, B, C)$$
$$S(\underline{D}, E)$$
$$T(\underline{F}, G, H)$$

For each of the views $V_1, \ldots, V_5$ below indicate their attributes, and list a set of functional dependencies (FDs) such that all the FDs that hold in the view can be inferred from these.

$$\begin{aligned} V_1 &= R \bowtie_{B=D} S \\ V_2 &= S \bowtie_{E=G} T \\ V_3 &= \sigma_{H=55}(T) \\ V_4 &= \Pi_{AB}(R) \\ V_5 &= \gamma_{H,sum(G) \text{ as } K}(T) \end{aligned}$$

For example, your answer could look as follows:

$$V_9(A, B, D, F, G) \qquad AB \rightarrow DFG; \;\; F \rightarrow B$$

(b) Consider a table $R(A, B, C, D, E, F)$ and the following functional dependencies:

$$\begin{aligned} BC &\rightarrow A \\ BDE &\rightarrow F \\ EF &\rightarrow B \end{aligned}$$

Compute a BCNF representation of R. Explain your steps.

(c) The MomPopDairy Company has a database with a single table:

```
Orders(customerID, customerName, customerAddress, milkQuantity, date)
```

In their current operation, every time a customer calls to place an order for milk delivery, the seller enters a new record in `Orders` with all the customer information. The company has operated for about ten years, and during this time there have been about 10,000 records inserted in `Orders`. Now, the owners want to create a Web interface to allow customers to place their own orders online: existing customers will provide their customer ID then can place their order, while new

customers will provide their name and address and will be assigned a customer ID that they can use for future orders. The company owners would like to keep all the old data, but would also like to enable the new application. They hire you as a database consultant to advise them on how to manage their database so that it can support the new Web application. You observe quickly that their schema is not in normal form. Advise the MomPopDiary Company on their options. Give them two options on how to design the Web application and/or restructure the database, explaining the immediate costs versus long term benefits tradeoff. Keep your explanations short: for each of the two option, given your answer in 1-3 sentences, then briefly enumerate the pros and cons.

3. (**20** points) **Transactions**

(a) Consider the four schedules below, where s_i means *transaction i starts*, and c_i means *transaction i commits*.

$$s_1, s_2, s_3, w_1(A), w_1(C), r_2(A), w_2(B), c_2, r_3(B), r_3(C), c_1, c_3 \quad (1)$$

$$s_1, s_2, s_3, w_1(A), r_1(C), r_2(A), w_2(B), r_3(B), r_3(C), c_1, c_2, c_3 \quad (2)$$

$$s_1, s_2, s_3, w_1(A), r_2(A), r_3(C), r_3(D), r_2(B), w_3(B), r_2(C), w_1(D), c_1, c_2, c_3 \quad (3)$$

$$s_1, s_2, s_3, w_1(A), w_1(C), c_1, r_2(A), r_3(C), w_2(B), c_2, r_3(B), c_3 \quad (4)$$

For each of the schedule indicate which of the following applies:

i. The schedule is not conflict serializable: in this case indicate a cycle in the serialization graph.

ii. The schedule is conflict serializable but non-recoverable: in this case indicate a violation of the recoverability condition. Your answer should be something like this: "if transaction 4 were to abort instead of committing, then we need to abort transaction 7 because [explain why] and we cannot do this because [explain why]" [you need to replace transactions 7, 4 with transactions in the schedules above].

iii. The schedule is conflict serializable, recoverable, but does not avoid cascading aborts; in this case give an example of a cascading abort. Your answer should be something like this: "if transaction 4 where to abort instead of committing, then we need to abort transaction 7 because [explain why], and as a consequence we also need to abort transaction 5 because [explain why]".

iv. The schedule is conflict serializable and avoids cascading aborts.

(b) Indicate for each of the concurrency control managers below whether it is guaranteed to produce a schedule that is (a) conflict serializable but not necessarily recoverable, (b) conflict serializable and recoverable but does not necessarily avoid cascading aborts, (c) conflict-serializable and avoids cascading aborts

i. 2PL.

ii. Strict 2PL.

iii. Timestamp based concurrency control (with the commit bit).

iv. Validation-based concurrency control.

(c) Consider the ARIES recovery algorithm. Answer the following questions:

i. Indicate if the following statement is true of false. "At the end of the analysis phase, the Dirty Page Table contains the exact list of all pages dirty at the moment of the crash."

ii. During the UNDO phase of the recovery, the ARIES system writes CLR records in the log. Suppose the system crashes during the recovery. At restart, how are the CLR records used ? Check one of the answers below:

- They are used during the ANALYSIS phase.
- They are used during the REDO phase.
- They are used during the UNDO phase.
- They are ignored.

4. (**10** points) **Indexes**

Consider a relation `R(A,B,C)`. Assume the following statistics:

$$\begin{aligned} B(R) &= 1000 \\ T(R) &= 10000 \\ V(R,A) &= 20 \end{aligned}$$

Thus, on average $10000/1000 = 10$ records `(A,B,C)` fit in one block. There are two indexes on `R`:

- A $B+$-tree, clustered index on `A`
- A $B+$-tree, unclustered index on `B,A`. On average, a leaf nodes contains 250 `(B, A, TID)` triples, where TID is a tuple id.

Consider the query:

```
select B
from R
where A='abcd'
```

Indicate the I/O cost for each of the physical plans below. You may assume that the depth of each $B+$ tree is 3, that is, the query processor needs to read two blocks before reaching a leaf node of the $B+$ tree.

(a) Scan sequentially the table `R`, apply the predicate `A='abcd'` on-the-fly.

(b) Use the clustered index on `A` to retrieve the records `A='abcd'`.

(c) Use the unclustered index as a "covering index" (that is, the query is answered by reading *only* the unclustered index, and without reading the table `R`; see the lecture notes).

5. (**20** points) **Query Execution and Optimization**

(a) Show a relational algebra plan that is equivalent to the following SQL query:

```
select y.cid, count(*)
from Product x, Company y
where x.manufacturer = y.cid
  and y.address='Seattle'
  and not exists (select *
                  from Product z
                   where z.manufacturer = y.cid
                     and z.name = x.name
                     and z.startDate > x.startDate)
group by y.cid
```

(b) Consider two tables $R(A, B)$ and $S(C, D)$ with the following statistics:

$$\begin{aligned} B(R) &= 5 \\ T(R) &= 200 \\ V(R, A) &= 10 \\ B(S) &= 100 \\ T(S) &= 400 \\ V(S, C) &= 50 \\ M &= 1000 \end{aligned}$$

There is an unclustered index on $R.A$ and a clustered index on $S.C$. Consider the logical plan:

$$P = \sigma_{A=77}(R) \bowtie_{B=C} S$$

There are two logical operators, $s = \sigma_{A=77}$ and $j = \bowtie_{B=C}$, and for each we consider two physical operators:

$$\begin{aligned} s_1 &= \text{sequential table scan} \\ s_2 &= \text{index-based selection using } R.A \\ j_1 &= \text{main memory hash join} \\ j_2 &= \text{index-based join using the index } S.C \end{aligned}$$

Both s_1 and s_2 are pipelined, i.e. the result of the select operator is not materialized. For each of the resulting four physical plans compute its cost in terms number of disc I/0s, expressed as a function of the statistics above, then indicate its numerical value. Your answer should consists of four expressions of the form $\texttt{COST}(s_1 j_1) = B(R)B(S)/M + V(R, A) = 544$ (not the real answer). You may ignore the cost of accessing an index, i.e. assume that all intermediate nodes of a $B+$ tree are in the buffer pool.

i. $\texttt{COST}(s_1 j_1) =$

ii. $\texttt{COST}(s_2 j_1) =$

iii. $\texttt{COST}(s_1 j_2) =$

iv. $\texttt{COST}(s_2 j_2) =$

(c) Consider two tables $R(A, B)$ and $S(C, D)$, and the query plan P below:

$$P \quad = \quad \sigma_{A>9}(\gamma_{A,sum(D) \text{ as } D}(R \bowtie_{B=C} S))$$

Indicate which of the following three query plans P_1, P_2, P_3 are equivalent to P. The symbol $\ltimes$ represents semijoin, $R \ltimes_{\texttt{condition}} S = \Pi_{\texttt{Attributes}(R)}(R \bowtie_{\texttt{condition}} S)$.

$$P_1 \quad = \quad \gamma_{A,sum(D) \text{ as } D}(\sigma_{A>9}(R) \bowtie_{B=C} \gamma_{C,sum(D) \text{ as } D}(S))$$

$$P_2 \quad = \quad \gamma_{A,sum(D) \text{ as } D}(\gamma_{A,sum(B) \text{ as } B}(\sigma_{A>9}(R)) \bowtie_{B=C} (S))$$

$$P_3 \quad = \quad \gamma_{A,sum(D) \text{ as } D}(\sigma_{A>9}(R) \bowtie_{B=C} \gamma_{C,sum(D) \text{ as } D}(S \ltimes_{B=C} \sigma_{A>9}(R)))$$

(d) Consider the following query, where $\bowtie$ denotes the natural join:

$$R(A, B) \bowtie S(B, C) \bowtie T(C, D) \bowtie U(D, E)$$

Here we only consider left linear plans, and do not distinguish between different join orders, i.e. $R(A, B) \bowtie S(B, C)$ is the same as $S(B, C) \bowtie R(A, B)$.

i. Show two different left linear plans without cartesian products.

ii. How many different plans without cartesian product exists for this query ?

6. (**10** points) **XML/XPath/XQuery**

Consider an XML instance having the following DTD:

```
<!ELEMENT doc (movie)*>
<!ELEMENT movie (title, year, actor*)>
<!ELEMENT actor (name, gender?)>
```

Movie titles are unique in the data. All `<gender>` elements are either `<gender> male </gender>` or `<gender> female </gender>`. Furthermore the actor's names are unique. That is, if `<name> Bacon </name>` occurs under different movies then it is the same actor, and all occurrences will have the same `<gender>`, or will miss the `<gender>` element. That is, we may find occurrences of the form:

```
<actor> <name> Bacon </name> <gender> male </gender> </actor>
```

or of the form:

```
<actor> <name> Bacon </name> </actor>
```

but the data will not contain

```
<actor> <name> Bacon </name> <gender> female </gender> </actor>
```

because in that case there would be two genders for the same actor.

(a) Write an XPath expression that computes all the movies where "Bacon" acted. Your expression should return an answer of the following form:

```
<title> a movie title here... </title>
<title> another title here... </title>
. . .
```

Note that you have to write an XPath expression, not an XQuery expression.

(b) Write an XQuery (or XPath) expression that computes the gender of the actor named "Subrahmanian". Your XQuery expresion should return either the single element `<gender> male </gender>` or the single element `<gender> female </gender>` or nothing at all (if no gender information is found for Subrahmanian).

(c) Write an XQuery expression that transforms the data into another data having the following DTD:

```
<!ELEMENT doc (actor*)>
<!ELEMENT actor (name, gender?, movie*)>
<!ELEMENT movie (title, year)>
```

Actors should occur uniquely, their gender should be listed only once, if it is available, and the actor element should include all movies that they acted in.