Prof. Li-Yan Yuan

CMPUT 391: Database Management Systems

Solutions to Final Examination

December 15, 2000

It is a close-book examination and the time for the test is 120 minutes. There are ten (10) questions over three (3) pages. The value of each question is indicated in [ ] and the total is 100. Good luck to all of you.

1. Summarize the strength of the relational, object-oriented, and object-relational database management systems, in terms of data type, query language, and performance. (At most three sentences.) [8]

   Solution: The relational DBMS enjoys a powerful query language, the object oriented DBMS support user defined, rich object data types, and excellent performance, and the object-oriented DBMS supports rich, user defined object data types, as the object-oriented system, and enjoys a powerful query language, as the object-oriented system.

2. Consider a database consisting of the following tables.

```
employee( e_id, e_name, position, salary )
works( e_id, d_id, pct_time )
dept( d_id, budget, manager_id )
```

   The first table indicates the job title and salary for each employee, the second shows the percentage of time that a given employee works for a given department, and the last one indicates the budget and boss of each department.

   Write an SQL query to list the names of all employees whose salary exceeds the budget of all the departments that he/she works for. [10]

   Solution:

```
select e_name
from employee
where e_salary > any ( select max(budget)
                       from  dept, works
                       where employee.e_id = works.e_id and
                             works.d_id = dept.d_id
                     )
```

3. Consider the database in the previous question again. Write SQL3 triggers to ensure that no employee has salary higher than that of the manager of one or more departments that he/she works for. [10]

   Solution:

```
create trigger salary_cap
before insert or update salary on employee
for each row
declare
   m_salary integer;
begin
   select max(e.salary) into m_salary
   from employee e, works w. dept d
   where w.e_id = new.e_id and w.d_id = d.d_id and
         d.manager_id = e.e_id;
```

```
      if new.salary > m_salary
         then raise_application_error(-2000, 'employee's salary cannot
              exceeds that of his/her manager' );
      end if
   end
```

Similar triggers are also needed for the update on work and dept.

4. Let $R = (StudentID, Name, Birthday, Advisor, Dept, Course, Grade, Hobby)$ and F contain the following functional dependencies

   StudentID → Name Birthday Advisor Dept;
   StudentID Course → Grade;
   Advisor → Dept.

   (a) Find a candidate key for R.

   (b) Find a join loss-less, dependency preserving and 3NF decomposition of R.

   (c) Explain whether your database schema obtained in (b) suffers from the problem of repetition of information. [8]

   Solution:

   (a) The only candidate key for R is {StudentID, Course, Hobby}.

   (b) The database with the following 4 relations is a join loss-less, dependency preserving and 3NF decomposition of R.

   ```
   (StudentId, Name, Birthday, Advisor)
   (StudentID, Course, Grade),
   (Advisor, Dept)
   (StudentId, Course, Hobby)
   ```

   (c) The last table in the above decomposition suffers from the obvious repetition because a student make take 5 courses and enjoys 10 different hobbies.

5. Consider the set of attributes $R = ABCDEG$ and the FD set $F = \{AB \rightarrow C, AC \rightarrow B, AD \rightarrow E, B \rightarrow D, BC \rightarrow A, E \rightarrow G\}$. Which of the following decompositions of R is (1) dependence-preserving ? (2) join lossless?

   (a) $\{AB, BC, ABDE, EG\}$
   Solution: This is neither dependency preserving or join-loss.
   It is easy to see that $AB \rightarrow C$ is not preserved. Further, consider the following instance r over R.

A	B	C	D	E	G
a1	b	c1	d	e1	g1
a2	b	c2	d	e2	g2

   Then the join of all the projects of r over relations schemas in the database contains tuples not in r.

   (b) $\{ABC, ACDE, ADG\}$ [8]
   Solution: It is not dependency preserving for $E \rightarrow G$ cannot be preserved but it is join loss-less because it can be obtained from R by decomposition based on $AC \rightarrow B$ first and then by $AD \rightarrow G$.

6. The SQL3 specifies four (4) different levels of isolation, i.e., READ UNCOMMITTED, READ COMMITTED, REPEATABLE READ, and (ANOMALY) SERIALIZABLE.

   (a) Present a sample schedule S such that S observes REPEATABLE READ but not SERIALIZABLE.

   (b) Indicate whether your schedule is permitted under the two phase locking protocol?

   (c) If you believe that the two phase locking protocol does not guarantee SERIALIZABLE, describe an approach to fix the problem. Otherwise, demonstrate how the two phase locking protocol prevents your schedule.

   Indicate any reasonable assumptions you may have. [10]

   Solution:

   Any schedule with phantom but without no dirty read and non-repeatable read observes REPEATABLE READ but not SERIALIZABLE. A sample schedule with tow transactions to access a relation $take = (Student, Course, Grade)$ is given below.

T1	**T2**
select * from take where course = 391	
	insert into take('Sarah', 391, 'B')
select * from take where course = 391	

   The phantom phenomenon can be prevented by the two phase locking protocol only if the logic lock is implemented.

7. Consider relations $r_1(A, B)$ and $r_2(B, C)$. Assume that the only (B-tree/hashtable) indexes available are on A for r_1 and on B for r_2. Describe an efficient algorithm for computing $r_1 \bowtie r_2$. ($\bowtie$ stands for natural join.) You may use a pseudo language to describe your algorithm. [10]

   Solution:

```
while ( r2 is non-empty ) {
   sequentially load one tuple
         t2 = < b, c > from r2;
   key_search r1 for t1 = < A, b >
   and join t2 with t2 if found
}
```

8. Consider the following database consisting of six transactions of items. [12]

transaction	**items**
t_1	pen, ink, diary, soap, camera, film
t_2	pen, ink, diary, camera, film
t_3	pen, diary,film
t_4	pen, ink, soap, film
t_5	camera, film, ink
t_6	ink, film, soap

   (a) Find all the association rules $X \Rightarrow Y$, where X has exactly one item, and Y is non-empty, with confidence >= 65% and support > 30%. [12]

It is very tedious if you try to calculate the support and confidence for each and every candidate association rules. Considering the fact that if $X \Rightarrow Y$ is a strong rule so is $X \Rightarrow Y'$ for each

$Y' \subseteq Y$, we need only to consider the maximum Y such that $X \Rightarrow Y$ satisfies the given support and confidence.

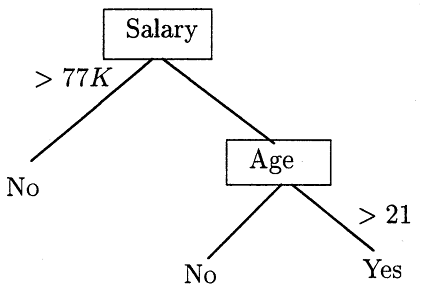
The following table lists all such strong rules/

rule	**support**	**confidence**
$\{pen\} \Rightarrow \{ink, film\}$	50	75
$\{pen\} \Rightarrow \{diary, film\}$	50	75
$\{ink\} \Rightarrow \{film\}$	83	100
$\{diary\} \Rightarrow \{pen, ink, camera, film\}$	33	67
$\{soap\} \Rightarrow \{pen, ink, film\}$	33	67
$\{camera\} \Rightarrow \{pen, ink, diary, film\}$,	33	67
$\{film\} \Rightarrow \{pen\}$,	33	67
$\{film\} \Rightarrow \{ink\}$,	33	67

9. Consider the following table containing information about the marketing campaign of the *Florida Recount* magazine. The first three columns show the age, salary, and residency state/province of a potential customer and the subscription column shows whether the person subscribed to the magazine. Please use this data to construct a decision tree that helps to predict whether a person is going to subscribe to the magazine. The simpler the decision tree, the higher the mark. [12]

age	**salary**	**state**	**subscription**
37	25k	FL	Yes
39	80k	TX	No
64	120k	TN	No
56	50k	TX	Yes
52	43k	AB	Yes
35	90k	FL	No
32	54k	AB	Yes
40	59k	FL	Yes
55	78k	FL	No
43	19k	TX	Yes
21	25k	TX	No
28	34k	AB	Yes

Solution: The following is one of the simplest.



10. Consider the following XML document. Define a relational database schema suitable for storing the information in the document in Oracle, and populate the database according to the XML document.

    Use the normal form theory to justify your design. [12]

```
<?xml version="1.0" encoding="utf-8"?>
<book isbn="0836217462" format = "hardcover">
   <title>
        Being a Dog Is a Full-Time Job
   </title>
   <authors>
        <author  aid = "12345">
             <name>
                  Charles M. Schulz
             </name>
             <email>
                 schulz@hotmail.com
             </email>
        </author>
   </authors>
   <published>
        1997
   </published>
</book>
<book isbn="23650395454" format = "paperback">
   <title>
        The Character of Physical law
   </title>
   <authors>
        <author aid = "54321">
             <name>
                 Richard Brewka
             </name>
             <email>
                 brewka@www.com
             </email>
         </author>
         <author  aid = "12345">
             <name>
                 Charles M. Schulz
             </name>
             <email>
                Schulz@hotmail.com
             </email>
        </author>
   </authors>
   <published>
       1979
   </published>
</book>
```