

CSE 444 Final --- Spring 2007

Tuesday, June 5, 10:30-12:20

Name _____ Student ID _____

#	1	2	3	4	Total (100)
Score					

Time: 1 hour 50 minutes.

Total points: 100.

1. [25 points] An online store maintains a database about their products and customers with the following schema:

Product(pid, weight) /* pid = product id, weight = in lb */

Order(pid, cid, quantity) /* quantity = an integer */

Customer(cid, age) /* cid = customer id */

- a. Write a SQL query that returns the cid's of all customers that have placed at least one order over 1000lb.

- b. Write a SQL query that returns the cid's of all customers whose total orders exceeds 1000lb.

- c. The company hires you to analyze customer trends. But instead of giving to you the three base tables, they send you the following three views:

HeavyProduct(pid) -- products with weight > 5

LargeOrder(pid, weight) -- products having orders with quantity > 100

HotOrders(pid, cid, quantity, age) –orders by customers with age < 30.

The views are defined as follows:

```
CREATE VIEW HeavyProduct AS
```

```
    SELECT x.pid  
    FROM Product x  
    WHERE x.weight > 5
```

```
CREATE VIEW LargeOrder AS
```

```
    SELECT DISTINCT x.pid, x.weight  
    FROM Product x, Order y  
    WHERE x.pid = y.pid AND y.quantity > 100
```

```
CREATE VIEW HotOrders AS
```

```
    SELECT y.pid, y.cid, y.quantity, z.age  
    FROM Order y, Customer z  
    WHERE y.cid = z.cid AND z.age < 30
```

In order to perform the required analysis you need to run the four SQL queries below. However, since you do not have the base tables but only the three views, you need to rewrite each of the four queries to express them over the views. For each of the queries below expressed over **Product**, **Order**, **Customer**, you have to write an equivalent SQL query over the views **HeavyProduct**, **LargeOrder**, **HotOrders**, or indicate that

such a rewriting is not possible. For example if the query were:

```
Qa = SELECT DISTINCT y.cid  
      FROM Order y, Customer z  
      WHERE y.cid = z.cid AND y.quantity > 40 AND z.age < 30
```

then you would answer:

```
Qa' = SELECT DISTINCT u.cid  
      FROM HotOrders u  
      WHERE u.quantity > 40
```

while if the query where:

```
Qb = SELECT DISTINCT y.cid  
      FROM Order y, Customer z  
      WHERE y.cid = z.cid AND y.quantity > 40 AND z.age < 35
```

then you would answer NOT POSSIBLE since Qb cannot be answer from the three views.

Q1 =

SELECT DISTINCT x.pid

FROM **Product** x, **Order** y, **Customer** z

WHERE x.pid = y.pid AND y.cid = z.cid

AND x.weight > 5 AND y.quantity > 50 AND z.age < 25

Q2 =

SELECT DISTINCT x.pid

FROM **Product** x, **Order** y, **Customer** z

WHERE x.pid = y.pid AND y.cid = z.cid

AND x.weight > 7 AND y.quantity > 50 AND z.age < 25

Q3 =

```
SELECT DISTINCT x.pid
```

```
FROM Product x, Order y, Customer z
```

```
WHERE x.pid = y.pid AND y.cid = z.cid
```

```
    AND x.weight > 2 AND y.quantity > 180 AND z.age < 30
```

Q4 =

```
SELECT DISTINCT x.pid
```

```
FROM Product x, Order y, Customer z
```

```
WHERE x.pid = y.pid AND y.cid = z.cid
```

```
    AND x.weight > 5 AND y.quantity > 100 AND z.age < 35
```


2. [25 points] XML. 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 expression 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, with their correct gender, and should include all movies that they acted in.

3. [25 points] Consider the two logs below, called LogA and LogB:

1	<START T1>
2	<T1, A, a>
3	<T1, B, b>
4	<START T2>
5	<T2 C, c>
6	<START T3>
7	<T3 D, d>
8	<COMMIT T1>
9	<START CKPT(T2,T3)>
10	<T2,E,e>
11	<START T4>
12	<T4,F,f>
13	<T3,G,g>
14	<COMMIT T3>
15	<END CKPT>
16	<COMMIT T2>
17	<COMMIT T4>

LogA

1	<START T1>
2	<T1, A, a>
3	<T1, B, b>
4	<START T2>
5	<T2 C, c>
6	<START T3>
7	<T3 D, d>
8	<COMMIT T1>
9	<START CKPT(T2,T3)>
10	<T2,E,e>
11	<START T4>
12	<T4,F,f>
13	<T3,G,g>
14	<COMMIT T3>
15	<COMMIT T2>
16	<END CKPT>
17	<COMMIT T4>

LogB

(i). One of the logs is a UNDO log, the other is a REDO log. Indicate which one is the UNDO log and which is the REDO log:

The UNDO log is: _____

The REDO log is: _____

Note: you must answer this question correctly in order to be able to answer the other questions.

(ii). Consider the UNDO log. For each of the elements A, B, ..., G, indicate when the system might have issued the corresponding OUTPUT statement. For each of the you will answer with an interval, e.g.

“<OUTPUT B> may be issued between time steps 7 and 14” (that is, you are saying that <OUTPUT B> may be issued any time after <T3,D,d> but before <COMMIT T3>; note: this is not necessarily the right answer for <OUTPUT B>). If you want to indicate that the OUTPUT may happen after the end of the log, write either “7,18” or “7, ∞”.

<OUTPUT A>	
<OUTPUT B>	
<OUTPUT C>	
<OUTPUT D>	
<OUTPUT E>	
<OUTPUT F>	
<OUTPUT G>	

(iii). Repeat the same question for the REDO log:

<OUTPUT A>	
<OUTPUT B>	
<OUTPUT C>	
<OUTPUT D>	
<OUTPUT E>	
<OUTPUT F>	
<OUTPUT G>	

(iv). Consider a concurrency control manager by timestamps. Below are several sequences of events, including start events, where st_i means that transaction T_i starts and co_i means T_i commits. These sequences represent real time, and the timestamp-based scheduler will allocate timestamps to transactions in the order of their starts. In each case below tell what happens with the last write request.

You have to choose between one of the following four possible answers:

- (1) the request is **accepted**,
- (2) the request is **ignored**,
- (3) the transaction is **delayed**,
- (4) the transaction is **rolled back**.

[1]. $st_1; st_2; st_3; r_1(A); r_2(B); r_2(C); r_3(B); co_2; w_3(B); w_3(C)$

The system will perform the following action for $w_3(C)$: _____

[2]. $st_1; st_2; r_1(A), r_2(B); w_2(A); co_2; w_1(B)$

The system will perform the following action for $w_1(B)$: _____

[3]. $st_1; st_3; st_2; r_1(A); r_2(B); r_3(B); w_3(A); w_2(B); co_3; w_1(A)$

The system will perform the following action for $w_1(A)$: _____

[4]. $st_1; r_1(A); w_1(A); st_2; r_2(C); w_2(B); r_2(A); w_1(B)$

The system will perform the following action for $w_1(B)$: _____

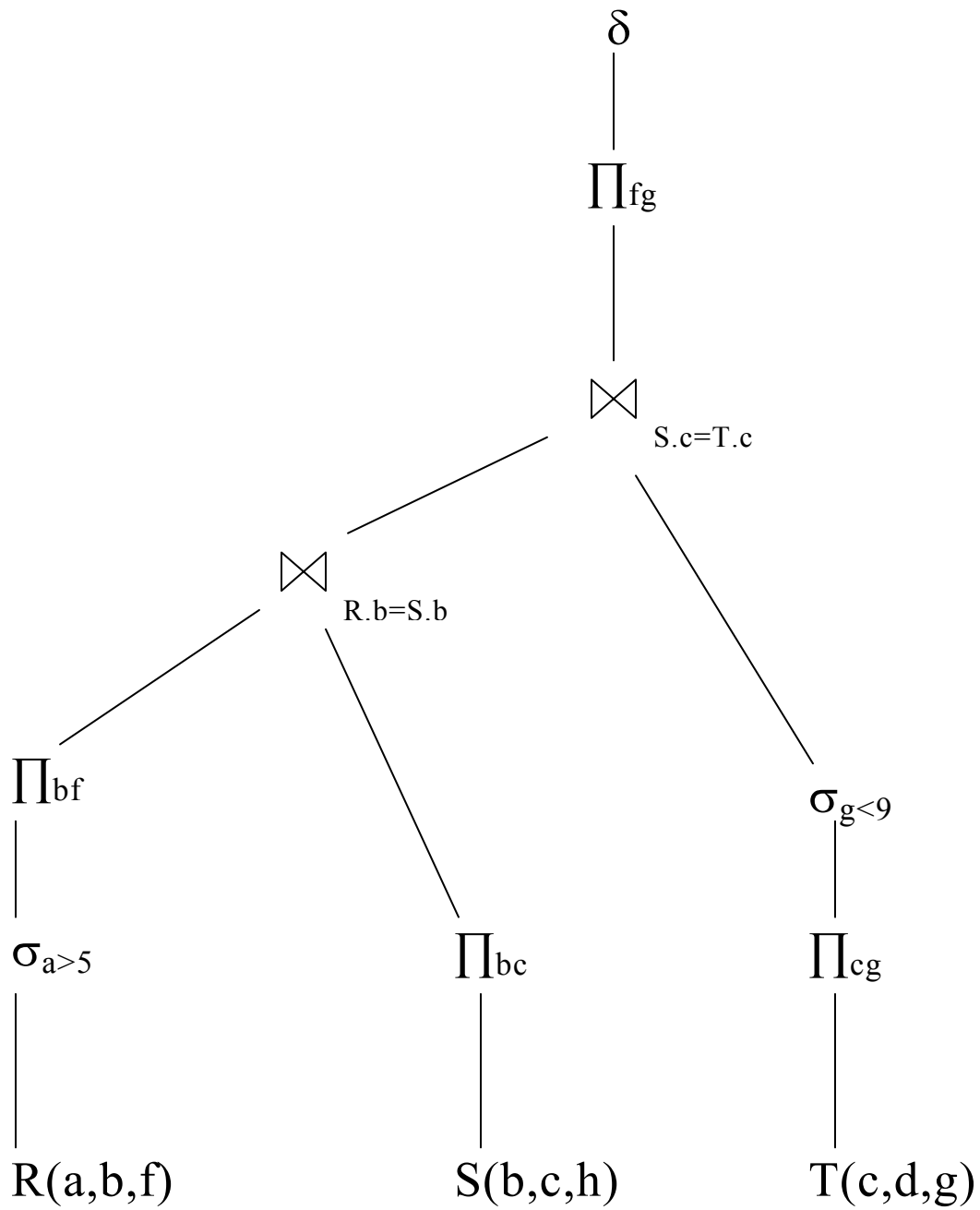
4. [25 points] Consider three tables R(a,b,f), S(b,c,h), T(c,d,g).

(a) Consider the following SQL query:

```
SELECT  R.a, sum(T.d)
FROM    R, S, T
WHERE   R.b = S.b AND S.c = T.c AND R.f = 5 AND T.g = 9
GROUP BY R.a
HAVING  sum(S.h) > 10
```

Draw a logical plan for the query. You may chose any plan as long as it is correct (i.e. no need to worry about efficiency).

(b) Write a SQL query that is equivalent to the logical plan below:



[This page is intentionally left blank; you may use it to write your answer.]