CSE444 Final

December 8, 2008

Name: .....

Problem	Points
Q1 [35 points] Q2 [35 points] Q3 [15 points] Q4 [15 points]	
Total [100 points]	

1 [35 points] SQL

The following schema contains chess players, and their games:

```
Player(PID, name, city)
Game(PID1, PID2)
```

An entry in Game means that PID1 has won over PID2. There are no draws. There can be multiple games, i.e. both PID1 won over PID2 and vice versa.

a. 'Joe Champion' is a famous player. (His name is unique in Player.) For each city count the number of players in that city that have won against 'Joe Champion' at least once.

b. Find all cities where every player has won over ‘Joe Champion’.

c. Find all cities where no player ever lost against ‘Joe Champion’.

d. Denote Q1 and Q2 the two queries in questions b and c. Indicate which of the following are true:

$$Q1 \subseteq Q2$$
$$Q1 = Q2$$
$$Q1 \supseteq Q2$$

2 [35 points] Database Internals

a. **Transactions.** The scheduler uses shared locks for reads and exclusive locks for writes and follows the strict 2PL policy: locks are acquired on a per need basis, and released at commit time. There are three transactions wishing to execute the following:

```
T1:  ST1;  R1(A), R1(B), W1(A), W1(B), CO1.
T2:  ST2:  R2(B), R2(C), W2(B), W2(C), CO2.
T3:  ST3:  R3(C), R3(A), W3(C), W3(A), CO3.
```

Consider the partial schedules below. For each schedule indicate whether (a) it will end in a deadlock no matter how the scheduler continues to schedule the transactions; in this case show a possible continuation of the schedule until it ends in a deadlock, (b) it may be completed successfully, but may also result in a deadlock; in this case show both a successful continuation of the schedule and one that ends in a deadlock. (c) it will complete successfully no matter what the scheduler does; in this case show a successful completion of the schedule.

```
Schedule 1
ST1, ST2, ST3, R1(A), R2(B), R3(C)
```

```
Schedule 2:
ST1, ST2, ST3, R2(B), R3(C), R3(A), W3(C)
```

```
Schedule 3:
ST1, ST2, ST3, R1(A), R2(B), R2(C), W2(B), W2(C)
```

Suppose now that the scheduler uses only exclusive locks, both for reads and writes. As an application writer you are asked to rewrite transaction T3 eliminate the possibility of deadlocks. Make a minimal change in the order of the operations in T3 such that a deadlock can never occur.

b. **Relational Algebra.**

Consider the query below:

```
SELECT R.key, sum(T.val)
FROM R, S, T
WHERE R.key = S.fk and S.key = T.fk
  and R.A = 'abc' and T.B = 'cde'
GROUP BY R.key
```

The attributes R.key, S.key, and T.key are keys, while S.fk and T.fk are foreign keys. Suppose:

```
T(R) = 10000 tuples
T(S) = 100000 tuples
T(T) = 1000000 tuples
```

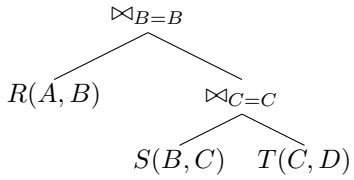
In each of the two cases below, show an optimal logical relational algebra plan.

b.1 `V(R, A) = 10, V(T, B) = 10000`

b.2 `V(R, A) = 1000, V(T, B) = 100`

c. **Query evaluation, indexes**

Consider the following logical plan:



Consider the following parameters:

```
M = 100
B(R) = 8000000
B(S) = 7000000
B(T) = 5000
B(S ⋈ T) = 5000
```

You will use partitioned hash join to process this query. You are allowed to interleave the two steps (partition and hash) for the two joins, as necessary. Show an optimal order of execution. Indicate the following:

- order of the two partition and two join steps
- the number of buckets used for each partition
- the size of the buckets (assume perfectly uniform distribution)
- which operations are pipelined
- indicate the total cost of this query plan

d. **Database tuning, indexes.**

Consider two indexes on the relation R(A,B,C,D):

- clustered index on (A,B) called CI
- unclustered index on (B, C) called UI

For each query below indicate which of the following is the most efficient way to execute the query: (i) by using index CI (ii) by using index UI, (iii) either by using CI or UI depending on the data, (iv) not using an index at all.

In each case below indicate which index is most efficient for the query at hand. If no index is usable for the query, indicate so.

```
SELECT * FROM R WHERE R.B = 55 and R.D = 99
```

```
SELECT * FROM R WHERE R.A = 22 and R.D = 99 and R.C = 33
```

```
SELECT * FROM R WHERE R.A > 11 and R.A < 22 and R.B = 55 and R.C = 33
```

3 [15 points] Advanced techniques: Hash maps and Bloom Filters

Data supplier S1 has $n = 1M$ $(= 10^6)$ documents. Data supplier S2 has also $n = 1M$ documents. Each document has $1k$ bytes. They have 50 documents in common and they want to compute these. They will proceed as follows:

- S1 computes a hash map M with cn bits, where c=8 and sends it to S2
- S2 checks its items in M and sends all matches to S1
- S1 computes the result and sends the matching 50 documents to S2

Indicate the total number of bytes transferred over the network in each step assuming (a) the hash map is a standard hash table, (b) the hash map is a bloom filter. [You may use the formulas in the lecture notes].

4 [15 points] XML

Consider the XML document below:

```
<aa>
  <bb>
    <cc> 1 </cc>
    <dd> 2 </dd>
    <bb> 3 </bb>
  </bb>
  <cc>
    <dd> 4 </dd>
    <cc> 5 </cc>
    <bb> 6 </bb>
  </cc>
  <cc>
    <dd> 7 </dd>
    <cc> 8 </cc>
  </cc>
</aa>
```

For each question below, write an XPath expression that returns precisely the values indicated. Your XPath expression should be restricted to navigation and qualifiers only: do not use numerical predicates. For example `//dd/text()` and `/*/*[bb]/*/text()` are acceptable answers but `//*[text()=4]/text()` is not.

i. Return 3, 6

ii. Return 4, 5, 6, 7, 8

iii. Return 4, 5, 6