CSE 344 Final Examination

December 16, 2015, 8:30am - 10:20am

Name: ________________________________

<table>
<thead>
<tr>
<th>Question</th>
<th>Points</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>Total:</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>

• This exam is CLOSED book and CLOSED devices.
• You are allowed TWO letter-size pages with notes (both sides).
• You have 1h:50 minutes; budget time carefully.
• Please read all questions carefully before answering them.
• Some questions are easier, others harder; if a question sounds hard, skip it and return later.
• Good luck!
1 XML, XPath, and XQuery

1. (20 points)

(a) (10 points) Consider the following XML document stored in a file called data.xml:

```xml
<cluster>
  <machine id='M1'>
    <software>
      <dbms>MySQL</dbms>
      <dbms>PostgreSQL</dbms>
    </software>
    <hardware>
      <memory>16</memory>
      <cores>4</cores>
    </hardware>
  </machine>

  <machine id='M2'>
    <software>
      <dbms>MySQL</dbms>
    </software>
    <hardware>
      <memory>8</memory>
      <cores>2</cores>
    </hardware>
  </machine>

  <machine id='M3'>
    <software>
      <dbms>PostgreSQL</dbms>
    </software>
    <hardware>
      <memory>64</memory>
      <cores>16</cores>
    </hardware>
  </machine>

  <machine id='M4'>
    <software>
      <dbms>MongoDB</dbms>
    </software>
    <hardware>
      <memory>64</memory>
      <cores>16</cores>
    </hardware>
  </machine>
</cluster>
```
Write **XPath** expressions that compute the following:

1. The id of the machines that run MySQL (formatting of result is up to you):

2. The amount of memory available on machine M4 (formatting of result is up to you):
(b) (10 points) Write an XQuery expression that will transform data.xml into the following document:

```xml
<result>
  <dbms>
    <name>MySQL</name>
    <nb_servers>2</nb_servers>
    <cores>4</cores>
    <cores>2</cores>
  </dbms>
  <dbms>
    <name>PostgreSQL</name>
    <nb_servers>2</nb_servers>
    <cores>4</cores>
    <cores>16</cores>
  </dbms>
  <dbms>
    <name>MongoDB</name>
    <nb_servers>1</nb_servers>
    <cores>16</cores>
  </dbms>
</result>
```
2. (30 points)

(a) (10 points) Design an E/R diagram describing the following domain:

- A person has attributes \text{id} (key), \text{fname}, and \text{lname}.
- A company has attributes \text{id} (key) and \text{name}. Each company name is unique.
- A party has attributes \text{pid}, \text{date}, and \text{duration}. The date attribute cannot be NULL. A party cannot last longer than 4 hours.
- A party is booked by either a person or a company. We need to capture information about both when the party takes place and when the party was booked. The booking date must be earlier than the party date. A party is booked by zero or one organizers. A person or company can book many parties.
- A party occurs in either a house, an apartment, or a suite. All these locations are uniquely identified with a location id (lid). A house has an attribute street address. The address is an atomic string attribute. An apartment additionally has an apartment number. A suite has a street address and a suite number. A party occurs in zero or one place. A place can host multiple parties.
- A company occupies zero or one suite and a suite hosts zero or one company (i.e., the application does not store historical data only current data).
- A person lives in either a house or an apartment. A person lives in zero or one place. A given place can host many people.

\textbf{Answer} (Draw an E/R Diagram on the next page):
Answer (Draw an E/R Diagram):
(b) (10 points) Write the CREATE TABLE statements necessary to capture the subset of the above ER diagram that corresponds to party, person, and company entities as well as the books relationship. Include all primary key, foreign key, and other constraints. Avoid creating unnecessary tables when possible. Dates can be represented with the `datetime` type.

**Answer** (Write CREATE TABLE statements for SUBSET of ER diagram):
(c) (10 points) Consider the following relational schema and set of functional dependencies.

\[ R(A, B, C, D, E, F, G) \]

with functional dependencies:

- \( E \rightarrow C \)
- \( G \rightarrow AD \)
- \( B \rightarrow E \)
- \( C \rightarrow BF \)

- Give one example of non-trivial functional dependency implied by the ones above:

  **Answer (Example FD):**

- Compute \( E^+ \), the closure of \( E \).

  **Answer \((E^+)\):**

- Why do we bother to decompose relations into BCNF? What does this normal form ensure?

  **Answer (Explanation):**
• Decompose R into BCNF. Show your work for partial credit. Your answer should consist of a list of table names and attributes and an indication of the keys in each table (underlined attributes).

**Answer** (Decompose R into BCNF):
3 Transactions

3. (20 points)
   (a) (10 points) Consider transactions executing concurrently on the same instance of SQL Server. For each of the execution traces below, circle the isolation levels (if any) that will allow the given interleaved execution to occur.

BEGIN TRANSACTION

SELECT * from Customers

SELECT * from Products

COMMIT

BEGIN TRANSACTION

SELECT * from Customers

SELECT * from Products

COMMIT

Serializazable  Repeatable Read  Read Committed
BEGIN TRANSACTION

UPDATE Customers SET id = 20
WHERE id = 10

UPDATE Products SET price = 0
WHERE price is NULL

SELECT * from Customers

SELECT * from Products

COMMIT

COMMIT

Serializable  Repeatable Read  Read Committed
BEGIN TRANSACTION

UPDATE Customers SET id = 10
WHERE id = 20

UPDATE Products SET price = 10
WHERE price = 0

SELECT * from Customers

SELECT * from Products

COMMIT

BEGIN TRANSACTION

UPDATE Customers SET id = 10
WHERE id = 20

UPDATE Products SET price = 10
WHERE price = 0

SELECT * from Customers

COMMIT

Serialization  Repeatable Read  Read Committed
BEGIN TRANSACTION

SELECT * from Customers

BEGIN TRANSACTION

INSERT INTO Customers VALUES(1)

SELECT * from Customers

COMMIT

COMMIT

Serializable  Repeatable Read  Read Committed
(b) (5 points) Consider the following transaction schedules. For each schedule, draw the precedence graph and indicate if it is conflict-serializable or not.

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

**Answer** (YES/NO):

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

**Answer** (YES/NO):
(c) (5 points) What is strict 2PL?
4 Parallel Data Processing

4. (30 points)

(a) (10 points) Consider relations $R(a, b)$, $S(c, d)$, and $T(e, f)$. All three are horizontally partitioned across $N = 3$ machines as shown in the diagram below. Each machine locally stores approximately $\frac{1}{N}$ of the tuples in $R$, $S$, and $T$. The tuples are randomly organized across machines (i.e., $R$ is block partitioned across machines).

Show a relational algebra plan for the following query and how it will be executed across the $N = 3$ machines. **Use hash-join (a.k.a shuffle-join) operators.** Include operators that need to re-shuffle data and add a note explaining how these operators will re-shuffle that data.

```
SELECT *
FROM R, S, T
WHERE R.b = S.c
AND S.d = T.e
AND (R.a - T.f) > 100
```

**Answer** (Draw the parallel query plan):

Machine 1

1/3 of R, S, T

Machine 2

1/3 of R, S, T

Machine 3

1/3 of R, S, T
(b) (10 points) Now consider the case where the R relation is very large and both S and T are very small. Show a plan that uses broadcast joins to compute the result of the query.
(c) (10 points) Explain how the query would be executed in MapReduce (not Pig). Make sure to specify the computations performed in the map and the reduce functions. Use hash-joins (shuffle joins). No need to give the pseudocode. Just state what each function does and what it outputs in plain text format.

**Answer** (Describe the map and reduce functions of the executed MapReduce job(s)):