Boston University
CS 460/660- Introduction to Database Systems
Final Exam – Spring 2005
(with solutions)

Student ID: __________
Name: ___________________________________________________

Instructions
• This exam is open book and notes.
• You have 120 minutes to complete it. There are 5 questions.
• Return BOTH question-sheet and answer booklet.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Points</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>15</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>35</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>20</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>15</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>15</td>
</tr>
<tr>
<td><strong>Total:</strong></td>
<td></td>
<td><strong>100</strong></td>
</tr>
</tbody>
</table>
Problem 1 (15 points)

Consider the following database schema:

Employee (fname, lname, ssn, bdate, address, sex, salary, mgrssn, dnumber)
Department (dname, dnumber, mgrssn, mngstartdate)
Project (pname, pnumber, plocation, dnumber)
Works_On (ssn, pnumber, hours)
Dependent (ssn, dependent_name, sex, bdate, relationship)

The above relations store information about a company. The meaning of most of the relations and attributes is straightforward. For example, the first relation stores information about employees. The mgrssn is the SSN of the manager (supervisor) of the given employee. The department relation stores information about departments. Also, each project is associated with a given department and is located in a single location. Finally, the Works_On and the Dependent relations are store information about which project each employee works and the dependents of each employee.

Using the above schema write the following queries in SQL:

(a) Find the names (first and last name) of all employees who work on every project.
(b) Find the average salary for each department, for all departments that have more than 5 employees.
(c) Find the names of all department managers who have no dependents.

Answer:

(a) SELECT fname, lname
    FROM Employee E
    WHERE NOT EXISTS( (SELECT DISTINCT pnumber
        FROM project)
        EXCEPT
        (SELECT DISTINCT pnumber
        FROM Project P, Works_On W
        WHERE P.pnumber = W.pnumber AND E.ssn = W.ssn))

(b) SELECT dname, avg(salary)
    FROM Employee E, Department D
    WHERE E.dnumber = D.dnumber
    GROUP BY dname
    HAVING COUNT(*)>5

(c) SELECT fname, lname
    FROM Employee E, Department D
    WHERE E.ssn = D.mgrssn AND NOT EXISTS ( SELECT ssn
        FROM Dependent P
        WHERE P.ssn = E.ssn)
Problem 2 (35 points)

In the following schedules, Ri(A) stands for a Read(A) operation by transaction i; Wi(A) stands for a Write(A) operation by transaction i.

1. For each of the following schedules show if it is conflict-serializable and give a conflict-equivalent serial schedule. Show all the conflict operations.
   
   (a) R1(A) W1(A) R2(A) R2(B) W3(B) W2(C) R4(A) R4(B) R4(C) R2(D) R3(E)
   (b) R1(A) R4(A) W1(A) W3(B) R2(A) R2(B) W2(C) R4(B) R4(C) R2(D) R3(E)

2. Consider the following schedule example where an item is missing:

   W3(B) R2(A) W1(A) R(?) R2(B) W1(B)

   and let:
   
   I. Producible using 2 Phase Locking
   II. Conflict Serializable

   Chose one answer for the following cases:
   
a) If ? = A, this schedule is which of the following:
   a. I & II b. I only c. II only d. neither I nor II

   b) If ? = B, this schedule is which of the following:
   a. I & II b. I only c. II only d. neither I nor II

   c) If ? = C, this schedule is which of the following:
   a. I & II b. I only c. II only d. neither I nor II

3. Suppose that we run the following transactions using the validation based protocol. The following table lists the transactions involved, together with their read and write sets:

<table>
<thead>
<tr>
<th>Transactions</th>
<th>Read Set</th>
<th>Write Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>{a,b}</td>
<td>{b, c}</td>
</tr>
<tr>
<td>T2</td>
<td>{a,b,c}</td>
<td>{h}</td>
</tr>
<tr>
<td>T3</td>
<td>{b}</td>
<td>{d, e}</td>
</tr>
<tr>
<td>T4</td>
<td>{c}</td>
<td>{f, g}</td>
</tr>
<tr>
<td>T5</td>
<td>{a}</td>
<td>{d, f}</td>
</tr>
<tr>
<td>T6</td>
<td>{g}</td>
<td>{e, g}</td>
</tr>
</tbody>
</table>

   The following sequence of events takes place. (No other transactions run before or concurrently with T1…T6)
1. T1, T2, T3, T4 start (in this order)
2. T1 initiates validation
3. T5, T6 start (in this order)
4. T5 initiates validation
5. T3 initiates validation
6. T4 initiates validation
7. T2 initiates validation
8. T1, T2, T3 finish (if they were not aborted earlier)
9. T6 initiates validation
10. T4, T5, T6 finish (if they were not aborted earlier)

Give the transactions that committed and the ones that aborted in the schedule above.

**Answer:**

1. (a) There is no cycle in the PG, so schedule is conflict serializable and equivalent to:
   {T1, T2, T3, T4}
   (b) There is a cycle in the PG. It is not conflict serializable.

2. (a) d
   (b) c
   (c) a

3. Committed successfully: T1, T5, T6
   Aborted: T2, T3, T4
Problem 3 (20 points)

Consider the following schema:

**Product**: (id(4), name(16), manufacturer(20), category(10), color(10), webpage(40))
**Sales**: (pid(4), quantity(4), shippingaddress(20), date(12), shippingmethod(10))

Each attribute has a fixed length, with the size (in bytes) indicated by the number in parentheses. The number of tuples in each relation is: T(Product) = 1,000,000 and T(Sales) = 2,000,000. The size of one disk block is 1000 bytes, and there are 101 buffer blocks available in main memory.

a. Compute the number of blocks taken by each table B(Product) and B(Sales).

b. Consider the logical plan below:

```
\Pi_{name, \text{date}}
\sigma_{\text{category}="\text{Toy}" \text{ AND quantity}>100}
```

Assume the following physical plan: the join is implemented as a hash join, its result is pipelined into the selection operator, and, from there, pipelined into the projection operators. Compute the total cost of this physical plan.

c. Derive a new logical plan by pushing selections and projections down as far as possible (you have to draw a plan).

d. Consider a physical plan for your new logical plan in which all selections and projects are pipelined and the join is a block-nested loop join. Further assume that 1% of all Products are in category “Toy” and that 20% of all Sales have a quantity over 100. Compute the cost of your plan.
Answer:

(a) \( b_P=100K \) and \( b_S=100K \)

(b) Since \( \left\lceil \frac{b_P}{M-2} \right\rceil > M \), we need to do recursive partitioning. Using the formula for the cost of the hash join using recursive partitioning we get that the cost of the join is: 1000000 I/Os. Note that the selection and projection operations are pipelined and therefore there are no additional I/O costs for these operators.

(c) We push the selection and the projections down. Apply selection on the Product relation on the “Toy” attribute and a projection on the id and name attributes. Also apply the selection on the Sales relation on the quantity attribute and a projection on the id and data. Then do the join of the results and project on the name and data.

(d) Let Product’ and Sales’ be the left and right operand of the join respectively. We have:

- \( T(\text{Product'}) = 10,000 \), record size = 20, records per block = 50
- \( B(\text{Product'}) = \frac{10,000}{50} = 200 \)
- \( T(\text{Sales'}) = 400,000 \), record size = 16, records per block = \( \frac{1000}{16} \)
- \( B(\text{Sales'}) = \frac{400,000}{(1000/16)} = 6400 \)

It is more advantageous to use Product’ as outer relation. That is, the bock-nested loop will read data from Product, perform the projection and selection on the fly and store 100 blocks of records from Product’ in main memory. For each such set it will do a full scan of Sales, perform the selection and projection, then the join. Total cost:

\[
B(\text{Product}) + B(\text{Product'})/100 * B(\text{Sales}) = 100,000 + 200/100*100,000=100,000 + 200,000=300,000
\]
Problem 4 (15 points)

Consider the following log corresponding to a particular schedule at the time of a system crash for four transactions T1, T2, T3, and T4. Suppose that we use the immediate modification protocol with checkpointing. Describe the recovery process from the system crash. Show all the Undo and Redo operations and the values of the data items A, B, C, and D after the recovery.

Answer:

Undo list \{T2, T3\}, Redo List \{T4\}. We perform Undo first:

- T3: C \rightarrow 30
- T2: D \rightarrow 15
- T2: B \rightarrow 12

And then Redo:

- T4: D \rightarrow 15
- T4: A \rightarrow 20

Values of A, B, C, and D: 20, 12, 15, 30
Problem 5 (15 points)

Consider the following sequence of numeric keys:

\[(20, 11, 14, 31, 47, 23, 56, 12, 34, 35, 41)\]

Construct a B+-tree for this sequence. Initially, assume that the tree is empty. Also assume that the maximum number of pointers allowed per tree-node is four (4) (the order of the tree is four).

Once you constructed the B+-tree, delete key values “34” and “23” (one after the other). Show all your work for both insertions and deletions.

Answer:

Just use the insertion and deletion algorithms for the B+-tree.