University of Waterloo
Midterm Examination #2

Spring, 2007

Student Name: ______________________________

Student ID Number: ______________________________

Course Abbreviation and Number	CS448/648
Course Title	Database Systems Implementation
Sections	01 (09:30)
Instructor	K. Salem

Date of Exam	July 11, 2007
Time Period	09:30-10:20
Duration of Exam	50 minutes
Number of Exam Pages (including this cover sheet)	5 pages
Exam Type	Closed Book
Additional Materials Allowed	None

Question 1: (10 marks)	Question 2: (12 marks)	Question 3: (12 marks)	Question 4: (6 marks)
Total: (marks)			

1. (10 total marks)

For both parts of this question, use the notation $r_i[x]$ to refer to a read by transaction T_i on object x, $w_i[x]$ to refer to a write of x by transaction T_i, and c_i to refer to the commit of T_i.

a. (5 marks)
Give a *simple* ($\leq$ 10 operations) example of a transaction history that is (conflict) serializable but not recoverable.

b. (5 marks)
Give a *simple* ($\leq$ 10 operations) example of transaction history that is strict but not (conflict) serializable.

2. (12 total marks)

Consider a database containing a relation R, which has T tuples and occupies B blocks of space on the disk. There are two B+-tree indexes defined on R: a clustered index on attribute $R.a$ and an unclustered index on attribute $R.b$. $R.a$ and $R.b$ are uncorrelated. The following query against R is being executed by the database management system:

```
select * from R order by R.b
```

a. (6 marks)
Suppose that the database system's query plan involves reading R using a sequential scan and then sorting the relation on attribute $R.b$ using an external merge sort. Assume that merge sort operator has sufficient memory to hold only M blocks of data, where $M < B$. What is the I/O cost of this plan, i.e., how many disk block I/O operations will be required to execute the plan? Include both the cost of sequential scan and the cost of the sort.

b. (6 marks)
Suppose instead that the database system's plan is to read R's tuples in sorted order using the unclustered index on $R.b$. Estimate the I/O cost of this plan. Assume that the system has a buffer cache with room for M disk blocks ($M < B$), and that the buffer cache initially contains M of the B blocks of relation R.

3. (12 total marks)

Consider the following query, which is a simplified version of one of the queries from the TPC-H benchmark

```
SELECT C.custkey, SUM(L.extendedprice)
FROM customer C, order O, lineitem L, nation N
WHERE C.custkey = O.custkey
  AND L.orderkey = O.orderkey
  AND O.orderdate > DATE('1995-01-01')
  AND L.returnflag = 'R'
  AND C.nationkey = N.nationkey
GROUP BY C.custkey
ORDER BY C.custkey
```

a. (4 marks)

List all of the “interesting orders” that should be considered by a dynamic programming query optimizer as it generates a plan for this query.

b. (8 marks)

The table below lists some plans that could be considered by a dynamic programming query optimizer as it tries to generate plans for joining the `customer` and `order` tables in the query shown above. Suppose that the values in the second column of the table represent the optimizer's cost estimates for these plans. In the third column of the table, indicate the order, if any, in which the tuples will be produced by this plan. For example, if you believe that the plan will produce output that is ordered on `O.orderdate`, write “`O.orderdate`” in the third column of that plan's row. If the output tuples will not be sorted on any attribute, write “NONE”. In the fourth column of the table, indicate whether the dynamic programming optimizer would keep the plan or would prune it. If you think that the optimizer would keep the plan, write “KEEP”. If you think that the optimizer would prune the plan, write “PRUNE”.

Plan	Cost	Output Order	KEEP/PRUNE
Merge join. One join input is an Index scan of `customer` using a clustered index on `C.custkey`. The other input is a sequential scan of `order`, which is then sorted on `O.custkey` using an external merge sort.	1000		
Index nested loop join. Outer input is an index scan on `customer` using a clustered index on `C.custkey`. Inner input is from an unclustered index on `orders` on `O.custkey`.	800		
Index nested loop join. Outer input is a sequential scan of `customer`. Inner input is from an unclustered index on `orders` on `O.custkey`.	2000		
Index nested loop join. Outer input is a sequential scan of `orders`. Inner input is from a clustered index on `customer` on `C.custkey`.	300		

4. (6 total marks)

a. (4 marks)
Briefly describe one drawback of hash joins relative to nested loop joins. Also, briefly describe one drawback of sort-merge joins relative to nested loop joins.

b. (2 marks)
What does it mean to say that a query execution plan is *pipelined*? Briefly define this term.