CS 631 / IT 603: ITDMBS: End Semester Exam, 22 Nov 2005

Duration: 3 Hrs Answer Part A on the answer sheet itself, Part B in the answer book

1. Part A: Short answers

   (a) Give instances of relations to show that each of the following pairs of expressions are not equivalent:

      i. $\Pi_A(R - S)$ and $\Pi_A(R) - \Pi_A(S)$. ...3

      ii. $\sigma_\theta(E_1 \supset\!\bowtie E_2)$ and $E_1 \supset\!\bowtie \sigma_\theta(E_2)$, where θ uses only attributes from E_2. ...3

   (b) Suppose I have a query of the form

```
select * from r, s
where r.A=s.B
order by r.A stop after 10
```

   which fetches only the first 10 rows of the join result. Under what conditions can the condition "order by r.A stop after 10" be pushed down onto relation r? ...3

   (c) What is physiological logging? ...2

   (d) Give a small example to show why the "read any copy, write all available copies" algorithm does not work correctly. ...3

   (e) Assuming an appropriate DTD, what exactly does the path expression $/a/b//c$ return, and in what order? ...3

(f) Explain very briefly how nodes can be numbered to efficiently check if a node ci is a descendant of node bj. ...3

Total Marks = 20

Part B

2. Halloween problem: Update queries that involve a selection on an updated column (e.g. give a 10 percent raise to all employees whose salary is $\geq$ 100,000) must be handled carefully if the update is done while finding qualifying employees.

   (a) Explain what problem could arise if updates are done while finding qualifying tuples. ...3

   (b) Give a simple (but perhaps expensive) way of avoiding the above problem. ...3

   (c) An alternative way of handling the problem is to filter out plans that have this problem. Given an update query

   ```
   update r set A = E1
   where C1
   ```

   where E1 is an expression and C1 a predicate, and a plan P for this query which does an index scan, and updates tuples during the scan, give conditions under which plan P can be inferred to be safe. ...3

3. Join extensions.

   (a) Suppose you have two relations $r(A, B)$ and $s(C, D)$, with attributes A and C of type integer. Suppose you wish to match each r tuple to the 5 s tuples with whose C values are closest to the A value of the r tuple (this is a simple form of nearest neighbour join). Suggest how to modify merge join to perform this join. ...3

   (b) Now suppose that attributes A and C above are of type "point", i.e. they have two sub-attributes $A.X, A.Y$ and $C.X, C.Y$ respectively, and the distance measure is the Euclidean distance.

   How do you perform the same 5-nearest neighbours join using an R-tree on $s.C$ *assuming for simplicity that r has only one tuple.*

   As part of your answer, briefly explain the key optimizations applied to this problem in the spatial join paper. ...4

   (c) Suppose now that there are many tuples in r; assume also that r has an R-tree index on $r.A$. If we use the same algorithm as in the previous part, in what order should we consider r tuples to get more buffer hits when accessing the R-tree on $s.C$? Explain your answer briefly. ...2

4. Aggregate materialization:

   (a) Suppose you have a relation *sales(date, amount)* and wish to quickly answer queries that require the sum of sales between any two specified dates (i.e. the query gives the date range, which is not known ahead of time). Suggest what information you can precompute, to answer such queries very fast (using a single subtraction operation), without having to scan the entire relation, but at the same time not causing an excessive storage overhead. Assume the data never gets updated. ...3

   (b) Now suppose you wish to support updates/inserts/deletes on the *sales* relation. What will be the cost of an update using your solution above? ...1

   (c) Suggest how to modify a B-tree by storing extra information at internal nodes, such that range queries as above can be efficiently answered, but at the same time inserts/updates/deletes can also be efficiently handled. ...5

5. The multi-version 2PL algorithm (used for example in Oracle) treats read-only transactions specially.

   (a) Explain briefly how read-only transactions are handled in this protocol. ...3.

   (b) A transaction that has some read queries initially followed by updates should be treated as an update transaction. But Oracle simply treats every read query as if it is part of a read-only transaction. Explain using a small example why doing so may result in a non-serializable execution. ...4

   (c) Why might have motivated Oracle to do such a dangerous thing? (Stupidity or craziness are not valid answers!) ...2

   Note: To save programmers from this danger, Oracle has a special syntax "select for update" to indicate that the query should not be treated as a read-only query.

6. Concurrency control on materialized views.

   (a) Updates to materialized views must be performed as part of the transaction that updates the base relations. Explain why serializability is affected otherwise. ...4

   (b) Suppose there is a relation $r(A, B)$ and a materialized view v defined as $\mathcal{G}_{\text{sum}(B)}(r)$. Suppose also that there are lots of transactions that insert tuples into r. What performance problem can concurrency control on the materialized view v cause? ...2

   (c) A solution suggested for the above problem is to allow early lock release on the materialized view.

      i. How should undo be done if locks on v are released early? ...2

      ii. A lock mode I (for increment) has been proposed to solve the above problem. Updates to the materialized view get an I lock (and hold it up to end of transaction) instead of getting an X lock; but they additionally get a short term X lock (latch) when actually updating v and release the latch immediately, even before end of transaction. Give a compatibility matrix of S, X and I lock modes, to ensure serializability. Explain your answer. ...4

7. Aries:

   (a) Logs are usually kept on a log disk. Suppose the log space is nearly full when an abort/crash occurs. What danger does the Aries approach of logging during recovery pose? ...3

   (b) Old log records can be deleted when no longer required for recovery (they may be archived for recovery from a dump, but ignore archiving for now). When using the Aries algorithm, what can you do to ensure that as much of the log is freed up as possible (without doing anything to active transactions). Explain your answer. ...3

   (c) A long-running transaction can make it difficult to recover log space. Explain why. ...3

   (d) Ignoring logical undo overheads, suggest how to compute how much log space will be required to abort a given transaction. ...3

Total Marks = 80