CS632 Midsem, Spring 2009, Feb 21, 2009, 2.30-4.30 PM

Total Marks: 60

NOTE: Answer all subparts of a question together, do not split them up.

1. BANKS

   The original backward expanding search algorithm uses an iterator to generate nodes in order of distance from a given node (containing a keyword). In contrast the single-iterator version of backward expanding search has a single iterator per keyword.

   (a) Briefly, what is the effect, in terms of answers generated, of having a single iterator per keyword, instead of one per node containing the key word? ...4

   (b) Suppose the edge score of an answer tree rooted at node r is $f(d(r,1), d(r,2), .., d(r,m))$, where $d(n,i)$ is the distance from node n to keyword i, and the query has m keywords. For example, f could add up these distances; unlike the Bhalotia et al paper, we assume we are not taking the inverse of the score. We assume that f is monotone.

   Suppose also that for each keyword i, we have a bound m_i such that all nodes at distance $\leq m_i$ from keyword i have already been explored.

   i. Given a node for which we know the shortest path to some keywords, but not all, explain how to compute a lower bound the edge score of any possible answer rooted at that node. ...3

   ii. Give a bound on the score of any node that has not yet been explored. ...3

2. The MQO paper mentions that the exhaustive algorithm explores every subset of nodes in the DAG for materialization, and therefore the cost is doubly exponential in the size of the original query.

   (a) Explain how this cost is derived (assume for simplicity that the only operation is join). ...2

   (b) In fact, we don't really have to consider every subset of nodes, and the exhaustive algorithm is cheaper than claimed. What is the size of the largest subset we need to consider? Explain why, and what is the resultant complexity? ...4

   (c) Consider the definition of degree of sharing $E[x][z]$. The paper claims that if $E[root][z]$ is greater than 1 then z is sharable. Explain why by outlining how to get a plan with two occurrences of z. ...4

3. Subqueries: Give two different examples of SQL queries where the max1row operator can be removed from the Apply rewriting, based on static analysis. ...5

4. Eddies: Consider a query plan consisting of only selections, without any joins. If each operator in the plan is running on a separate processor, which of the policies described in the eddies paper (naive and lottery) do you think would work best? Explain why. ...5

5. Progressive query optimization: The performance study in the POP paper shows that reoptimization can occur when a plan execution is nearly complete, when it is best to carry on to completion. The problem is, how do you estimate how much more time the current plan execution will take?

   Consider the special case of evaluating $\sigma_\theta(R) \bowtie S$, where the join algorithm chosen is indexed NL, with S as the inner relation, and σ_θ is being evaluated on a relation scan of R. At any given point in the evaluation, suggest how to estimate how much more time the join will take? You can assume that the selectivity of θ is independent of the order in which tuples are retrieved from R. ...10

6. Batched Bindings. Provide the missing parts in the batched form of the following procedure. State any assumptions you make. ...10

```
procedure update-addr(custid, addr) {
    int lcount;

    SELECT count(cust-id) INTO lcount
    FROM custaddr
    WHERE cust-id=custid;

    if (lcount == 0)
        INSERT INTO custaddr VALUES (custid, addr);
    else
        UPDATE custaddr SET cust-addr = addr
        WHERE cust-id = custid;
}
```

Batched form:

```
update-addr-batched(Table(custid, addr) params) {
    Table t(custid, addr, lcount, cv);

    for each r in params {
        Record r;
        r.custid = custid;
        r.addr   = addr;
        t.add(r);
    }

    MERGE INTO t USING  <.. fill this in ..>  AS bres
               ON t.custid=bres.custid
               WHEN MATCHED THEN UPDATE SET t.lcout = bres.lcount;

    for each r in t {
        r.cv = (r.lcount == 0);
    }

    INSERT INTO custaddr <.. fill this in ..>

    MERGE INTO custaddr <.. fill this in ..>
}
```

7. Consider the diverse query results paper. The diversity measure in this paper assumes a strict ordering of attributes. Consider the special case where attributes A and B have the same priority, and no other attributes contribute to diversity. Assume also that there is a similarity function $sim_A(v1, v2)$ defining the similarity of two values for A, and likewise for B

   (a) Give at least two ways of defining similarity of tuples, giving equal weightage to A and B. (Hint: consider geometric distances) ...4

   (b) Given a (fully materialized) set of tuples, suggest a heuristic for finding a diverse set. Don't worry if the heuristic is not optimal, just provide something reasonable. ...6

Total Marks = 60