Name: AndrewId:

15-415 Final Exam
Fall 2011 (Kesden)

Notes:

- Questions differing in length of answer anticipated; shorter (but fully correct) answers are generally better than longer ones
- Partial credit might be helped by exposing more of your thought process, rather than less
- Call, any hour, with any questions: 412-268-1590 (Greg's cell)

XML/Semi-Structured Data

1. Given an XML file, we could choose to use either DOM or SAX to process it. What is the difference in their models? When would we select each?

2. Is XPath generally based on DOM or SAX? What about XQuery? Why?

“Big Data” Distributed Databases

3. Traditional Relational databases model data using an abstraction known as a *relation* or, more informally as a *table*. How do HBase and Cassandra model their data?

4. Based on the data model, what type of operation common in SQL databases do HBase and Cassandra assume will be relatively rare (*Hint:* Consider what they don’t do well that SQL does)?

5. Based on the data model, what type of operation do HBase and Cassandra assume will be relatively frequent (*Hint:* Consider what they do very effectively)?

6. Why are “Joins” considered painful operations in “Big Data” Databases such as HBase and Cassandra?

7. What is the difference in the underlying implementation of HBase and Cassandra? Does this affect their performance characteristics? If so, in broad strokes, distinguish when they might be expected to perform very differently. If not, explain how the different implementations achieve similar performance.

Query Optimization

8. Consider the following query (yes, substr does what you think it does):

```
select * from r,s
where substr(r.lastName,10) = substring(s.lastName,10);
```

   Describe two different query plans that might be generated by the compiler to execute this query. In other words, with respect to the relational algebra portion of the query (not the string munging), please describe solutions to the problem using two different data structures or ways of solving the problem generally available within a database, e.g. hashing, indexing, merging, etc.

9. Given some relation, *r*, with some attributes *size* and *name*, please explain how a histogram can be used to select those *names* associated with sizes greater than some arbitrary *x*.

10. When are the stats stored within the catalogue updated? Why is this strategy generally employed?

Functional Dependencies, Normal Forms and Normalization

11. You are given a table R(A, B, C) with the following FDs:

    AB -> C
    B -> C.

    Can we deduce from these that A -> C holds? If yes, explain your reasoning. If no, give a counter-example with 3 tuples or less.

12. Consider the following 4 functional dependencies and *Table 1*, which illustrates relation R(A, B, C, D, E) at a particular point in time. Assume that *Relation R* can change over time through insertions, deletions and updates, but that the schema will remain unchanged.

Functional dependencies:

12.1 A -> CD
12.2 AC -> B
12.3 AB -> CD
12.4 BD -> CE

tuple-id	A	B	C	D	E
T1	4	7	22	48	1
T2	5	6	22	49	7
T3	5	7	24	53	9
T4	3	2	10	23	8
T5	6	0	12	30	3
T6	3	2	10	23	0
T7	2	3	10	22	9
T8	5	1	12	29	1
T9	5	6	22	49	2

Table 1

Will the above four functional dependencies hold true for this *and any future* instance of this relation? For each of the functional dependencies, please choose one of (A), (B), (C) and (D) --and-- support your answer as described:

(A) Always holds (now and in the future) – explain your reasoning
(B) Does not hold (now) – give at least one tuple-id that violates the functional dependency
(C) Holds now, by chance (Not guaranteed to hold in the future) – This is here only to prevent confusion. If you want to choose this, then choose (D). Really.
(D) Need to perform a SQL query to find out – provide the SQL

13. Consider the functional dependencies set S = {AB -> C, C -> B, C -> D}.

13.1 Given *S*, is the relation R_1 (A, B, C) in 3NF? If yes, explain how you know. If not, specify at least one functional dependency which violates 3NF.

13.2 Given *S*, consider relation R_2(A, B, C, D). Decompose it into a collection of BCNF relations. Give these relations.