CS411 Database Systems

Spring 2009, Prof. Chang

Department of Computer Science
University of Illinois at Urbana-Champaign

Final Examination

May 8, 2009

Time Limit: 180 minutes

Problem 1 (*32 points*) Misc. Concepts

(1) False

If two relations are both in BCNF, their join must also be in BCNF.

(2) True

Transaction management consists of two main functional components: *concurrency control* and *failure recovery*.

(3) True

SQL Injection attacks a Web site by manipulating the user input to cause harmful SQL commands to be executed at the backend database.

(4) False

Relational algebra was invented to formalize the underlying operations of the SQL language.

(5) True

When translating an E-R diagram to the relational model, there are multiple ways to translate a sub-class relationship.

(6) False

With respect to a set of integers, there exists a unique structure to index them in a B+ tree.

(7) True

In determining a query plan involving joins, by focusing on only left-deep join trees, we are *not* guaranteed to generate the optimal query plan.

(8) True

Regardless of UNDO or REDO, a logging system must write the corresponding log entry *before* any update of database values on disk.

(9) False

In cost-based optimization, dynamic programming is a technique that helps us to estimate the cost of a query plan.

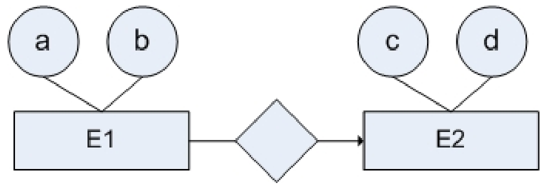

Figure 1: E-R diagram

(10) True

For the same operations (*e.g.*, sorting, grouping), two-pass algorithms generally require less memory buffer than their corresponding one-pass algorithms.

(11) True

An important requirement for database indexing is the ability of the index to maintain an appropriate structure as the database changes over time.

(12) False

The *Explain Plan* facility in Oracle SQL shows the execution plan and its actual cost by executing the plan.

(13) False

We can use *Hints* to tell Oracle query optimizer what to do– Further, a Hint can adapt to the changes of a database to lead to good query plans over time.

(14) True

PostGIS implements the OpenGIS SFSQL and thus provides geo-spatial data types such as *Point* and *Polygon* as well as functions like *Distance* and *Within*.

(15) False

A typical "denial of service" attack is to attack a database server by sending complex SQL queries that would cause the underlying query optimizer to hang and thus result in the intended denial.

(16) False

Schema normalization is a technique that will lead to more efficient query processing.

Problem 2 (*16 points*) Short Answer Questions

(1) Yes, because it enables index join on $S.a$

(2) No, an RA can describe different execution procedures.

(3) A,B: (1,3),(2,3),(3,1) and (A,C,D):(1,2,2),(2,2,4),(3,3,6),(3,1,6). (A,D) and (ABC) are also right.

(4) Refer to Figure 1 .

(5) In a left-deep join tree such as $(((R_0 \bowtie R_1) \bowtie R_2)... \bowtie R_{n-1})$ each R_i $(i = 0 \cdots n-1)$ has $(n-i)$ choices. Thus the final answer is: $n(n-1)..1 = n!$.

(6) Isolation: each transaction must appear to be executed as if no other transaction is executing at the same time.

(7) Every extension will certainly alleviate over utilized buckets. Other correct advantages are acceptable as well.

(8) 1. Since a is a primary key and 100 is a possible value for a.

Problem 3 (*10 points*) Schema Decomposition

(a) AB, BC, BDE

(b) Yes, in 3NF. All the attributes on the right-hand-side are part of key.

(Notice, we grade this problem based on your answer to part (a).)

(c) No. $DE \rightarrow A$ violates BCNF. We decompose based on this, and get $R_1 = ADE$, and $R_2 = BCDE$.

R_2 is still not in BCNF, because of $CD \rightarrow E$. Further decompose based on this FD, will give us $R_{21} = CDE$, and $R_{22} = BCD$.

So the final decomposed relations are: $R_1 = ADE$, $R_{21} = CDE$, and $R_{22} = BCD$.

(Notice, there are other ways to decompose. Full credit as long as the decomposition is right.)

Problem 4 (*10 points*) Query Languages

(a) $\pi_{name}((\sigma_{color=Red}Product) \bowtie Order \bowtie Customer)$

(b) SELECT SUM(quantity)

FROM Order, Customer

WHERE Order.cid = Customer.cid AND Customer.age > 70

(c) SELECT pid

FROM Order

GROUPBY pid

HAVING SUM(quantity) ≥ ALL (SELECT SUM(quantity) FROM Order GROUPBY pid)

Problem 5 (*10 points*) Indexing: B+ tree

(a) See Figure 2.

Points:

- Key 5 inserted (1 point)
- Appropriate leftmost leaf node split (1 point. Split (1,2) or (1,2,3) are both acceptable.)
- Appropriate index keys (2 points)

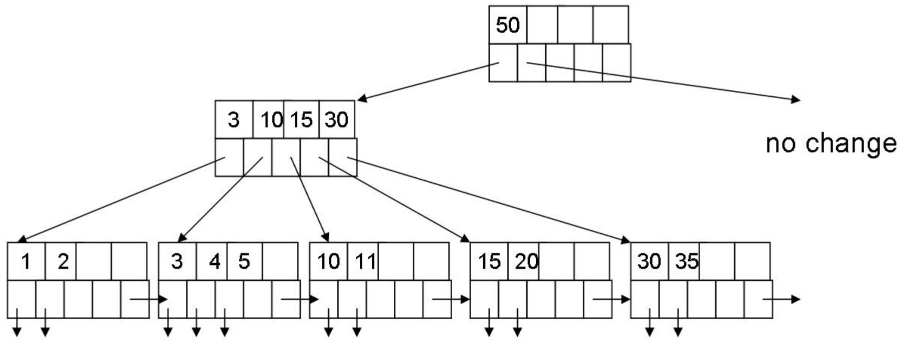

Figure 2: After Inserting Key 5

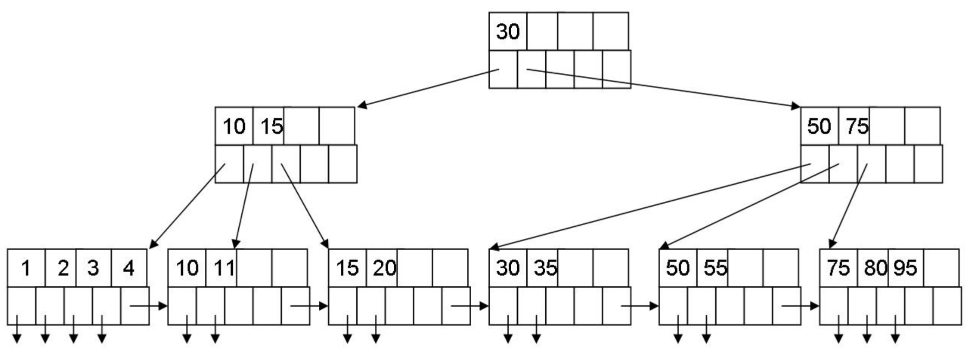

Figure 3: After Deleting Key 90

(b) See Figure 3.

Points:

- Key 90 deleted (1 point)
- Right leaf nodes (75,80) and (95) must be merged (1 point)
- Internal node (75,80) must be adjusted (1 point)
- Appropriate index keys (1 point)

(c)
1. Start at the root and follow left pointer.
2. Since 40 is greater than 30, follow the rightmost pointer from the node (10,15,30).
3. At the leaf nodes, continue to move right until it reaches the rightmost leaf node.

Problem 6 (*12 points*) Query Processing

(a) 3(B(R)+B(S)) = 4500 blocks, or 5(B(R)+B(S)) = 7500 blocks

(b) 3(B(R)+B(S)) = 4500 blocks

(c) We effectively have to perform two nested-loop joins of 500 and 250 blocks, respectively, using 101 blocks of memory. Such a join takes 250 + 500*250/100 = 1500 disk I/O's, so two of them takes 3000. To this number, we must add the cost of sorting the two relations, which takes four disk I/O's per block of the relations, or another 6000. The total disk I/O cost is thus 9000.

Points:

- Algorithm (3 points)
- Cost (3 points)

Problem 7 (*17 points*) Query Optimization

(a) Assume two relations $R(A,B) = (1,2)$ and $S(A,B) = (1,4)$. We have:
$\pi_a(R-S) = (1)$, but $\pi_a R - \pi_a S = \emptyset$

(b) (1) Use an index-scan using the nonclustering index on c. Since V(R,c) = 5000, only one tuple should be retrieved. Filter the retrieved tuple for a=1 and b=3. The expected disk I/O cost is 1.

(2) Use an index-scan using the nonclustering index on b. Since V(R,b) = 1000, 5 blocks should be retrieved. Filter the retrieved blocks for a=1 and c<3. The expected disk I/O cost is 5.

Points:

- Query plan (2 points)
- Cost (2 points)

(c) Iteration 1, queries with 1 table: Author, Write, Book, Publisher. Totally 4.

Iteration 2, queries with 2 tables: AW, WB, BP. Totally 3.

Iteration 3, queries with 3 tables: AWB, WBP. Totally 2.

Iteration 4, queries with 4 tables: AWBP. Totally 1.

Thus, the total number of sub-queries will be 10.

Points:

- The correct number of sub-queries for each iteration (1.5 points)

<u>Problem 8</u> (*13 points*) Failure Recovery

(a)
1. Between logID 11 and 12, $\langle STARTCKPT(T3, T4)\rangle$
2. Between logID 15 and 16, $\langle ENDCKPT\rangle$

Points:

- START CKPT (1 point)
- END CKPT (1 point)

(b)
1. Log portion needs to be inspected: from crash point until START CKPT
2. Transactions need to be undone: T5

Points:

- Appropriate inspected log portion (2 point)
- Correct transactions to be undone (1 point)

(c)
1. Between logID 4 and 5, $\langle STARTCKPT(T1, T2)\rangle$
2. After START CKPT, $\langle ENDCKPT\rangle$

Points:

- START CKPT (1 point)
- END CKPT (1 point)

(d)
1. Log portion needs to be inspected: from logID 1 to crash point
2. Transactions need to be undone: T1,T2,T3,T4

Points:

- Appropriate inspected log portion (2 point)
- Correct transactions to be undone (1 point)

(e)
1. Log portion needs to be inspected: from logID 1 to crash point
2. Transactions need to be undone: T1

Points:

- Appropriate inspected log portion (2 point)
- Correct transactions to be undone (1 point)