CS 564 Final Exam Fall 2016
Answers

A: RELATIONAL ALGEBRA, SQL & NORMALIZATION **[25pts]**

I. **[5pts]** Consider a relational schema with two relations, $R(A, B)$ and $S(B, C, D)$, and the following query in Relational Algebra:

$$q = \pi_A(R \bowtie \sigma_{C=1}(S))$$

Which of the following queries are equivalent to q? Clearly **circle** all the correct options and only the correct options.

(a) $\pi_A(\sigma_{C=1}(R \bowtie S))$

(b) $\pi_A(R) \bowtie \sigma_{C=1}(S)$

(c) $\pi_A(\pi_{A,B}(R) \bowtie \pi_B(\sigma_{C=1}(S)))$

(d) $\pi_A((R \bowtie S) - (R \bowtie \sigma_{C\neq 1}(S)))$

(e) $\pi_A(R \bowtie S) - \pi_A(R \bowtie \sigma_{C\neq 1}(S))$

ANSWER: (a), (c), (d)

II. **[5pts]** Consider a relation $R(A, B, C, D, E)$ with the following functional dependencies:

$$A \to B, C \qquad\qquad C \to D$$

Which of the following decompositions are lossless-join? Clearly **circle** all the correct options and only the correct options.

(a) $R_1(A, B, C, D)$ & $R_2(D, E)$

(b) $R_1(A, B)$ & $R_2(C, D, E)$

(c) $R_1(A, B, C)$ & $R_2(A, D, E)$

(d) $R_1(A, B, C, E)$ and $R_2(C, D)$

(e) $R_1(A, B, C)$, $R_2(C, D)$ and $R_3(A, E)$

ANSWER: (c), (d), (e)

III. **[15pts]** Consider the following schema used to store reservations in a restaurant:

Customer (cid, firstname, lastname, email)

Table (tid, size)

Booking (bid, date, cid, tid, partySize)
Booking.cid is a foreign key referring to Customer.cid.
Booking.tid is a foreign key referring to Table.tid.

For the following two questions, write a **single** SQL query (it should be one statement):

1. **[7pts]** Get the emails of the customers who have booked a table with size at least 10.

   ANSWER:

2. **[8pts]** For each customer get the first name, last name, and cumulative party size for bookings during the year 2015. Output only the **top 20** customers, in **decreasing order** of the cumulative party size. You can assume that the date is stored in the form "MM-DD-YYYY".

   ANSWER:

B: STORAGE AND INDEXING **[25pts]**

I. **[10pts]** For the following questions, **clearly circle** True or False.

1. The access time when reading a block from disk is dominated by the transfer time.

   FALSE

2. The buffer manager writes a page to the disk as soon as the pin count becomes zero.

   FALSE

3. The free space in a page where we store fixed-length records using a packed organization is contiguous.

   TRUE

4. Given a relation $R(A, B, C)$, it is not possible to create two clustered indexes with search keys A and B respectively.

   TRUE

5. Creating a new index always improves the performance of the DBMS.

   FALSE

II. **[5pts]** Consider the relation $R(A, B, C, D, E)$ and suppose we want to evaluate the following predicate:

$$(A = 10) \text{ AND } (B > 30) \text{ AND } (C = 20)$$

Which of the following indexes matches the predicate? Clearly **circle** all the correct options and only the correct options.

(a) hash index on (A)

(b) hash index on (A, B)

(c) hash index on (C, A)

(d) B+ tree index on (D, C)

(e) B+ tree index on (C, D)

ANSWER: (a), (c), (e)

III. **[10pts]** Consider a B+ tree that has order $d = 2$ (so every node can hold at most 4 search key values) and is initially empty. Draw the B+ tree after the following values have been inserted in order in the B+ tree:

$$10, 11, 12, 13, 14, 1, 2, 3, 20, 21, 30, 31, 40, 50.$$

ANSWER:

C: OPERATOR IMPLEMENTATION **[25pts]**

II. **[10pts]** We are given a relation R that we want to sort using the general external sort algorithm. Assume that we use replacement sort during the initial pass, and we create sorted runs of size $2B$. In our case, the buffer pool has size $B = 11$. Compute the maximum size of R (in pages) that can be sorted in 2 and 3 passes respectively (you need to compute two numbers). Explain your answer in detail.

ANSWER: For 2 passes, we must have $N/2B \leq B$, which gets us $N \leq 220$ pages. For 3 passes we get $N/2B \leq (B-1)^2$, so $N \leq 2200$.

II. **[15pts]** We are given two relations: R with 1,000 pages and S with 2,000 pages. We are performing a key-foreign key join of R and S wherein S has the foreign key attribute. What is the **smallest size** B of the buffer pool for which the block nested loop join has smaller I/O cost than the hash join? Assume that the fudge factor for constructing a hash table is $f = 1.4$. Explain your answer in detail.

ANSWER: The cost of the BNLJ is $1,000 + k \cdot 2,000$, where k is the number of passes. The cost of the hash join if data fits in memory is $9,000$. So for the BNLJ to be faster, $k \leq 4$, which gets us $B \geq 252$. Notice that for this value of B, hash join can indeed be executed in 2 passes.

D: QUERY OPTIMIZATION **[15pts]**

Consider the following database schema:

Company (<u>cid</u>, cname, location)

Employee (<u>eid</u>, ename, age, cid)
Employee.cid is a foreign key referring to Company.cid

Company has 1,000 tuples, and each record is 50 bytes long. Employee has 200,000 tuples, and each record is 20 bytes long. Each page can hold 5,000 bytes. The buffer pool has 102 frames. Assume that there are 500 distinct values for the attribute "location" and that its distribution is uniform.

Consider the following SQL query:

```
SELECT  DISTINCT E.cname
FROM    Employee E, Company C
WHERE   E.cid = C.cid
AND     C.location = "Madison" ;
```

Propose a physical plan that evaluates the above query **without materializing** any intermediate result. Compute its total I/O cost and briefly explain each component of the total

cost. Ignore the cost of writing the output to disk.

ANSWER: A simple plan is to scan the Company relation (10 pages), do the selection, and then do a BNLJ or HJ with Company. This join can be done since the intermediate result is very small. We need to scan Employee once for the join (800 pages). Then we do a projection and duplicate elimination in memory. The total cost is 810 pages.

E: TRANSACTION MANAGEMENT **[10pts]**

I. **[4pts]** For the following questions, **clearly circle** either True or False.

1. Suppose transaction T_1 has an S lock on tuple t. If transaction T_2 attempts to get an X lock on t, it will be successful.

   FALSE

2. A FORCE buffer pool policy guarantees that all transactions will satisfy durability.

   TRUE

II. **[6pts]** Consider the following two transactions:

$$T1 : W(A), R(C), W(A), Commit$$
$$T2 : R(B), R(A), Commit$$

and the following interleaved schedule of the two transactions:

$$W_{T1}(A), R_{T2}(B), R_{T1}(C), R_{T2}(A), W_{T1}(A), Commit_{T1}, Commit_{T2}$$

Can this schedule be the result of a non-strict 2-Phase Locking protocol? Explain your answer in detail.

ANSWER: No, it cannot be. T_1 needs to release its exclusive lock on A so that T_2 can read A, but then it cannot acquire the lock to write A again.