Name	**NetID**

CS411: Database Systems **Spring 2016**

Final Exam, May 10

- READ THESE INSTRUCTIONS CAREFULLY BEFORE YOU START. DO NOT turn this page UNTIL the proctor instructs you to.
- First: write your name and NetID at the top of this sheet.
- The exam lasts for 120 minutes, i.e., from 7–9pm.
- You can bring one letter-size, double-sided, hand-written cheat sheet. Printing/scanning references is not allowed, and nor is a magnifying glass.
- We will not answer any questions during the exam. If you need to make any assumptions for any of the questions, please feel free to do so and then clarify the assumption in your answer.
- The examination contains both objective type (True/False) and long answer questions. All questions are compulsory.
- For the objective type questions, **please circle the right answer, and provide a short description justifying your choice**. If you need extra space for any calculations, feel free to use the back side of each page.
- For the long answer questions, please answer in the space provided; if you need more space, feel free to use the back side of each page. You should not need more space, and we will not provide more space, so please use your space wisely. Show all necessary steps as part of your calculation to get partial credit.
- The maximum score you can obtain is $20 + 15 + 15 + 15 + 20 + 15 = 100$.
- You must stop writing when time is called by the proctors.
- **Cheating: No.**

Question	1	2	3	4	5	6	Total
Points							

Objective-Type Questions - 20 points

Please **circle** the right answer, and provide a short 1–2 line description justifying your choice.

1. [2] If X and Y are entity sets, then a relationship between them is always a subset of $X \times Y$.

   True False

2. [2] The theta join is equivalent to a natural join followed by the selection operation.

   True False

3. [2] Since two-attribute relations are always in BCNF, we can break any relation into two-attribute relations to obtain a BCNF decomposition.

   True False

4. [2] For any relation, any of its 3NF decompositions will always preserve its functional dependencies whereas some of its BCNF decompositions may not.

   True False

5. [2] The projection operator is faster in bag semantics than in set semantics.

   True False

6. [2] Extensions in linear hash tables can be costly and disruptive, because after an extension the extended table or the hash table may no longer fit in memory.

   True False

7. [2] For large relations, the duplicate elimination operation (DISTINCT) can be processed in one-pass by maintaining the set of unique values in memory.

   True False

8. [2] Say a relation R(A, B) has T tuples, and has m distinct values of A. Then the size following a selection with condition (A > 3) on R is between 0 and T - m + 1.

   True False

9. [2] In undo-logging, if transaction T modifies X, then the <T,X,V> entry must be written to the log and the log must be flushed to the disk before X is written to disk.

   True False

10. [2] If we perform recovery for the second time in undo logging, we should ensure to not perform the same undo command twice since this may lead to inconsistent data.

   True False

SQL - 15 points

Consider the following relational schema for a Movies database:

- Actor(actor_id, first_name, last_name, gender)
- Acting(actor,movie)
- Movie(movie_id, name, released_date, length, genre)

You can make the following assumptions:

- Gender in the Actor table can be either 'F' or 'M'
- actor and movie in the Acting table are foreign keys to actor_id in the Actor table and movie_id in the Movie table respectively. The two attributes also form the joint Primary Key of the Acting table.

Specify the following queries using SQL:

1. [4] What is the average length of all the action movies?

2. [4] How many actresses does the movie 'The Godfather' have?

3. [7] Consider relations R(A,B) and S(B,C). Are the following two SQL queries equivalent on all databases? If not, give a simple counterexample.

```
select R.A
from R,S
where R.B = S.B and S.C = 4

select R.A
from R
where R.B in (select S.B from S where S.C = 4)
```

Indexing - 15 points

Consider the B+ tree shown in Figure 1, and show the effect of the following operations, **in sequence**.

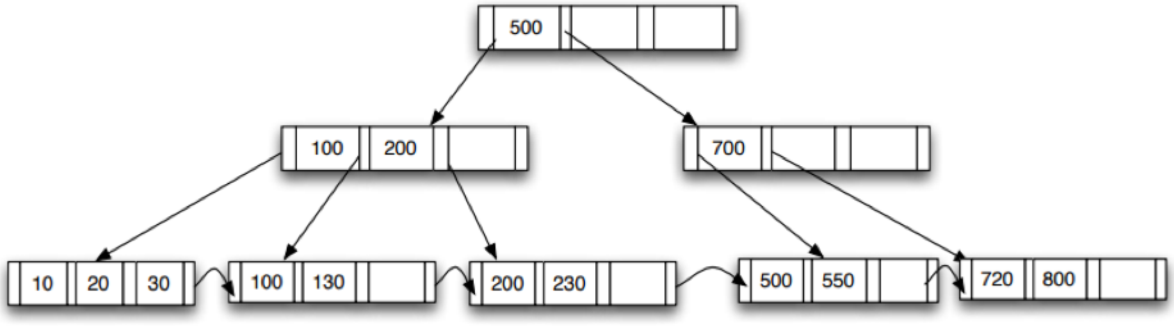

Figure 1: Original B+ tree

1. [5] Insert 80 into the B+ tree shown in Figure 1, and show the updated B+ tree below.

2. [5] Insert 250 to the updated B+ tree from question 1, and show the updated B+ tree below.

3. [5] Insert 280 to the updated B+ tree from question 2, and show the updated B+ tree below.

Query Execution - 15 points

1. Consider three relations R(w,x) , S(x,y) and U(y,z). It is given that B(R) = 1000, B(S) = 2000, and B(U) = 3000. Assume that all the relations are clustered. Imagine that you have completed the Dynamic Programming algorithm for query optimization, and you have identified that the best logical plan is $(R \bowtie S) \bowtie U$. Now, you need to determine the best physical plan: choice of join operators, and pipelining/materialization, taking memory considerations into account.

   For each of the following scenarios, compute the lowest cost (in terms of I/O operations) required to compute $(R \bowtie S) \bowtie U$. In each case, provide a succinct justification by providing details of the physical plan (join algorithm, pipelining/materialization) you would use.

   (a) [5] Infinite memory availability. That is, assume you have no constraints on M at all.

   (b) [5] Highly constrained memory. Specifically, assume that M = 3, so you can hold only 3 blocks in memory at a time.

2. [5] Consider two relations r and s. It is given that the relation r has **more** number of tuples than relation s. For each of the following questions, **circle** all the correct options. Note: more than one may be correct.

   (a) If relation s is treated as the outer table in the block nested loop join, which of the following are true:

      i. cost of the join operation is increased

      ii. number of iterations is reduced

      iii. this approach is more optimal than having relation r as the outer table

      iv. relation s needs to fit in memory

   (b) If relation s is treated as the build relation in hash join, which of the following are true:

      i. cost of the join operation is reduced

      ii. cost of the join operation remains unchanged if both relations can simultaneously fit in memory

      iii. relation r needs to fit in memory

      iv. relation s needs to fit in memory

Query Optimization - 20 points

1. Assume five relations A, B, C, D, E, having only one common attribute l; note that they may have other attributes themselves. Answer the questions below under the following assumptions: "containment of values" and "preservation of value sets".

T(A) = 2000	T(B) = 1000	T(C) = 6000	T(D) = 5000	T(E) = 3000
V(A, l) = 500	V(B, l) = 400	V(C, l) = 3000	V(D, l) = 2500	V(E, l) = 1500

   (a) [5] Estimate the number of tuples returned as a result of the following expression:

$$((A \bowtie B) \bowtie (C \bowtie D)) \bowtie E$$

   Provide details of your calculation. (That is, start with the innermost parenthesis, and work your way out, arguing how you obtained each step.

(b) [5] Estimate the number of tuples returned as a result of the following alternative expression:

$$(A \bowtie (B \bowtie (C \bowtie (D \bowtie E))))$$

Provide details of your calculation. Using the results of (a) and (b), what conclusions can we draw?

2. [10] Consider the following relations and assumptions about the number of blocks in each relation:

   B(A) = 300

   B(B) = 400

   B(C) = 25

   B(D) = 500

   Use the dynamic programing algorithm as taught in the class to determine the optimal plan to join the above relations efficiently. Show your work by completing the following table (each step in the dynamic programming algorithm should be one row):

Subset	Size	Lowest Cost	Lowest-cost Plan
AB			
AC			
AD			
BC			
BD			
CD			
ABC			
ABD			
ACD			
BCD			
ABCD			

   Details: when doing the calculations, you should

   (a) Use the following formula of size estimation as discussed in the class: $B(R1 \bowtie R2) = B(R1) * B(R2) * 0.01$, where R1 and R2 are any two relations.

   (b) Assume that the join operation is not symmetric, i.e., the plan (R1 R2) is not the same as the plan (R2 R1). When choosing between these two plans, select the one that has the smaller table on the left.

Transaction Management - 15 points

Consider the transaction log provided below in Table 1, and answer the questions that follow.

LogID	Log
1	<START T1>
2	<T1, A, 4>
3	<START T2>
4	<START T3>
5	<T3, B, 7>
6	<T2, C, 9>
7	<T3, G, 15>
8	<T2, H, 13>
9	<ABORT T3>
10	<T2, D, 9>
11	<START T4>
12	<T4, J, 13>
13	<T1, K, 12>
14	<COMMIT T1>
15	<T4, E, 3>
16	<T4, F, 9>
17	<COMMIT T4>

Table 1: Transaction log

Note: For questions 1–4, assume the given log sequence is an UNDO log.

1. [2] What is the latest time—i.e., before which LogID—for transaction T1 to output variable A onto disk? Give the LogID for your answer.

2. [2] At the end of the log, what is the value of Variable B?

3. [3] Suppose we want to start check-pointing after LogID 9. Between which two log records would we start check-pointing, and what should the log entry (for check-pointing) look like? At the earliest time possible, between which two log records should we stop check pointing, and what should the log entry look like?

4. [3] Given the check points you have created in Question 3, suppose the system crashes right after LogID 13. Show which transactions/actions (e.g.: <T1, A, 15>) need to be undone and in what order.

Note: For questions 5–6, assume the given log sequence is a REDO log.

5. [2] What does the record, <T4, J, 13> (which is at LogID 12), indicate? Consider the original log from Table 1.

6. [3] Suppose the system crashes right after LogID 15. Show which transactions/actions (e.g.: <T4, J, 13>) need to be redone and in what order. Consider the original log from Table 1.

Computational Complexity - Bonus Section

Show that the class of questions for which some algorithm can provide an answer in polynomial time is equivalent to the class of questions for which an answer can be *verified* in polynomial time. In other words, show that $\mathbf{P} = \mathbf{NP}$.

Just kidding. Please use this page for rough work.