W4111 Introduction to Databases
Fall 2015
Final Exam

Closed Book, 1/2 sheet paper both sides
Duration: 75 minutes

Professor: Eugene Wu
Wednesday, December 9, 2015

Your Name: _____

Your UNI: _____

UNI of person to your left: _____

UNI of person to your right: _____

Problem	Points	Score
1	22	
2	14	
3	14	
4	28	
5	20	
6	0	3.5
Write your UNI in the top right right corner of each page.	2	
TOTAL	100	

Good Luck!

1 (22 pts) Terms and Definitions

1.1 (10 pts) True or False

(2.5 points each) In the following questions, circle True or False and include a 1 line description of why. **Draw a box around your final answer**.

1. Given a database $\mathcal{D}$, it is possible to develop software to automatically identify the functional dependencies using techniques such as correlation analysis. **True/False**

2. A query optimizer such as the Selinger Optimizer finds the query plan with the minimum estimated execution cost. **True/False**

3. A search key is a type of candidate key that is optimized for searching over. **True/False**

4. A table exhibits redundancy when 2 or more rows have the same value for 1 or more attributes in the schema. **True/False**

1.2 (12 pts) Short Answers

(3 points each) In *at most* one short sentence each, answer the following questions as they relate to database management systems. **Draw a box around your final answer**.

1. In database recovery, what ACID transaction property is UNDO designed to ensure?

2. In database recovery, what ACID transaction property is REDO designed to ensure?

3. Can the following sequence of operations and log writes guarantee correct recovery after a crash at any point in time? Why or Why not?

```
1. BEGIN;
2. Read(A)
3. Write(B)
4. Write(A)
5. flush log record for "Write(A)" to disk
6. flush A to disk
7. flush log record for "Write(B)" to disk
8. flush B to disk
9. flush log record for COMMIT to disk
10. COMMIT;
```

4. What is the purpose of a database cursor?

2 (14 points) SQL

Consider the following database schema containing actors, movies, and which actor acted in which movie:

```
CREATE TABLE actors(
  aid  int PRIMARY KEY,
  is_tall bool NOT NULL,
  name text UNIQUE NOT NULL
);

CREATE TABLE movies(
  mid  int PRIMARY KEY,
  name text NOT NULL,
  year int NOT NULL
);

CREATE TABLE actedin(
  actor_id int REFERENCES actors(aid),
  movie_id int REFERENCES movies(mid),
  PRIMARY KEY (actor_id, movie_id)
);
```

Write the SQL statements to answer the following statements. You may use temporary tables. **Draw a box around your final answer, that is what we will grade!**

1. (6 pts) Find the names of the movies whose cast contains all tall actors. For example, if there are a total of 5 tall actors, and movie M's cast has 10 actors, and all 5 tall actors are in M's cast, then M should be in the result set.

2. (8 pts) Find the ids of all unique pairs of actors that have acted together in two different movies where at least one of the actors is tall. We consider (A, B) to be the same as (B, A), so if (A, B) is in the result, then (B, A) should NOT be in the result.

3 (14 points) Functional Dependencies

Consider the table `ABCDEF`, where each letter represents an attribute in the schema. Let the table have the following functional dependencies

$$A \rightarrow BCDEF \tag{1}$$
$$B \rightarrow CD \tag{2}$$
$$E \rightarrow DF \tag{3}$$
$$FB \rightarrow D \tag{4}$$

Draw a box around your final answer – that is what we will grade!

1. (6 pts) Write one (of possibly many) BCNF decompositions of the table `ABCDEF` given the functional dependencies above.

2. (2 pts) ***Briefly*** describe whether the decomposition exhibits any anomalies, and if so, list them. (This answer is contingent on a correct decomposition above).

3. (6 pts) Write the minimal cover of the above set of functional dependencies.

4 (28 pts) Statistics and Access Paths

Given the tables `S(sid, name, age)` and `T(tid, topic)`, where there are secondary B+Tree indexes on `S(sid)` and `T(tid)`, and a secondary B+Tree index on `S(age)`. Also consider the following statistics about the tables.

```
NCARD(S)      = 500
ICARD(S.sid)  = 100
ICARD(S.age)  = 50
minmax(S.age) = [0, 50)   -- from 0 to 49, inclusive

NCARD(T)      = 1000
ICARD(T.tid)  = 100

Page size     = 1020 bytes
Pointer size  = 10 bytes
Key size      = 10 bytes
Tuple size    = 50 bytes  -- both S and T tuples have same size
Fill Factor   = 0.5
default selectivity: 0.1
```

Assume that leaf pages store (key, pointer) pairs. Assume that a tuple is fully stored within its page (and not split across multiple pages). The following questions refer to the query below:

```
SELECT *
FROM S JOIN T ON S.sid = T.tid
WHERE age < 5
```

Draw a box around your answer, that is what we will grade. Clearly list your assumptions, if any.

1. (3 pts) What is the selectivity of the `age < 5` predicate?

2. (3 pts) What is the expected cardinality of the query?

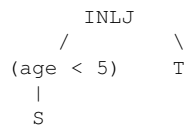
3. (6 pts) What is the best access method for S while taking into account the predicate `age < 5`? How many disk IOs does it cost?

4. (16 pts) The following questions ask you to estimate the cost of different physical query plans (note, this is not Selinger's `bestjoin()` estimate, but the normal plan cost estimates). We assume the following as part of the cost estimates:

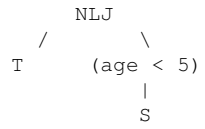
   - If the selection operator is on the outer table, it is pipelined to the parent join operator.
   - If the selection operator is on the inner table, we assume that the selection operator is executed to completion and its results are stored in a temporary table. The temporary table is *then* used as the inner table in the join. In this case, take into account the cost of writing the temporary to disk, and reading it in the parent operator.

   Draw a box around your final cost estimate. Clearly state any assumptions that you make.

   (a) (8 pts) Compute the cost estimate of the following index nested loops join plan:



(b) (8 pts) Compute the cost estimate of the following nested loops join plan:



5 (20 points) Query Optimization and Transactions

5.1 (8 pts) Selinger Optimization

This question involves the Selinger optimizer discussed in lecture. We want to join tables A, B, and C, and want to determine the optimal join order. There is a secondary B+ tree index on B, and a secondary B+tree index on C.

Assume Selinger has partially run and determined the following optimal join orderings, where NL = Nested Loops, INL = Index Nested Loops, SM = Sort Merge.

```
bestjoin(A,B) = (A NL B)
bestjoin(B,C) = (B NL C)
bestjoin(A,C) = (A INL C)
```

(2 pts each) Circle true or false for each of the following questions. Include a brief one sentence explanation.

1. The Selinger Optimizer would consider the plan `(C NL (A NL B))`. (True / False)

2. The Selinger Optimizer would consider the plan `((B NL C) INL A)`. (True / False)

3. The Selinger Optimizer would consider the plan `((B NL A) NL C)`. (True / False)

4. The Selinger Optimizer would consider the plan `((A INL C) INL B)`. (True / False)

5.2 (12 pts) Transaction Schedules

(1 pt per correct statement) Consider the following transactions schedules, along with statements about the schedule. Draw a circle around each true statement.

1.
```
T1: R(A)       W(B)              R(C)
T2:       W(A)       R(C)   W(B)       R(C) W(C)
```

   (a) The schedule is serial.
   (b) The schedule is serializable.
   (c) The schedule is conflict serializable.
   (d) The schedule exhibits the Dirty Read anomaly.
   (e) The schedule exhibits the Unrepeatable Read anomaly.
   (f) The schedule exhibits the Lost Writes anomaly.

2.
```
T1:            r(A)       w(B)
T2: r(B) w(A)                   w(B)
T3:                 r(A)            w(B)
```

   (a) The schedule is serial.
   (b) The schedule is serializable.
   (c) The schedule is conflict serializable.

3.
```
T1:        R(A)        W(B)                   R(C)
T2: W(A)         R(C)        W(B) R(C) W(C)
```

   (a) The schedule exhibits the Dirty Read anomaly.
   (b) The schedule exhibits the Unrepeatable Read anomaly.
   (c) The schedule exhibits the Lost Writes anomaly.

6 (Extra credit)

1. (3.5 points) Draw something here, or don't. Whatever you want. You'll get the points regardless.