W4111 Introduction to Databases
Fall 2016
Midterm 2 Exam Solutions

Closed Book, 1 page notes: 8.5x11” letter paper, both sides
614 Schermerhorn if **LAST DIGIT** of your UNI is < 5
303 Mudd otherwise
Duration: 75 minutes
Total Points: 130

Instructor: Eugene Wu
Dec 5, 2016

Instructions (Read Carefully)

This exam includes a double-sided **solutions page** where you must write your final answers. The rest of the exam booklet contains questions and will not be read for grading—you may use the pages as scratch paper.

Good Luck!

Regrade requests must be made by 10AM 12/12.
If we added your score incorrectly, we will correct it.
If there is an error in the solutions, please let us know with a **private piazza message**.
If you want a regrade of a question, we will *regrade the entire exam* carefully.

0 General Information

1. **(3 points)** If you are in 614 Schermerhorn and the last digit of your UNI is < 5, or you are in 303 Mudd and last digit of your UNI is ≥ 5. You do not need to write anything to receive these points.

2. You may choose to use a friend's answer instead of your own for **Problem 5.3**. If you do so, your answer score for that question will be the maximum of your own answer and your friend's. To be a friend, you must correctly write **both their name AND UNI**.

   (a) Your **friend's** Name

   (b) Your **friend's** UNI

1 (28 points) Query Execution

Consider a database with the following statistics, tables, and indexes. Assume all attributes are integers.

Fill Factor	0.5
Dir Entry Size	10 bytes
Tuple Size for all tables	100 bytes
Page Size	1000 bytes
ICARD(T1.a)	10
ICARD(T2.b)	10
ICARD(T3.c)	100
ICARD(T3.d)	10
minmax(T1.a)	[0, 99]
minmax(T2.b)	[0, 9]
minmax(T3.c)	[0, 99]
minmax(T3.d)	[0, 10]
no overflow pages	

Table Name	**Cardinality**	**Indexes**
T1	10 tuples	Secondary tree on T1(a)
T2	1000 tuples	Primary hash on T2(b)
		Primary tree on T2(b)
T3	100k tuples	Primary hash on T3(c)

1. **(3 points)** Consider the query $\sigma_{a=10}(T1)$. What is the best access path for this query?

Sequential Scan

2. **(1 points)** How many pages does your answer in the previous question cost?

2 pages

3. **(2 points)** What is the selectivity of $(T1) \bowtie_{T1.a=T3.c} (T3)$?

$$sel_{T1.a=T3.c} = \frac{1}{max(100, 100)} = 0.01 \tag{1}$$

4. **(2 points)** What is the selectivity of $\sigma_{c<50 \wedge d=10}(T3)$?

$$sel_{c<50} = \frac{50}{100} = 0.5 \tag{2}$$
$$sel_{d=10} = \frac{1}{10} = 0.1 \tag{3}$$
$$sel = sel_{c<50} \times sel_{d=10} = 0.05 \tag{4}$$

5. **(10 points)** Consider the optimal execution plan for the following SQL query. Describe the plan below by circling the options for each part of the plan.

```
SELECT * FROM T1, T3
WHERE T1.a = T3.c AND T3.d = 10
```

(a) Outer table: (a) T1 (b) T3

(b) Inner table: (a) T1 (b) T3

(c) Join Algorithm: (a) Nest Loops (b) Index Nested Loops

(d) Access path for outer: (a) Heap Scan (b) Hash Index (c) Primary BTree (d) Secondary BTree

(e) Access path for inner: (a) Heap Scan (b) Hash Index (c) Primary BTree (d) Secondary BTree

a, b, b, a, b

$\sigma_{d=10}(T1$ INL $T3)$

6. **(10 points)** Consider the join between all three tables below, and the join template below.

```
SELECT *
FROM   T1, T2, T3
WHERE  T1.a = T2.b AND T2.b = T3.c AND T3.c = T1.a
```

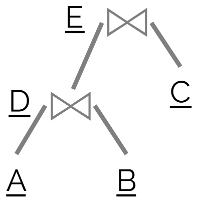


Figure 1: Join template, with missing tables and join algorithms.

Name	Description
NL	Nested loops join
INL-P	Index nested loops using a primary tree index
INL-S	Index nested loops using a secondary tree index
INL-H	primary hash index

Table 1: Shorthand names for join algorithms.

Consider the optimal query plan, as selected by the Selinger optimizer described in lecture, to execute the above SQL query that matches the join template in Figure 1. For A, B, C, write the appropriate table name. For D, E, write the name of join algorithm (Table 1)—you may assume that `hashjoin` is not considered by the optimizer.

A: T1
B: T2
C: T3
D: INL-H
E: INL-H

2 (12 points) Hip Hop is Alive

The Wu-Tang Clan is a popular hip-hop group from Shaolin. The original group already had many members, and with the affilated members (termed Killer Bees), it's very hard to keep track of the members and their album releases. It's gotten to the point that Method Man has to constantly holla "My peoples are you with me, where you at?"

Let's help the Clan find their members, by querying the following database:

```
members(
  mid int primary key,
  name text,      // name of the member
  is_a_bee bool,  // false if an original member
                  // true if an affiliate member (a Killer Bee)
  city text       // city where member lives
)
albums(
  aid int primary key,
  name text,      // name of the album record
  review int,     // 0 means bad, 10 means best
  revenue float
)
performs_on(      // a tuple denotes that member performed on album
  mid int references members(mid),
  aid int references albums(aid),
  primary key(mid, aid)
)
```

1. **(4 points)** Where are the peoples at? Find the city that has the most number of Killer Bees.

We accepted any query that resembled:

```
SELECT    city
FROM      members
WHERE     is_a_bee = true
GROUP BY  city
ORDER BY  count(*) desc
LIMIT     1
```

2. **(8 points)** "GZA" is the name of a founding Wu-Tang member. He wants to know if improving the quality of his albums leads to more sales. Consider all albums that GZA performed on—compute the average revenue for each review level. Return the review and average revenue in ascending order by review.

We accepted queries that performed the appropriate joins, filtered for GZA, and performed the group by aggregation.

```
SELECT    review, avg(revenue)
FROM      performs_on as p, albums as a, members as m
WHERE     p.mid = m.mid AND p.aid = a.aid AND m.name = 'GZA'
GROUP BY  review
ORDER BY  review asc;
```

3 (22 points) Functional Dependencies

Consider the relation $R(ABCDE)$ and the following functional dependencies
$\mathcal{F} = \{$A→BC, BC→D, ADC→CE, E→CD $\}$

1. **(5 points)** Check the boxes for all true statements, or check $\emptyset$ if none are true. Recall that $\mathcal{F}^+$ denotes the closure and $\mathcal{F}^{min}$ denotes the minimum cover.

   A $\mathcal{F}^{min} = \mathcal{F}^+$

   B $AC \to B$ is in $\mathcal{F}^+$

   C $A \to CD$ is in $\mathcal{F}^+$

   D $A \to E$ is in $\mathcal{F}^{min}$

   E $(\mathcal{F}^+)^+ = \mathcal{F}^+$

False, True, True, True, True.

0 points for $\emptyset$

2. **(5 points)** Check the box for all true statements, or check $\emptyset$ if none are true.

   A A is a candidate key for the relation

   B B is a candidate key for the relation

   C C is a candidate key for the relation

   D D is a candidate key for the relation

   E E is a candidate key for the relation

True, False, False, False, False

0 points for $\emptyset$

3. **(6 points)** For the following projections of the relation, check the box for all redundancy free projections, or check $\emptyset$ if none are redundancy free.

   A ABC

   B BCD

   C CDE

True, True, True

0 points for $\emptyset$

4. **(6 points)** Consider the minimal cover $\mathcal{F}^{min}$ and the projection $ACDE$. Alice decomposes it into AE and ECD. Check the box for all true statements, or check $\emptyset$ if none are true.

   A The decomposition is lossy

   B The decomposition is dependency preserving

   C Both decomposed tables have candidate keys

False, False, True

0 points for ∅

4 (4 points) Locks

Suppose table T contains 10 records, id is the primary key and Alice executes the following query as a transaction:

```
UPDATE T
SET    a = a + 10
WHERE  id = 10
```

1. **(2 points)** How many tuple read locks will this transaction acquire?

10 or 5.

11 was partially accepted.

2. **(2 points)** Alice installs a primary B+ tree index on `T(id)`. If the database uses this index as part of query execution, how many tuple read locks will this transaction acquire?

1

2 was partially accepted.

5 (16 points) Concurrency

1. **(6 points)** Given the following schedule, check the box for all true statements, or ∅ if none are true.

```
T1: R(B) W(B)           W(A) COMMIT
T2:           W(A) W(B)             W(B) COMMIT
```

A Schedule is Serializable

B Schedule is Conflict Serializable

C Schedule is allowed under Strict Two Phase Locking.

False, False, False

Due to grading error, we give full marks to all students, applied *after* any curves are computed, if any.

2. **(4 points)** Given the following two transactions, write a schedule that exhibits dirty read and lost write anomalies, but **does not** exhibit any unrepeatable read anomalies.

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

one possible solution:

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

3. **(6 points)** Given the following two transactions, write a schedule that is serializable but **not** conflict serializable.

```
T1: r(B) w(A) w(A)
T2: r(C) w(B) w(A)
```

Due to misgrade, everyone was awarded full marks for this question during grading.

The following is the same as T2, T1

```
T1:           r(B) w(A)      w(A)
T2: r(C) w(B)           w(A)
```

6 (17 points) Recovery

1. **(5 Points)** Consider the following sequence of operations. Check the statements where, if a crash occurs immediately after the statement, the database cannot correctly recover. Otherwise, check $\emptyset$ if the database can correctly recover.

```
1. write A
2. flush A
3. flush log record for "Write(A)"
4. flush log record for "COMMIT"
5. COMMIT
```

After operation 2.

2. **(8 Points)** Consider the following contents of the recovery log, assuming that each object corresponds to a single page and the database state is initially $X = 0, Y = 10$.

```
1. BEGIN T1
2. T1: X = X + 10
3. BEGIN T2
4. T1: X = X - 1
5. T2: Y = X - 2
6. T1 COMMIT
```

The on-disk state shows that $X = 0, Y = 7$. Answer the following questions:

A Which operations would need to be re-executed during REDO?

B Which operations would need to be undone during UNDO?

C After recovery completes, what is the correct value of X?

D After recovery completes, what is the correct value of Y?

A: 2, 4
B: 5
C: 9
D: 10

3. **(4 Points)** Check the boxes that the REDO phase of recovery is used for, or ∅ if none.

   A Atomicity

   B Consistency

   C Isolation

   D Durability

True, False, False, True

7 (18) Short Answers

(3 points each) Answer the following questions in *at most 2 short sentences* each.

These questions were graded rather strictly.

1. Describe the importance of normal forms.

Normal forms are systematic methods to eliminate redundancy in a schema.

Keyword is redundancy. “Duplicate” with a proper explanation is also ok.

2. Describe the difference between per-operator and pipelined query execution?

Per-operator runs each operator in the query plan to completion one at a time, and materializes the results. In contrast, pipelined execution can run operators simultaneously.

Key phrases are “materialize intermediates”, “per operator executes each operator to completion”, and “streamlined/overlapping” for pipelined execution.

3. Alice executes a complex, long running `SELECT` query and turns off all concurrency control. Which concurrency anomalies, if any, could her query be susceptible to? If her query is safe from concurrecy-based anomalies, simply write *None*.

Dirty Reads and Unrepeatable Reads.

Points deducted if non-anomalies are present in the answer.

4. Explain why a database system would use a *page* as the unit of data transfer.

If the database stores data on disk drives, then pages are used to balance between seek costs and transfer costs.

Key phrases include "disk IO", "seek time", "optimize".

5. Assuming their search keys are the same, describe a case where a secondary B+ tree index can be used to execute a `SELECT` query faster than a primary B+ tree index. If such a case does not exist, simply write *None*

If the query only accesses attributes in the search key, then there may be fewer leaf nodes to read in the secondary index. For instance, a `COUNT(*)` query.

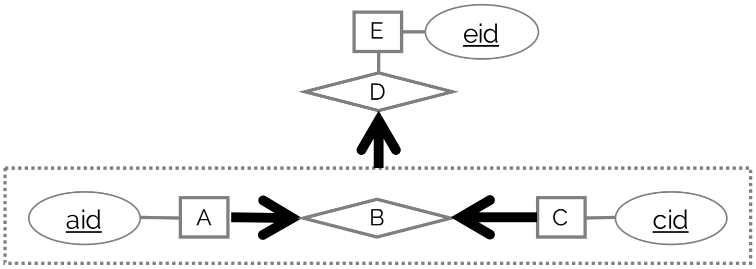
Grader looked for an example as well as a reasoning for lower IO/page access costs.

6. What was most useful topic that you learned in this course and why?

8 (10 points) Going back to the ER

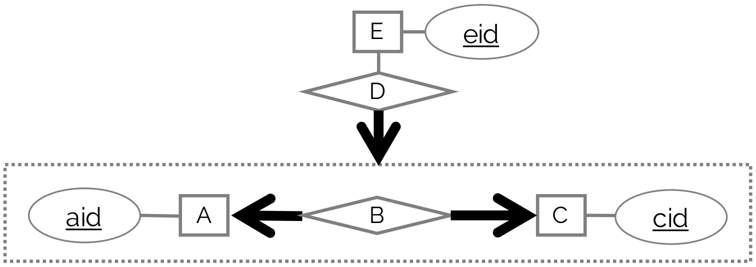
In this problem, you will translate each ER diagram into its corresponding SQL schema definition. If the ER diagram is incorrect, simply state "incorrect". If some aspect of the diagram cannot be expressed for whatever reason, simply state the fact. Do not assume that any tables have been defined. For readability, thick lines are also colored black, and thin lines are also colored grey.

1. **(5 points)**



```
ABCD(
  aid int UNIQUE NOT NULL,
  cid int UNIQUE NOT NULL,
  eid int NOT NULL,
  primary key(aid, cid)
)
E(
  eid int primary key
)
```

2. **(5 points)**



Incorrect