CS 145 Introduction to Databases
Autumn Quarter 2016
Stanford University

Midterm Examination

Thursday October 27, 2016

80 minutes

	Problem	Full Points	Bonus Points	Your Score
1	Pet Store for the AIs	30	10	
2	They have an ER for the dramatic!	15		
3	Where are my keys?	20		
4	Decompositions.	20	10	
5	Bob's Bagel Bite Shop	20		
	Total	90	20	

The exam contains **21** pages including this cover page. The last page provides you some information that may be useful in the exam. Feel free to tear the last page out.

Instructions:

- This exam is **closed book i.e. no laptops, notes, textbooks, etc. during the exam**
- There is a cheetsheet of definitions on the last page.
- In all cases, and especially if you're stuck or unsure of your answers, **explain your work!** We'll give partial credit for good explanations of what you were trying to do
- Keep the point values of questions in mind- don't get too stuck on any one part! **Our advice is to skip the bonus problems on your first pass!**

Name: ______________________________

SUNETID: ______________________________

I have read and will abide by the Stanford honor code:

Signature: ______________________________

1 Pet Store for the AIs [20 points]

Imagine the following setup. You run a retail shop. Now don't get me wrong, it's fancy: you specialize in selling products via either brain-computer IoT interfaces or to AIs that keep humans as pets. Your niche in this obviously crowded space is that all your products are still made by people (how retro!). Of course, your products and orders are stored in a classical database.

```
Buyer(bid,name,phone,city,is-ai)
Sale(buyer-id, seller-name,pid, number)
Product(pid, name, price, category, cid)
Company(cid, name, country, debt)
```

You may assume that all fields are not `NULL`. You may abbreviate tables with their first letter, if you choose. For your reference, we'll replicate this schema on each page. Some more information about the schema is below:

- The key of buyer is bid. The attribute is-ai takes the value true when the buyer is an AI.
- In Sale, a buyer-id refers to Buyer.bid, and seller-name is a person's name, and the pid refers to Product.pid. The number of units is the number of units of that product sold.
- A product has an id, and a price. The Product.cid refers to Company.cid
- Company has a key cid, a name, and a country in which iti ias based.

1.1 Part 1: Watch out for our Northern Flank! [5 points]

Find all the distinct names of all companies that are based in Canada.

1.2 Part 2: They probably owe us money! [5 points]

```
Buyer(bid,name,phone,city,is-ai)
Sale(buyer-id, seller-name,pid, number)
Product(pid, name, price, category, cid)
Company(cid, name, country, debt)
```

Find the distinct names of all companies that are based in Canda and that sold a product to an *AI* based in Cupertino.

1.3 Part 3: HUGELY successful companies [10 points]

```
Buyer(bid,name,phone,city,is-ai)
Sale(buyer-id, seller-name,pid, number)
Product(pid, name, price, category, cid)
Company(cid, name, country, debt)
```

Find the names of all companies that have sold at least five distinct products.

1.4 Part 4: Only the most luxurious! [10 points]

```
Buyer(bid,name,phone,city,is-ai)
Sale(buyer-id, seller-name,pid, number)
Product(pid, name, price, category, cid)
Company(cid, name, country, debt)
```

Find the names of all companies such that *every* product they sell have ever sold cost more than 1 million dollars. Companies that have not sold any products should not be counted, as they are losers.

1.5 BONUS: The hottest startups ... [10 points]

```
Buyer(bid,name,phone,city,is-ai)
Sale(buyer-id, seller-name,pid, number)
Product(pid, name, price, category, cid)
Company(cid, name, country, debt)
```

Find the names of all companies such that have not sold even a single product.

2 They have an ER for the Dramatic! [15 points]

Design an ER diagram for a theatre ticket seller. It contains the following kinds of objects together with the listed attributes:

- Actors(name, debut_date, nationality)
- Directors(name, beret_size, smoke)
- Product(title, cash-balance)
- Venue(name, address, capacity)

In addition the following constraints must be modeled:

- Each actor or director is uniquely modeled by their names.
- A director may direct a production, and an actor may appear in a production in a role.
- Every production is required to have at most one director (but may have no director)
- Every venue has exactly one production.
- Not every actor is in a production (for shame!)
- Every production has at least one actor.

Be sure to specify **keys, multiplicity, and participation constraints** in addition to the ER basics (entity sets, relationships and attributes). If in doubt, explain what you're trying to do!

Warning: Do not spend too much time on this question. We will grade it in a straightforward way: did you include the required constraints? The required attributes, entity sets, and relationships. Is your ER syntax valid?

Extra space:

3 Where are my keys? [20 points]

For the first three parts of this problem, consider the relation $R(U, V, W, X, Y, Z)$ with functional dependencies:

1. $\{W, Y\} \rightarrow \{Z\}$
2. $\{U\} \rightarrow \{X, Y\}$
3. $\{V, X\} \rightarrow \{W\}$
4. $\{Y\} \rightarrow \{V\}$

For all parts of this question: keep in mind that explaining your reasoning / showing your work, while not required, is extremely helpful if we are trying to give you partial credit!

3.1 Part 1 [5 points]

Consider the instance of relation R below. Fill in the blanks so that the instance is consistent with the FDs above. When the FDs imply a specific value for a blank space, fill in that value, otherwise if they do not imply a specific value, fill it in with a question mark:

U	V	W	X	Y	Z
0	0	1	2	3	4
1	0		2	2	
	0	1		3	4
2		1	2	2	5

3.2 Part 2 [5 points]

Find the closure of $\{X, Y\}$, $\{X, Y\}^+$:

3.3 Part 3 [5 points]

List **one** key implied by the given FDs, and **explain why it is a key based on the definition:**

3.4 MVDs!

Consider the instance of relation R below. Fill in the blanks so that the instance is consistent with the following MVDs:

- $\{U\} \twoheadrightarrow \{W, Y\}$
- $\{V\} \twoheadrightarrow \{X, Z\}$

When the MVDs do not imply a specific value for a blank space, fill it in with a question mark:

U	V	W	X	Y	Z
0		2	3		5
	2		4	5	6
0	2	2	4	4	6
0	1	3	3	5	

4 Decompositions.

4.1 Part 1: BCNF is what everybody is talking about . . . [5 points]

For this problem, consider the relation $T(A, B, C, D, E)$ with the following functional dependencies:

1. $\{A\} \to \{B, C\}$
2. $\{C\} \to \{B\}$
3. $\{B\} \to \{D\}$
4. $\{E\} \to \{C\}$

Decompose T into two or more tables so that it is in BCNF, and briefly justify that your end result is indeed in BCNF:

4.2 Part 2: Lossy your mind [5 points]

Consider the following decomposition table $R(A, B, C)$ and its decomposition into $S(A, B)$ and $T(A, C)$. Given an example an instance of R such that when decomposed into S and T, the join is lossy, i.e., the join of S and T on A returns more tuples than were present in R.

4.3 Part 3: Don't Lossy your mind [5 points]

Consider the following decomposition table $R(A, B, C, D)$ and its decomposition into $S(A, B)$ and $T(A, C)$. Give a sufficient condition (a constraint) such that decomposing into S and T are lossless, i.e., the join of S and T on A will return *exactly* R for any input set of tuples satisfying the constraint.

4.4 Bonus: I lost something... [10 points]

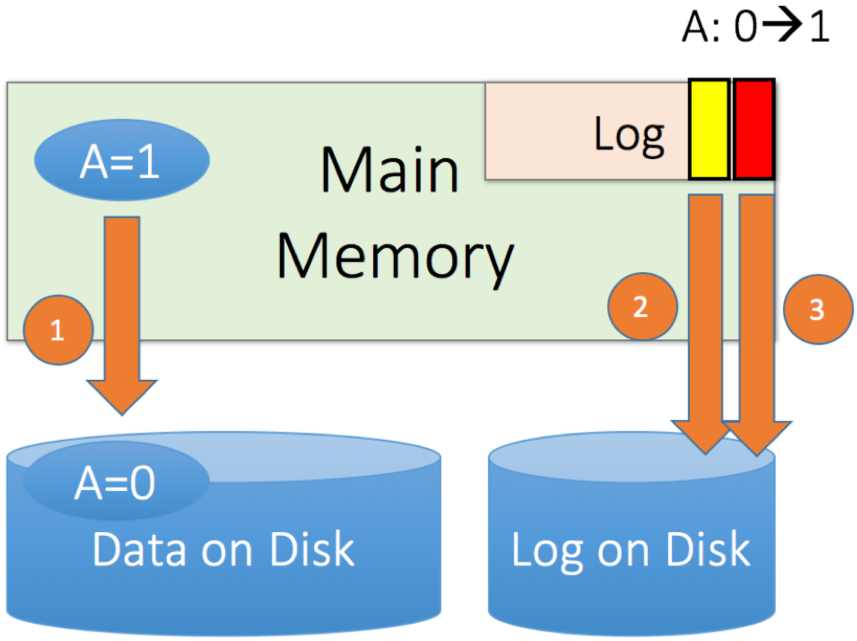
Recall in the in-class activity, we "lost a dependency" doing BCNF. More precisely, we decomposed a relation R into BCNF with two tables S and T. To check some functional dependency on R, we had to join S and T together (neither table contained all the attributes of the functional dependency)! Give an example (the schema of R and a set of FDs that exhibit this phenomenon.

5 Bob's Bagel Bite Shop [20 points]

Bob's Bagel Bite Shop, the premiere purveyor of homemade bagel bites in the bay area, has decided to start a loyalty points program, where customers can earn *"bitecoins"*. Chef Bob is a big fan of transactions and upholding ACID guarantees, but to save on money, he decides to skip buying a full-featured SQL DBMS and instead implement everything himself... oh boy.

[This space left intentionally blank... draw your favorite animal eating fruit here?]

5.1 Part 1: Write what you feel! [10 points]



Bob implements the following three procedures, but isn't quite sure how to order them:

1. The values in main memory written by the TXN are flushed to disk
2. The log of the TXN's actions is flushed to disk
3. The commit record (indicating that the TXN was committed) is flushed to disk

5.1.1 [3 points]

Consider the following order: 1,3,2. List the points at which a system crash would cause issues, and explain why, referring to ACID terms (one sentence each):

5.1.2 [3 points]

Consider the following order: 1,2,3. List the points at which a system crash would cause issues, and explain why, referring to ACID terms (one sentence each):

5.1.3 [4 points]

What is the correct order for WAL? Again, refer to ACID terms, and explain with one sentence.

5.2 Part 2: Conflicting Bites [10 points]

Bob also does not have any sort of locking, and must interleave transactions by hand. Consider the time-line below, showing three TXNs operating over two bitecoin accounts, A and B:

5.2.1 [5 points]

Label at least three *conflicting* pairs of actions:

5.2.2 [5 points]

Describe the serial order that this schedule corresponds to. If this schedule is *not serializable*, first circle two operations whose horizontal position could be swapped to make this schedule serializable.

[This page left intentionally blank]

6 Possibly Useful Information

- **Canonical SQL Statement:**

```
SELECT <attributes>
FROM <tables>
WHERE <conditions>
GROUP BY <attributes>
HAVING <conditions>
```

- **Functional Dependency (FD):** For a relation R, and sets of attributes X and Y, the functional dependency $X \to Y$ holds if for any $t_1, t_2 \in R$, $t_1[X] = t_2[X] \implies t_1[Y] = t_2[Y]$.
- **Armstrong's Axioms:** Let the A_is, B_js, and C_ks be attributes:
  1. *Split/Combine:* If $A_1, ..., A_n \to B_j$ for $j = 1, ..., m$, then this is equivalent to $A_1, ..., A_n \to B_1, ..., B_m$ and vice-versa
  2. *Reduction/Trivial:* $A_1, ..., A_n \to A_i$ for any $i = 1, ..., n$
  3. *Transitive Closure:* If $A_1, ..., A_n \to B_1, ..., B_m$ and $B_1, ..., B_m \to C_1, ..., C_p$ then $A_1, ..., A_n \to C_1, ..., C_p$
- **Closure:** Given a set of attributes X and a set of FDs F, the closure X^+ is the set of all attributes y such that $X \to y$.
- **Superkey:** Given a relation R, a superkey is a set of attributes X such that X^+ is equal to the full set of attributes of R.
- **Key:** A key is a minimal superkey, i.e. a superkey where no subset of it is also a superkey.
- **Boyce-Codd Normal Form (BCNF):** A relation R is in BCNF if for all sets of attributes X, either $X^+ = X$ (X is trivial) or $X^+ =$ the set of all attributes (X is a superkey).
- **Conflicts:** Two actions conflict if they are part of different TXNs, involve the same variable, and at least one of them is a write.
- **Serializable:** A schedule is serializable if it is equivalent to some serial ordering.
- **Multi-Value Dependency (MVD):** Given a relation R with a set of attributes A, two sets of attributes $X, Y \subseteq A$, we say that the MVD $X \twoheadrightarrow Y$ holds if for any tuples $t_1, t_2 \in R$ such that $t_1[X] = t_2[X]$, there is a tuple t_3 such that:
  - $t_3[X] = t_1[X]$
  - $t_3[Y] = t_1[Y]$
  - $t_3[A \setminus Y] = t_2[A \setminus Y]$