Prof. Brewer
Fall 2015

CS 186
Databases

Midterm 2

PRINT your name: ____________________________, ____________________________
(last) (first)

I am aware of the Berkeley Campus Code of Student Conduct and acknowledge that academic misconduct will be reported to the Center for Student Conduct.

SIGN your name: ____

PRINT your class account login: `cs186-`______ and SID: __________________________

Your TA's name: ____

Your section time: ____

Login of the person sitting to your left: ____________________ Login of the person sitting to your right: ____________________

You should receive a double-sided answer sheet in addition to this packet. Mark your name and login on the answer sheet, and in the blanks above. **All answers must be marked on the answer sheet – do not show any work unless asked.** You have 80 minutes for this exam – do not spend too much time on any single problem. Turn in both this packet and the answer sheet when you are done.

Do not turn this page until your instructor tells you to do so.

Question:	1	2	3	4	Total
Points:	36	22	27	15	100
Score:					

Problem 1 ***Relational Algebra and Query Optimization*** **(36 points)**

Part A. Relational Algebra and SQL

Not all lines on the answer sheet will be used. Refer to the following Schema for the entirety of Problem 1:

Country		Instructor		Flight	
cid	int	**id**	int	**fid**	int
cname	string	name	string	departure*	int
popularity	int	budget	int	destination*	int
...		cid*	int	price	int

*A few other notes:

- We shorthand Country as C, Instructor as I, Flight as F.
- Bolded items are primary keys of the respective tables.
- I.cid is a foreign key referencing C.cid.
- F.departure is a foreign key referencing C.cid.
- F.destination is a foreign key referencing C.cid.

For Problems 1 and 2, answers do not have to be optimal.

1. [**6 points**] Translate the following relational algebra statement into SQL:

   (a) $\pi_{C.cid}\big(\sigma_{C.popularity>5}\big(I \bowtie_{I.cid=F.departure} F\big) \bowtie_{F.destination=C.cid} C\big)$

   (b) $\big((\pi_{I.cid}(I \bowtie_{I.cid=F.departure} F)) \cap (\pi_{C.cid}(F \bowtie_{F.destination=C.cid} (\sigma_{C.popularity>5} C)))\big)$

2. [**6 points**] Translate the following statements into relational algebra:

   (a)
   ```
   SELECT I.name
   FROM Instructor I, Country C, Flight F
   WHERE I.budget = F.price
      AND C.cid = I.cid
      AND C.popularity < 10;
   ```

   (b)
   ```
   SELECT C1.cid, C2.cid
   FROM Country C1, Country C2, Flight F
   WHERE C1.cid = F.departure
       AND C2.cid = F.destination
       AND C1.popularity < C2.popularity
   ```

Part B. Query Optimization

For this problem, assume that the size of a page is 4 KB, and that there are 66 pages in the buffer pool. To make this question simpler, we will also ignore Sort-Merge Joins and Hash Joins. In System R optimizers, cross joins are not considered unless they cannot be avoided.

The statistics for each table are given in the Appendix in the back of the exam.

Suppose we join 3 tables with the following query:

```
SELECT C.cname
FROM Country C, Instructor I, Flight F
WHERE I.cid = F.departure
AND C.cid = F.destination
AND I.budget >= 1000
AND F.price < 1000;
```

1. [**5 points**] Walking through all possible single table access plans:
   (a) Find the costs of file scans
   (b) Find the costs of any other plans
2. [**2 points**] What single table access plans are kept?
3. [**2 points**] What two-table joins are considered? (You don't need to state the specific type of join.)
4. [**10 points**] What two-table access plans are kept? (**Do** state the specific type of join for each.)

Assume the following values for each ordered pair:

Pair	IOs
$(C \bowtie I) \bowtie F$	250,000
$(C \bowtie F) \bowtie I$	36,000
$(F \bowtie C) \bowtie I$	40,000
$(F \bowtie I) \bowtie C$	50,000
$(I \bowtie C) \bowtie F$	300,000
$(I \bowtie F) \bowtie C$	45,000

5. [**3 points**] Given that we follow the same rules for System R and are evaluating the query above, what three table access plans are considered?
6. [**2 points**] What is the final plan chosen?

Problem 2 ***FDs and Relational Decompositions*** **(22 points)**

Consider the schema R(A, B, C, D, E, X, Y, Z) with functional dependencies:

$$\mathrm{F} = \{AB \rightarrow CD, B \rightarrow DX, Y \rightarrow ABE, E \rightarrow X\}$$

For Questions 1-3 [**10 points**], circle True or False *on the answer sheet.* Using the space provided, provide a *short* but convincing explanation if you circled True, or a counterexample or explanation if you circled False.

1. **True/False:** R is in Boyce Codd Normal Form (BCNF).
2. **True/False:** Multiple decompositions are necessary to decompose R into BCNF (that is, we must repeat the algorithm described in lecture, regardless of which FD is processed first).
3. Suppose we decompose R into Q(A, B, C, D), S(A, B, E, Y), and T(X, Y, Z):
   (a) **True/False:** This is a lossless join decomposition.
   (b) **True/False:** This is a dependency preserving decomposition.
   (c) **True/False:** The resulting relations are in BCNF.
4. [**2 points**] What is the attribute closure for AB?
5. [**2 points**] What candidate keys *with the minimum number of attributes* are inferred from F? List all if there are multiple.
6. [**3 points**] Determine the minimal cover for F.

Now consider the schema R(A, B, C, D, E, F) with functional dependencies:

$$\mathrm{F} = A \rightarrow E, B \rightarrow DF, C \rightarrow B, E \rightarrow C$$

7. [**5 points**] Suppose we decompose R into BCNF form using the algorithm described in lecture.
   (a) **What are the resulting relations?** Process the FDs *from left to right.*
   (b) **Is this decomposition lossless?** Explain concisely and convincingly.
   (c) **List all dependencies that are not preserved.** If all dependencies are preserved, write "All preserved."

Problem 3 ***Transactions and Concurrency Control*** **(27 points)**

1. [**8 points**] Consider the following transaction T1.
Assume that A=50 and B=1000 before T1 begins execution, and that other transactions may be running concurrently.

T1	R(B)	R(A)	B := A+B	W(B)	A := A+B	W(A)	B := A+B	W(B)	COMMIT

   (a) List **one possible** outcome of B if the given ACID property is **not** preserved. If you believe such an outcome cannot be determined from the given information, write "needs more information".

      i. Atomicity

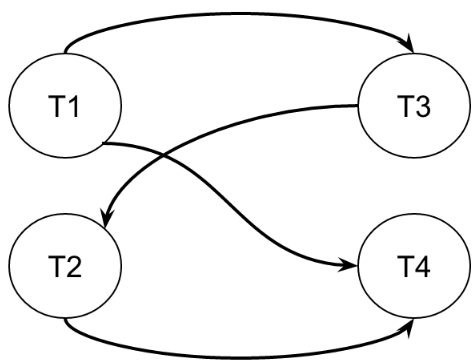
      ii. Consistency

      iii. Durability

   (b) Now, assume there is another transaction T2 that is to run concurrently with T1. List **one possible** outcome of B after both transactions have run if isolation is not preserved.

T2	R(B)	B := B*3	W(B)	COMMIT

2. [**8 points**] Consider the following dependency graph:



   (a) Is this schedule conflict serializable? If yes, list **all possible** serial orderings of T1, T2, T3, and T4. If not, state why not.

   (b) Write a possible schedule for T1, T2, T3, and T4 that results in the above dependency graph. Assume you can read/write to variables A, B, and C. **You may not need to use all of the variables, and use the minimum number of columns possible.**

3. [**6 points**] Assume you have the following locking schedule for transactions T1, T2, T3, and T4 that utilizes multiple granularity locks. Assume A, B, C, and D are all separate tables and that no locks are released within the given timeframe.

T1					X(A)						
T2		SIX(B)	IS(C)				IS(D)				
T3				IX(C)				IS(B)			SIX(A)
T4	IS(A)					IS(B)			IX(D)	SIX(C)	

(a) Draw the arrows for the waits-for-graph at the end of the schedule.

(b) Does the schedule lead to deadlock? If yes, list the transactions that could be aborted to resolve the deadlock. Do not list transactions that do not necessarily need to be aborted. If no, explain why.

4. [**5 points**] Assume you have the following schedule for transactions T1, T2, and T3 that adheres to the *non-strict* 2-phase locking protocol (2PL).

time	1	2	3	4	5	6	7	8	9
T1	R(A)	R(B)			W(B)				
T2			R(A)				W(B)	R(C)	W(C)
T3				R(A)		R(C)			

For the following questions, if you want to specify that T1 can release a lock immediately after timestamp 1, write "1". Locks cannot be acquired at the timestep at which they were unlocked, and exclusive locks are only acquired if necessary.

(a) What is the earliest time at which T1 could release its exclusive lock on B?

(b) What is the earliest time at which T2 could acquire its exclusive lock on C?

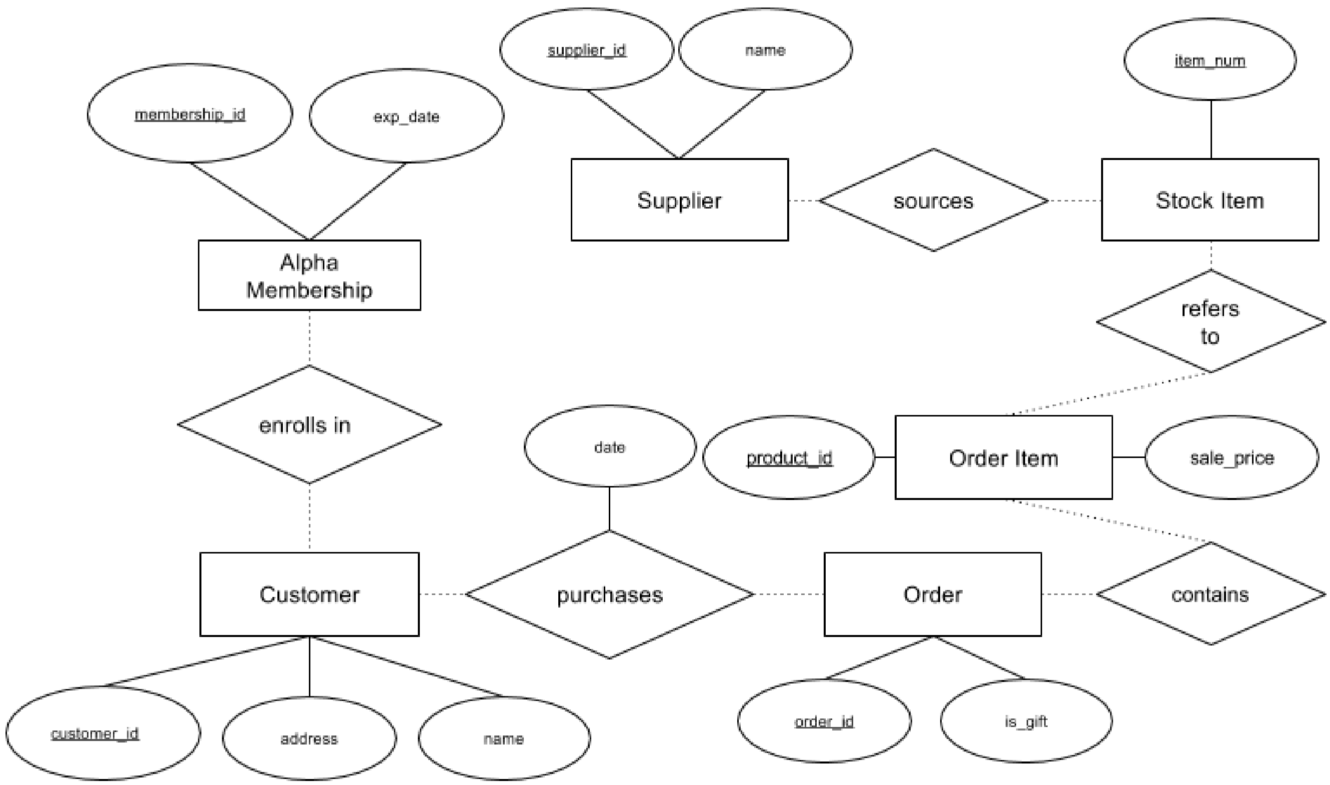
(c) What is the latest time at which T2 could release its shared lock on A?

(d) What is the earliest time at which T3 could release its shared lock on A?

(e) What is the earliest time at which T3 can acquire its shared lock on A?

Problem 4 *ER Diagrams* (15 points)

You're working for a hot new startup that is trying to redefine the way we buy and sell products online. In order for the service to run, the company needs to be able to store a catalog of its items and orders. Part of your first task at the budding company is to design an entity-relationship diagram so that your team can translate it into a database schema.



Your company needs to keep track of its Stock Items.

1. Each Stock Item refers to a specific item that the company currently has in its warehouse. For example, if your company currently has 5 Model A bicycles, then there would be 5 Stock Items - one for each bicycle.
2. However, it is not necessarily true that all Stock Items of the same model are sourced from the same Supplier. Following the previous example, it could be true that Supplier 1 supplies 3 of the bicycles while Supplier 2 supplies the other 2. Your database maintains records of suppliers, even if you don't source anything from them.
3. Customers may choose to enroll in a special Membership program called Alpha Membership (only one membership account per customer), which provides extra perks. However, Membership is not required and customers can still use the service without it. Memberships are not created prior to enrollment.
4. Customers can also make purchases of Orders on your service.
5. Each Order contains several (at least one) Order Items. Order Items are identified by orders, and each Order Item refers to a specific Stock Item in the warehouse.

A. partial participation, non-key
B. partial participation, key
C. total participation, non-key
D. total participation, key

1. [**10 points**] Given the lettered options above, choose the best constraint for each of the following relationships:
   (a) Between Enrolls In and Alpha Membership
   (b) Between Customer and Enrolls In
   (c) Between Customer and Purchases
   (d) Between Purchases and Order
   (e) Between Order and Contains
   (f) Between Contains and Order Item
   (g) Between Order Item and refers to
   (h) Between refers to and Stock Item
   (i) Between Stock Item and Sources
   (j) Between Supplier and Sources
2. [**5 points**] Can some entities be converted into weak entities in this ER diagram? Briefly explain why, and identify all strong and weak entity pair you would create (make sure you specify which is which).

Appendix: Query Optimization

Country		Instructor		Flight	
cid	int	**id**	int	**fid**	int
cname	string	name	string	departure*	int
popularity	int	budget	int	destination*	int
...		cid*	int	price	int

All B+ trees are Alternative 2. You may also assume that all B+ tree indices are filled.

Table 1: Schemas

Table	NPages	Tuples/Page	Indices
Country	32	8	
Instructor	256	8	B+ Unclustered on Budget (4 page)
Flight	4096	256	B+ Clustered on Price (16 pages)

You may assume a uniform distribution for non-key attributes.

Table 2: Country

	cid	popularity
min	0	0
max	255	99
numDistinct	256	100

Table 3: Instructor

	id	name	budget	cid
min	0	Abby	0	0
max	$2^{11} - 1$	Xavier	1999	255
numDistinct	2^{11}	200	2000	256

Table 4: Flight

	fid	departure	destination	price
min	0	0	0	0
max	$2^{20} - 1$	255	255	1999
numDistinct	2^{20}	256	256	2000