Midterm 2: CS186, Fall 2015 - Solutions

Question 1A: Relational Algebra

1a.
SELECT C.cid
FROM Instructor I, Country C, Flight F
WHERE I.cid = F.departure
AND C.cid = F.destination
AND C.popularity > 5

1b.
SELECT C.cid
FROM Instructor I, Country C, Flight F1, Flight F2
WHERE I.cid = F1.departure
AND F2.destination = C.cid
AND C.popularity >5
AND I.cid = C.cid

2a.
$\pi_{I.name}(F \bowtie_{I.budget=F.price}(I \bowtie_{C.cid=I.cid}(\sigma_{C.popularity<10}C))$

2b.
$\pi_{C1.cid,C2.cid}(F \bowtie_{F.departure=C1.cid \wedge F.destination=C2.cid}(\rho(C1, C) \bowtie_{C1.popularity<C2.popularity}\rho(C2,C))$

Question 1B: Query Optimization

Question 1a - Cost of File Scans:

Each cost is the NPages of each relation.

Country	Instructor	Flight
32	256	4096

Question 1b - Cost of any other plans:

We follow this to get to our answers:

1. Determine whether or not there indexes with matching selections for single table
   a. Matching selection refers to attributes that are both in the WHERE clause and the key of an index
2. If exists, use appropriate calculation for single table accesses

Country	Instructor	Flight
N/A	B+ Unclustered on Budget: RF = 1000/2000 = ½ RF*(NPages(Index) + NTuples(Rel)) = 0.5(4 + 8*256) = 1026	B+ Clustered on Price: RF = 1000/2000 = ½ 0.5(16 + 4096) = 2056

Question 2 - Single table Access Plans Kept:

Country	Instructor	Flight
File Scan	File Scan	B+ Clustered

Question 3 - Two-table joins Considered:

We enumerate all possibilities, and eliminate those that are cross joins:

C ⋈ F	~~I ⋈ C~~	F ⋈ I
~~C ⋈ I~~	I ⋈ F	F ⋈ C

Question 4 **-** Two table plans kept:

We use the single table access plans from Question 2 - and we consider all possible join methods. We simplify things by not considering Sort Merge and Hash join. We can begin by compiling what we know from the previous single table plans. For the cardinality, we have to consider the reduction factor.

	Access Cost (IOs)	Cardinality (Pages)
Country	$32 = 2^5$	32
Instructor	$256 = 2^8$	$(256)(0.5) = 2^7$
Flight	2056	$(2\text{^}12)(0.5) = 2^{11}$

If we had more indices, we would want to calculate the access cost of each index. If we recall the Discussion Worksheet for Week 8, $NTuples_{potential}$ is akin to our [Cardinality of Previous].

In general, for a **nested loop join**, calculating ((Previous) ⋈ F):
[Access Cost of Previous] + [Cardinality of Previous] * [Access Cost F]

For CNLJ, the Cardinality is by page. For INLJ, the cardinality is by Tuple. *However, there are no indices that we can join on (we have no index on I.cid, C.cid, F.departure, nor F.destination).*

Approximations would have been fine for parts (ie, 2056 ~ 2^{11})

C ⋈ F	**Chunk Nested Loop Join** [Access Cost C] + [Cardinality of Previous] * [Access Cost F] / (Buffers - 2) $(32) + (2^5)(2056) / (2^6) = 1060$

I ⋈ F	**Chunk Nested Loop Join** [Access Cost I] + [Cardinality of Previous] * [Access Cost F] / (Buffers - 2) $(2^8) + (2^7)(2056) / (2^6)$ = 4112 + 256 = 4368
F ⋈ I	**Chunk Nested Loop Join** [Access Cost F] + [Cardinality of Previous] * [Access Cost I] / (Buffers - 2) $(2056) + (2^{11})(2^8) / (2^6)$ = 8096 + 2056
F ⋈ C	**Chunk Nested Loop Join** [Access Cost F] + [Cardinality of Previous] * [Access Cost C] / (Buffers - 2) $(2056) + (2^{11})(2^5) / (2^6)$ = 1024 + 2056 = 3080

We will keep: I ⋈ F, C ⋈ F.

Question 5: Three-Table Access Plans

We will consider (C ⋈ F) ⋈ I and (I ⋈ F) ⋈ C.

This must follow from your answer from Question 4.

Question 6:

Final plan should be (C ⋈ F) ⋈ I.
This must follow from your answer from Question 5.

Question 2: FDs + Relational Decompositions

Consider the schema R(A, B, C, D, E, X, Y, Z) with functional dependencies
F = { AB -> CD, B -> DX, Y -> ABE, E -> X }

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

1. T/F: R is in BCNF. For example, AB is not a superkey.
2. T/F: Multiple decompositions are necessary to decompose R into BCNF.
   Decompose with Y -> ABE => ABEY, CDXYZ
3. Suppose we decompose R into Q(A, B, C, D), S(A, B, E, Y), and T(X, Y, Z).
   a. T/F: This is a lossless join decomposition.
      BCNF: ABCDEXYZ -> ABCD ABEXYZ -> ABCD ABEY XYZ
   b. T/F: This is a dependency preserving decomposition.
      B -> X and E -> X are not preserved.
   c. T/F: The resulting tables are in BCNF. B -> D violates BCNF constraints.

4. What is the attribute closure of AB?
   AB+ = {ABCDX}

5. What candidate keys with the minimal number of attributes are inferred from F?
   { YZ }

6. Determine the minimal cover for F.
   AB -> C, B -> D, B -> X, Y -> A, Y -> B, Y -> E, E -> X

Now consider the schema R(A, B, C, D, E, F) with functional dependencies
F = {A -> E, B -> DF, C -> B, E -> C}

7. Find the BCNF decomposition for R, by handling the dependencies *from left to right*.
   ABCE, BDF
   ACE, BC, BDF
   Final Answer: AE, CE, BC, BDF

   a. Is this decomposition lossless?
      Yes. BCNF decompositions are lossless.
   b. List all dependencies that are not preserved. If there are none, write “None”
      None.

Question 3: Transactions

Consider the following transaction T1. List **one possible** outcomes 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".
Assume that A=50 and B=1000 before T1 begins execution.

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

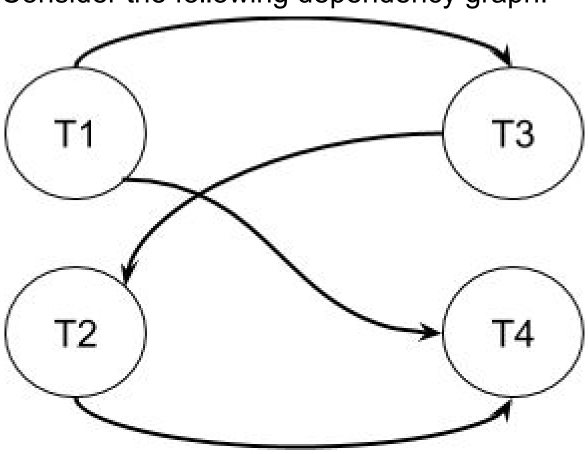
Atomicity: 1050, 2050, 1100, 1000
Consistency: Needs more information
Durability: 1000

Now, assume there is another transaction T2 that is to run concurrently with T1. List **one possible** outcomes of B after both transactions have run if isolation is not preserved. Again, assume that A=50 and B=1000 before both transactions begin execution and that B can not be read/written by two transactions at exactly the same time.

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

3000, 3150, 6350

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 no, state why not.
   T1, T3, T2, T4

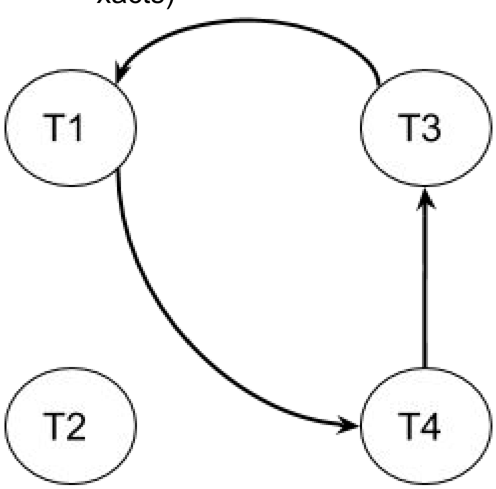
b. Write a possible schedule for T1, T2, T3, and T4 that results in the above dependency graph. Assume you can read/write to the variables A, B, C. Use the minimum number of reads and writes possible. **You may not need to use all of the columns.**

T1	W(A)										COM
T2					W(B)						COM
T3		R(A)		R(B)							COM
T4			R(A)			R(B)					COM

3. Assume you have the following locking schedule for transactions T1, T2, T3, and T4 which utilizes multiple granularity locks.

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 this schedule's waits-for graph. (4 pts, 2 for timestamp,3 for aborted xacts)



b. Does the schedule lead to deadlock?

Yes. Abort T1, T3, or T4

4. A schedule for transactions T1, T2, and T3 is given below.

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, assume that all transactions adhere to the non-strict 2-phase locking protocol. Assume that locks cannot be acquired at the timestep at which they were unlocked, and that exclusive locks are only acquired if necessary. If you want to specify that T1 can release a lock immediately after timestamp 1, write “1”.

i. What is the earliest timestamp at which T1 could release its exclusive lock on B? 5
ii. What is the earliest timestamp at which T2 could acquire its exclusive lock on C? 7
iii. What is the latest timestamp at which T2 could release its shared lock on A? 9
iv. What is the earliest timestamp at which T3 could release its shared lock on A? 4
v. What is the earliest timestamp at which T3 can acquire its shared lock on A? 1

Question 4: ER Diagrams

A. partial participation, non-key
B. partial participation, key
C. total participation, non-key
D. total participation, key
Given the lettered options above, choose the best constraint for each of the following relationships:

(a) Between Enrolls In and Alpha Membership
D. Each membership is unique and participates in exactly one enrollment.
(b) Between Customer and Enrolls In
B. Each customer can choose to enroll in a membership, but does not have to. If they choose to enroll, they can enroll in at most one.
(c) Between Customer and Purchases
A. A customer can make 0 or more purchases.
(d) Between Purchases and Order
D. Any Purchase refers to a single order.
(e) Between Order and Contains
C. An Order must Contain at least one Order Item.
(f) Between Contains and Order Item
D. An order item is contained in exactly one order.
(g) Between Order Item and refers to
D. An Order Item must refer to exactly one Stock Item.
(h) Between refers to and Stock Item
A. A Stock Item can be referred to by 0 or more Order Items.
(i) Between Stock Item and Sources
D. A Stock Item can only be Sourced from a single Supplier.
(j) Between Supplier and Sources
A. A Supplier is stored in our db even if it does not source any Stock Items.

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).
There were many possible conversions. Here are some valid ones:
1. Order Items can be converted to a weak entity for Orders since any Order Item is uniquely identified by its Order and must be tied to exactly one Order.
2. Membership can be converted to a weak entity for Customer.