CS143–Winter 2020—FINAL EXAM

Your Name Your Student ID PageNo:

(Write your ID and the page number on the top of every page of the final.)

On the first page you must write & sign the following Academic Honesty statement: **I thereby certify with my signature that I completed this exam entirely on my own, without reference to any prohibited sources or materials, nor communications with anyone.**

Signed: ____________________

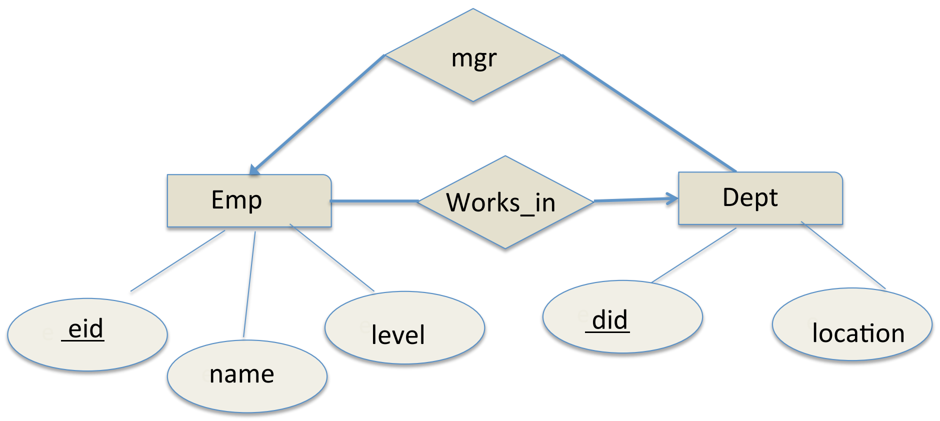
Total number of pages in your submitted final: ____

General Instructions

- The exam will be posted on and submitted to CCLE like previous assignments.
- You should write your answers on blank paper pages (letter-size recommended) and scan/take a picture of them with your device (e.g. your cell-phone). Please write neatly and do not spread your answers to the same problem several pages. If your hand-writing is hard to read you should use a text editor to prepare the PDF document that you'll submit to CCLE.
- For submissions, there will be a grace period of 30 Minutes (beyond the regular three hours).
- Make sure that you submission contains cover page with your Name, ID, total number of pages, and the signed statement certifying your academic honesty. Also write your ID and the page number on top of every page.
- Your exam will be proctored via Zoom. You will need to keep plenty of social distancing around you and only the exam pages and the cheat-sheet on your desk (one double-sided cheat sheet allowed).
- You must have the camera and the video enabled. Make sure that you have the bandwidth required to support this. Your mike should on mute, but you can unmute and use it to ask special question of questions of general interest
- For questions, Piazza will be on, but Chat is much better. Also, you can use chat to ask for the cell-phone number of a TA: if everything else fails, my office number is 1-310-454-7120.

Problem A. 20 points, 5 points each question. Given the ER-diagram

Problem A: Question A1



A1. Generate a BCNF schema with minimal relation count and show the relations with all their keys underlined (as in $\texttt{Mytable}(\underline{\texttt{A1}}, \texttt{A2}, \texttt{A3}, \ldots)$.

Answer: $\texttt{Emp}(\underline{\texttt{eid}}, \texttt{name}, \texttt{level})$; $\texttt{Dept}(\underline{\texttt{did}}, \texttt{location}, \texttt{mgrid})$; $\texttt{Works\_in}(\underline{\texttt{eid}}, \texttt{did})$
The first and the last table have the same key. They can be joined without loosing any FD (FK constraints are also preserved.) producing:

$$\texttt{Emp}(\underline{\texttt{eid}}, \texttt{name}, \texttt{level}, \texttt{did});\quad \texttt{Dept}(\underline{\texttt{did}}, \texttt{location}, \texttt{mgr})$$

A2. Describe in words how the E-R changes if an employee can work in several departments.

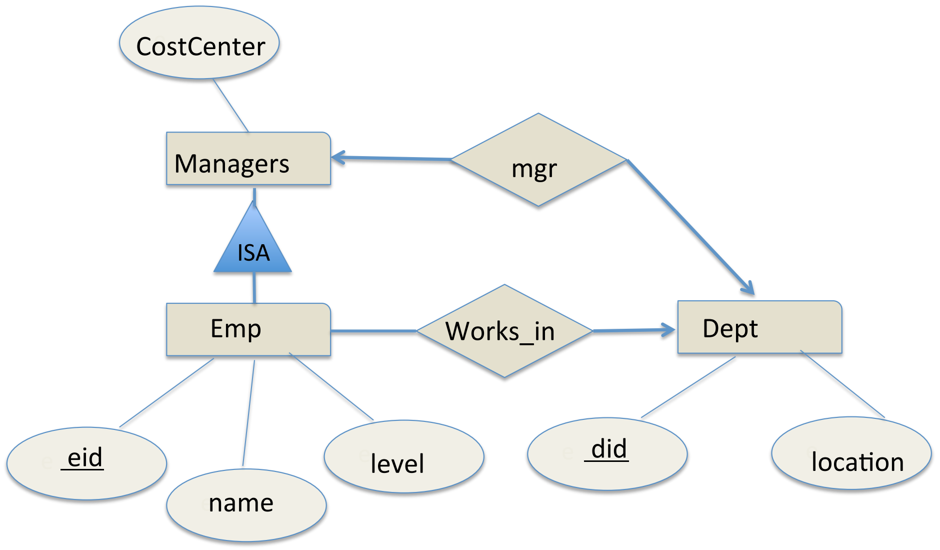
Answer: The head of the arrow leading from Works_in into Dept will be removed.

A3. Generate a BCNF schema with minimal relation count and keys underlined for the ER diagram modified as described in Q2.

Answer: $\texttt{Emp}(\underline{\texttt{eid}}, \texttt{name}, \texttt{level})$; $\texttt{Dept}(\underline{\texttt{did}}, \texttt{location}, \texttt{mgr})$; $\texttt{Works\_in}(\underline{\texttt{eid}, \texttt{did}})$
This has a minimal BCNF count since a join of Emp and Dept would produce violations of BCNF: e.g. $\texttt{did} \rightarrow \texttt{location}$ where $\texttt{did}$ is not a key.

A4. Consider now the ER-diagram below where Managers have an additional attribute (say `CostCenter`) which regular employees do not have (however managers are still considered to work in the department they manage).

Problem A: Question A4



From this new ER diagram, generate a BCNF schema with minimal relation count and keys underlined

Answer: Directly from the ER-diagram we can derive the following schema:

$\texttt{Emp}(\underline{\texttt{eid}}, \texttt{name}, \texttt{level}, \texttt{did});\ \texttt{Dept}(\underline{\texttt{did}}, \texttt{location}, \underline{\underline{\texttt{eid\_of\_mgr}}}); \texttt{Mgr}(\underline{\texttt{mgreid}}, \texttt{CostCenter}).$

But the ER-diagram shows that the relationship between Managers and Dept is one-to-one whereby the last two relations can be joined while preserving BCNF. Thus the answer is:

$$\texttt{Emp}(\underline{\texttt{eid}}, \texttt{name}, \texttt{level}, \texttt{did});\quad \texttt{Dept}(\underline{\texttt{did}}, \texttt{location}, \underline{\underline{\texttt{eid\_of\_mgr}}}, \texttt{CostCenter})$$

Problem B. 20 Points: 5 per question.

Consider the relational scheme $R(A, B, C, D, E)$ with the following FDs:

1. $AB \rightarrow C$
2. $C \rightarrow B$
3. $C \rightarrow D$
4. $C \rightarrow E$
5. $ED \rightarrow E$
6. $B \rightarrow D$

B1. Is any of these FDs trivial?

Answer: FD 5 is trivial. We must ignore it in answering the remaining questions.

B2. Is $R(A, B, C, D, E)$ BCNF? (You must explain the reasons for your answer.`Please test the FDs in reverse order, i.e. starting from FD 6 and moving up .`

Answer: Thus consider 8. $B \rightarrow D$. Now $B+ = \{B, D\}$ Thus B is neither a key, nor a superkey: BCNF is violated

B3. Provide a lossless decomposition of the original schema into a BCNF schema (underline the keys of the BCNF relations). `Test the FDs in reverse order!`

Answer: For FD 6 we found that $B+ = \{B, D\}$ and we decomposed R into:
$r1(B, D); r2(A, B, C, E)$
Now in r2 we test FD 4. and find $C+ = \{C, B, E\}$: A is missing, and $r2$ is not BCNF.
We decompose r2 into: $r21(C, B, E)$; $r22(C, A)$.
Now in $r21$, FD 3 no longer holds, and C is a key. Thus we got our BCNF relations, and we underline the keys:
$r21(\underline{B}, D)$; $r21(\underline{C}, B, E)$; $r22(\underline{C, A})$.

B4. Is the BCNF decomposition you just produced FD-preserving? (Explain the reasons for your answer.)

Answer: FDs 2, 4 and 6 are preserved by the keys. 3 is the composition of 2 and 6 and therefore it is preserved. Now, using FDs 2,4, 6 and even 3 we can compute $AB+ = \{A, B, D\}$. Since C is not there, FD 1 is lost.

Problem C 20 Points: 5 poins per question. T1 is the oldest transaction, and T3 is the youngest.

T1	T2	T3
write D		
		read D
	write C	
read C		
		write A
		read C
	read A	
write A		

C1. Is this first schedule conflict-serializable? (Justify your answer using the applicable graph.)

Answer: The serializability graph as the following arcs: $T1 \rightarrow T3$ on D, and $T3 \rightarrow T1$ on A. This is a cycle and thus this is not a conflict-serializable schedule.

In the next two questions, assume that the transaction manager uses a **strict 2PL protocol** where each shared/exclusive lock is set just before the read/write action requiring it. The transactions commit immediately they complete their last read or write.

C2. If we do not use any deadlock prevention strategy, will the resulting transactions (i) complete, or (ii) deadlock? If your answer is (i), show a completed schedule; if your answer is (ii), then show the schedule up to the deadlock.
Also state clearly and concisely what happen to each transaction: e.g., abort, restart, wait-for resource, kill other transactions, etc.

Answer: The sequence is: T1: X-lock(D), T3 blocks on S-lock(D), T2: X-lock(C), T1: blocks on S-lock(C), T2: S-lock(A) and completes. Then T1 completes; finally, T3 completes as well.

C3. If we use a wound-wait deadlock prevention strategy will the resulting transactions (i) complete, or (ii) deadlock? If your answer is (i), show a completed schedule (there could be more than one–any correct one will do); if your answer is (ii), then show the schedule up to the deadlock.

Answer: The resulting schedule is as follows: T3 waits for T1 to release the lock on D. Then by writing on C, T1 forces T2 to roll back. Thus T1 completes first, ad then T3 completes and finally T2. (But if T2 is restated quickly, then T3 will wait for T2 to finish its read A, before it can complete.

C4. Now assume that we are using a timestamp-based concurrency control method, with Thomas' write rule. Which transactions will abort (and why) and which will complete?

Answer: T1 will abort on read-C which has a write-timestamp that of T2. Then T2 will abort on read A which has as write timestamp that of T3. Thus T3 complete.

Then T1 and T2 will restart and complete in some order.

Problem D: 20 points: 2 per question. Answer the questions by a list such as D1:Yes|No, D2:Yes|No, . . . , D10:Yes|No. An explanation is not required.

In general, what are the reasons for normalizing relational schemas?

D1 Removing update anomalies. **Answer:** Yes

D2 facilitating maintenance of integrity constraints: **Answer:** Yes

D3 Making query execution more efficient: **Answer:** No

What does a conservative 2PL protocol assures?

D4 Absence of dirty reads? **Answer:** No

D5 Absence of deadlocks? **Answer:** Yes

In Transaction Recovery after a crash:

D6 The Deferred Update log protocol eliminates the need for any REDO. **Answer:** No

D7 The execution of a CHECKPOINT forces all the running transactions to commit or abort. **Answer:** No (it instead force the buffers for all running transactions to be written out to safe storage (i.e., disk).

Our Database Transactions use a Time-Stamp based protocol:

D8 Our transactions use a Time-Stamp based protocol.
Can these transactions incur a deadlock situation? **Answer:** No (Time-stamped based protocols are free of deadlocks).

D9 Do rolled-back transactions keep their original timestamp? **Answer:** No

D10 Does this protocol guarantee cascadeless schedules? **Answer:** No

Problem E: 20 points: 5 points per question.

Recovering from a crash using the immediate database modification protocol, the database system finds the following log (stored in three pages):

```
T0 start
T0  C, 900, 700
T1 start
Checkpoint L1
-------------log page boundary
T2 start
T1 C, 700, 955
T1 B, 400, 800
T1 commit
T3 start
T0 Commit
Checkpoint L2
T2 A, 160, 200
-------------log page boundary
T3 B, 880, 600
T4 start
T4 C, 955, 2400
T5 start
T4 commit
T5 D, 2100, 4000
```

E1. For each checkpoint the recovery system maintains a list of transactions that satisfy a certain propert. What property is that, and which are the the transactions in the lists L1 and L2 shown next to the two `Checkpoint` in the above schedule?

Answer: That is the list of the current transactions, i.e. of those that have not committed yet. Thus: L1= [T0, T1], L2= [T2, T3].

E2. How many pages of the log will the system read for recovery (explain)?

Answer: Walking back from the end, at the last checkpoint T2 and T3 are still uncommitted. To find their start we must visit the first page.**Answer:** 2 pages.

E3. Which transactions will be undone and which will be redone?

Answer: Undone(no commit): T2, T3, T5 .
Redone (commit after last checkpoint): T4
Ignored (commit before last checkpoint): T0, T1.

E4. What are the values of `A`,`B` and `C` be at the end of a successful recovery (explain the reason for your statement)?

Answer: A=160 because (T2 A, 160, 200) which was then undone;
B=880, because (T3 B, 880, 600) which was then undone;

C= 2400, because (T4 C, 955, 2400) was redone
D= 2100 since (T5 D, 2100, 4000) was undone.