HW5: Database design and transactions

TOTAL points possible: 100 = 15+20+33+12+20

(1) [15 pts] Consider the following instance of a relational table S with attributes X,Y,Z. Then circle the correct answer (yes / no) plus add the explanation for your answer to (b).

S

X	Y	Z
a	20	T
a	10	F
b	30	T
c	30	T
b	20	T

(a) Which of the following dependencies are violated by the instances of S?

i. [2 points] Yes No : $X \rightarrow Y$ is violated.
ii. [2 points] Yes No : $Z \rightarrow X$ is violated.
iii. [2 points] Yes No : $Y \rightarrow Z$ is violated.
iv. [2 points] Yes No : $XY \rightarrow Z$ is violated.
v. [2 points] Yes No : $YZ \rightarrow X$ is violated.
vi. [2 points] Yes No : $XZ \rightarrow Y$ is violated.

(b) [3 points] By only observing the instance of S in Table 1, can you identify the functional dependencies that hold on schema S? Why? Write your answer below.

Yes No

(2) [20 points] For this set of questions, consider the table X with attributes {A, B, C, D, E, F}. Let the following functional dependencies *FD* be defined over the relation *X*:

$$A \rightarrow B$$
$$B \rightarrow CD$$
$$E \rightarrow F$$

(a) [2 points] Provide the attribute closure of *{AB}*.

(b) Consider the decomposition AB, BCD, EF, i.e. we have three tables, one of which has attributes {A,B}, etc. Mark 'True' or 'False':

i. [3 points] True False : It is lossless

ii. [3 points] True False : It is dependency-preserving

(c) Consider the decomposition $AB, BCDF, EF$. Mark 'True' or 'False':

iii. [3 points] True False : It is lossless

iv. [3 points] True False : It is dependency-preserving

(d) Consider the decomposition $ABCEF, EBD$. Mark 'True' or 'False':

v. [3 points] True False : It is lossless

vi. [3 points] True False : It is dependency-preserving

(3) [33 points] Consider the relation E with attributes {*P, Q, R, S*}. Suppose that the following functional dependencies hold:

$$PQ \rightarrow R \quad (7)$$
$$PQ \rightarrow S \quad (8)$$
$$R \rightarrow P \quad (9)$$
$$S \rightarrow Q \quad (10)$$

(a) [6 points] List *all* the candidate key(s) for *E*.

(b) [2 points] Is the relation *E* in BCNF? Yes No

(c) From the list below, select all applicable choices to justify whether E is (or is not) in BCNF. Note: when we refer to the "main requirement" for BCNF, we mean: every determinant is a super key.

i. [1 point] True False : All FD's satisfy the main requirement.
ii. [1 point] True False : FD (7) violates the main requirement.
iii. [1 point] True False : FD (8) violates the main requirement.
iv. [1 point] True False : FD (9) violates the main requirement.
v. [1 point] True False : FD (10) violates the main requirement.

(d) [2 points] Is the relation *E* in 3NF? Yes No

(e) From the list below, select all applicable choices to justify whether E is (or is not) in 3NF. Note: when we refer to the "secondary requirement" for 3NF, we mean: for every FD X → A, the dependent A is part of a candidate key.

vi. [1 point] True False : All FD's satisfy the secondary requirement.
vii. [1 point] True False : FD (7) violates the secondary requirement.
viii. [1 point] True False : FD (8) violates the secondary requirement.
ix. [1 point] True False : FD (9) violates the secondary requirement.
x. [1 point] True False : FD (10) violates the secondary requirement.

(f) [5 points] Give a 3NF decomposition of E that is lossless, dependency preserving, and has as few tables as possible.

(g) [8 points] Give a BCNF decomposition of E that is lossless, and has as few tables as possible.

(4) [12 pts] Serializability and 2PL: Consider the schedule given below in Table 1 with 3 transactions. R(·) and W(·) stand for 'Read' and 'Write', T_x for transaction x, and t_x for time x.

time	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
T_1	R(A)	W(A)	W(E)	R(B)		W(B)	R(C)	W(C)	R(D)	W(D)
T_2			R(A)		W(A)	R(D)	W(D)	W(E)		W(A)
T_3				R(C)	W(C)			R(F)	W(F)	

(a) [1 point] Is this schedule serial?

Yes No

(b) [4 points] Give the dependency graph of this schedule. List each edge in the dependency graph like this: $T_x \rightarrow T_y$ because of Z (i.e., Transaction T_y reads/writes Z which was last written by T_x.). Order the edges in ascending order with respect to x.

(c) [1 point] Is this schedule conflict serializable?

Yes No

(d) [3 points] If you answer "yes" to (iii), provide the equivalent serial schedule. If you answer "no", briefly explain why.

(e) [3 points] Is this schedule possible under 2PL?

Yes No

(5) [20 pts] Deadlock Detection and Prevention. Consider the following lock requests where

- S(·) and X(·) stand for 'shared lock' and 'exclusive lock', respectively.
- T_1, T_2, T_3, and T_4 represent four transactions.
- LM represents a 'lock manager'.
- Make the assumption that transactions will never release a granted lock.

time	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8
T_1	S(A)		S(B)					
T_2		X(B)						X(D)
T_3				S(C)	X(D)		X(A)	
T_4						X(C)		
LM	g							

(a) [3 points] For the lock requests in Table 3, determine which lock request will be granted, blocked or aborted by the lock manager (LM), if it has no deadlock prevention policy. Please write 'g' for grant (the LM allows this lock given all prior locks) and 'b' for block (the LM does not allow this lock). An example is given in the first column where the lock manager grants the shared lock to T1. Insert your answers into the table above.

(b) [4 points] Give the wait-for graph for the lock requests in above table. List each edge in the graph like this: $T_x \rightarrow T_y$ because of Z (i.e., T_x is waiting for T_y to release its lock on resource Z). Order the edges in ascending order with respect to x.

(c) [3 points] Determine whether there exists a deadlock in the lock requests in Table 3, and briefly explain why.

1. Collaboration Policy

We randomly assign you to groups of 2 or 3 students on Blackboard. You will have to submit your individual homework, but we allow you to exchange ideas and brainstorm on solution ideas *within* the randomly assigned group of students. If you received help from one of your assigned team mates, then acknowledge the help of your team mate at the end of this file. Any received help from within the group does not count against you. At the end of the semester, we ask everyone to rate the helpfulness of the team mates you were paired up throughout the semester.

New for this homework, you are also allowed to collaborate with *anybody* during office hours and in the presence of the TAs. Thus, you can come to office hours, freely discuss solutions with any other student, finish your homework with others together, it just needs to be during office hours. We only ask you to acknowledge the other students with whom you collaborated (again, we will *not* deduce any points from your homework solution for you having collaborated with anybody during office hours and acknowledged them).

2. Submission policy

Please complete the homework on paper and then scan your solutions to PDF, e.g. by creating pictures from your submission and transforming them to PDF. A tool that I highly recommend and we are using in my lab for scanning handwritten proofs and scribbles is "Office Lens" by Microsoft. You can find it for free in the Apple and Android stores (I am not paid my Microsoft...)

3. Acknowledgements

Add here your acknowledgements