Weilan Yu 005119471

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.

Total number of pages submitted
5

A

1. Emp (eid, name, level, did)
   Dept (did, location, mgr_id) mgr_id → eid

2. If an employee can work in several department, then the arrow from works_in relationship to Dept entity would be a straight line.

3. Emp (eid, name, level)
   Dept (did, location, mgr_id) mgr_id → eid
   Works_in (eid, did)

4. Emp (eid, name, level)
   Dept (did, location)
   mgr (mgr_id, did) mgr_id → eid
   Managers (costcenter, eid)

B.

1. Yes. 5 is trivial

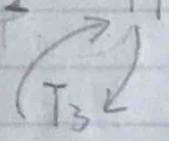
2. No, because $B^+ = \{B, D\}$ not a key

3. $B^+ = \{B, D\} \Rightarrow (\underline{B}, D)\ (A, B, C, E)$
   $C^+ = \{B, C, D, E\} \Rightarrow (B, \underline{C}, D, E)\ (\underline{A}, \underline{C})$
   So we have $(\underline{B}, D)$, $(B, \underline{C}, D, E)$, $(\underline{A}, \underline{C})$

4. Not I-D-preserving.
$B^+ = \{D\}$ $F_1 = \{B \to D\}$
$C^+ = \{B, C, D, E\}$ $F_2 = \{C \to BDE\}$
We are missing $AB \to C$.

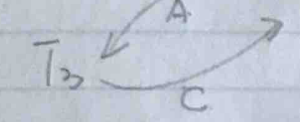
C.

1. No
$T_1 \to T_3$ at D
$T_2 \to T_1$ at C
$T_3 \to T_2$ at A
$T_2 \to T_1$ at A
$T_3 \to T_1$ at A
$\Rightarrow$



There's a cycle, so it's not conflict-serializable.

2. There's a deadlock. T_3 obtains lock-X(A). and $T_1 \xrightarrow{C} T_2$ wait for lock(C), then T_2 waits for lock(A).



T_1 wait-for T_2 to release lock(C).
T_3 wait-for T_2 to release lock(C)
T_2 wait-for T_3 to release lock(A)

3. Complete

T_2 holds lock-X(C), T_1 is older transaction. it wounds T_2 and get lock-S(C). T_2 roll back. T_1 would hold lock-X(C) until it finish all transaction for strict 2PL protocal. T_3 holds lock-X(A), then requests lock-S(C). it will wait for A to finish. T_1 wounds T_3 and get lock(A). T_1 finishes, then T_2, T_3.

A possible schedule can be T_1, T_2, T_3

4. $TS(t_1) < Ts(t_2) < Ts(t_3)$

$W\_TS(D) := t_1$

$t_3 > W\_TS(D)$ Ok

$W\_TS(C) := t_2$

$t_1 < W\_TS(C)$ Do nothing

$W\_TS(A) := t_3$

$t_3 > W\_TS(C)$ Ok

$t_2 <= W\_TS(C)$ abort

$t_1 < W\_TS(A)$ abort

So T_1, T_2 will abort, T_3 will complete.

D

D_1	Yes	D_4	NO	D_6	NO	D_8	NO
D_2	NO	D_5	Yes	D_7	Yes	D_9	Yes
D_3	NO					D_{10}	NO

E

1. Transactions are in active state
   $L_1 = (T_0, T_1)$
   $L_2 = (T_2, T_3)$

2. 2.
   In the last checkpoint, L_2, T_2, T_3 haven't commited. So we only need to check 2 log page.

3. Undone: T_5, T_2, T_3
   Redone: T_4.

4. From (3), T_4 will be redone. So C will be recovered successfully.
   $C = 2400$