Tutorial

Week 3

1. Examine the table shown below.

branchNo	branchAddress	telNo	mgrStaffNo	mgrName
B001	8 Jefferson Way, Portland, OR 97201	503-555-3618	S1500	Tom Daniels
B002	City Center Plaza, Seattle, WA 98122	206-555-6756	S0010	Mary Martinez
B003	14 - 8th Avenue, New York, NY 10012	212-371-3000	S0145	Art Peters
B003	14 - 8th Avenue, New York, NY 10012	212-371-3000	S0306	Jane Smith
B004	16 - 14th Avenue, Seattle, WA 98128	206-555-3131	S2250	Sally Stern

a) Which normal form is this table in? Why?
b) Describe and illustrate the process of normalizing the data shown in this table to third normal form (3NF).
c) Identify the primary and foreign keys in your 3NF relations.

a) Which normal form is this table in? Why?

Solution:

The functional dependencies in this relation are:

Fd1: branchNo -> branchAddress, telNo

Fd2: mgrStaffNo -> mgrName

Candidate key is (branchNo, mgrStaffNo) so primary key is (branchNo, mgrStaffNo).

Both Fd1 and Fd2 are partical dependencies on primary key attributes (violates the condition for 2NF), so this relation is in 1NF.

b) Describe and illustrate the process of normalizing the data shown in this table to third normal form (3NF).
c) Identify the primary and foreign keys in your 3NF relations

Solution:

b) To normalize a relation to 3NF, the following steps are needed:

1. identify functional dependencies for each relation
2. identify primary keys for each relation
3. remove partial dependencies and transitive dependencies on primary key by removing the partially/transitively dependent attributes to a new relation, with a copy of their determinants.

Both Fd1 and Fd2 are partial dependencies on primary key, so removing them into:

Branch(branchNo, branchAddress, telNo)

MgrStaff(mgrStaffNo, mgrName)

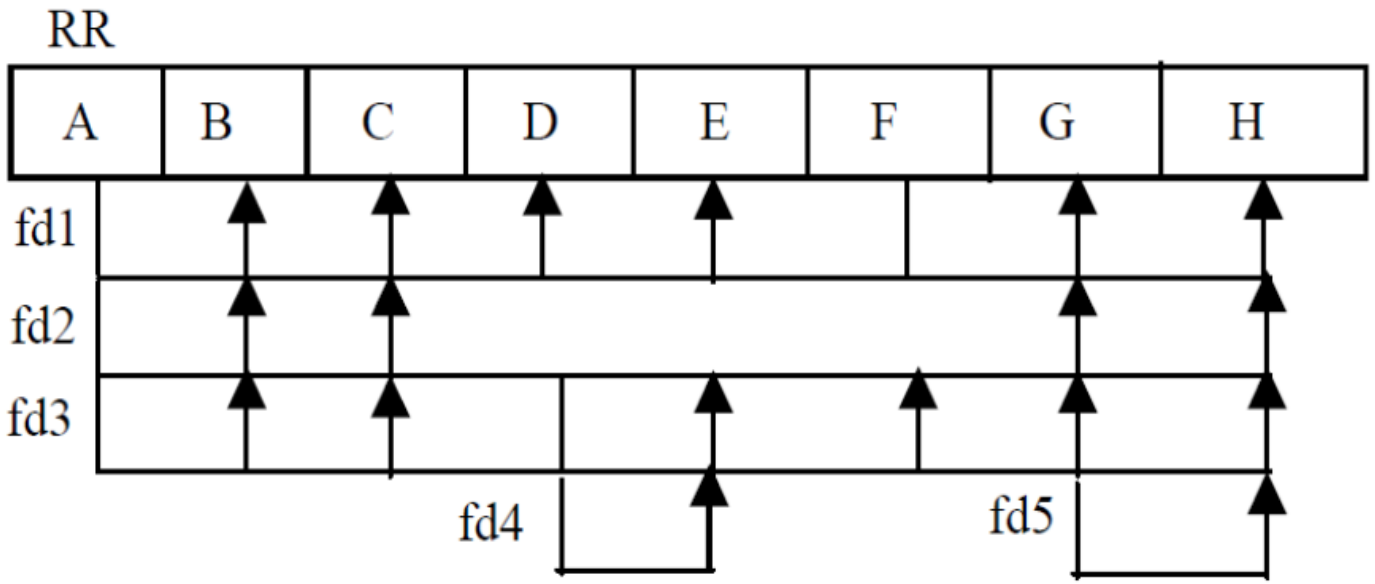
BranchManager(branchNo, mgrStaffNo)

The above relations now have no transitive or partial dependencies, so they are in 3NF.

c) Primary key for Branch is branchNo, Primary key for MgrStaff is mgrStaffNo, Primary key for BranchManager is (branchNo, mgrStaffNo)

Foreign key for BranchManager relation is branchNo (referencing to Branch relation) and mgrStaffNo (referencing to MgrStaff relation).

2.Given the following relation schema and its functional dependencies:



a) Specify candidate keys and state the primary key.

b) Assuming that the relation is in first normal form (1NF), describe and illustrate the process of normalizing the relation schema to second (2NF) and third (3NF) normal forms. Identify the primary and foreign keys in your schemas.

a) Specify candidate keys and state the primary key.

b) Assuming that the relation is in first normal form (1NF), describe and illustrate the process of normalizing the relation schema to second (2NF) and third (3NF) normal forms. Identify the primary and foreign keys in your schemas.

Solution:

a) Functional dependencies are:

Fd1: A, F -> B,C,D,E,G,H

Fd2: A -> B, C, G,H

Fd3: A, D -> B,C,E,F,G,H

Fd4: D -> E

Fd5: G -> H

Candidate keys are (A, F) and (A, D), and choose (A, F) as primary key.

b) (*See question 1 b) solution for the description of normalizing to 2NF and 3NF*)

Fd2 is a partitial dependency on primary key, and fd4 and fd4 are transitive dependencies.

First removing partial dependency Fd2, and get new relations:

R1 (A, B, C, G, H) primary key is A, R2 (A, D, E, F) primary key is (A, F)

Relation R1 and R2 are now in 2NF.

R1 has transitive dependency Fd5, R2 has transitive dependency Fd4, so removing them to new relations:

R3 (G, H) primary key is G, R4 (A, B, C, G) primary key is A, foreign key is G referencing to R3,

R5 (D, E) primary key is E, R6 (A, D, F) primary key is A, F, foreign key is D referencing to R5.

Relation R3, R4, R5 and R6 are now in 3NF.

3. Answer the following questions for the schedule of two transactions T1 and T2.

Time	T1	T2
t_1		begin_transaction
t_2		read(X)
t_3		X = X+100
t_4	begin_transaction	write(X)
t_5	read(X)	…
t_6	X = X*10	rollback
t_7	write(X)	
t_8	commit	

a) What is the problem of updating X in the two transactions T1 and T2? Explain your answer.

b) Rewrite T1 and T2 in **Figure 1** using Two-Phase Locking (2PL).

a) What is the problem of updating X in the two transactions T1 and T2? Explain your answer.

Solution:

- This is the **uncommitted dependency problem** caused by **concurrency**.
- This problem occurs when one transaction can see intermediate results of another transaction before it has committed.

(to be continued)

Time	T1	T2	X
t_1		begin_transaction	100
t_2		read(X)	100
t_3		X = X+100	100
t_4	begin_transaction	write(X)	200
t_5	read(X)	…	200
t_6	X = X*10	rollback	100
t_7	write(X)		2000
t_8	commit		2000

Continued solution:

For example:

Suppose X is 100.

T2 updates X to 200 but it aborts, so X should be back at original value of 100.

T1 has read new value of X (200) and uses value as basis of *10, giving a new balance of **2000, instead of 1000**.

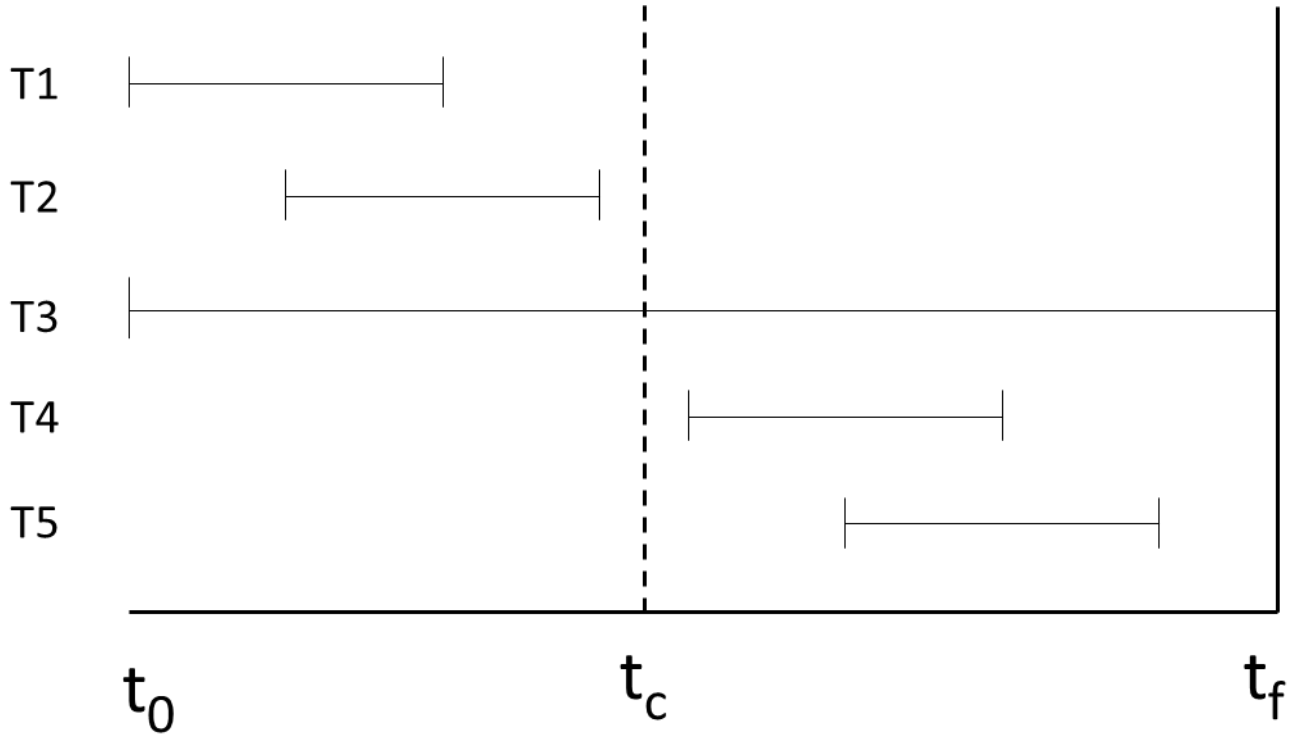
b) Rewrite T1 and T2 in **Figure 1** using Two-Phase Locking (2PL).

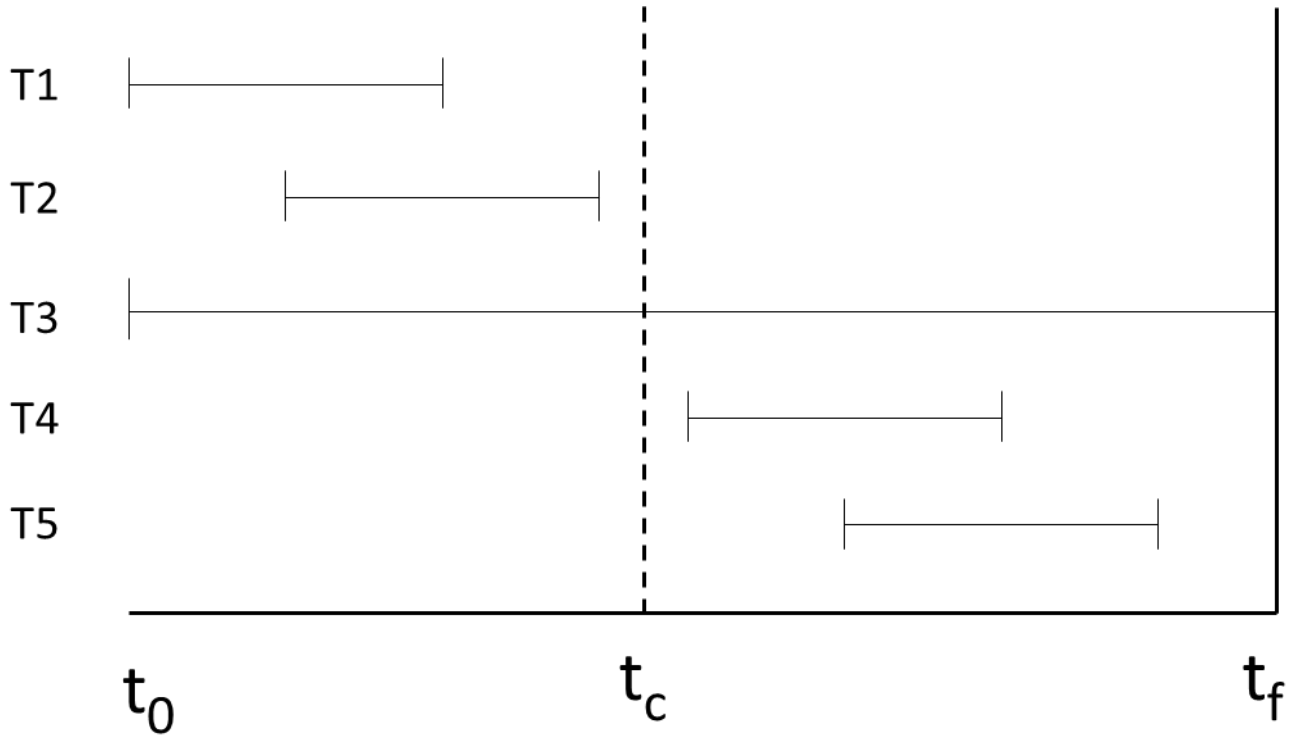
Solution: (Red part indicates the operations for 2PL)

Time	T1	T2
t_1		begin_transaction
t_2		read(X)
t_3		X = X+100
t_4	begin_transaction	write(X)
t_5	read(X)	…
t_6	X = X*10	rollback
t_7	write(X)	
t_8	commit	

Time	T1	T2
t_1		begin_transaction
t_2		Write_lock(X)
t_3		read(X)
t_4		X = X+100
t_5	begin_transaction	write(X)
t_6	Write_lock(X)	…
t_7	WAIT	Rollback/Unlock(X)
t_8	read(X)	
t_9	X = X*10	
t_{10}	write(X)	
t_{11}	Commit/Unlock(X)	

4. The following figure illustrate a number of transactions being processed from t0 until tf when system had a power failure. At tc a checkpoint was done. Explain for each transactions in the figure below how recovery is done.





Solution:

- Checkpoint at time tc, changes made by T1 and T2 have been written to secondary storage.
- Thus: only redo T4 and T5, undo transactions T3