ASSIGNMENT 2

QUESTION 1

Consider a relation $R(A, B, C, D, E, G, H, I, J)$ and its FD set F = $\{A \rightarrow DE, B \rightarrow GI, E \rightarrow CD, CE \rightarrow ADH, H \rightarrow G, AH \rightarrow I\}$.

1) **Check if $A \rightarrow I \in F^+$.**

   To check if $A \rightarrow I \in$ F^+ we need to find A^+
   So,
   A^+ = ADECHGI
   Therefore $A \rightarrow I \in F^+$

2) **Find a candidate key for R.**

   B^+ = BGI $\notin F$
   A^+ = ADECHGI $\notin F$
   J^+ = J $\notin F$
   $(ABJ)^+$ = ABJDEGICH $\in F$
   $(BEJ)^+$ = BEJGICDAH $\in F$
   $(BC)^+$ = BCGI $\notin F$

   Hence we have two candidate keys in the given relation R as follows
   a. ABJ
   b. BEJ

3) **Determine the highest normal form of R with respect to F. Justify your answer.**

   Candidate keys are ABJ, BEJ
   Hence prime attributes are A, B, E, J and non-prime attributes are C, D, G, H, I
   <u>Checking for 2NF</u>

   $F = \{ A \rightarrow DE, B \rightarrow GI, E \rightarrow CD \}$ shows that non-prime attributes are not fully functionally dependent on the candidate keys.
   Hence the given relation is not in 2NF and thus cannot be in any higher normal form than Second Normal Form (2NF).

   Therefore the highest normal form of the relation R is 1NF.

4) Find a minimal cover F_m for F.

$F = \{A \rightarrow DE, B \rightarrow GI, E \rightarrow CD, CE \rightarrow ADH, H \rightarrow G, AH \rightarrow I\}$.

Step 1: Split the FDs such that R.H.S contains a single attribute

F_m = { A → D, A → E, B → G, B → I, E → C, E → D, CE → A, CE → D, CE → H, H → G, AH → I}

Step 2: Find the redundant FDs and delete them from the set

A → D and E → D are redundant because we have A^+ = AECDHGI and E^+ = ECADHGI
Hence after deleting them we get

F_m = { A → E, B → G, B → I, E → C, CE → A, CE → D, CE → H, H → G, AH → I }

Step 3: Find the redundant attributes on L.H.S and delete them from the set

In CE → A, CE → D, CE → H. C is redundant as we have E^+ = EC
In AH → I. H is redundant as we have A^+ = ADECHGI
Hence after deleting them we get

F_m = { A → E, B → G, B → I, E → C, E → A, E → D, E → H, H → G, A → I }

Now we can combine them into more simpler FDs

Therefore
F_m = { A → EI, B → GI, E → ACDH, H → G }

5) Decompose into a set of 3NF relations if it is not in 3NF step by step. Make sure your decomposition is dependency-preserving and lossless-join.

Using the minimal cover we have F_m = { A → EI, B → GI, E → ACDH, H → G }
Decomposing F_m into relations we get
R_1 (A, E, I)
R_2 (B, G, I)
R_3 (A, E, C, D, H)
R_4 (H, G)

Also we can easily combine R1 and R3 as A and E are common prime attributes, we get our new relations as :-
R_1 (B, G, I)
R_2 (A, E, C, D, H, I)
R_3 (H, G)

Dependency-preserving check

Here, the given relation $R(A, B, C, D, E, G, H, I, J)$ with FD set $F = \{A \rightarrow DE, B \rightarrow GI, E \rightarrow CD, CE \rightarrow ADH, H \rightarrow G, AH \rightarrow I\}$ is decomposed into R_1 (B,G,I) and R_2 (A,E,C,D,H,I) and R_3 (H,G) which is dependency preserving because FD
$F = \{A \rightarrow DE\}$ is a part of R_2 (A, E, C, D, H, I).
$F = \{B \rightarrow GI\}$ is a part of R_1 (B, G, I).
$F = \{E \rightarrow CD\}$ is a part of R_2 (A, E, C, D, H, I).
$F = \{CE \rightarrow ADH\}$ is a part of R_2 (A, E, C, D, H, I).
$F = \{H \rightarrow G\}$ is a part of R_3 (H, G)
$F = \{AH \rightarrow I\}$ is a part of R_2 (A, E, C, D, H, I)

Hence the decomposition is dependency-preserving.

Lossless-join check

	A	B	C	D	E	G	H	I	J
R_1	b	a	b	b	b	a	b	a	b
R_2	a	b	a	a	a	~~b~~ a	a	a	b
R_3	b	b	b	b	b	a	a	b	b

By the above table we get that the decomposition in lossy and hence to make it lossless we need to add another relation R4 (A, B, J) because ABJ is our candidate key.

	A	B	C	D	E	G	H	I	J
R_1	b	a	b	b	b	a	b	a	b
R_2	a	b	a	a	a	~~b~~ a	a	a	b
R_3	b	b	b	b	b	a	a	b	b
R_4	a	a	~~b~~ a	~~b~~ a	~~b~~ a	~~b~~ a	~~b~~ a	~~b~~ a	~~b~~ a

As we can see the last row is complete 'a' hence now the decomposition is lossless.
Further lossless-join checks include :-

1. $Att(R_1) \cup Att(R_2) \cup Att(R_3) \cup Att(R_4)$ = (A, B, C, D, E, G, H, I, J) = Att(R)
2. $Att(R_1) \cap Att(R_2) \cap Att(R_3) \cap Att(R_4) \neq \emptyset$

Therefore a dependency-preserving and lossless-join 3NF decomposition is

R_1 (B, G, I)
R_2 (A, E, C, D, H, I)
R_3 (H, G)
R_4 (A, B, J)

QUESTION 2

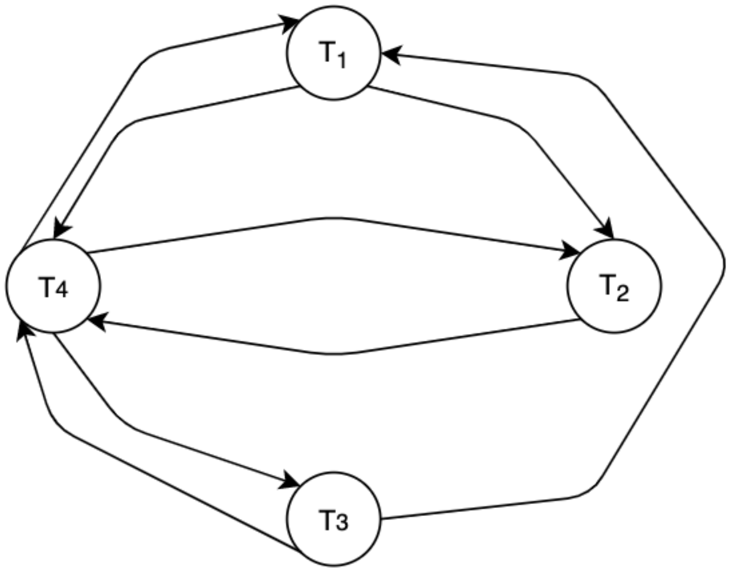
Consider the schedule below. Here, R(*) and W(*) stand for 'Read' and 'Write', respectively. T_1, T_2 , T_3 and T_4 represent four transactions and t_i represents a time slot.

Time	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}
T_1	R(B)					R(A)	W(B)				W(A)	
T_2								R(B)				W(B)
T_3			R(A)	W(A)								
T_4		R(A)			W(A)				R(B)	W(B)		

Each transaction begins at the time slot of its first Read and commits right after its last Write (same time slot).
Regarding the following questions, give and justify your answers.

1) **Is the transaction schedule conflict serializable? Give the precedence graph to justify your answer.**

<u>Precedence graph</u>



The precedence graph is cyclic in nature therefore the transaction is NOT conflict serializable.

2) **Give a serial schedule of these four transactions.**

Time	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}
T_1	R(B)	R(A)	W(B)	W(A)								
T_2					R(B)	W(B)						
T_3							R(A)	W(A)				
T_4									R(A)	W(A)	R(B)	W(B)

3) **Lock the transactions T_1 and T_2 according to the simple locking scheme. You only need to consider the order of the operations, not the timestamps.**

T_1	T_2
write_lock(B)	
R(B)	
write_lock(A)	
R(A)	
W(B)	
unlock(B)	
	write_lock(B)
	R(B)
W(A)	
unlock(A)	
	W(B)
	unlock(B)