Set1:

1. With an example explain the properties of the transactions that the database system should maintain. (2M)
   Ans: ACID properties takinh example of a schedule have 2 transactions with read and write operation
2. Find the candidate keys for the relation R(A,B,C,G,H,I) with the set of functional dependencies, F={A->B; A->C; CG->H; CG->I; B->H}. What do you mean by two functional dependency sets are equivalent?

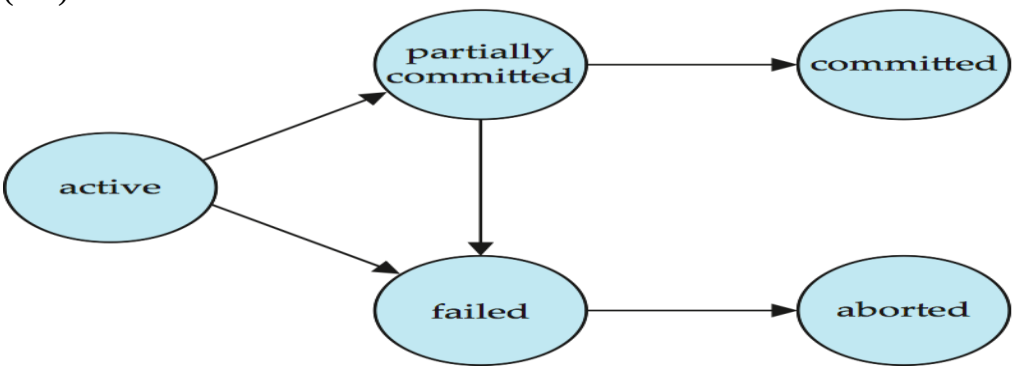
Ans: find the attributes which r not part of the RHS of any of the FDs. Here, AG. Next find the closure of it , AG+, AG+ includes all the attributes of R hence, AG is candidate key.

Next find the closure of rest of the attributes and their combinations. For eg, B+, C+, H+, I+, BC+, BH+ etc.. but none of these combinations determines all R. Hence only AG is the key.

Given F and G functional dependency sets and if F covers G and G covers F then they r equivalent. F covers G, iff all the functional dependencies of G are derivable using F set. [1M]

Set2:

2. Explain with a figure the transaction model depicting the various states of a transaction. (2M)




Explain every state.

3. Find Minimal cover for the given F : ABCD → E, E → D, AC → D, A → B **[3 Marks]**

**Ans:** Method 1:

Step 1: Reduce RHS (Here RHS of all FD's are already reduced.)

Step 2: Reduce LHS (i.e remove extraneous attributes)

Method 1: Consider ABCD, find ABCD+ = ABCDE

Now find closure of different combinations of ABCD and check whether it is possible to give you ABCD+.

i.e find out A+, B+, C+, D+, AB+, AC+, AD+, BC+, BD+, CD+ , ABC+, BCD+, CDA+

Here none of the closure gives ABCD+, hence there is no extraneous attributes in ABCD->E.

Method 2:

Consider ABCD, and Find the closure of different combinations of ABCD, and identify the one which will derive all attributes from the LHS of ABCD->E, Then we will replace ABCD->E by that one

i.e A+, B+, C+, D+, AB+, AC+, AD+, BC+, BD+, CD+ , ABC+, BCD+, CDA+

Now, ACD+ = ACDB which gives LHS of ABCD->E, hence we can replace ABCD-> E by ACD->E

Further, AC+ = ACDB, hence Replace ACD->E by AC->E

Therefore o/p of step 2 is : AC->E, E → D, AC → D, A → B

Step 3: Remove redundant FD's

Using trasitive rule we can have AC->D from AC->E and E->D

Hence Ac->D is redundant.

Hence minimal cover using Mathod 2 is AC->E, E->D, A->B

Using Mathod1, the given FD's can't be reducible.

Set3:

4. What problems occur when concurrent transactions execute in an **uncontrolled** manner?(2M)

Ans: Problems occur when concurrent transactions execute in an **uncontrolled** manner:

- The Lost Update
- The Temporary Update (or Dirty Read)
- The Incorrect Summary
- Unrepeatable Read:

With explanation

5. Consider the following relation for published books:

BOOK (Book_title, Authorname, Book_type, Listprice, Author_affil, Publisher)

Suppose the following dependencies exist:

Book_title -> Publisher, Book_type

Book_type -> Listprice

Author_name -> Author-affil

What normal form is the relation in? Explain your answer. [**1 M]**
Apply normalization until you cannot decompose the relations further. State the reasons behind each decomposition. **[2 M]**

**Ans**: The key for this relation is Book_title,Authorname. This relation is in 1NF and not in 2NF as no attributes are FFD on the key. It is also not in 3NF.

(b) Apply normalization until you cannot decompose the relations further. State the reasons behind each decomposition.

2NF decomposition:

Book0(Book_title, Authorname)

Book1(Book_title, Publisher, Book_type, Listprice)

Book2(Authorname, Author_affil)

This decomposition eliminates the partial dependencies.

3NF decomposition:

Book0(Book_title, Authorname)

Book1-1(Book_title, Publisher, Book_type)

Book1-2(Book_type, Listprice)

Book2(Authorname, Author_affil)

This decomposition eliminates the transitive dependency of Listprice

Set 4:

6. Find all possible candidate keys for the given F, F={ A->BC, B->CFH, CH->G, E->A, B->EG} **[2 Marks]**

Ans: All attributes have incoming edge. Hence, consider or start with single attribute.

A+=ABCFHGE is a key, B+=BCFHEAG is a key, C+=C is not a key,

E+=EABCFHG is a key, F+=F is not a key, G+=G is not a key, H+=H is not a key.

CF+=CF is not a key , CG+=CG is not a key, CH+=CHG is not a key, FG+FG is not a key, FH+=FH is not a key, GH+=GH is not a key.

CFG+=CFG is not a key, CGH+=CGH is not a key, FGH+=FGH is not a key, HCF+=HCFG is not a key nCr= n!/n-r! r! = 24/6= 4 combinations

CFGH==CFGH is not a key.

7. Can the following schedule be view serializable? If yes, give its view equivalent schedule.(3M)

| **T1** | **T2** | **T3** |
|---|---|---|
| R(A) | | |
| W(A) | | |
| | R(A) | |
| | R(B) | |
| W(B) | | |
| | | R ( C) |
| | R ( C) | |
| | W( C) | |

Ans: show its not conflict serializable, write the table for initial read, final update and writes(updates), write the sequence of transaction for each variable. Finally get the view serial schedule

Ans:t3->t2->t1

Set5

1. Consider R = (A, B, C, D, E), is decomposed into R1 = (A, D, E), R2 = (A, B, C), and F is { A → BC, CD → E, B → D, E → A}. Check whether this decomposition is a lossless-join decomposition(mention all the steps clearly). **[2M]**

   Ans: **Ans:** R1 ∩ R2 = A; (A → BC) => (A → ABC) => (R1 ∩ R2 → R1) => this is a lossless-join decomposition.
   **They have to show it using matrix method.**

2. Can the following schedule be view serializable? If yes, give its view equivalent schedule.(3M)

| **T1** | **T2** | **T3** |
|---|---|---|
| | R(A) | |
| | W(A) | |
| | | R (C ) |
| W(C ) | | |
| | | W (A) |
| | W ( C) | |
| R(A) | | |
| R(B) | | |
| W(A ) | | |
| W(B) | | |

Ans: show its not conflict serializable, write the table for initial read, final update and writes(updates), write the sequence of transaction for each variable. Finally get the view serial schedule

A:t2-t3-t1

B:t1

C:t3-t1-t2

Since sequence conflict, view serialiazable schedule is not possible.