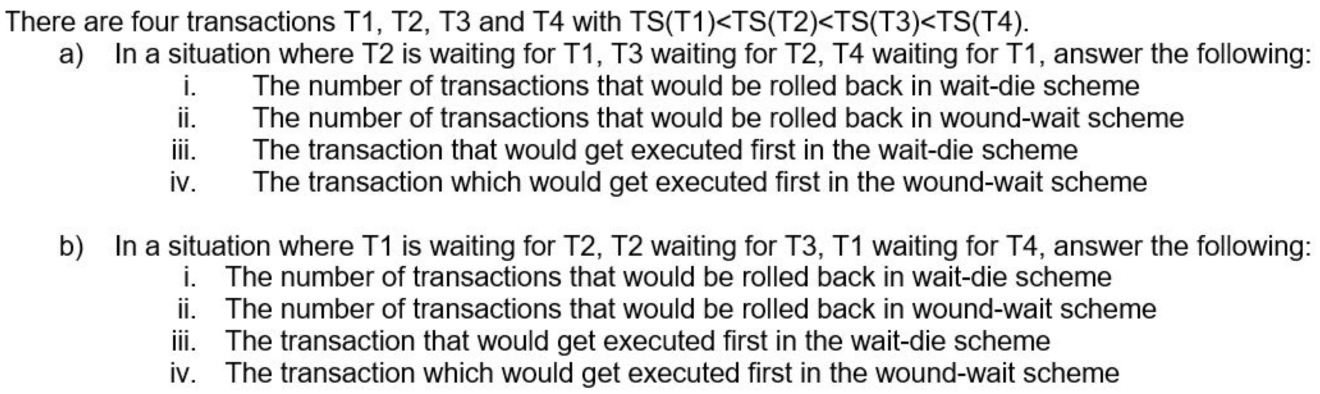
Question-wise Details

Section #1

Question 1: **Time:** 8 Min 26 Sec **Marks: 2 / 5**



There are four transactions T1, T2, T3 and T4 with TS(T1)<TS(T2)<TS(T3)<TS(T4).

a) In a situation where T2 is waiting for T1, T3 waiting for T2, T4 waiting for T1, answer the following:
   i. The number of transactions that would be rolled back in wait-die scheme
   ii. The number of transactions that would be rolled back in wound-wait scheme
   iii. The transaction that would get executed first in the wait-die scheme
   iv. The transaction which would get executed first in the wound-wait scheme

b) In a situation where T1 is waiting for T2, T2 waiting for T3, T1 waiting for T4, answer the following:
   i. The number of transactions that would be rolled back in wait-die scheme
   ii. The number of transactions that would be rolled back in wound-wait scheme
   iii. The transaction that would get executed first in the wait-die scheme
   iv. The transaction which would get executed first in the wound-wait scheme

Response:

a. i) in wait-die, the younger one will rollback in case of a dead lock and the older one waits in case of conflict

ii) 3

iii)T1 assuming that younger one wont force older one who already has resources to abort

iv)oldest gets priority hence T1

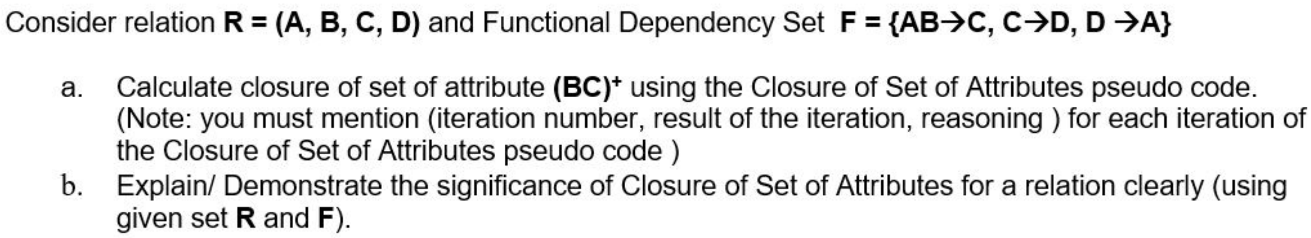
b. ii) 3

iii)T4 or T3 as they are youngest and have no conflict for resources

iv)T1 as its oldest and will force younger to abort/rollback

Words : 72

Question 2: **Time:** 6 Min 4 Sec **Marks: 3 / 5**



Consider relation **R = (A, B, C, D)** and Functional Dependency Set **F = {AB→C, C→D, D →A}**

a. Calculate closure of set of attribute **(BC)⁺** using the Closure of Set of Attributes pseudo code. (Note: you must mention (iteration number, result of the iteration, reasoning) for each iteration of the Closure of Set of Attributes pseudo code)

b. Explain/ Demonstrate the significance of Closure of Set of Attributes for a relation clearly (using given set **R** and **F**).

Response:

result = {BC}

iteration1 : take AB -> C
result : BC
reason : AB not a subset, nothing is done

2 : take C -> D
result : BCD
reason : C is a subset of result hence we add it to result

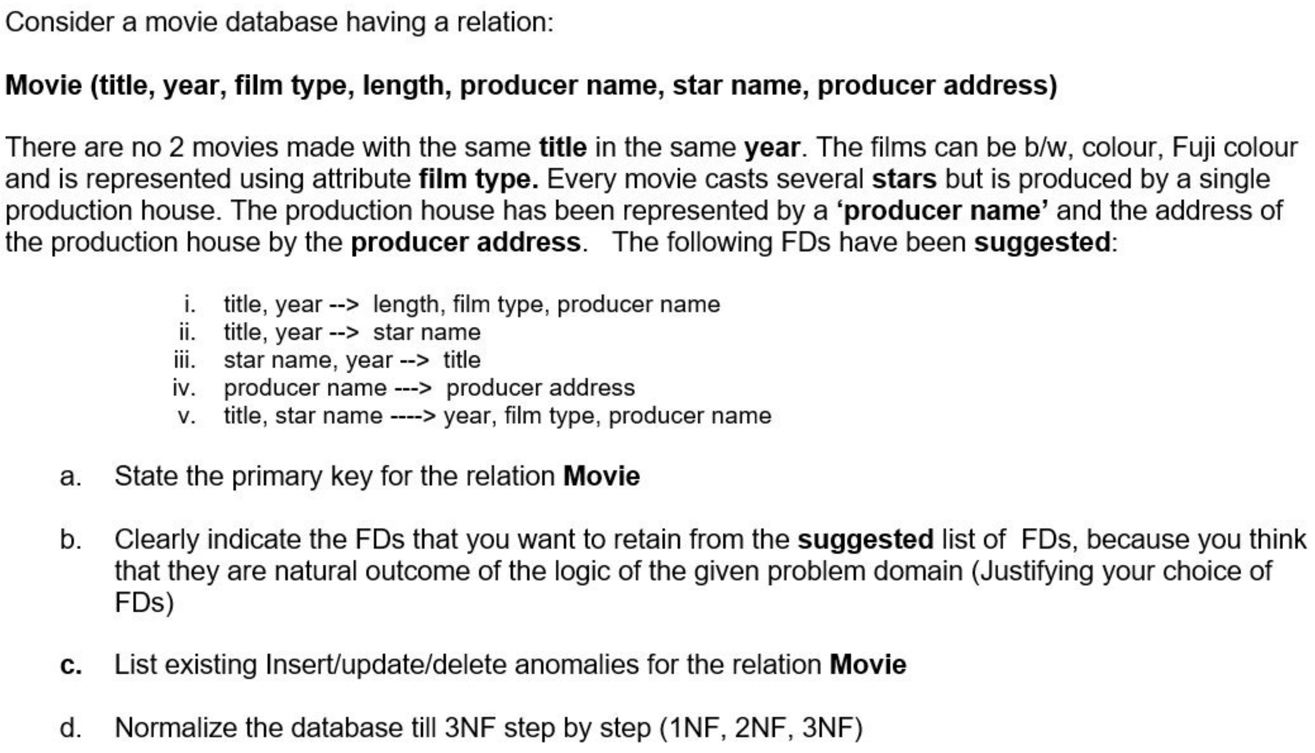
3. take D->A
result : ABCD
reason : D is a subset of result so we add A to result

no more changes to result.

Here, closure of BC is ABCD i.e, all attributes of R. So, we can take it as the Candidate Key for R as it can uniquely identify all the attributes in R. We use closure of a set of attributes to find out Candidate Keys and therefore primary keys. This helps in maintaining the database.

Words : 124

Question 3: **Time:** 22 Min 12 Sec **Marks: 1 / 5**



Consider a movie database having a relation:

Movie (title, year, film type, length, producer name, star name, producer address)

There are no 2 movies made with the same **title** in the same **year**. The films can be b/w, colour, Fuji colour and is represented using attribute **film type.** Every movie casts several **stars** but is produced by a single production house. The production house has been represented by a **'producer name'** and the address of the production house by the **producer address**. The following FDs have been **suggested**:

i. title, year --> length, film type, producer name
ii. title, year --> star name
iii. star name, year --> title
iv. producer name ---> producer address
v. title, star name ----> year, film type, producer name

a. State the primary key for the relation **Movie**

b. Clearly indicate the FDs that you want to retain from the **suggested** list of FDs, because you think that they are natural outcome of the logic of the given problem domain (Justifying your choice of FDs)

c. List existing Insert/update/delete anomalies for the relation **Movie**

d. Normalize the database till 3NF step by step (1NF, 2NF, 3NF)

Response:

1. The primary key is {title, year} as it can uniquely identify all the tuples

2. FDs :

- producer name -> producer address
    - The production house name will determine its address
- title, year -> length, film type, producer name
- title, year -> Starname
    - above two FDs will ensure that closure of PK is R
    - Every movie casts several stars
- starname, year -> title
    - this shows the movies the star has acted in, in a given year

we will remove title, starname -> year, film type, producer type as multiple stars can act in one movie and same star can act in multiple movies with same titles in different years

3. Insert Anomaly :

- Suppose you are keeping tab of older movies and you know just the title. Since Year cant be null, you wont be able to insert the details about that movie
- If two production houses collaborated on a movie, either you insert value of one production house or insert mutiple tuples.

Update Anomaly :

- If you want to the year for a given movie, you will have to do that for multiple tuples for that pair of (title, movie) with different actors

Delete Anomaly :

- Suppose a new production house has made just one movie.
- On deleting that movie tuple, you also end up deleting the information about that production house

4. The database is Movie(title, year, film type, length, producer name, star name, producer address)

1NF :

Assuming only one production house makes one movie, we divide it into:
Movie(title, year, film type, length, producer name, producer address)
Movie_star(title, year, star)

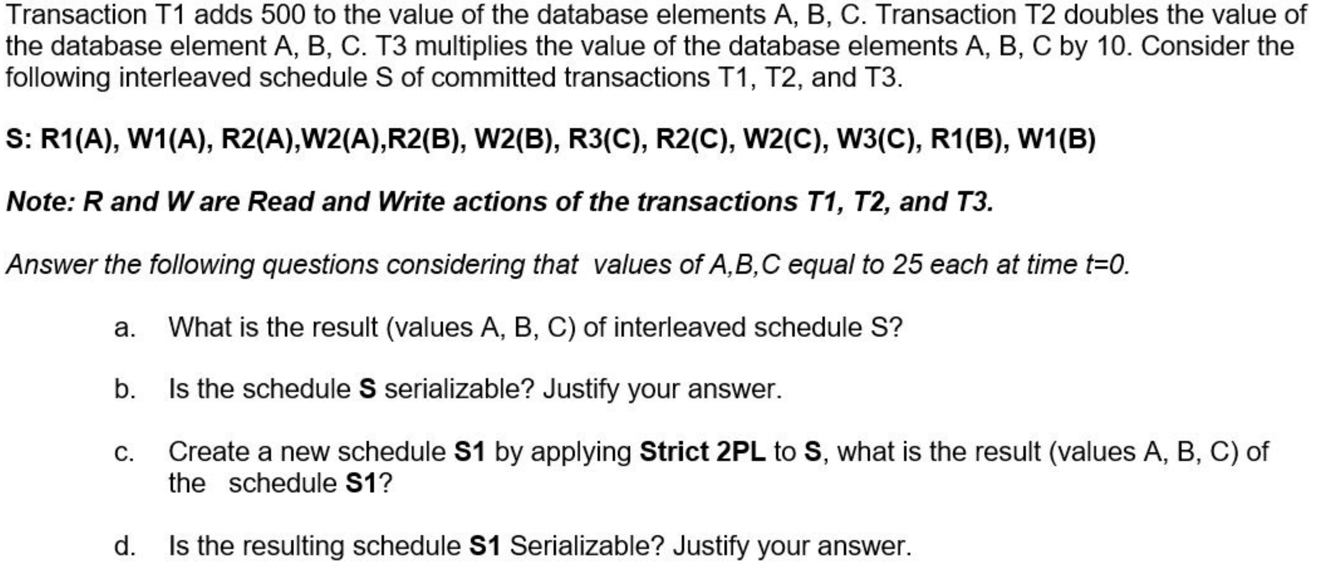
since the table has atomic attribute and defined PK, it is in 1NF

2NF :

Words : 270

Question 4:

Time: 23 Min 18 Sec **Marks: 5 / 5**



Transaction T1 adds 500 to the value of the database elements A, B, C. Transaction T2 doubles the value of the database element A, B, C. T3 multiplies the value of the database elements A, B, C by 10. Consider the following interleaved schedule S of committed transactions T1, T2, and T3.

S: R1(A), W1(A), R2(A),W2(A),R2(B), W2(B), R3(C), R2(C), W2(C), W3(C), R1(B), W1(B)

Note: R and W are Read and Write actions of the transactions T1, T2, and T3.

Answer the following questions considering that values of A,B,C equal to 25 each at time t=0.

a. What is the result (values A, B, C) of interleaved schedule S?

b. Is the schedule **S** serializable? Justify your answer.

c. Create a new schedule **S1** by applying **Strict 2PL** to **S**, what is the result (values A, B, C) of the schedule **S1**?

d. Is the resulting schedule **S1** Serializable? Justify your answer.

Response:

a. A = 1050, B = 550, C = 250

b. Making a precedence graph of the Transactions, there exists a loop between T1 and T2 due to the W1(A), W2(A), W2(B), W1(B), so no it is not conflict serializable.It does not have blind reads so it is not eligible to be not view serializable

c. In strict P2L, the transaction leaves the exclusive lock only after commits. Assuming that T2 has to wait for T1 to finish and meanwhile T3 can edit value of C, values of A = 1050, B = 1050 and C = 500

d. S1 : R1(A), **W1(A),R3(C),** W3(C), R1(B), **W1(B), R2(A)**, W2(A), R2(B), W2(B) R2(C), W2(C)

Yes s1 is serializable as the adjacent pairs (bold/italics) above can be swapped untill we get :

R1A, W1A, R1(B), W1B, R3C, W3C, R2A, W2A, R2B, W2B, R2(C), W2(C)

the order of transactions is T1 -> T3 -> T2

Words : 151