**Assignment 2**

**Question 1**

Given

,

derive :

1. (given)
2. (by F5(Projectivity) from 1)
3. (given)
4. (by F3(Transitivity) from 2 and 3)
5. (given)
6. (by F5(Projectivity) from 5)
7. (given)
8. (by F5(Projectivity) from 7)
9. (by F3(Transitivity) from 6 and 8)
10. (by F2((Augmentation) from 6 and 9); that is,
11. (by F3(Transitivity) from 10 and 4)

Therefore, .

Step 1:

Let

Step 2:

Try to remove A, H, C

Thus

Thus

Thus

Step 3:

Try to remove B, E, J

Thus cannot be removed

So is a candidate key for *R*

1. According to 1):

We can prove:

1. (given)
2. (by F5(Projectivity) from 1)
3. (given)
4. (by F3(Transitivity) from 2 and 3)
5. (given)
6. (by F5(Projectivity) from 5)
7. (by F2((Augmentation) from 6); that is,
8. (by F3 (Transitivity) from 7 and 4)

Thus:

The attributes of are transitively dependent on .

Therefore:

𝑅 with respect to *F* does not satisfies Third Normal Form (3NF)

1. Because:

(According to 1))

is a candidate key for *R* (According to 2)

Thus,

is partially dependent on

Therefore,

𝑅 with respect to *F* does not satisfies Second Normal Form (2NF)

1. All of attribute values are atomic

Therefore,

𝑅 with respect to *F* satisfies First Normal Form (1NF)

Hence, the highest normal form of 𝑅 with respect to 𝐹 is First Normal Form (1NF)

Step 1:

Step 2:

; thus is not inferred by.

Hence, cannot be replaced by .

; thus is inferred by.

Hence, can be replaced by .

; thus is not inferred by.

Hence, cannot be replaced by .

; thus is inferred by.

Hence, can be replaced by .

; thus is not inferred by.

Hence, cannot be replaced by .

; thus is inferred by.

Hence, can be replaced by .

; thus is not inferred by.

Hence, cannot be replaced by .

; thus is inferred by.

Hence, can be replaced by .

Step 3:

; thus is inferred by

That is, is redundant.

Thus, we can remove from to obtain .

; thus is not inferred by

That is, is not redundant.

Iteratively,

Thus,

Candidate key:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E | G | H | I | J |
|  | b | a | b | b | b | a | b | a | b |
|  | a | b | b | b | a | b | b | a | b |
|  | a | b | a | a | a | b | a | b | b |
|  | b | b | b | b | b | a | a | b | b |
|  | b | a | b | b | a | a | b | b | b |

According to the rule of testing for the lossless join property:

Base on key :

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E | G | H | I | J |
|  | b | a | b | b | b | a | b | a | b |
|  | a | b | b | b | a | b | b | a | b |
|  | a | b | a | a | a | b | a | b | b |
|  | b | b | b | b | b | a | a | b | b |
|  | a | a | a | a | a | a | a | a | a |

Therefore,

The decomposition of is lossless-join.

can infer

can infer

can infer

can infer

can infer

Therefore,

The decomposition of is dependency-preserving.

**Question 2**



According to the schedule graph, the graph is a cyclic.

Therefore, the transaction schedule is not conflict serializable.

T1 T2 T3 T4

|  |  |  |  |
| --- | --- | --- | --- |
| R(B) |  |  |  |
| W(B) |  |  |  |
| R(A) |  |  |  |
| W(A) |  |  |  |
|  | R(B) |  |  |
|  | W(B) |  |  |
|  |  | R(A) |  |
|  |  | W(A) |  |
|  |  |  | R(A) |
|  |  |  | W(A) |
|  |  |  | R(B) |
|  |  |  | W(B) |

T1 T2

|  |  |
| --- | --- |
| Write\_lock(B) |  |
| Write\_lock(A) |  |
| R(B) |  |
| R(A) |  |
| W(B) |  |
| Unlock(B) |  |
| W(A) | Write\_lock(B) |
| Unlock(A) | R(B) |
|  | W(B) |
|  | Unlock(B) |