**Part A-Normalization and functional dependencies**

Consider the following relation R(A, B, C, D, E) and functional dependencies F that holdover this relation.

F=D→C, A

B, A→C

B→A

C→A

E→A

a) Determine all candidate keys of R. {B,E,D}

b) compute the attribute cover of X={C, B} according to F.

{ C, A, B}

c) compute the canonical cover of F, Show each step of the generation according to the algorithm shown in class.

D→C,A ( A is extraneous) since C→A using decomposition and transitivity, so D→ C

B, A→C ( A is extraneous) since B→A using decomposition and pesutransitivity, so B→C

E→A

Fc

C→A

B→A

D→ C

B→C

E→A

d) in which normal form is relation R(recall that a relation can be in multiple normal forms).

* **2NF**
* **3NF**
* **BCNF**

D→C, A Violates 2NF so it is in 1NF

Use 3NF decomposition algorithm to decompose R into 3NF Ri

R1 {C,A}

R2 {B,A}

R3{D, C}

R4{B,C}

R5{E,A}

**Part B-Concurrency Control**

For each of the following schedules determine which properties this schedule has and provide explanation.

E.g., a schedule may be *recoverable* and *cascade-less* or *conflict or view serializable*. Consider the following notation for operations of transactions:

w1(A) Transaction 1 wrote item A

r1(A) transaction 1 read item A

c1 transaction 1 commits

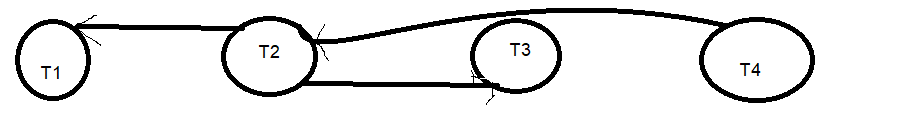
a1 transaction 1 aborts

S1=r3(B), w2(C), w4(A), w1(C), w2(A), c2, r3(A), c1, c3, c4

S2=r2(A), r1(B), w2(A), r2(B), r3(A), w1(B), c1, w3(A), c3, w2(B), c2

S3=r1(A), r1(B), r2(C), w2(C), w1(B), r2(A), w2(A), c1, w2(B), c2

S1 Conflict Serializable no cyclic in the order of precedence graph, therefore it is also view serializable. Not recoverable as the R3(A) commits before c4 and might read unrecoverable data. It is also not cascadeless



S2

|  |  |  |
| --- | --- | --- |
| T1 | T2 | T3 |
|  | R (A) |  |
| R(B) |  |  |
|  | W(A) |  |
|  | R(B) |  |
|  |  | R(A) |
| W(B) |  |  |
| C |  |  |
|  |  | W(A) |
|  |  | C3 |
|  | W(B) |  |
|  | C |  |

R1 (B) can’t be swapped to get T1 starts after T2 finishes due to the conflict of W1(B) and W2(B), also swaps can’t be done to have T2 starts after T1 finishes due the R2(B) and W1(B) conflict. So, it is not conflict serializable

|  |  |  |
| --- | --- | --- |
|  | A | B |
| First read | T2 | T1, T2 |
| Read after updates | T3 |  |
| Final write | T3 | T2 |

T1-.> T2 -> T3

|  |  |  |
| --- | --- | --- |
|  | A | B |
| First read | T2 | T1 |
| Read after updates | T3 | T2 |
| Final write | T3 | T2 |

Aren’t view equivalent not view serializable

C3 commit before C2 thus the r3(A) commits before the W2(A) is final hence not recoverable nor Cascaless

S3 swap can be done to get t1 finishes before T2 starts -> conflict seriaza