TIFFANY WONG
A20442087
CS425

HOMEWORK #4

PART A - NORMALIZATION & FUNCTIONAL DEPENDENCIES

Consider the following relation R(A, B, C, D, E) & functional dependencies F that hold over this relation.

F = { D → C, A — D determines C & A
B, A → C — B & A determines C
B → A — B determines A
C → A — C determines A
E → A } — E determines A

(a) Determine all candidate keys of R.

→ look for the attributes that cannot be determined because then they should be present in the candidate keys

BDE

→ see if other attributes are required to find the set of candidate keys
↳ find closure of BDE
↳ check attributes that can be determined from this set

$(BDE)^+ = BDECA$ → C & A can be determined by BDE

candidate keys of R = BDE

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

→ find all attributes that can be determined by X

$(CB)^+ = CBA$

attribute cover of X = CBA

(c) Compute the canonical cover of F. Show each step of the generation according to the algorithm in class.

$F_c = \{ D \to CA, BA \to C, B \to A, C \to A, E \to A \}$

$A^+ = A$
$B^+ = BAC$
$C^+ = CA$
$D^+ = DCA$
$E^+ = EA$

- 1st iteration:
  1) apply union rule to combine RHS

nothing to combine bc LHS are all unique

2) find extraneous attributes

→ A is extraneous in $D \to CA$
↳ check if the result of deleting A from $D \to CA$ is implied by other dependencies

Yes, $C \to A$ is already present.

→ set is now $F_c = \{D \to C, BA \to C, B \to A, C \to A, E \to A\}$

- 2nd iteration:
  1) apply union rule to combine RHS

nothing to combine bc LHS are all unique

2) find extraneous attributes

→ A is extraneous in $BA \to C$
↳ check if the result of deleting A from $BA \to C$ is implied by other dependencies

Yes because $C \to A$ exists, so $BA \to C = B \to C$.

→ set is now $F_c = \{D \to C, B \to C, B \to A, C \to A, E \to A\}$

- 3rd iteration
  1) apply union rule to combine RHS

→ combine $B \to C$ & $B \to A$ into $B \to CA$

→ set is now $F_c = \{D \to C, B \to CA, C \to A, E \to A\}$

2) find extraneous attributes

→ A is extraneous in $B \to CA$
↳ check if the result of deleting A from $B \to CA$ is implied by other dependencies

Yes because $C \to A$ exists & if $C \to A$, then $B \to C$ is enough

→ set is now $F_c = \{D \to C, B \to C, C \to A, E \to A\}$

The canonical cover is: $F_c = \{D \to C, B \to C, C \to A, E \to A\}$.

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

① determine candidate keys

→ From (a) = BDE

② divide all attributes into prime & non-prime

→ prime = attributes that are part of candidate key

prime = BDE

non-prime = AC

③ check 1NF, 2NF, 3NF, & BCNF

- 1NF - yes because the relation does not contain composite or multi-valued attributes
- 2NF - no because non-prime attributes depends on parts of the candidate key

relation R is in 1NF

Use 3NF decomposition algorithm to decompose R into 3NF R_i.

- 1st step : R(A, E)
- 2nd step : R(A, C)
- 3rd step : R(B, C)
- 4th step : R(B, D, E)
- 5th step : R(C, D)

PART B - CONCURRENCY CONTROL

PROPERTIES:
- view/conflict serializability
- recoverable/non-recoverable
- cascading rollbacks / cascadeless

] every cascadeless schedule is recoverable

★ if confict, then view; but view does not guarantee conflict!

① S1 = r3(B), w2(C), w4(A), w1(C), w2(A), c2, r3(A), c1, c3, c4

t_1	t_2	t_3	t_4
		r_B	
	w_C		
			w_A
w_C			
	w_A		
	commit		
		r_A	
commit			
		commit	
			commit

→ non-recoverable because t_3 reads A after t_2 has commited writing to A.

→ not cascadeless because t_2 reads A after t_2 writes to A & already committed. If there's an error in t_3, it won't be able to get the t_3 write to A info/data back

→ view serializable & conflict serializable

② S2 = r2(A), r1(B), w2(A), r2(B), r3(A), w1(B), c1, w3(A), c3, w2(B), c2

t_1	t_2	t_3
	r_A	
r_B		
	w_A	
	r_B	
		r_A
w_B		
commit		
		w_A
		commit
	w_B	
	commit	

→ non-recoverable

→ not cascadeless

→ not conflict serializability

→ not view serializability

③ S3 = r1(A) r1(B) r2(C) w2(C) w1(B) r2(A) w2(A) c1 w2(B) c2

t_1	t_2
r_A	
r_B	
	r_C
	w_C
w_B	
	r_A
	w_A

→ recoverable
→ cascadeless
→ view serializable
→ conflict serializable

Commit

w_B
Commit