**Part 3**

Question 1

(a)

Relation schema *R* has attributes ABCDEFGHIJK

For BCNF to hold, the LHS of a FD must be a super key.

S = { I → DGF, H → CEA, BI → J, B → H, CI → K }

|  |  |  |  |
| --- | --- | --- | --- |
| LHS of FD | Closure | Superkey? | Does this FD fulfill BCNF? |
| I | I+ = IDGF | × | × |
| H | H+ = HCEA | × | × |
| BI | BI+ = BIJHDGFCEAK = ABCDEFGHIJK | ✓ | ✓ |
| B | B+ = BHCEA | × | × |
| CI | CI+ = CIKDGF | × | × |

Out of the given FDs, only BI → J fulfills BCNF, all other FDs violate BCNF.

(b)

Decomposing *R* using FD I → DGF.

I+ = IDGF, so this yields two relations:   
R1 = DFGI with FDs SR1 = { I → DGF }  
R2 = R – I+ + I = ABCEHIJK   
 with FDs SR2 = { H → ACE, BI → J, B → H, CI → K}

Project the FDs onto R1 = DFGI

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| D | F | G | I | Closure | FDs |
| ✓ |  |  |  | D+ = D | Nothing |
|  | ✓ |  |  | F+ = F | Nothing |
|  |  | ✓ |  | G+ = G | Nothing |
|  |  |  | ✓ | I+ = IDGF | I → DGF; I is a superkey of R1 |
| Supersets of I | | | | Irrelevant | Can only generate weaker FDs |
| ✓ | ✓ |  |  | DF+ = DF | Nothing |
| ✓ |  | ✓ |  | DG+ = DG | Nothing |
|  | ✓ | ✓ |  | FG+ = FG | Nothing |
| ✓ | ✓ | ✓ |  | DFG+ = DFG | Nothing |

This relation satisfies BCNF

Project the FDs onto R2 = ABCEHIJK

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A | B | C | E | H | I | J | K | Closure | FDs |
| ✓ |  |  |  |  |  |  |  | A+ = A | Nothing |
|  | ✓ |  |  |  |  |  |  | B+ = BHCEA | B → ACEH violates BCNF, stop projection |

Decompose R2 further.

Decompose R2 using FD B → ACEH. B+ = BHCEA, so this yields two relations:

R3 = ABCEH with FDs SR3 = { H → ACE, B → H }

R4 = R2 – B+ + B = BIKJ with FDs SR4 = { BI → J }

Project the FDs onto R3 = ABCEH

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| A | B | C | E | H | Closure | FDs |
|  | ✓ |  |  |  | B+ = BHCEA | B → ACEH; B is a superkey of R3 |
|  |  |  |  | ✓ | H+ = HCEA | H → ACE violates BCNF, stop projection |

Decompose R3 using FD H → ACE. H+ = HCEA, so this yields two relations:

R5 = ACEH

R6 = R3 – H+ + H = BH

Project the FDs onto R5 = ACEH

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| A | C | E | H | Closure | FDs |
| ✓ |  |  |  | A+ = A | Nothing |
|  | ✓ |  |  | B+ = B | Nothing |
|  |  | ✓ |  | E+ = E | Nothing |
|  |  |  | ✓ | H+ = HCEA | H → ACE; H is a superkey of R5 |
| Supersets of H | | | | Irrelevant | Can only generate weaker FDs |
| ✓ | ✓ |  |  | AC+ = AC | Nothing |
| ✓ |  | ✓ |  | AE+ = AE | Nothing |
|  | ✓ | ✓ |  | CE+ = CE | Nothing |
| ✓ | ✓ | ✓ |  | ACE+ = ACE | Nothing |

This relation satisfies BCNF

Project the FDs onto R6 = BH

|  |  |  |  |
| --- | --- | --- | --- |
| B | H | Closure | FDs |
| ✓ |  | B+ = BHCEA | B → H; B is a superkey of R6 |
|  | ✓ | H+ = HCEA | Nothing |

This relation satisfies BCNF

Return to R4 = BIJK, project the FDs.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| B | I | J | K | Closure | FDs |
| ✓ |  |  |  | B+ = BHCEA | Nothing |
|  | ✓ |  |  | I+ = IDGF | Nothing |
|  |  | ✓ |  | J+ = J | Nothing |
|  |  |  | ✓ | K+ = K | Nothing |
| ✓ | ✓ |  |  | BI+ = BIHJCEADGFK | BI → JK; BI is a superkey of R4 |
| Supersets of H | | | | Irrelevant | Can only generate weaker FDs |
| ✓ |  |  | ✓ | BK+ = BKHCEA | Nothing |
| ✓ |  | ✓ |  | BJ+ = BJHCEA | Nothing |
|  | ✓ | ✓ |  | IJ+ = IJDGF | Nothing |
|  | ✓ |  | ✓ | IK+ = IKDGF | Nothing |
|  |  | ✓ | ✓ | JK+ = JK | Nothing |

This relation satisfies BCNF

The final decomposition is: (in alphabetical order)

1. R5 = ACEH with FD H → ACE
2. R6 = BH with FD B → H
3. R4 = BIJK with FD BI → JK
4. R1 = DFGI with FD I → DFG

Question 2:

(a)

Relation schema *P* has attributes ABCDEFGH

T = { ACDE → B, BF → AD, B → CF, CD → AF, ABF → CDH }

Simplify to singleton RHS to set S1

1. ACDE → B
2. BF → A
3. BF → D
4. B → C
5. B → F
6. CD → A
7. CD → F
8. ABF → C
9. ABF → D
10. ABF → H

S1 = { ACDE → B, BF → A, BF → D, B → C, B → F, CD → A, CD → F, ABF → C, ABF → D, ABF → H }

Looking for redundant FDs

|  |  |  |  |
| --- | --- | --- | --- |
| FD | Exclude from S1 when computing closure | Closure | Decision |
| 1 | 1 | No way to get B without this FD | Keep |
| 2 | 2 | BF+ = BFDCA… | Discard |
| 3 | 2, 3 | BF+ = BCF, can’t get D | Keep |
| 4 | 2, 4 | B+ = BFD, can’t get C | Keep |
| 5 | 2, 5 | B+ = BC, can’t get F | Keep |
| 6 | 2, 6 | CD+ = CDF, can’t get A | Keep |
| 7 | 2, 7 | CD+ = CDA, can’t get F | Keep |
| 8 | 2, 8 | ABF+ = ABFCDH | Discard |
| 9 | 2, 8, 9 | ABF+ = ABFCDH | Discard |
| 10 | 2, 8, 9, 10 | No way to get H without this FD | Keep |

Call the remaining FDs S2:

1. ACDE → B

3 BF → D

4 B → C

5 B → F

6 CD → A

7 CD → F

10 ABF → H

S2 = {ACDE → B, BF → D, B → C, B → F, CD → A, CD → F, ABF → H}

We will try reducing the LHS of any FDs with multiple attributes on LHS. We will close over the full set S2, including the FD being considered for simplification.

1. ACDE → B  
   A+ = A, can’t reduce LHS to A  
   C+ = C, can’t reduce LHS to C  
   D+ = D, can’t reduce LHS to D  
   E+ = E, can’t reduce LHS to E  
   AC+ = AC, can’t reduce LHS to AC  
   AD+ = AD, can’t reduce LHS to AD  
   AE+ = AE, can’t reduce LHS to AE  
   ACD+ = ACDF, can’t reduce LHS to ACD  
   ACE+ = ACE, can’t reduce LHS to ACD  
   CDE+ = CDEAFB, we can reduce the LHS to CDE

3 BF → D   
 B+ = BCFD, we can reduce LHS to B

6 CD → A  
 C+ = C, can’t reduce LHS to C  
 D+ = D, can’t reduce LHS to D  
 This FD remains unchanged.

7 CD → F  
 C+ = C, can’t reduce LHS to C  
 D+ = D, can’t reduce LHS to D  
 This FD remains unchanged.

10 ABF → H  
 A+ = A, can’t reduce LHS to A  
 B+ = BCFDAH, we can reduce LHS to B   
  
Call the FDs with reduced LHS S3:

1’ CDE → B  
3’ B → D  
4 B → C  
5 B → F  
6 CD → A  
7 CD → F  
10’ B → H

S3 = { CDE → B, B → D, B → C, B → F, CD → A, CD → F, B → H }

We now have new FDs (1’, 3’ and 10’) so we attempt to simplify further.

|  |  |  |  |
| --- | --- | --- | --- |
| FD | Exclude from S1 when computing closure | Closure | Decision |
| 1’ | 1’ | No way to get B without this FD | Keep |
| 3’ | 3’ | No way to get D without this FD | Keep |
| 4 | 4 | No way to get C without this FD | Keep |
| 5 | 5 | B+ = BDCAFH | Discard |
| 6 | 5, 6 | No way to get A without this FD | Keep |
| 7 | 5, 7 | No way to get F without this FD | Keep |
| 10’ | 5, 10’ | No way to get H without this FD | Keep |

Thus, we can now remove FD #5 from the set of relations.

This set, let’s call it S4, is minimal basis:

4 B → C  
3’ B → D  
10’ B → H  
6 CD → A  
7 CD → F  
1’ CDE → B

Impossible to simplify further.

Minimal basis for T, S4 = { B → C, B → D , B → H , CD → A, CD → F, CDE → B}

(b)

From S4,

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Appears on | | Conclusion |
| LHS | RHS |
| G |  |  | Must be in every key |
| E | ✓ |  | Must be in every key |
| F, H, A |  | ✓ | Are not in any key |
| B, C, D | ✓ | ✓ | We should check |

We will consider all combinations of attributes B, C, and D, and we must add is G and E to each combination, since they must be in every key.

* EGB+ = EGBCDHAF = ABCDEFGH, SO EGB = BEG is a key
* EGC+ = EGC, this is not a key
* EGD+ = EGD, this is not a key
* EGCD+ = EGCDAFBH = ABCDEFGH, SO EGCD = CDEG is a key

All other possible combinations have either EGB or EGCD and therefore must be non-minimal superkeys.

So, there are two keys for P: (rearranged alphabetically)

1. BEG
2. CDEG

(c)

We will merge the RHSs of S4, and call these FDs S5:

CDE → B  
B → CDH  
CD → AF

Resulting set of relations, with attributes:

R1(B,C,D,E), R2(B, C, D, H), R3(A, C, D, F)

Since none of the keys is a subset of R1, R2, or R3, we need to add one of them, say R4(B,E,G) because BEG is a key as per part B above.

The final set of relations (decomposition of relation P in 3NF) is:

R1(B, C, D, E),  
R2(B, C, D, H),   
R3(A, C, D, F),   
R4(B, E, G)

(d)

Each relation was formed from an FD in S5, so the LHS for each FD is a superkey for that relation.

However, we can look for other FDs because of whom BCNF is violated and then this schema is known to allow redundancy.

To check, we project the FDs from S5 onto these relations.

Projecting FDs onto R1 = ABCDE

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| B | C | D | E | Closure | FDs |
| ✓ |  |  |  | B+= BCDHAF  = ABCDFH | B → CDHAF; this is not a superkey of R1 because it is missing E |

Thus, because not all LHS of each FD form superkeys in their relations, redundancy based on functional dependency is present. Therefore BCNF is violated and thus this schema allows redundancy. This is expected as 3NF does not necessarily guarantee no redundancy.