**CSC343 Assignment 3 Part2**

**Fenglun Wu: 1002596684 and Qiangyu Zheng: 1002144128**

1.

(a)

|  |  |  |
| --- | --- | --- |
|  |  | LPR is not a superkey, so it violates BCNF. |
|  |  | LR is not a superkey, so it violates BCNF. |
|  |  | M is not a superkey, so it violates BCNF. |
|  |  | MR is not a superkey, so it violates BCNF. |

Therefore, LPR→Q, LR→ST, M→LO, MR→N. All FDs violate BCNF.

(b)

First, we decompose R using FD LPR→Q.

R1: LPRQST and R2: LMNOPR

Project all FDs on to R1.

Since no single attribute can indicate something other than itself in R1.

Therefore, no need to check a single attribute.

The first FD that violates this relation which includes two attributes is LR→ST

We abort the projection. We must decompose R1 further.

R3: LRST

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| L | R | S | T | closure | FDs |
|  |  |  |  | L+ = L | nothing |
|  |  |  |  | R+ = R | nothing |
|  |  |  |  | S+ = S | nothing |
|  |  |  |  | T+ = T | nothing |
|  |  |  |  | LR+ = LRST | LR is a superkey of R3 |

R3 satisfies BCNF.

R4: LPRQ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| L | P | R | Q | closure | FDs |
|  |  |  |  | L+ = L | nothing |
|  |  |  |  | P+ = P | nothing |
|  |  |  |  | R+ = R | nothing |
|  |  |  |  | Q+ = Q | nothing |
|  |  |  |  | LPR+ = LPRQ | LPR is a superkey of R4 |

R4 satisfies BCNF.

R2: LMNOPR

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| L | M | N | O | P | R | closure | FDs |
|  |  |  |  |  |  | L+ = L | nothing |
|  |  |  |  |  |  | M+ = MLO | Violates BCNF |

Decompose R2 using M→LO

R5: MLO

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| M | L | O | closure | FDs |
|  |  |  | M+ = MLO | M is a superkey in R5 |
|  |  |  | L+ = L | nothing |
|  |  |  | O+ = O | nothing |

R5 satisfies BCNF.

R6: MNPR

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| M | N | P | R | closure | FDs |
|  |  |  |  | M+ = M | nothing |
|  |  |  |  | N+ = N | nothing |
|  |  |  |  | P+ = P | nothing |
|  |  |  |  | R+ = R | nothing |
|  |  |  |  | MR+ = MRN | Violates BCNF |

Decompose R6 using MR→N

R7: MRN

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| M | R | N | closure | FDs |
|  |  |  | M+ = M | nothing |
|  |  |  | R+ = R | nothing |
|  |  |  | N+ = N | nothing |
|  |  |  | MR+ = MRN | MR is a superkey of R7. |

R7 satisfies BCNF.

R8: MPR with no FDs.

Final answer:

R3 = LRST, FDs: LR→ST

R4 = LPQR, FDs: LPR→Q

R5 = LMO, FDs: M→LO

R7 = MNR, FDs: MR→N

R8: MPR with no FDs.

2.

(a)

Step 1: Split the RHSs to get our initial set of FDs, S1:

(1) AB → C

(2) AB → D

(3) ACDE → B

(4) ACDE → F

(5) B → A

(6) B → C

(7) B → D

(8) CD → A

(9) CD → F

(10) CDE → F

(11) CDE → G

(12) EB → D

Step 2: For each FD, try to reduce the LHS:

1. B+ = ACD, so we can reduce the LHS of this FD, yielding the new FD: B → C, which is already existed, then remove
2. B+ = ACD, so we can reduce the LHS of this FD, yielding the new FD: B → D, which is already existed, then remove
3. CD+ = ACDF, CDE+ = ABCDEFG, so we can reduce the LHS of this FD, yielding the new FD: CDE → B
4. CD+ = ACDF, so we can reduce the LHS of this FD, yielding the new FD: CD → F
5. Only one attribute on the LHS, we cannot reduce the LHS
6. Only one attribute
7. Only one attribute
8. C+ = C, D+ = D, we cannot reduce the LHS
9. C+ = C, D+ = D, we cannot reduce the LHS.
10. CD+ = ACDF, so we can reduce the LHS of this FD, yielding the new FD: CD → F which is already existed, then remove
11. CD+ = ACDF, C+ = C, D+ = D, E+ = E, so we cannot reduce the LHS of this FD
12. E+ = E, B+ = ACD, so we can reduce the LHS of this FD, yielding the new FD: B → D, which is repeated, remove

Our new set of FDs, S2 is:

1. CDE → B
2. B → A
3. B → C
4. B → D
5. CD → A
6. CD → F
7. CDE → G

Step 3: Try to eliminate each FD

1. CDE+S2-(1) = ACDEFG. We need
2. B+S2-(2) = ABCDF. We can remove
3. B+S2-(3) = BD. We need
4. B+S2-(4) = BC. We need
5. CD+S2-(5) = ACD. We need
6. CD+S2-(6) = CDF. We need
7. CDE+S2-(7) = ABCDEF. We need

After combination, our minimal basis is:

{ CDE → BG, B → CD, CD → AF }

ABCDEFGH

(b)

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Appears on | | Conclusion |
| LHS | RHS |
| H | ­- | - | Must be in every key |
| E | √ | - | Must be in every key |
| AFG | - | √ | Is not in any key |
| BCD | √ | √ | Must check |

We only need to consider all combinations of B, C, D. For each, we must add in E, H, since they are in every key.

CDEH+ = ABCDEFGH

BEH+ = ABCDEFGH

All other possibilities include BEH and CDEH, so we’ve done.

Therefore, BEH is a key.

(c)

Apply 3NF algorithm:

our minimal basis is:

{ CDE → BG, B → CD, CD → AF }

𝑅1:(𝐵,𝐶,𝐷,𝐸,𝐺) with FD 𝐶𝐷𝐸→𝐵𝐺.

𝑅2:(𝐵,𝐶,𝐷) with FD 𝐵→𝐶𝐷.

𝑅3:(𝐴,𝐶,𝐷,𝐹) with FD 𝐶𝐷→𝐴𝐹.

Since the attributes BD occur within R1, we do not need to keep the relation R2.

There is no key in these relations, we need to add a relation contains a key.

R4: (B, E, H)

Then, the final set of relations is: R1(𝐵,𝐶,𝐷,𝐸,𝐺), R3(𝐴,𝐶,𝐷,𝐹) and R4 (B,E,H)

(d)

Our schema allows redundancy.

Because we can find a relation that violate BCNF:

B → CD will project onto the relation R1. And B+ = ABCDF, so B is not a superkey of this relation. Then our schema allows redundancy.