University of Toronto
CSC343, Fall 2023

Assignment 3: Part 2

Sophia Lee - 1007698608

Aryaman Arora - 1007858873

Question 1: BCNF

$$R = ABCDEFG$$
$$S = \{AC \to D, BG \to E, D \to CFG, DG \to B, G \to F\}$$

(a) We can determine which FDs violate BCNF using the Closure Test. Any FD whose LHS is not a superkey violates BCNF.

FD	Closure	Superkey?
$AC \to D$	$AC^+ = ACDFGBFE$	Yes
$BG \to E$	$BG^+ = BGEF$	No
$D \to CFG$	$D^+ = DCFGBE$	No
$DG \to B$	$DG^+ = DGBCFE$	No
$G \to F$	$G^+ = GF$	No

As shown in the Closure test, the following FDs violate BCNF: $BG \to E$, $D \to CFG$, $DG \to B$, $G \to F$

(b) We now apply BCNF Decomposition algorithm on R.
Loop through each FD that violates BCNF:

- $BG \to E$ violates BCNF because BG is not a superkey. Split R along BG and project S onto the new relations
- $R_1 = ABCDG, R_2 = BGEF$
- Project S onto R_1 using closure test:
  $AC^+ = ACD$, $BG^+ = BG$, $D^+ = DCGB$, $DG^+ = DGB$ (remove trivial FDs and redundancies)
  $S_1 = \{AC \to D, D \to CG, DG \to B\}$
- Project S onto R_2 using closure test:
  $BG^+ = BGEF$, $G^+ = GF$ (remove G→F redundancy)
  $S_2 = \{BG \to E, G \to F\}$

BCNF Decomposition on R_1

- $D \to CG$ violates BCNF because $D^+ = DCGB$ is not a superkey for R_1. Split R_1 along D and project S_1 onto the new relations
- $R_3 = AD, R_4 = BCDG$
- Project S_1 onto R_3 using closure test:
  $A^+ = A$, $D^+ = D$ (remove trivial FDs)
  $S_3 = \{\}$
- Project S_1 onto R_4 using closure test:
  $D^+ = DCGB$, $DG^+ = DGB$, (remove redundancy)
  $S_4 = \{D \to CG, DG \to B\}$
- Notice that R_3 and R_4 will not be decomposed any further. R_3 is a two attribute relation which trivially satisfies BCNF as discussed in class. R_4's two FDs both include attribute D, which is shown to by itself be a superkey, thus all R_4's FDs are superkeys and won't need further decomposition.

BCNF Decomposition on R_2

- $G \to F$ violates BCNF because $G^+ = GF$ is not a superkey for R_2. Split R_2 along G and project S_2 onto the new relations
- $R_5 = BEG, R_6 = GF$
- Project S_2 onto R_5 using closure test:
  $BG^+ = BGE$, so $S_5 = \{BG \to E\}$
- Project S_2 onto R_6 using closure test:
  $G^+ = GF$, so $S_6 = \{G \to F\}$
- Notice that R_5 and R_6 will not be decomposed any further. R_6 is a two attribute relation which trivially satisfies BCNF as discussed in class. R_5's only FD is a superkey.

Recursion has finished, BCNF Decomposition Algorithm has terminated. Final relations with FD sets:
$R_3 = AD, S_3 = \{\}$
$R_4 = BCDG, S_4 = \{D \to CG, DG \to B\}$
$R_5 = BEG, S_5 = \{BG \to E\}$
$R_6 = FG, S_6 = \{G \to F\}$

(c) The resulting decomposition does not preserve the original dependencies. For example, consider the original FD, $D \to CFG$ which is logically equivalent to these three FDs $D \to C, D \to F, D \to G$. Since D and F attributes are no longer ever in the same table, this functional dependency cannot be maintained. We are able to update the relations with values of D and F that do not follow this original dependency.

(d) I will now perform the chase test to prove that the decomposition is lossless.
First make an example join of tables R_3, R_4, R_5, R_6. Note the 1 values are all placeholders, assume they are all arbitrary and distinct for the chase test.

A	B	C	D	E	F	G
a	1	1	d	1	1	1
1	b	c	d	1	1	g
1	b	1	1	e	1	g
1	1	1	1	1	f	g

Now restrict the values based on original FDs to see if we can get an instance of abcdefg in the join. Otherwise, this means spurious tuples were created.

$D \to CFG, G \to F$

A	B	C	D	E	F	G
a	1	c	d	1	f	g
1	b	c	d	1	f	g
1	b	1	1	e	f	g
1	1	1	1	1	f	g

$DG \to B, BG \to E$

A	B	C	D	E	F	G
a	b	c	d	e	f	g
1	b	c	d	e	f	g
1	b	1	1	e	f	g
1	1	1	1	1	f	g

We have shown that the tuple abcdefg needed to exist given our FDs. This means that this decomposition passes the Chase test and is lossless! (Proves that if a tuple exists in the join, then it must exist in the original.)

Question 2: 3NF

$$A = LMNOPQRS$$

$$B = \{N \to M, NO \to LR, NQR \to MP, P \to R, Q \to NO\}$$

(a) Here are the steps to find a minimal basis:

1. Break down the RHS:
   $B_1 = \{N \to M, NO \to L, NO \to R, NQR \to M, NQR \to P, P \to R, Q \to N, Q \to O\}$
2. Reduce LHS (if possible) by calculating closures of multiattribute LHS:
   $N^+ = NM$, $O^+ = O$, $Q^+ = QNOLRM$, $R^+ = R$
   This means neither N nor O by themselves can close L nor R, so the joint LHS of NO stays. Q, however is sufficient to close much of the attributes, meaning most LHS with Q can be reduced.

   $$B_2 = \{N \to M, NO \to L, NO \to R, N \to M, Q \to M, Q \to P, P \to R, Q \to N, Q \to O\}$$

3. Remove unnecessary FDs (I will first remove exact repetition):

   $$B_2 = \{1 : N \to M, 2 : NO \to L, 3 : NO \to R, 4 : Q \to M, 5 : Q \to P, 6 : P \to R, 7 : Q \to N, 8 : Q \to O\}$$

FD	Exclude these from B_2 when computing closure	Closure	Decision
1	1	$N^+ = N$	keep
2	2	$NO^+ = NOMR$	keep
3	3	$NO^+ = NOML$	keep
4	4	$Q^+ = QNOPMLR$	discard
5	4, 5	$Q^+ = QMNOMRL$	keep
6	4, 6	$P^+ = P$	keep
7	4, 7	$Q^+ = QO$	keep
8	4, 8	$Q^+ = QNMRP$	keep

4. Final minimal basis combining FDs with same LHS:

   $$B_f = \{N \to M, NO \to LR, P \to R, Q \to NOP\}$$

(b) Find all keys:

1. Shortcut 1: Attribute on RHS and never on LHS is never part of a key: M, L, R are never part of any key.
2. Shortcut 2: S is never part of any FD, thus it is part of every key.
3. Compute single attribute closures (plus S)
   $NS^+ = MNS$, $OS^+ = OS$, $PS^+ = PRS$, $QS^+ = LMNOPQRS$
   We have found one key: QS. This means, no other key can contain both Q and S, otherwise it won't be minimal.
4. Notice that since S must be in every key, and since we can't have both Q and S in another key, Q cannot be in any other key. However, Q never appears on any RHS of the FD set, meaning we can't get to Q without Q. Thus, there are no more keys possible.

Final only key: QS

(c) Do 3NF Synthesis:

1. First construct relations that are the union of LHS and RHS for each FD:

$$R1(N, M), R2(N, O, L, R), R3(P, R), R4(Q, N, O, P)$$

2. Check if any of these relations contains a superkey. Since S is needed to be a minimal key, none of these relations are a superkey.
3. Since none of the relations contain a superkey, add a relation who's schema is a key to get the final set of relations:

$$R1(N, M), R2(N, O, L, R), R3(P, R), R4(Q, N, O, P), R5(Q, S)$$

(d) Looking at the set of minimal FDs, there is no strict subset of a relation that is part of an FD. This means that there is no FD that is not a superkey of a relation (none violate BCNF), and thus there is no redundancy. Each FD is a superkey in the relations 1-4 and each cannot be projected onto any other relation.