University of Toronto
CSC343, Fall 2020

Assignment 2:
Changyan Xu (1004802181) Jiaming Yang (1006458575)

Question 1

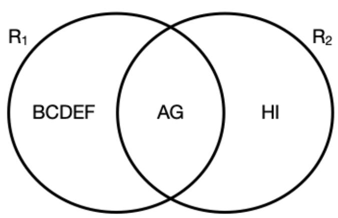
(a)

$AEG^+ = AEGHI$	Thus $AEG \to F$ violates BCNF	since AEG is not a superkey
$B^+ = BAD$	Thus $B \to AD$ violates BCNF	since B is not a superkey
$AG^+ = AGHI$	Thus $AG \to HI$ violates BCNF	since AG is not a superkey
$BG^+ = ABGDHI$	Thus $BG \to D$ violates BCNF	since BG is not a superkey

They all violates BCNF, because none of the closure of the LHS gives us ABCDEFGHI.

(b)

$R : ABCDEFGHI$
from #1(a), we have $AG \to HI$ that violates BCNF
Thus, we will break R by $AG \to HI$:



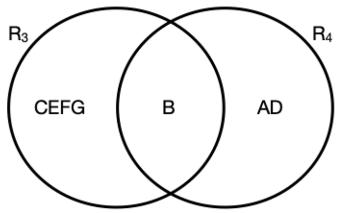
$R_2 : AGHI$
projection of the FDs on R_2.

A	G	H	I	closure	FD
✓	✓			$AG^+ = AGHI$ since none of the other possible combinations appear on the LHS of any FDs, they won't give us new FDs.	$AG \to HI$ (AG is a superkey) $\emptyset$

$R_1 : ABCDEFG$
projection of the FDs on R_1.

A	B	C	D	E	F	G	closure	FD
✓							$A^+ = A$	$\emptyset$
	✓						$B^+ = BAD$	$B \to AD$ (not a superkey)

Thus, we will break R_1 further.



$R_3 : BCEFG$
projection of the FDs on R_3.

B	C	E	F	G	closure	FD
✓					$B^+ = BAD$	$\emptyset$
✓				✓	$BG^+ = ABGDHI$	$\emptyset$ (since none of ADHI is in R_3)
					since none of the other possible combinations appear on the LHS of any FDs, they won't give us new FDs.	$\emptyset$

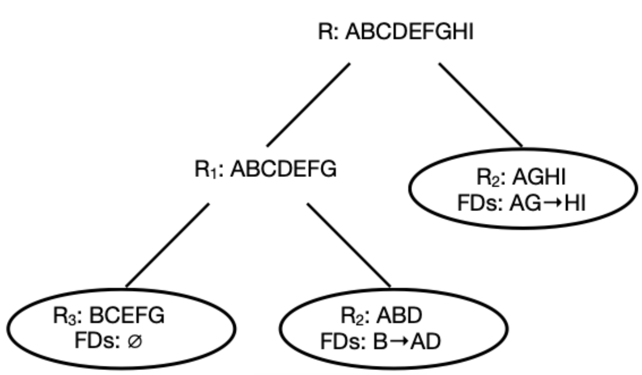
Thus, R_3 does not violate BCNF, because it has no FDs, which satisfies that the LHSs of all FDs are superkeys.

$R_4 : ABD$
projection of the FDs on R_4.

A	B	D	closure	FD
	✓		$B^+ = BAD$	$B \to AD$ (B is a superkey)
			since none of the other possible combinations appear on the LHS of any FDs, they won't give us new FDs.	$\emptyset$

Thus, R_4 does not violate BCNF, because it satisfies that the LHSs of all FDs are superkeys.

Thus, the final result looks like:



(c)

My schema doesn't preserve dependencies
we originally have 4 FDs:

$$AEG \to F, B \to AD, AG \to HI, BG \to D$$

And after the BCNF decomposition, we're left with $B \to AD, AG \to HI$
And from $B \to AD$ and $AG \to HI$

$AEG^+ = AEGHI \leftarrow$ (doesn't contain F)

$BG^+ = ADBG \leftarrow$ (can get $BG \to D$)

Thus we cannot derive $AEG \to F$ from $B \to AD$ and $AG \to HI$, and we lost this dependency

(d)

Relation R: ABCDEFGHI has FDs:

$$AEG \to F,\ B \to AD,\ AG \to HI,\ BG \to D$$

We decomposed R into relations BCEFG(R_3), ABD(R_4), AGHI(R_2)

Assume that R_2, R_3, R_4 have these tuples:

R_2:

A	G	H	I
a	g	h	i

R_3:

B	C	E	F	G
b	c	e	f	g

R_4:

A	B	D
a	b	d

R:

A	B	C	D	E	F	G	H	I
$\cancel{3}$a	b	c	$\cancel{3}$ d	e	f	g	$\cancel{3}$ h	$\cancel{3}$ i
a	b	3	d	3	3	3	3	3
a	3	3	3	3	3	g	h	i

Thus, after we do a natural join on R_2, R_3, R_4, we will have $abcdefghi$ in R.
Therefore, my schema is a lossless-join decomposition.

Question 2

2a)
Step1: split RHSs of the FDs

(a) LNOP→ M, (b) M → N, (c) M → Q, (d) NO →L, (e) NO →Q, (f) MNQ →L, (g) MNQ →O,
(h) LMQ →N, (i) LMQ→O, (j) LMQ →S

Step2: try to remove attributes (LHSs)

(a) NO^+ = NOLQ
Since no other combination of LNOP appears on the LHSs of any FDs, except for the whole LNOP, we know that we won't be able to get anything from them.
∴ keep this FD

(d)(e) N^+= N, O^+=O
∴ keep this FD

(f) M^+ = MNQ**L**OS
∴ simplify (f) to M → L

(g) M^+ = MNQL**O**S
∴ simplify (g) to M → O

(h) M^+ = M**N**QLOS
∴ simplify (h) to M → N

(i) M^+ = MNQL**O**S
∴ simplify (i) to M → O

(j) M^+ = MNQLO**S**
∴ simplify (j) to M → S

The rest of the FDs only have a single element on the LHS, thus keep them

FDs left: B2 = { (a) LNOP→ M, (b) M → N, ~~(c) M → Q~~, (d) NO →L, (e) NO →Q , ~~(f) M → L~~ , (g) M → O,
(h) M → S }

Step3: try to eliminate entire FDs

(a) $LNOP^+_{B2-(a)}$ = LNOPQ   no M, need this FD

(b) $M^+_{B2-(b)}$ = MOQLS   no N, need this FD

(c) $M^+_{B2-(c)}$ = MNLOS**Q**   ∴ we can remove this FD

(d) $NO^+_{B2-(c)-(d)}$ = NOQ   no L , need this FD

(e) $NO^+_{B2-(c)-(e)}$ = NOL   no Q , need this FD

(f) $M^+_{B2-(c)-(f)}$ = MNOS**L**Q   ∴ we can remove this FD

We don't need to consider (g), (h) because no other FDs (after we remove c & f) have O or S on their RHSs
Final answer: a minimal basis for B is B = { LNOP→ M, M → NOS, NO →LQ }

2b)

P, R are not on the RHSs of any FDs, thus they must appear in every key.
Q , S are on the RHS of some FD, but not on the LHS of any FDs, thus they can't be in any key

We are then left with L, M, N, O to analyze

$\begin{cases} PR^+ = PR \\ MPR^+ = LMNOPQRS \end{cases}$ $\Longrightarrow$ thus MPR is a key, since none of it's subset is a superkey

$\begin{cases} NPR^+ = NPR \\ OPR^+ = OPR \\ NOPR^+ = LMNOPQRS \end{cases}$ $\Longrightarrow$ thus NOPR is a key, since none of it's subset is a superkey

$LPR^+ = LPR$
$LNPR^+ = LNPR$
$LOPR^+ = LOPR$
Thus, all keys for A are MPR and NOPR

2c)
From the minimal basis form 2a), we get the relations after step 2 of the 3NF synthesis:
A1: LMNOP A2:MNOS A3: LNOQ

Since no relation is a superkey for A, we'll add a relation 'MPR' because MPR is a key we found in 2b)

Thus we decompose A into:
A1: LMNOP A2:MNOS A3: LNOQ A4: MPR

2d)
Yes, my schema allows redundancy

We have B = { LNOP→ M, M→ NQ, NO→ LQ, MNQ→ LO, LMQ→ NOS }

We project these FDs onto A1:

L M N O P	closures	FDs
√	$L^+ = L$	
√	$M^+ = MNQLOS$	M→ NOL no P, thus M is not a superkey

So A1 allows redundancy because
Consider the following example

A1

L	M	N	O	P
1	2	3	4	5
1	2	3	4	6

This is a valid instance of A1 according to the FDs. However, it contains redundancy. Because if we know M, we know what NOL are. So we don't need to include this information twice.