Nayeong Park (1000796739)

1. Suppose we have a relation on attributes A, B, C, D, E, and F, and these functional dependencies hold:
S = { B → DE, BF → C, CF → B, DF → AE }.
Write your closures in alphabetical order. For example, rather than BDFA, write ABDF.

(a) Compute B+.
B+ = BDE
(b) Compute CF +.
CF+ = ABCDEF
(c) Compute DF +.
DF+ = ADEF
(d) Compute BC+.
BC+ = BCDE
(e) Compute ABC+.
ABC+ = ABCDE

2. Again, suppose we have a relation on attributes A, B, C, D, E, and F, and these functional dependencies hold:
S = { B → DE, BF → C, CF → B, DF → AE }. Show your rough work.

(a) Does it follow from S that B → A?
No. B+ = BDE; B → A does not follow from S.
(b) Does it follow from S that CF → E?
Yes. CF+ = ABCDEF; CF → E follows from S.
(c) Does it follow from S that DF → B?
No. DF+ = ADEF; DF → B does not follow from S.
(d) Does it follow from S that BD → C?
No. BD+ = BDE; BD → C does not follow from S.
(e) Does it follow from S that BFC → A?
Yes. BFC+ = ABCDEF; BFC → A follows from S.

3. Consider relation R(A, B, C, D, E, F) with functional dependencies S.
S = {CD → A, B → EF, A → BC, F → D}

(a) Which functional dependencies indicate a violation of BCNF?
CD+ = ABCDEF
B+ = BDEF
A+ = ABCDEF
F+ = DF
∴ **B → EF, F → D violates BCNF**
(b) Create an instance of R that satisfies its FDs and has redundant data. Identify the redundancy.
Thought exercise: what does it have to do with the FDs?

A	B	C	D	E	F
1	2	3	4	5	6
2	5	1	4	3	6

Since some of the FD's violates BCNF, it allows the table to have duplicates which keeps redundant data. Having BCNF FD's will remove redundancy because the attributes are determined by key attributes (no duplicates allowed)

Nayeong Park (1000796739)

(c) Apply the first step of the BCNF decomposition algorithm and indicate what two new relations will replace R. Show your rough work.

1. **Choose any FD that violates BCNF: B → EF**
2. **Break into two relation where B → EF will satisfy BCNF condition. The other relation will include B and the rest attributes other than E and F**
   A. **R1 (B, E, F)**
   B. **R2 (B, A, C, D)**

(d) Project the FDs onto these two relations. You do not have to show your rough work for this part

For R1: { B → EF }

B	E	F	Closure	FD
X			B+ = BDEF	B → EF (superkey)
	X		E+ = E	N/A
		X	F+ = DF	N/A
	X	X	EF+ = DEF	N/A

For R2: { A → BCD }

A	B	C	D	Closure	FD
X				A+ = ABCDEF	A → BCD (superkey)
	X			B+ = BDEF	B → D, weaker than superkey
		X		C+ = C	N/A
			X	D+ = D	N/A
X	X			A is key, ignore all supersets of A	
X		X			
X			X		
	X	X		BC+ = ABCDEF	BC → AD, weaker than superkey
	X		X	BD+ = BDEF	N/A
		X	X	CD+ = ABCDEF	CD → AB, weaker than superkey
X	X	X		A is key, ignore all supersets of A	
X		X	X		
	X	X	X	BCD+ = ABCDEF	BCD → A, weaker than superkey
X	X	X	X	A is key, ignore all supersets of A	

(e) Is the new schema, with these two relations, in BCNF, or would we have to recurse and continue decomposing? Explain.

The new relations are in BCNF, as the FDs are superkey.