Jiana Javier 1000438298
Frank Lai 999088623

Part 2: Functional Dependencies, Decompositions, Normal Forms

1)

a) BH → AD
    BH+ = ABCDEFGH //does not violate; BH superkey
D → BH
    D+ = ABCDEFGH //does not violate; D superkey
BCE → F
    BCE+ = BCEF //violates
F → C
    F+ = CF //violates
A → GEF
    A+ = ACEGF //violates

Answer : **BCE → F, F → C, A → GEF violates BCNF**

b) **BCNF Decomposition**

BH+ = ABCDEFGH (does not violate)
D+ = ABCDEFGH (does not violate)

A+ =ACEFG (violates)
Decomposition:
R1 = ACEFG
R2 = ABDH

Project FDs onto R1(ACEFG)
~~R1(ACEFG)~~ :
A+ = ACEFG -- A → GEF, F → C; A superkey, does not violate
C+ = C --nothing
E+ = E --nothing
F+ = CF -- F → C violates, abort
Decomposition
R3 = FC
R4 = AEFG

Project FDs onto R3(FC)
F+ = FC -- F → C; F superkey, does not violate
C+ = C --nothing
FC+ = FC --nothing
R3 satisfies BCNF

Jiana Javier 1000438298
Frank Lai 999088623

Project FDs onto R4(AEFG)
A+ = AGEF, A → GEF; A superkey, does not violate
E+ = E --nothing
F+ = FC -- nothing
G+ = G --nothing
dont need to check supersets of A (weaker)
EF+ = EFC --nothing
FG+ = CFG --nothing
EG+ = EG --nothing
R4 satisfies BCNF

Return to R2. Project FDs onto R2(ABDH)
R2(ABDH)
A+ = AGEFC -- nothing
B+ = B --nothing
D+ = ABDH -- D → BH, BH → AD; D superkey, does not violate
H+ = H -- nothing
AB+ = ABCEFG -- nothing
dont need to check supersets of D (weaker)
AH+ = AGEFCH -- nothing
BH+ = BHAD, D → BH, BH → AD; BH superkey doesn't violate
R2 satisfies BCNF

Answer :
BCNF Decomposition Result:
R2(A,B,D,H), R3(C,F), R4(A,E,F,G)

2)

a) Answer:
D+ = ABCDEFG; D is a key for R and no superset of D can be a key.

The rest of the FDs except for EF → B contain D so I don't need to consider those.
And EF is not a key since EF+ = EFB.

b) S1

1 DBE → F	DBE+=ABCDEFG, discard
2 DBE → C	DBE+=ABCDEFG, discard
3 CD → A	CD+=CDFABGE, discard
4 CD → F	CD+=ABCDEFG, discard
5 D → A	no other way to get A, keep
6 D → B	no other way to get B, keep
7 D → G	no other way to get G, keep
8 BADE → C	no other way to get C, keep
9 ABD → E	no other way to get E, keep

10 D → F no other way to get F, keep
11 EF → B no other way to get B, keep

S2
5 D → A no other way to get A, keep
6 D → B no other way to get B, keep
7 D → G no other way to get G, keep
8 ~~BADE~~ → C can be simplified to D → C
9 ~~AB~~D → E can be simplified to D → E
10 D → F no other way to get F, keep
11 EF → B no other way to get B, keep

Answer:
The minimal basis for S is:
D → A
D → C
D → E
D → F
D → G
EF → B

c) Merge the right hand sides.

D → ACEFG
EF → B

The set of relations that would result would have these attributes:
R1(A,C,D,E,F,G) R2(B,E,F)

None of the attributes completely overlap. So we can't eliminate any relations.
D is a key for R so there is no need to add another relation that includes a key.

So the final set of relations is:
R1(A,C,D,E,F,G), R2(B,E,F)

d) Because we formed each relation from an FD, the LHS of those FDs are indeed superkeys for their relations. However there may be other FDs that violate BCNF and therefore allow redundancy. Find out by projecting FDs onto each relation.

EF → B projects onto R2. EF+ = EFB (does not violate), E+=E (does not violate), B+= B (does not violate), BE+ = BE (does not violate), BF+ = BF (does not violate).

D → ACEFG projects onto R1. D+ = ABCDEFG (does not violate) but the subset EF+ = EFB which is not a superkey so it violates BCNF.

So yes, this schema allows redundancy.