1. Consider a relation schema R with attributes *ABCDEFGHIJK* with functional dependencies S.

$$S = \{ I \rightarrow DGF, H \rightarrow CEA, BI \rightarrow J, B \rightarrow H, CI \rightarrow K \}$$

(a) State which of the given FDs violate BCNF.

I+ = DFGI (I not superkey)

H+ = ACEH (H not superkey)

BI+ = ACBDEFGHIJK (BI superkey)

B+ = ABCEH (B not superkey)

CI+ = CDFGIK (CI not superkey)

∴ { I → DGF, H → CEA, B → H, CI → K } violates BCNF

(b) Employ the BCNF decomposition algorithm to obtain a lossless and redundancy-preventing decomposition of relation R into a collection of relations that are in BCNF. Make sure it is clear which relations are in the final decomposition, and don't forget to project the dependencies onto each relation in that final decomposition. Because there are choice points in the algorithm, there may be more than one correct answer. List the final relations in alphabetical order (order the attributes alphabetically within a relation, and order the relations alphabetically).

R = { R1 = DFG**I** FD: {I → DGF} **I** is now a superkey
R2 = **I**ABCEHJK FD: {H → CEA, BI → J, B → H, CI → K}

R2 = { R3 = ACE**H** FD: {H → CEA} H is now a superkey
R4 = **HI**BJK FD: {BI → J, B → H, CI → K}

R4 = { R5 = **BI**J FD: {BI → J} BI is now a superkey
R6 = **HI**BK FD: { B → H, CI → K}

R6 = { R7 = **B**H FD: {B → H} B is now a superkey
R8 = **I**BK FD: { CI → K}

R8 = **CI**K FD: { CI → K } CI is now a superkey

Projection of FD's

For R1: { I → DGF }

D	F	G	I	Closure	FD
X				D+ = D	ϕ
	X			F+ = F	ϕ
		X		G+ = G	ϕ
			X	I+ = DFG	I → DFG (superkey)

For R3: { H → CEA }

A	C	E	H	Closure	FD
X				A+ = A	ϕ
	X			C+ = C	ϕ
		X		E+ = E	ϕ
			X	H+ = ACE	H → ACE (superkey)
X	X			AC+ = AC	ϕ
X		X		AE+ = AE	ϕ
X			X	H is key, ignore all subsets of H	
	X	X		CE+ = CE	ϕ
	X		X	H is key, ignore all subsets of H	
		X	X		
X	X	X		ACE+ = ACE	ϕ
X		X	X	H is key, ignore all subsets of H	
X	X		X		
	X	X	X		
X	X	X	X		

For R5: { BI → J }

B	I	J	Closure	FD
X			B+ = ABCEH	N/A
	X		I+ = DFG	N/A
		X	J+ = J	ϕ
X	X		BI+ = ABCDEFGHIJK	BI → J (superkey)
X		X	BJ+ = ABCEHJ	N/A
	X	X	IJ+ = DFGIJ	N/A
X	X	X	BI is key, ignore all subsets of BI	

For R8: { CI → K }

C	I	K	Closure	FD
X			C+ = C	ϕ
	X		I+ = DFG	N/A
		X	K+ = K	ϕ
X	X		CI+ = CDFGIK	CI → K (superkey)
X		X	CK+ = CK	ϕ
	X	X	IK+ = DFGIK	N/A
X	X	X	CI is key, ignore all subsets of CI	

For R7: { B → H }

B	H	Closure	FD
X		B+ = BH	B → H (superkey)
	X	H+ = ACEH	N/A
X	X	B is key, ignore all subsets of B	

Relations	FD
R = ACEH	H → CEA
R = BH	B → H
R = BIJ	BI → J
R = CIK	CI → K
R = DFGI	I → DGF

2. Consider a relation P with attributes *ABCDEFGH* and functional dependencies T.

T = { ACDE → B, BF → AD, B → CF, CD → AF, ABF → CDH }

Show all of your steps so that we can give part marks where appropriate. There are no marks for simply a correct answer. If you take any shortcuts, you must explain why they are justified.

(a) Compute a minimal basis for T. In your final answer, put the FDs into alphabetical order. Within a single FD, this means stating an FD as XY → A, not as Y X → A. Also, list the FDs in alphabetical order ascending according to the left-hand side, then by the right-hand side. This means, WX → A comes before WXZ → A which comes before WXZ → B.

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

- ACDE → B
- BF → A
- BF → D
- B → C
- B → F
- CD → A
- CD → F
- ABF → C
- ABF → D
- ABF → H

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

- ACDE → B

A	C	D	E	Closure	Reduction
X				A+ = A	*N/A*
	X			C+ = C	*N/A*
		X		D+ = D	*N/A*
			X	E+ = E	*N/A*

A	C	D	E	Closure	Reduction
X	X			AC+ = AC	*N/A*
X		X		AD+ = AD	*N/A*
X			X	AE+ = AE	*N/A*
	X	X		CD+ = ACDF	*N/A*
	X		X	CE+ = CE	*N/A*
		X	X	DE+ = DE	*N/A*
X	X	X		ACD+ = ACDF	*N/A*
X		X	X	ADE+ = ADE	*N/A*
X	X		X	ACE+ = ACE	*N/A*
	X	X	X	CDE+ = ABCDEFH	CDE → B

- BF → A

A	B	F	Closure	Reduction
X			A+ = A	*N/A*
	X		B+ = ABCDFH	B → A

- BF → D

B	D	F	Closure	Reduction
X			B+ = ABCDFH	B → D

- B → C

  Singleton, cannot reduce

- B → F

  Singleton, cannot reduce

- CD → A

A	C	D	Closure	Reduction
X			A+ = A	*N/A*
	X		C+ = C	*N/A*
		X	D+ = D	*N/A*
	X	X	CD+ = ACDF	CD → A, cannot reduce

- CD → F

C	D	F	Closure	Reduction
X			A+ = A	*N/A*
	X		C+ = C	*N/A*
		X	D+ = D	*N/A*
X	X		CD+ = ACDF	CD → F, cannot reduce

- ABF → C

A	B	C	F	Closure	Reduction
X				A+ = A	*N/A*
	X			B+ = ABCDFH	B → C

- ABF → D

A	B	C	F	Closure	Reduction
X				A+ = A	*N/A*
	X			B+ = ABCDFH	B → D

- ABF → H

A	B	C	F	Closure	Reduction
X				A+ = A	*N/A*
	X			B+ = ABCDFH	B → H

∴ S2 = { CDE → B, B → A, B → D, CD → A, CD → F, B → C, B → H }

Step 3: Try to eliminate each FD.

- CDE $+_{S2-(\mathbf{CDE \rightarrow B})}$ = ACDEF → missing B, cannot eliminate
- B $+_{S2-(\mathbf{B \rightarrow A})}$ = ABCDFH → contains A, can be eliminated
- B $+_{S2-(\mathbf{B \rightarrow D})}$ = ABCH → missing D, cannot eliminate
- CD $+_{S2-(\mathbf{CD \rightarrow A})}$ = CDF → missing A, cannot eliminate
- CD $+_{S2-(\mathbf{CD \rightarrow F})}$ = ACD → missing F, cannot eliminate
- B $+_{S2-(\mathbf{B \rightarrow C})}$ = ABDH→ missing C, cannot eliminate
- B $+_{S2-(\mathbf{B \rightarrow H})}$ = ABCDF→ missing H, cannot eliminate

∴ S = { B → C, B → D, B → H, CD → A, CD → F, CDE → B }

∴ S = { B → CDH, CD → AF, CDE → B }

(b) Using your minimal basis from the last subquestion, compute all keys for P.

Attributes	Appearance		Key
	LHS	RHS	
A	-	X	NO
B	X	X	Need to check
C	X	X	Need to check
D	X	X	Need to check
E	X	-	YES
F	-	X	NO
G	-	-	YES
H	-	X	NO

Check B,C,D:

BEG+ = ABCDEFGH (B is a key)

CEG+ = CEG (C is not a key)

DEG+ = DEG (D is not a key)

∴ **BEG is key**

(c) Employ the 3NF synthesis algorithm to obtain a lossless and dependency-preserving decomposition of relation P into a collection of relations that are in 3NF. Do not "over normalize", this means that you should combine all FDs with the same left-hand side into a single relation. If your schema includes one relation that is a subset of another, remove the smaller one.

S = { B → CDH, CD → AF, CDE → B }

R1(B,C,D,H) R2(A,C,D,F) R3(B,C,D,E) + R4(B,E,G)

(d) Does your schema allow redundancy? Explain how you know that it does or does not.

It allows redundancy; FD's from original set (before reducing to minimal basis) will project onto new 3NF relations but may violate BCNF condition if the FDs are not superkeys