PART II

1. a) compute closures to determine if any FD violate BCNF:

(LPR)+ = LPQRST violates BCNF
(LR)+ = LRST violates BCNF
M+ = LMO violates BCNF
(MR)+ = LMNORST violates BCNF
None of the FDs satisfy BCNF constraint.

b) BCNF decomposition:

First, we pick a violating constraint.
LPR->Q is a one of the violation of BCNF, and (LPR)+ = LPQRST
The original relation is split into two relations
R1(L, P, Q, R, S, T) and R2(L, M, N, O, P, R)

We first project the original FDs onto R1

Attributes	Closure	FD
L	L+ = L	None
P	P+ = P	None
Q	Q+ = Q	None
R	R+ = R	None
S	S+ = S	None
T	T+ = T	None
LP	(LP)+ = LP	None
LQ	(LQ)+ = LQ	None
LR	(LR)+ = LRST	LR->ST Violate BCNF

So R1 can be furthermore split into two relations using the violating FD.
R3(L, R, S, T) and R4(L, P, Q, R)
We do the same projection on R3 and R4

Work continues on the next page...

Projection on R3

Attributes	Closure	FD
L	L+ = L	None
R	R+ = R	None
S	S+ = S	None
T	T+ = T	None
LR	(LR)+ = LRST	LR->ST key
LS	(LS)+ = LS	None
LT	(LT)+ = LT	None
RS	(RS)+ = RS	None
RT	(RT)+ = RT	None
ST	(ST)+ = ST	None
LRS	Superkey of LR	
LRT	Superkey of LR	
LST	(LST)+ = LST	None
RST	(RST)+ = RST	None

Projection on R4:

Attribute	Closure	FD
L	L+ = L	None
P	P+ = P	None
Q	Q+ = Q	None
R	R+ = R	None
LP	(LP)+ = LP	None
LQ	(LQ)+ = LQ	None
LR	(LR)+ = LRST	None

PQ	(PQ)+ = PQ	None
PR	(PR)+ = PR	None
QR	(QR)+ = QR	None
LPQ	(LPQ)+ = LPQ	None
LPR	(LPR)+ = LPQR	LPR->Q key
LQR	(LQR)+ = LQRST	None
PQR	(PQR)+ = PQR	None

R3 and R4 do not have any more BCNF violation, we leave it as it is.

Now, we come back to look at R2
L, M, N, O, P, R
Projection on R2

Attribute	Closure	FD
L	L+ = L	None
M	M+ = LMO	M->LO violates BCNF

We can split R2 into two relations using the violating FD.
R5(L, M, O) and R6(M, N, P, R)
We can now project R5 and R6 to see any other BCNF violation

Projection on R5

Attribute	Closure	FD
L	L+ = L	None
M	M+ = LMO	M->LO key
O	O+ = O	None
LM	Superkey of M	None
LO	(LO)+ = LO	None
MO	Superkey of M	None

Projection on R6:

Attribute	Closure	FD
M	M+ = MLO	None
N	N+ = N	None
P	P+ = P	None
R	R+ = R	None
MN	(MN)+ = LMNO	None
MP	(MP)+ = LMPO	None
MR	(MR)+ = LMRN...	MR->N violates BCNF

As a result, R5 satisfies BCNF, but R6 does not.
R6 can be split into two relations using violating FD.
R7(M, N, R) and R8(M, P, R)

Projection on R7:

Attribute	Closure	FD
M	M+ = MLO	None
N	N+ = N	None
R	R+ = R	None
MN	MN+ = LMNO	None
MR	MR+ = MNR	MR->N key
NR	NR+ = NR	None

Projection on R8:

Attribute	Closure	FD
M	M+ = MLO	None
R	R+ = R	None
P	P+ = P	None

MR	(MR)+ = MNR	None
MP	(MP)+ = LMOP	None
RP	(RP)+ = RP	None

There is no violation in both R7 and R8

For the final result, we will keep R3, R4, R5, R7, R8
R3(L, R, S, T) with FD LR->ST
R4(L, P, Q, R) with FD LPR-> Q
R5(L, M, O) with FD M-> LO
R7(M, N, R) with FD MR-> N
R8(M, P, R) with no FD

2. a) Simplify FDs to singleton RHS and to reduce redundancy

FD #	FD	Exclude from original singleton FDs to compute closure	closure	decision
1	AB->C	1	(AB)+ = ABDC...	discard
2	AB->D	1,2	(AB)+ = ABCD...	discard
3	ACDE->B	1,2,3	No way to get B without this FD	keep
4	ACDE->F	1,2,4	(ACDE)+ = BACDEF...	discard
5	B->A	1,2,4,5	B+ = BCDA...	discard
6	B->C	1,2,4,5,6	No way to get C without this FD	keep
7	B->D	1,2,4,5,7	B+ = BC	keep
8	CD->A	1,2,4,5,8	(CD)+ = CDF	keep
9	CD->F	1,2,4,5,9	(CD)+ = CDA	keep
10	CDE->F	1,2,4,5,10	(CDE)+ = CDAF...	discard
11	CDE->G	1,2,4,5,10,11	No way to get G without this FD	keep
12	EB->D	1,2,4,5,10,12	(EB)+ -> EBACD...	discard

After, we will get the new set of simplified FDs as below, as S1
3 ACDE->B
6 B->C
7 B->D
8 CD->A
9 CD->F
11 CDE->G

Now we compute non-singleton LHS for any further redundancy
3 ACDE->B

A+ = A   C+ = C   D+ = D   E+ = E
(AC)+ = AC   (AD)+ = AD   (AE)+ = AE
(CD)+ = CDAF   (CE)+ = CE   (DE)+ = DE
(ACD)+ = ACDF   (ACE)+ = ACE   (ADE)+ = ADE
(CDE)+ = CDEAFB so we can reduce LHS to CDE

8 CD->A

C+ = C and D+ = D, so the FD remains as it is.

9 CD->F

C+ = C and D+ = D, so the FD remains as it is.

11 CDE->G

C+ = C   D+= D   E+ = E
(CD)+ = CDAF   (CE)+ = CE   (DE)+ = DE
So we cannot reduce LHS, the FD remains as it is.

Let S3 be the set of FDs

FD #	FD	Exclude from original singleton FDs to compute closure	closure	decision
3’	CDE->B	3’	No way to get B without this FD	keep
6	B->C	6	No way to get C without this FD	keep
7	B->D	7	B+ = BC	keep
8	CD->A	8	(CD)+ = CDF	keep

9	CD->F	9	(CD)+ = CDA	keep
11	CDE->G	11	No way to get G without this FD	keep

No further simplifications are possible.
As result, we will have the minimal basis of {CDE->B, B->C, B->D, CD->A, CD->F, CDE->G}

b) Our minimum basis from a) is {CDE->B, B->C, B->D, CD->A, CD->F, CDE->G}

Attribute	Appear on LHS	Appear on RHS	Conclusion
A,G,F	No	Yes	Not in any key
E	Yes	No	Must be in every key
H	No	No	Must be in every key
B,C,D	Yes	Yes	Must check

So we now consider all combinations of B,C,D, and for each we have to add attribute E and H.

(BEH)+ = BEHCDAFG   *Note that BEH will be a superkey, so we have eliminated a lot of cases
(CEH)+ = CEH
(DEH)+ = DEH
(CDEH)+ = CDEHBAFG

Therefore, the set of key will be BEH and CDEH.

c) From part a), we will have a revised set of FDs

Let our final revised FDs be {CDE->BG, B->CD, CD->AF}
For each of the FD
R1(C, D, E, B, G) with FD CDE->BG
R2(B, C, D) with FD B->CD
R3(C, D, A, F) with FD CD->AF

We do not necessarily need R2 since R2 happened in R1.
However, there is no key constraint in P, so we would add another relation R4
R4(B, E, H)

The final relations will be R1, R2, and R4.

d) Yes, this design allows redundancy. To show that, if we compute (CDE)+ (CDE)+ = CDEBGAF where attribute H is missing. CDE is not a superkey. This violate BCNF and therefore creates a redundancy.