1. Relation R has attributes $IJKLMNOP$ and functional dependencies:

$$S_P = \{M \to IJL, J \to LI, JN \to KM, M \to J, KLN \to M, K \to IJL, IJ \to K\}$$

(a) Find a minimal basis for R. List the attributes in each LHS in alphabetical order, each RHS in alphabetical order, and your entire set FDs in alphabetical order.

Step 1:
Rewrite the FD's so there is only one attribute on the RHS without any trivial FD's
a) M→I
b) M→J
c) M→L

d) J→L
e) J→I

f) JN→K
g) JN→M

h) M→J

i) KLN→M

j) K→I
k) K→J
l) K→L

m) IJ→K

Step 2:
Reduce the LHS of each FD
The LHS of the set of FD's, {a-e, h, j-l}, can not be reduced as they are singletons

f and g) The closure of J^+=IJL and the closure of N^+=N, thus using these closures will not help to reduce f and g. We are given IJ→K and JN→K.Therefore, f can be reduced as N and I do not matter when determining K.
∴ JN→K ⇒ J→K

i) The closure of K+=IJKL and therefore this FD's LHS can be reduced
∴ KLN→M ⇒ KN→M

m) The closure of J+=IJL so this FD's LHS can be reduced
∴ IJ→K ⇒ J→K

By eliminating repeats and using the reduced FD's, we now have

a) M→I
b) M→J
c) M→L
d) J→L
e) J→I
f) J→K
g) JN→M
h) KN→M
i) K→I
j) K→J
k) K→L

Step 3:
Try to eliminate each FD

a) M^+= JLIMK, I is in the closure. Thus we are able to remove this FD
b) M^+= ILM unable to obtain J, thus this FD is required
c) M^+= IJKLM, L is in the closure. Thus we are able to remove this FD
d) J^+=IJKL, L is in the closure. Thus we are able to remove this FD
e) J^+=JLKI, I is in the closure. Thus we are able to remove this FD
f) J^+=ILJ unable to obtain K, thus this FD is required
g) JN^+=JNKILM, M is in the closure. Thus we are able to remove this FD
h) KN^+=KNIJL, unable to obtain M, thus this FD is required
i) K^+=KJL, unable to obtain I, thus this FD is required
j) K^+=IL unable to obtain J, thus this FD is required
k) K^+=KIJ, unable to obtain L, thus this FD is required

The minimal basis for R is:
M→J
J→K
KN→M
K→IJL

(b) Find all keys for R.

Step 1:
Determine which attributes must belong in a key and will never belong in a key

O, P must always be in a key as they do not exist on the LHS or RHS of any FD
N must always be in a key as it only appears on the LHS of FD's that include N
I, L will never be part of a key as it only appears on the RHS of FD's that include I and L
J, K, M has to be checked

Step 2:
Check the attributes that have to be checked with the attributes that must be in a key
a) $JNOP^+$=JNOP

Using these FD's:
JN→KM
$JNOP^+$=JKMNOP

J→LI
$JNOP^+$=IJKLMNOP

∴ JNOP is a key

b) $KNOP^+$= KNOP

Using these FD's:
K→IJL
$KNOP^+$= IJKLNOP

JN→KM
$KNOP^+$= IJKLMNOP

∴ KNOP is a key

c) $MNOP^+$=MNOP

Using these FD's:
M→IJL
$MNOP^+$=IJLMNOP

JN→K
$MNOP^+$=IJKLMNOP

∴ MNOP is a key

All keys for R are:
JNOP
KNOP
MNOP

(c) Uses 3NF to find a lossless, dependency-preserving decomposition of R. Be sure to combine FDs with the same LHS into the same into a single relation, and omit relations that are subsets of other relations.

Recall the minimal basis from part a:

a) M→J

b) J→K

c) KN→M

d) K→IJL

Step 1:

Create a set of relations that satisfy the minimal basis above

a) R1(M, J)

b) R2(J, K)

c) R3 (K, M, N)

d) R4 (I, J, K, L)

Step 2:

Remove all redundant relations

a) Attributes M and J do not show up in any other relation, thus we are able to keep this relation

b) Attributes in R2, J and K, show up in R4, thus we are able to remove this relation

c) Attributes in R3 do not show up in any other relation, thus we are able to keep this relation

d) Attributes in R4 do not show up in any other relation, thus we are able to keep this relation

The resulting left relations are:

a) R1(M, J)

b) R3 (K, M, N)

c) R4 (I, J, K, L)

Step 3:

Check if any relation is a key. If there are no keys, add a relation that is a key

None of the relations, a-c, are keys as they are not included in part (b) of this question. Therefore we must add a key from part (b)

Add R5 (J, N, O, P)

The final set of relations are:

R1(M, J)

R2 (K, M, N)

R3 (I, J, K, L)

R4 (J, N, O, P)

(d) Does your schema allow redundancy? Explain why, or why not.

Step 1:
We must determine which FD's violate BCNF in the set of relations above in part c with the minimal basis we found in part a (projection onto the relations)

Recall the minimal basis from part a:

a) M→J
b) J→K
c) KN→M
d) K→IJL

Recall the set of relations that satisfy 3NF:

R1(M, J)
R2 (K, M, N)
R3 (I, J, K, L)
R4 (J, N, O, P)

R1: Using minimal FD, M→J, M is a superkey for R1. Therefore satisfying BCNF for R1

R2: Using minimal FD, KN→M, KN is a superkey for R2. Therefore satisfying BCNF for R2

R3: Using the minimal FD, K→IJL, K is a superkey for R3. Therefore satisfying BCNF for R3

R4: There is no FD in the result of a projection from the minimal FD's. Therefore satisfying BCNF for R4

When a relation does not satisfy BCNF, the relation allows redundancies as the FD is not a superkey. All the relations above satisfy BCNF, thus the schema does not allow redundancies.

2. Relation T contains attributes $CDEFGHIJ$ and functional dependencies:

$$S_T = \{C \to EH, DEI \to F, F \to D, EH \to CJ, J \to FGI\}$$

(a) Which of the FDs in S_T violate BCNF?

Find all closures of the LHS of the FD's and see if they are superkeys of relation T

<u>Closure of C:</u>
$C^+ = C$

$C \to EH$
$C^+ = CEH$

$EH \to CJ$
$C^+ = CEHJ$

$J \to FGI$
$C^+ = CEFGHIJ$

$F \to D$
$C^+ = CDEFGHIJ$

This FD satisfies BCNF as C is a superkey for relation T

<u>Closure of DEI:</u>
$DEI^+ = DEI$

$DEI \to F$
$DEI^+ = DEFI$

This FD violates BCNF as the closure of DEI is not a superkey for relation T

<u>Closure of F:</u>
$F^+ = F$

$F \to D$
$F^+ = FD$

This FD violates BCNF as the closure of F is not a superkey for relation T

<u>Closure of EH:</u>
$EH^+ = EH$

EH→CJ
EH^+= CEHJ

Because we already know C is a superkey, we can conclude this closure will also be a superkey for relation T. Therefore, this satisfies BCNF.

Closure of J:
J^+= J

J→FGI
J^+= FGIJ

F→D
J^+= DFGIJ

This FD violates BCNF as the closure of J is not a superkey for relation T

The FD's that violate BCNF are:
DEI→F
F→D
J→FGI

(b) Decompose T into a collection of relations that are each in BCNF, using the BCNF decomposition algorithm. Your decomposition should be lossless and redundancy-preserving. Show your intermediate steps, explaining any steps that you may legitimately omit. Project S_T onto your final set of relations, where every LHS and RHS is in alphabetical order, and each list of projected FDs itself should be in alphabetical order.

Step 1:
Decompose T to using the first FD that does not satisfy BCNF (DEI→F)
Using the closer of DEI obtained in part a, we are able to create two new relations:
DEI^+= DEFI
R1 (D, E, F, I)
R2 (C, D, E, G, H, I, J)

Step 2:
Project the FD's onto R1
D^+= D no FD's
E^+= E no FD's
F^+= FD obtained from part a violates BCNF as it is not a superkey to R1

Step 3:
Decompose R1 using the FD, F→D, as it violates BCNF
Using the FD of F, we are able to create two new relations:

R1 (E, F, I)
R2 (F, D)
R3 (C, D, E, G, H, I, J)

Step 4:
Project FD's to R2
R2 (F, D)
F^+= DF projected onto R2 becomes F^+= F
D^+= D no FD's
This relation satisfies BCNF with FD F→D

Step 5:
Project FD's to R1
R1 (E, F, I)
E^+= E no FD's
F^+= DF projected onto R1 becomes F^+= F
I^+= I no FD's
EF^+= EF no FD's
EI^+= EI no FD's
FI^+= FI no FD's
R1 satisfies BCNF as there is no FD's that exist in R1

Step 6:
Project FD's to R3 to see if R3 violates BCNF
R3 (C, D, E, G, H, I, J)
Because C is already a superkey, we are able to ignore the closure of C
D^+=D no FD's
E^+=E no FD's
G^+= G no FD's
H^+= H no FD's
I^+= I no FD's
J^+= DFGIJ projected onto this relation becomes J^+= DGIJ which violates BCNF

Step 7:
Decompose R3 using the FD, J→DGI as it violates BCNF
Using the FD of J, we are able to create two new relations:
R1 (E, F, I)
R2 (F, D)
R3 (C, E, H, J)
R4 (D, G, I, J)

Step 8:
Project FD's to R4

R4 (D, G, I, J)
$D^+ = D$ no FD's
$G^+ = G$ no FD's
$I^+ = I$ no FD's
$J^+ = DFGIJ$ projected onto this relation becomes $J^+ = DGIJ$
We can now ignore any combination of J as it is a superkey for this relation
$DG^+ = DG$ no FD's
$DI^+ = DI$ no FD's
$GI^+ = GI$ no FD's
R4 satisfies BCNF with FD J→DGI

Step 9:
Check R3 to see if any FD's violate BCNF
R3 (C, E, H, J)
We are able to ignore anything with C or EH as C or EH is already a superkey for relation T
$D^+ = D$ no FD's
$E^+ = E$ no FD's
$H^+ = H$ no FD's
$J^+ = DFGIJ$ projected onto R3 becomes $J^+ = J$
$EJ^+ = EFGIJ$ projected onto R3 becomes $EJ^+ = EJ$
$HJ^+ = FGHIJ$ projected onto R3 becomes $HJ^+ = HJ$
R3 satisfies BCNF with FD's EH→CJ and C→EH

As all the relations can not be further decomposed as all 4 relations satisfy BCNF.

Final decomposition:
R1 (E, F, I) with no FD's
R2 (F, D) with FD, F→D
R3 (C, E, H, J) with FD, C→EH and EH→CJ
R4 (D, G, I, J) with FD, J→DGI