Question1

(1) No,According to R’s FD set F={AB→CE,D→GH,E→BCD, C →DI, H→G, EH→I} C^+ ={C,G,H,I,D},which isn’t include J. In conclusion,”C→J” does not belong to F^+

(2) Candidate key: {A,B,J},{A,E,J}

(3) R= {A,B,C,D,E,G,H,I,J}

Set F ={ AB→CE,D→GH,E→BCD, C→DI, H→G, EH→I }

Step1:Reduce right side

F’ ={ AB→C , AB→E, D→G, D→H, E→B, E→C, E→D, C→D, C→I, H→G, EH→I }

Step2:Reduce left side

For AB→C

A^+ ={A},thus, A→C is not inferred by F’.Hence, AB→C can not be replaced by A→C

B^+ ={B},thus, B→C is not inferred by F’.Hence, AB→C can not be replaced by B→C

For AB→E

A^+ ={A},thus, A→E is not inferred by F’.Hence, AB→E can not be replaced by A→E

B^+ ={B},thus, B→E is not inferred by F’.Hence, AB→E can not be replaced by B→E

For EH→I

E^+ ={B,C,D,E,G,H,I},thus, E→I is inferred by F'.Hence, EH→I can not be replaced by E→I

H^+ ={G,H},thus, H→I is not inferred by F'.Hence, EH→I can not be replaced by H→I

In conclusion,

F''={AB→C , AB→E, D→G, D→H, E→B, E→C, E→D, C→D, C→I, H→G, E→I }

Step3:Remove redundant FDs

$AB^+|_{F''-\{\mathrm{AB\to C}\}}$={A,B,C,D,E,G,H,I},thus AB→C is redundant.Hence,we can remove AB→C from F'' to obtain F'''.

$AB^+|_{F''-\{\mathrm{AB\to E}\}}$={A,B,C,D,G,H,I},thus AB→E is not inferred by F''-{AB→E},which shows that AB→E is not redundant.

$D^+|_{F''-\{\mathrm{D\to G}\}}$={D,G,H},thus D→G is redundant.Thus, we can remove D→G from F'' to obtain F'''.

$D^+|_{F''-\{\mathrm{D\to H}\}}$={D,G},thus D→H is not inferred by F''-{D→H},which shows that D→H is not redundant.

$E^+|_{F''-\{\mathrm{E\to B}\}}$={E,C,D,G,H,I},thus E→B is not inferred by F''-{E→B},which shows that E→B is not redundant.

$E^+|_{F''-\{\mathrm{E\to C}\}}$={E,B,D,G,H,I},thus E→C is not inferred by F''-{E→C},which shows that E→C is not redundant.

$E^+|_{F''-\{\mathrm{E}\rightarrow\mathrm{D}\}}$={E,B,C,D,G,H,I},thus E→D is redundant.

Thus,we can remove E→D from F'' to obtain F'''.

$C^+|_{F''-\{\mathrm{C}\rightarrow\mathrm{D}\}}$={C,I},thus C→D is not inferred by F''-{C→D},which shows that C→D is not redundant.

$C^+|_{F''-\{\mathrm{C}\rightarrow\mathrm{I}\}}$={C,D,H,G},thus C→I is not inferred by F''-{C→I},which shows that C→I is not redundant.

$H^+|_{F''-\{\mathrm{H}\rightarrow\mathrm{G}\}}$={H},thus H→G is not inferred by F''-{H→G},which shows that H→G is not redundant.

$E^+|_{F''-\{\mathrm{E}\rightarrow\mathrm{I}\}}$={E,B,C,D,G,H,I},thus E→I is redundant.

Thus,we can remove E→I from F'' to obtain F'''.

In conclusion

F'''={ AB→E, D→H, E→B, E→C, C→D, C→I, H→G }

(4)

From part(3) we have the minimal cover:

F'''={ AB→E, D→H, E→B, E→C, C→D, C→I, H→G }

So,we can decompose R into

Decompositions	Inferred from
R_1={A,B,E}	AB→E
R_2={D,H}	D→H

R_3={B,C,E}	E→B, E→C,
R_4={C,D,I}	C→D, C→I
R_5={H,G}	H→G
R_6={A,B,J}	Nothing to infer J, A,B,J is super key

It is obvious that the decompositions are dependency-preserving since all of which are inferred from the minimal cover F’’’

To prove the lossless-join we need to draw this table below ,we fill up “a” in the light of R_1 to R_6

	A	B	C	D	E	G	H	I	J
R_1	a	a			a				
R_2				a			a		
R_3		a	a		a				
R_4			a	a				a	
R_5						a	a		
R_6	a	a							a

The column A and B have ‘a’ in R_1 and R_6 then,we seek a FD which contain A and B in the left side,obviously AB→E is we looking for.Therefore,we can change the table.

	A	B	C	D	E	G	H	I	J

R_1	a	a			a				
R_2				a			a		
R_3		a	a		a				
R_4			a	a				a	
R_5						a	a		
R_6	a	a			a				a

We can redo the same thing .This time we look at the column E,we can use E→C to change this table

	A	B	C	D	E	G	H	I	J
R_1	a	a			a				
R_2				a			a		
R_3		a	a		a				
R_4			a	a				a	
R_5						a	a		
R_6	a	a	a		a				a

Iteratively, for R_4 and R_6we use C→D, C→I to fill up ‘a’

	A	B	C	D	E	G	H	I	J
R_1	a	a			a				
R_2				a			a		
R_3		a	a		a				
R_4			a	a				a	
R_5						a	a		

R_6	a	a	a	a	a			a	a

Iteratively, for R_2 and R_6 we use D→H fill up ‘a’

	A	B	C	D	E	G	H	I	J
R_1	a	a			a				
R_2				a			a		
R_3		a	a		a				
R_4			a	a				a	
R_5						a	a		
R_6	a	a	a	a	a		a	a	a

Iteratively, for R_5 and R_6 we use H→G fill up ‘a’

	A	B	C	D	E	G	H	I	J
R_1	a	a			a				
R_2				a			a		
R_3		a	a		a				
R_4			a	a				a	
R_5						a	a		
R_6	a	a	a	a	a	a	a	a	a

For row R_6 ,this row can be filled up with ‘a’ in all columns

Hence,this decomposition is lossless-join

Question2

(1) The number of super keys is 96.

List 5 of them: {A,B,J},{A,B,C,J},{A,B,D,J},{A,E,J},{A,E,C,J}

(2) 2NF

Every attributes in R are atomic and all non-prime attributes are fully functionally dependent on the relation keys.But for FDs E→BCD and C→D,where exist E→C→D,C↛E.Hence the attributes of D are transitively dependent on E.Thus,R is not satisfy 3NF .Hence,the highest normal form of R with respect to F is 2NF.

(3) No,from Question1 Part(3),we can have minimal cover F''' = { AB→E, D→H, E→B, E→C, C→D, C→I, H→G },it is obvious that function dependency D→H and C→I cannot be preserved after this decomposition because R_1 ,R_2 and R_3 don't have C and I or D and H at same time.

(4) To prove the lossless-join we need to draw this table below ,we fill up "a" according to R_1 , R_2, R_3

	A	B	C	D	E	G	H	I	J
R_1	a	a	a	a	a				
R_2					a	a	a		
R_3					a			a	a

Firstly,we seek for column E ,and we use E→BCD to fill up ‘a’ in this table

	A	B	C	D	E	G	H	I	J
R_1	a	a	a	a	a				
R_2		A	a	a	a	a	a		
R_3		a	a	a	a			a	a

And then, we seek for column C, and we use C→I to fill up ‘a’ in this table

	A	B	C	D	E	G	H	I	J
R_1	a	a	a	a	a			a	
R_2		a	a	a	a	a	a	a	
R_3		a	a	a	a			a	a

And then, we seek for column D, and we use D→GH to fill up ‘a’ in this table

	A	B	C	D	E	G	H	I	J
R_1	a	a	a	a	a	a	a	a	
R_2		a	a	a	a	a	a	a	
R_3		a	a	a	a	a	a	a	a

Finally,we cannot find any rows which can be filled up with ‘a’ in all columns.Hence, this decomposition is not lossless-join.

(5) From Question1 Part4 we have such 3NF decomposition:

R_1={A,B,E},R_2={D,H},R_3={B,C,E},R_4={C,D,I},R_5={H,G} R_6 ={A,B,J}

Which is dependency-preserving and lossless-join.

For R_1={A,B,E}, AB→E is non-trivial and AB is superkey

For R_2={D,H}, D→H is non-trivial and D is superkey

For R_3={E,C,B}, E→B and E→C are non-trivial and E is superkey

For R_4={C,D,I}, C→D, C→I are non-trivial and C is superkey

For R_5={G,H}, H→G is non-trivial and H is superkey

Thus,each FD in this decomposition are non-trivial and left-side is superkey.Hence,this decomposition can satisfy BCNF