Q1.

1. Check if $C \rightarrow J$ ∈ F+

the closure of C is $C^{+}=\{C\}$ according to F, So $C \rightarrow J$ ∈ F+ is wrong ,because $C \rightarrow J$ does not belong to F^{+}

2. List all the candidate keys for R.

According to the algorithm to Compute All the Candidate Keys.

T := ∅, X ={A,B,J} , $\{X\text{-}A\}^{+}$ = {B, J} do not contain all attributes of R, $\{X\text{-}B\}^{+}$ = {A, J} do not contain all attributes of R, $\{X\text{-}J\}^{+}$ = {A, B, C, D, E, G, H, I } do not contain all attributes of R, T :=T ∪ X is {A, B, J}.

And do the same thing. X= {A, E, J}, then any attribute cannot be removed.

T ∪ X is {A, B, J} ∪{A ,E, J}

No available super key which does not contain candidate key in T can be founded.

So all candidate keys are (A, B, J), (A, E, J)

3. Find a minimal cover Fm for F.

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

F = $\{AB \rightarrow CE, D \rightarrow GH, E \rightarrow BCD, C \rightarrow DI, H \rightarrow G, EH \rightarrow I\}$.

Step 1: F' = {AB → C, AB→E, D → G, D→ H, E→ B, E→ C, E →D, $C \rightarrow D$, $C \rightarrow I$, $H \rightarrow G$, $EH \rightarrow I$ }

Step 2: AB → C

A+ = {A}; thus AB → C cannot be replaced by A → C.

B+ = {B}; thus AB → C cannot be replaced by B → C

AB→E

A+ = {A}; thus AB → E cannot be replaced by A → E.

B+ = {B}; thus AB →E cannot be replaced by B → E

$EH \rightarrow I$

E+ = {B, C, D, E, I, G, H,}; thus EH → I can be replaced by E →I

F'' = {AB → C, AB→E, D → G, D→ H, E→ B, E→ C, E →D, $E \rightarrow I$, $C \rightarrow D$,

$C \rightarrow I$, $H \rightarrow G$ }

Step 3:

AB+|F'' – {AB →C} = {A, B, C, D, E, G, H, I}; thus AB→C is inferred by F'' – {AB→C}. That is, AB→C is redundant.

AB+|F'' – {AB →E} = {A, B}; thus AB→E is not inferred by F'' –{AB→E}. That is, AB→C is not redundant.

D+|F'' – {D →G} = {D, G, H}; thus D→G is inferred by F'' –{D→G}. That is, D→G is redundant.

D+|F'' – {D →H} = {D}; thus D→G is not inferred by F'' –{D→G}. That is, D→G is not redundant.

E+|F'' – {E →B} = {E, C, D, I, H, G}; thus E→B is not inferred by F'' –{E→B}. That is, E→B is not redundant.

E+|F'' – {E →C} = {E, B, D, I, H, G}; thus E→C is not inferred by F'' –{E→C}.

That is, E→C is not redundant.

E+|F'' – {E →D} = {E, B, C, D, I, H, G}; thus E→D is inferred by F'' –{E→D}. That is, E→D is redundant.

E+|F'' – {E →I} = {E, B, C, D, I, H, G}; thus E→I is inferred by F'' –{E→I}. That is, E→I is redundant.

C+|F'' – {C →D} = {C, I}; thus C→D is not inferred by F'' –{C→D}. That is, C→D is not redundant.

C+|F'' – {C →I} = {C, D, H, G}; thus C→I is not inferred by F'' –{C→I}. That is, C→I is not redundant.

H+|F'' – {H →G} = {H}; thus H→G is not inferred by F'' –{H→G}. That is, H→G is not redundant.

Thus, Fmin = {AB→E, D→H, E→B, E→C, C→D, C →I, H →G}.

4. Decompose into a set of 3NF relations if it is not in 3NF. Make sure your decomposition is dependency-preserving and lossless-join. Justify your answers

We have decompositions from the minimal cover.

R1 = {A, B, E} inferred from: AB→E

R2= {D, H} inferred from: D→H

R3= {B, E, C} inferred from: E→B, E→C

R4= {C, D, I} inferred from: C→D, C→I

R5= {H, G} inferred from: H →G

R6= {A, B, J}   inferred from: A, B, J is a candidate key

It is dependency-preserving because all of these relations are inferred from minimal cover.

Now justify it is lossless-join, I will just ignore some steps and give the final situation and the blank space means b.

Initially, S is

	A	B	C	D	E	G	H	I	J
R1	a	a			a				
R2				a			a		
R3		a	a		a				
R4			a	a				a	
R5						a	a		
R6	a	a							a

After check some FD, it becomes

	A	B	C	D	E	G	H	I	J
R1	a	a	a		a				
R2				a			a		
R3		a	a		a				
R4			a	a			a	a	
R5						a	a		
R6	a	a	a		a				a

When the algorithm stops, it becomes

	A	B	C	D	E	G	H	I	J
R1	a	a	a	a	a		a	a	
R2				a		a	a		
R3		a	a	a	a			a	
R4			a	a		a	a	a	
R5						a	a		
R6	a	a	a	a	a	a	a	a	a

Now row 6 is entirely a’s, so the decomposition is lossless

Q2

1. How many super keys can be found for R? Compute the total number of super keys and list 5 of them

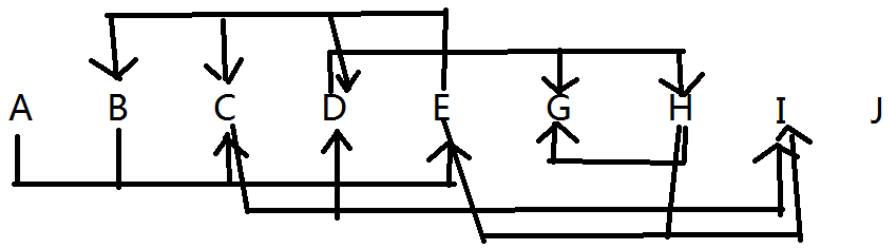
we need to find the candidate key first. It is much similar to the process in Q1, So I will omit the detailed process and give the result.

The candidate keys are (A, B, J), (A, E, J),

So, we have some combinations and we can choose other attribute from R. First, choose only one attribute, which equals to C_6^1 x $2-1$ =11 .we have 6 choices to get one element from the remaining 6 elements and minus one combination that has been computed twice. Then, choose 2 elements. Which equals to C_6^2 x $2-C_5^1$ =25 ,which need to minus the combination that have computed ,namely choose one element that may have combined with E. Then, compute the remaining possible combinations, choose 3, 4, 5, 6 from (C, D, E, G, H, I), which is just considered from (A,B,J). And I times 2, because we have another candidate keys, and which need to do the same thing. As for the rest combinations, I give the formula as follows. Choose 3: $C_6^3 \times 2 - C_5^2$ =30 . Choose 4 attributes: $C_6^4 \times 2 - C_5^3$ =20. Choose 5 attributes : $C_6^5 \times 2 - C_5^4$ =7. Choose 6 attributes $C_6^6 \times 2 - C_5^5$ =1. So the total number of super keys is 11+25+30+20+7+1+2(the two candidate keys, means choose no extra attributes) =96

List 5 of them: (A, B, J), (A, E, J), (A, B, C, J), (A, B, E, J), (A, D, B, J)

2. Determine the highest normal form of R with respect to F. Justify your answer



It is not 2NF, because non-prime attribute C is partially dependent on ABJ, based on $AB \rightarrow CE$ (AB is a part of candidate key). So C partially dependent on ABJ. So R is 1NF.

3. Regarding F, is the decomposition R1 = $\{ABCDE\}$, R2 = $\{EGH\}$, R3 = $\{EIJ\}$ of R dependency-preserving? Please justify your answer

F = $\{AB \rightarrow CE, D \rightarrow GH, E \rightarrow BCD, C \rightarrow DI, H \rightarrow G, EH \rightarrow I\}$. But we can infer other FD from F. E→GH

We have F1 = {AB → CE, E →BCD}, F2= {E→GH}, F3= {E →E, I →I, J →J}

F1 U F2 U F3 not equals to F, so it is not dependency preserving

4. Regarding F, is the decomposition R1 = $\{ABCDE\}$, R2 = $\{EGH\}$, R3 = $\{EIJ\}$ of R lossless-join? Please justify your answer

Initial:

	A	B	C	D	E	G	H	I	J
R1	a	a	a	a	a				
R2					a	a	a		
R3					a			a	a

Column E has the same value, so we add variables to column B,C,D based on $E \rightarrow BCD$

	A	B	C	D	E	G	H	I	J
R1	a	a	a	a	a				
R2		a	a	a	a	a	a		
R3		a	a	a	a			a	a

Column C has the same value, so we add variables to column D, I based on $C \rightarrow DI$

	A	B	C	D	E	G	H	I	J
R1	a	a	a	a	a			a	

R2		a	a	a	a	a	a	a	
R3		a	a	a	a			a	a

Column D has the same value, so we add variables to column G,H based on $D \rightarrow GH$

	A	B	C	D	E	G	H	I	J
R1	a	a	a	a	a	a	a	a	
R2		a	a	a	a	a	a	a	
R3		a	a	a	a	a	a	a	a

Now, we cannot change the S with any FD, so it is not lossless join

5. Decompose it into a collection of BCNF relations if it is not in BCNF. Make sure your decomposition is lossless-join and briefly justify your answers

A+={A}, B+={B}, C+={CDI}, D+={DGH}, E+={EBCD}, G+={G}, H+={HG}, I+={I}, J+={J}

Consider AB → CE, AB is not a super key, spilt it into R1{ABCE} R2{ABDGHIJ} E →BC. So R11={AB}, no FD, so it is BCNF R12={EBC}. E is candidate key, the only FD that applies is E-> BC, so it is BCNF. D+={DGH}, so R21={DGH}, the only FD that applies is D-> GH, so it is BCNF. R22={ABDIJ}, AB->DI, so the candidate key of R22 is (ABJ), R221=(ABDI), the only FD that applies is AB-> DI (derived from AB->CE, C->DI), so it is BCNF, R222=(ABJ), no FD, is BCNF. So the decomposition is R1={AB}, R2={EBC}, R3={DGH}, R4={ABDI}, R5=(ABJ)

Initially, S is

	A	B	C	D	E	G	H	I	J
R1	a	a							
R2			a	a	a		a		
R3				a		a	a		
R4	a	a		a				a	
R5	a	a							a

AB-> CE

	A	B	C	D	E	G	H	I	J
R1	a	a	a		a				
R2			a	a	a		a		
R3				a		a	a		
R4	a	a	a	a	a			a	
R5	a	a	a		a				a

E-> BCD

	A	B	C	D	E	G	H	I	J
R1	a	a	a	a	a				
R2		a	a	a	a		a		
R3				a		a	a		
R4	a	a	a	a	a			a	
R5	a	a	a	a	a				a

D-> GH

	A	B	C	D	E	G	H	I	J
R1	a	a	a	a	a	a	a		
R2		a	a	a	a	a	a		
R3				a		a	a		
R4	a	a	a	a	a	a	a	a	
R5	a	a	a	a	a	a	a		a

C-> DI

	A	B	C	D	E	G	H	I	J
R1	a	a	a	a	a	a	a	a	
R2		a	a	a	a	a	a	a	
R3				a		a	a		
R4	a	a	a	a	a	a	a	a	
R5	a	a	a	a	a	a	a	a	a

So, it is lossless-join, because row5 is entirely made up by a.