Homework 4

19.2

1. $A \to B$, $BC \to E$, $ED \to A$

$(CD)^+ = CD$

So CD is not a key.

$(BCD)^+ = B, C, D, E, A$

$(ECD)^+ = E, C, D, A, B$

$(ACD)^+ = A, C, D, B, E$

So, ACD, BCD, CDE are keys for R.

2. R is in 3NF, because A, B and E are all parts of keys. Transtive dependencies are not present.

3. R is not in BCNF, because L.H.S attributes A, BC and ED do not contain a key.

19.3

1. According to the Functional dependency, if every value of X is uniquely able to indentify value of y, then $X \to Y$ exist. if not, then not exist. Similarly, following functional dependency to check all other attributes. Thus, Finally we got that $X \to Y$, $Z \to Y$, $XZ \to Y$ that relation instance satisfies.

2. If last record in relation is changed from Z_3 to Z_2, all above functional dependency sets are unchanged and no new functional dependency is added. So, we can still check this use the question 1 rule.

19.5

1. $R_1(A, C, B, D, E)$, $A \to B$, $C \to D$

(a) 1NF

(b) BCNF decomposition: AB, CD, ACE

2. $R_2(A, B, F)$, $AC \rightarrow E$, $B \rightarrow F$

(a) 1NF

(b) BCNF decomposition: AB, BF

3. $R_3(A, D)$. $D \rightarrow G$, $G \rightarrow H$

(a) BCNF

(b) No need to decompose the relation R_3.

4. $R_4(D, C, H, G)$, $A \rightarrow I$, $I \rightarrow A$

(a) BCNF

(b) No need to decompose the relation R_4.

5. $R_5(A, I, C, E)$

(a) BCNF

(b) No need to decompose the relation R_5.

19.6

1.

A	B	C
1	2	3
4	2	3
5	3	3

(a) $A \rightarrow B$ V

(b) $BC \rightarrow A$ X

(c) $B \rightarrow C$ V

Ans: $BC \rightarrow A$ does not hold over schema S.

2. No. For example: $A \rightarrow B$, $B \rightarrow C$. We can say that certain dependencies are not violated, but we can't say that these dependencies hold with respect to S. Because, we need more tuples or instances to make sure.

19.7 1. $C \rightarrow D$, $C \rightarrow A$, $B \rightarrow C$

(a) $(A)^+ = A$

$(B)^+ = B, C, D, A$

$(C)^+ = C, D, A$

$(D)^+ = D$

Therefore, the candidate key is B.

(b) It satisfied 2NF not 3NF

(c) The functional that violating the properties of BCNF are $C \rightarrow D$, $C \rightarrow A$. Let the relation into BCNF. The decomposition into CD, CA, BC.

2. $B \rightarrow C$, $D \rightarrow A$

(a) $(A)^+ = A$

$(B)^+ = B, C$

$(C)^+ = C$

$(D)^+ = D, A$

The candidate key is BD.

(b) It satisfies 1NF not 2NF.

(c) The functional that violating the properties of BCNF are $B \rightarrow C$, $D \rightarrow A$. Let the relation into BCNF. The decomposition into DA, BD, BC.

3. $ABC \rightarrow D$, $D \rightarrow A$.

(a) $(ABC)^+ = A, B, C, D$

$(D)^+ = D, A$

$(BCD)^+ = B, C, D, A$

The candidate keys are ABC and BCD.

(b) It satisfies 3NF not BCNF.

(c) The functional that violating the properties of BCNF is $D \rightarrow A$. Let the relation into BCNF. The decomposition into DA, BCD. But functional dependency $ABC \rightarrow D$ will not be preserved. So, there is no BCNF decomposition.

4. $A \to B$, $BC \to D$. $A \to C$

(a) $(A)^+ = A, B, C, D$

$(BC)^+ = B, C, D$

The candidate key is A.

(b) It satisfies 2NF not 3NF.

(c) The functional that violating the properties of BCNF is $BC \to D$. Let the relation into BCNF. The decomposition into BCD, ABC.

5. $AB \to C$, $AB \to D$, $C \to A$, $D \to B$.

(a) $(AB)^+ = A, B, C, D$

$(C)^+ = C, A$

$(D)^+ = D, B$

The candidate keys are AB, CD, BC, AD.

(b) It satisfies 3NF not BCNF.

(c) The functional that violating the properties of BCNF are $C \to A$, $D \to B$. Let the relation into BCNF. The decomposition: DA, BCD. But functional dependency $AB \to C$, $AB \to D$ will not be preserved. Since $D \to B$, BCD is still not BCNF. So, there is no BCNF decomposition.

$\boxed{19.10}$ 1. $B \to C$, $D \to A$; decompose into Bc and AD.

(a) $(BD)^+ = A, B, C, D$

$(B)^+ = B, C$

$(D)^+ = D, A$

Candidate keys: BD.

(b) The decomposition into BC, AD is unsatisfactory. Because there are no comman attribute between Bc and AD. The cartesian product would be bigger than ABCD.

2. $AB \to C$, $C \to A$. $C \to D$; decompose into ACD and BC.

(a) $(AB)^+ = A, B, C, D$

$(CB)^+ = C, B, A, D$

candidate keys : AB , CB

(b) ACD and BC have common attribute C.

$R_1(A, C, D)$

$(AC)^+$	$(AD)^+$	$(CD)^+$	$(A)^+$	$(C)^+$	$(D)^+$
ACD	AD	CDA	A	CAD	D

$\Rightarrow F_1 (C \to AD)$, The super key is C in R_1.

$R_2(B, C)$

$(B)^+$	$(C)^+$
B	CAD

$\Rightarrow F_2 \; \Phi$

$F_1 \cup F_2 \equiv C \to AD \not\equiv AB \to C, \; C \to A, \; C \to D$.

It's not dependency preserving since $AB \to C$ is not preserved.
Therefore, This is not Lossless join decomposition.

3. $A \to BC$, $C \to AD$; decompose into ABC, AD.

(a) $(A)^+ = A, B, C, D$

$(C)^+ = C, A, D, B$

candidate key : A . C .

(b) ABC and AD have common attrabute A.

$R_1 (A, B, C)$

$(AB)^+$	$(AC)^+$	$(BC)^+$	$(A)^+$	$(B)^+$	$(C)^+$
ABCD	ACBD	BCAD	ABCD	B	CADB

$F_1 (A \to BC, C \to AB)$ A and C are super keys.

$R_2(A, D)$

A^+	D^+
ABCD	D

$F_2(A \to BC)$

$F_1 \cup F_2 \equiv A \to BC, C \to AB \not\equiv A \to BC, C \to AD.$

It's not dependency preserving since $C \to AD$ is not preserved.

Therefore, This is not Lossless join decomposition.

4. $A \to B, B \to C, C \to D$; decompose into AB and ACD.

(a) Candidate key : A.

(b) AB and ACD have common attribute A.

$R_1(A, B)$

$(A)^+$	$(B)^+$
ABCD	BCD

$F_1(A \to B)$ super key is A

$R_2(A, C, D)$

$(AC)^+$	$(AD)^+$	$(CD)^+$	$(A)^+$	$(C)^+$	$(D)^+$
ACBD	ADBC	CD	ABCD	CD	D

$F_2(A \to CD)$ super key is A

$F_1 \cup F_2 \equiv A \to B, A \to CD \not\equiv A \to B, B \to C, C \to D$

It's not dependency preserving since $B \to C$ is not preserved.

Therefore, This is not Lossless join decomposition.

5. $A \to B, B \to C. C \to D$; decompose into AB, AD and CD.

(a) Candidate key : A.

(b) AB and AD have common attribute A.

AD and CD have common attribute D.

$R_1 (A, B)$

A^+	B^+
ABCD	BCD

$F_1 (A \rightarrow B)$

$R_2 (A, D)$

A^+	D^+
ABCD	D

$F_2 (A \rightarrow D)$

$R_3 (C \rightarrow D)$

C^+	D^+
CD	D

$F_4 (C \rightarrow D)$

$F_1 \cup F_2 \equiv A \rightarrow B, B \rightarrow C, C \rightarrow D.$

Therefore, This is Lossless join decomposition, but this is not the best decomposition. AB, BC, CD is better.