Chapter 11, Relational Database Design Algorithms and Further Dependencies

• Normal forms are insufficient on their own as a criteria for a good relational database schema design.

• The relations in a database must collectively satisfy two other properties - dependency preservation property and lossless (or nonadditive) join property - to qualify as a good design.

11.1 Properties of Relational Decompositions 11.1.1 Relation Decomposition and Insufficiency of Normal Forms

• The relational database design algorithms discussed in this chapter start from a single universal relation schema $R = \{A_1, A_2, \ldots, A_n\}$ that includes all the attributes of the database. Using the functional dependencies F, the algorithms decompose R into a set of relation schemas $DECOMP = \{R_1, R_2, \ldots, R_m\}$ that will become the relational database schema.

• Examine an individual relation R_i to test whether it is in a higher normal form does not guarantee a good design (decomposition); rather, a set of relations that together form the relation database schema must possess certain additional properties to ensure a good design.

– Attribute preservation property: Each attribute in R will appear in at least one relation $R_i$in the decomposition so that no attributes are 'lost'; formally we have

$$\bigcup_{i=1}^{m} R_i = R$$

– Dependency preservation property: See Figure 10.12 (Fig 14.12 on e3). Dis cuss on 11.1.2

– Lossless (nonadditive) join property: If we decompose EMP PROJ in Figure 10.2 (Fig 14.2 on e3) to EMP LOCS and EMP PROJ1 in Figure 10.5 (Fig 14.5

on e3), it will violate the nonadditive join property. Discuss on 11.1.3

11.1.2 Decomposition and Dependency Preservation

- It is not necessary that the exact dependencies specified in F on R appear themselves in individual relations of the composition DECOMP. It is sufficient that the union of the dependencies that hold on individual relations in DECOMP be equivalent to F.

- The projection of F on R_i, denoted by $\pi_{R_i}(F)$, is the set of dependencies $X \rightarrow Y$ in F^+ such that the attributes in $X \cup Y$ are contained in R_i.

- A decomposition DECOMP = $\{R_1, R_2, \ldots, R_m\}$ of R is dependency-preserving with respect to F if

$$((\pi_{R_1}(F)) \cup \ldots \cup (\pi_{R_m}(F)))^+ = F^+$$

- claim 1: It is always possible to find a dependency-preserving decomposition DECOMP with respect to F such that each relation R_i in DECOMP is in 3NF.

11.1.3 Decomposition and Lossless (Nonadditive) Joins

- A decomposition DECOMP = $\{R_1, R_2, \ldots, R_m\}$ of R has the lossless join property with respect to the set of dependencies F on R if, for every relation state r of R that satisfies F, the following holds, where $*$ is the NATUAL JOIN of all the relations in DECOMP:

$$*(\pi_{R_1}(r), \ldots, \pi_{R_m}(r)) = r$$

- The decomposition of EMP PROJ(SSN, PNUMBER, HOURS, ENAME, PNAME, PLOCATION) from Figure 10.3 (Fig 14.3 on e3) into EMP LOCS(ENAME, PLO CATION) and EMP PROJ1(SSN, PNUMBER, HOURS, PNAME, PLOCATION) in Figure 10.5 (Fig 14.5 on e3) does not have the lossless join property.

- Another example: decompose R to R_1 and R_2 as follows.

R				R_1			R_2		
A	B	C	D	A	B	C	B	C	D
1	2	3	4	1	2	3	2	3	4
1	2	3	5	4	2	3	2	3	5

4 2 3 4 3 1 4 1 4 2
3 1 4 2

2

Is R equal to $R_1 * R_2$?

$R_1 * R_2$

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

- Algorithm 11.1 Testing for the lossless (nonadditive) join property

  Input: A universal relation R, a decomposition DECOMP = $\{R_1, R_2, . . . , R_m\}$ of R, and a set F of functional dependencies.

  – 1. Create an initial matrix S with one row i for each relation R_i in DECOMP, and one column j for each attribute A_j in R.

  – 2. Set S(i, j) := b_{ij} for all matrix entries.

  – 3. For each row i

  For each column j

  If R_i includes attribute A_j

  Then set S(i, j) := a_j

  – 4. Repeat the following loop until a complete loop execution results in no changes to S

  For each X → Y in F

  For all rows in S which has the same symbols in the columns corresponding to attributes in X

  make the symbols in each column that correspond to an attribute in Y be the same in all these rows as follows:

  if any of the rows has an “a” symbol for the column, set the other rows to the same “a” symbol in the column.

  If no “a” symbol exists for the attribute in any of the rows, choose one of the “b” symbols that appear in one of the rows for the attribute and set the other rows to that same “b” symbol in the column

– 5. If a row is made up entirely of “a” symbols, then the decomposition has the lossless join property; otherwise it does not.

• Example:

R = {SSN, ENAME, PNUMBER, PNAME, PLOCATION, HOURS} F = {SSN → ENAME, PNUMBER → {PNAME, PLOCATION}, {SSN, PNUMBER} → HOURS}

DECOMP = {R_1, R_2, R_3}

R_1 = {SSN, ENAME}

R_2 = {PNUMBER, PNAME, PLOCATION}

R_3 = {SSN, PNUMBER, HOURS}

SSN ENAME PNUMBER PNAME PLOCATION HOURS R_1 a_1 a_2 b_{13} b_{14} b_{15} b_{16} R_2 b_{21} b_{22} a_3 a_4 a_5 b_{26} R_3 a_1 b_{32} a_3 b_{34} b_{35} a_6

SSN ENAME PNUMBER PNAME PLOCATION HOURS R_1 a_1 a_2 b_{13} b_{14} b_{15} b_{16} R_2 b_{21} b_{22} a_3 a_4 a_5 b_{26} R_3 a_1 $b6_{32}$ a_2 a_3 $b6_{34}$ a_4 $b6_{35}$ a_5 a_6

– Try SSN → ENAME:

$a_1 \rightarrow 6b_{32}$ a_2

– Try PNUMBER → {PNAME, PLOCATION}:

$a_3 \rightarrow 6b_{34}$ a_4

$a_3 \rightarrow 6b_{35}$ a_5

In row 3, all symbols are a_i. The decomposition DECOMP has the lossless join property.

• Another example:

R = {A, B, C, D, E}

F = {A → C, B → C, C → D, DE → C, CE → A}

DECOMP = {R_1, R_2, R_3, R_4, R_5}

$R_1 = \{A, D\}$

$R_2 = \{A, B\}$

$R_3 = \{B, E\}$

$R_4 = \{C, D, E\}$

$R_5 = \{A, E\}$

A B C D E
R_1 a_1 b_{12} b_{13} a_4 b_{15}
R_2 a_1 a_2 b_{23} b_{24} b_{25}
R_3 b_{31} a_2 b_{33} b_{34} a_5
R_4 b_{41} b_{42} a_3 a_4 a_5
R_5 a_1 b_{52} b_{53} b_{54} a_5

– For each F Ds in F (first loop):

∗ Try $A \rightarrow C$:

A B C D E
R_1 a_1 b_{12} b_{13} a_4 b_{15}
R_2 a_1 a_2 $b6_{23}$ b_{13} b_{24} b_{25}
R_3 b_{31} a_2 b_{33} b_{34} a_5
R_4 b_{41} b_{42} a_3 a_4 a_5
R_5 a_1 b_{52} $b6_{53}$ b_{13} b_{54} a_5

∗ Try $B \rightarrow C$:

A B C D E
R_1 a_1 b_{12} b_{13} a_4 b_{15}
R_2 a_1 a_2 b_{13} b_{24} b_{25}
R_3 b_{31} a_2 $b6_{33}$ b_{13} b_{34} a_5
R_4 b_{41} b_{42} a_3 a_4 a_5
R_5 a_1 b_{52} b_{13} b_{54} a_5

∗ Try $C \rightarrow D$:

	A	B	C	D	E
R_1	a_1	b_{12}	b_{13}	a_4	b_{15}
R_2	a_1	a_2	b_{13}	$b6_{24}$ a_4	b_{25}
R_3	b_{31}	a_2	b_{13}	$b6_{34}$ a_4	a_5
R_4	b_{41}	b_{42}	a_3	a_4	a_5
R_5	a_1	b_{52}	b_{13}	$b6_{54}$ a_4	a_5

∗ Try $DE \rightarrow C$:

	A	B	C	D	E
R_1	a_1	b_{12}	b_{13}	a_4	b_{15}
R_2	a_1	a_2	b_{13}	a_4	b_{25}
R_3	b_{31}	a_2	$b6_{13}$ a_3	a_4	a_5
R_4	b_{41}	b_{42}	a_3	a_4	a_5
R_5	a_1	b_{52}	$b6_{13}$ a_3	a_4	a_5

∗ Try $CE \rightarrow A$:

	A	B	C	D	E
R_1	a_1	b_{12}	b_{13}	a_4	b_{15}
R_2	a_1	a_2	b_{13}	a_4	b_{25}
R_3	$b6_{31}$ a_1	a_2	a_3	a_4	a_5
R_4	$b6_{41}$ a_1	b_{42}	a_3	a_4	a_5
R_5	a_1	b_{52}	a_3	a_4	a_5

– The third row is made up entirely of a_i symbols. The decomposition DECOMP has the lossless join property.

- Another example:

  $R = \{A, B, C\}$

  $F = \{AB \rightarrow C, C \rightarrow B\}$

DECOMP = $\{R_1, R_2\}$

– If $R_1 = \{A, B\}$ and $R_2 = \{B, C\}$

	A	B	C
R_1	a_1	a_2	b_{13}
R_2	b_{21}	a_2	a_3

∗ For each F Ds in F (first loop):

· Try $AB \rightarrow C$:

	A	B	C
R_1	a_1	a_2	b_{13}
R_2	b_{21}	a_2	a_3

· Try $C \rightarrow B$:

	A	B	C
R_1	a_1	a_2	b_{13}
R_2	b_{21}	a_2	a_3

· Complete loop execution results in no changes to the matrix and the ma trix does not contain a row with all a_i symbols. This decomposition does not have lossless join property.

– If $R_1 = \{A, C\}$ and $R_2 = \{B, C\}$

	A	B	C
R_1	a_1	b_{12}	a_3
R_2	b_{21}	a_2	a_3

∗ For each F Ds in F (first loop):

· Try $AB \rightarrow C$:

	A	B	C
R_1	a_1	b_{12}	a_3
R_2	b_{21}	a_2	a_3

· Try C → B:

$$\begin{array}{llll} & A & B & C \\ R_1 & a_1 & b6_{12}\ a_2 & a_3 \\ R_2 & b_{21} & a_2 & a_3 \end{array}$$

· Row 1 contains all a_i symbols. This decomposition has the lossless join property.

– If $R_1 = \{A, B\}$ and $R_2 = \{A, C\}$

$$\begin{array}{llll} & A & B & C \\ R_1 & a_1 & a_2 & b_{13} \\ R_2 & a_1 & b_{22} & a_3 \end{array}$$

∗ For each F Ds in F (first loop):

· Try AB → C:

$$\begin{array}{llll} & A & B & C \\ R_1 & a_1 & a_2 & b_{13} \\ R_2 & a_1 & b_{22} & a_3 \end{array}$$

· Try C → B:

$$\begin{array}{llll} & A & B & C \\ R_1 & a_1 & a_2 & b_{13} \\ R_2 & a_1 & b_{22} & a_3 \end{array}$$

· Complete loop execution results in no changes to the matrix and the ma trix does not contain a row with all a_i symbols. This decomposition does not have lossless join property.

11.1.4 Testing Binary Decompositions for the Nonadditive Join Property

• Property LJ1 below is a handy way to decompose a relation into two relations.

– Property LJ1 A decomposition DECOMP = $\{R_1, R_2\}$ of R has the lossless join property with respect to a set of functional dependencies F on R if and only if either

  * The FD $((R_1 \cap R_2) \rightarrow (R_1 - R_2))$ is in F^+, or
  * The FD $((R_1 \cap R_2) \rightarrow (R_2 - R_1))$ is in F^+.

– Use LJ1 to decompose the previous example.

11.1.5 Successive Lossless (Nonadditive) Join Decompositions

- Claim 2: Preservarion of Nonadditivity in Successive Decompositions If a decomposition DECOMP = $\{R_1, R_2, \ldots, R_m\}$ od R has the nonadditive (lossless) join property with respect to a set of functional dependency F on R, and if a decompo sition $D_i = \{Q_1, Q_2, \ldots, Q_k\}$ of R_i has a nonadditive join property with respect to the projection of F on R_i, then the decomposition $D_2 = \{R_1, R_2, \ldots, R_{i-1}, Q_1, Q_2, \ldots, Q_k, R_{i+1}, \ldots, R_m\}$ of R has the nonadditive join property with respect to F.

11.2 Algorithms for Relational Database Schema Design

11.2.1 Dependency-Preserving Decomposition into 3NF Schemas

- Algorithm 11.2 Relational synthesis algorithm with dependency-preserving Input: A universal relation R and a set of functional dependencies F on the attributes of R.

  Output: A dependency-preserving decomposition DECOMP = $\{R_1, R_2, \ldots, R_n\}$ of R that all R_i's in DECOMP are in 3NF.

  – 1. Find a minimal cover G for F;

  – 2. For each left-hand-side X of a functional dependency that appears in G, create a relation schema in DECOMP with attributes $\{X \cup \{A_1\} \cup \{A_2\} \ldots \cup \{A_k\}\}$, where $X \rightarrow A_1, X \rightarrow A_2, \ldots, X \rightarrow A_k$ are the only dependencies in G with X as left-hand-side (X is the key of this relation);

– 3. Place any remaining attributes (that have not been placed in any relation) in a single relation schema to ensure the attribute preservation property.

• Example:

$R = \{A, B, C, D, E, H\}$,

$F = \{AE \rightarrow BC, B \rightarrow AD, CD \rightarrow E, E \rightarrow CD, A \rightarrow E\}$.

Find a dependency-preserving decomposition DECOMP = $\{R_1, R_2, \ldots, R_n\}$ of R such that each R_i in DECOMP is in 3NF.

– Step 1: A minimal cover $G = \{A \rightarrow B, A \rightarrow E, B \rightarrow A, CD \rightarrow E, E \rightarrow CD\}$ of F is derived from algorithm 10.2.

– Step 2: Decompose R to

$R_1 = \{A, B, E\}$ and $F_1 = \{A \rightarrow B, A \rightarrow E\}$

$R_2 = \{B, A\}$ and $F_2 = \{B \rightarrow A\}$

$R_3 = \{C, D, E\}$ and $F_3 = \{CD \rightarrow E\}$

$R_4 = \{E, C, D\}$ and $F_4 = \{E \rightarrow CD\}$

Combine R_3 and R_4 into one relation schema

$R_5 = \{C, D, E\}$ and $F_5 = \{CD \rightarrow E, E \rightarrow CD\}$.

– Step 3: There is one attribute H in $R - (R_1 \cup R_2 \cup R_5)$. Create another relation schema to contain this attribute.

$R_6 = \{H\}$ and $F_6 = \{\}$.

All relational schemas in the decomposition DECOMP = $\{R_1, R_2, R_5, R_6\}$ are in 3NF.

• Notice that the dependency are preserved: $\{F_1 \cup F_2 \cup F_5 \cup F_6\}^+ = F^+$. • Claim

3: Every relation schema created by Algorithm 11.2 is in 3NF.

11.2.2 Lossless (Nonadditive) Join Decomposition into BCNF Schemas

- Algorithm 11.3 Relational decomposition into BCNF relations with lossless join preperty

  Input: A universal relation R and a set of functional dependencies F on the attributes of R.

  – 1. Set DECOMP = {R};

  – 2. While there is a relation schema Q in DECOMP that is not in BCNF do {

  choose a relation schema Q in DECOMP that is not in BCNF;
  find a functional dependency $X \rightarrow Y$ in Q that violates BCNF;
  replace Q in DECOMP by two relation schemas $(Q - Y)$ and $(X \cup Y)$; };

  Why: Since $(Q - Y) \cap (X \cup Y) \rightarrow (X \cup Y) - (Q - Y)$ is equvalent to $X \rightarrow Y \in F^+$. By Property LJ1, the decompsition is lossless.

- Example: See Figure 10.11, 10.12, and Figure 10.13 (Fig 14.11, 14.12, 14.13 on e3).

- Example:

  R = {A, B, C}

  F = {AB → C, C → B}

  – Step 1: Let DECOMP = {{A, B, C}};

  – Step 2: {A, B, C} in DECOMP that is not in BCNF;

  Pick {A, B, C} in DECOMP;
  Pick C → B in {A, B, C} that violates BCNF;
  Replace {A, B, C} in DECOMP by {A, C} and {B, C};

  Step 2: DECOMP = {{A, C}, {B, C}} and both {A, C} and {B, C} are in BCNF;
  Therefore, the decomposition DECOMP = {{A, C}, {B, C}} has the lossless join property.

  (Try to use the Algorithm 11.1 to test this decomposition)

- Another example:

R = {A, B, C, D, E}

F = {AB → CDE, C → A, E → D}

– ‘ Step 1: Let DECOMP = {{A, B, C, D, E}};

– Step 2: {A, B, C, D, E} in DECOMP that is not in BCNF;

Pick {A, B, C, D, E} in DECOMP;

Pick C → A in {A, B, C, D, E} that violates BCNF;

Replace {A, B, C, D, E} in DECOMP by {B, C, D, E} and {A, C};

Step 2: DECOMP = {{B, C, D, E}, {A, C}} and {B, C, D, E} in DECOMP that is not in Pick {B, C, D, E} in DECOMP;

Pick E → D in {B, C, D, E} that violate BCNF;

Replace {B, C, D, E} in DECOMP by {B, C, E} and {D, E};

Step 2: DECOMP = {{A, C}, {B, C, E}, {D, E}} and all of them are in BCNF;

Therefore, the decomposition DECOMP has the lossless join property.

(Try to use the Algorithm 11.1 to test this decomposition)

- The same example, but try to pick a different FD first that violate BCNF

R = {A, B, C, D, E}

F = {AB → CDE, C → A, E → D}

– Step 1: Let DECOMP = {{A, B, C, D, E}};

– Step 2: {A, B, C, D, E} in DECOMP that is not in BCNF;

Pick {A, B, C, D, E} in DECOMP;

Pick E → D in {A, B, C, D, E} that violates BCNF;

Replace {A, B, C, D, E} in DECOMP by {A, B, C, E} and {D, E};

Step 2: DECOMP = {{A, B, C, E}, {D, E}} and {A, B, C, E} in DECOMP that is not in Pick {A, B, C, E} in DECOMP;

Pick C → A in {A, B, C, E} that violate BCNF;

Replace {A, B, C, E} in DECOMP by {B, C, E} and {A, C};

Step 2: DECOMP = {{A, C}, {B, C, E}, {D, E}} and all of them are in BCNF;

Therefore, the decomposition DECOMP has the lossless join property.

The order of F Ds to be applied for decomposition does not matter.

11.2.3 Dependency-Preserving and Nonadditive (Lossless) Join Decomposition into 3NF Schemas

- Algorithm 11.4 Relational synthesis into 3NF with dependency preservation and Nonadditive (lossless) join property

  Input: A universal relation R and a set of functional dependencies F on the attributes of R.

  Output: A dependency-preserving and lossless-join decomposition DECOMP = $\{R_1, R_2, \ldots, R_n\}$ of R that all R_i's in DECOMP are in 3NF.

  – 1. Find a minimal cover G for F (use algorithm 10.2).

  – 2. For each left-hand-side X of a functional dependency that appears in G create a relation schema in DECOMP with attributes $\{X \cup \{A_1\} \cup \{A_2\} \ldots \cup \{A_k\}\}$, where $X \rightarrow A_1, X \rightarrow A_2, \ldots, X \rightarrow A_k$ are the only dependencies in G with X as left-hand-side (X is the key of this relation)

  – 3. If none of the relation schemas in DECOMP contains a key of R, then create one more relation schema in D that contains attributes that form a key of R.

- Example:

  R = {A, B, C, D, E, H},

  F = {AE → BC, B → AD, CD → E, E → CD, A → E}.

  Find a dependency-preserving and lossless-join decomposition DECOMP = $\{R_1, R_2, \ldots, R_n\}$ of R such that each R_i in DECOMP is in 3NF.

  – Step 1: A minimal cover G = {A → B, A → E, B → A, CD → E, E → CD} of F is derived from algorithm 10.2.

– Step 2: Decompose R to

R_1 = {A, B, E} and F_1 = {A → B, A → E}

R_2 = {B, A} and F_2 = {B → A}

R_3 = {C, D, E} and F_3 = {CD → E}

R_4 = {E, C, D} and F_4 = {E → CD}

Combine R_3 and R_4 into one relation schema

R_5 = {C, D, E} and F_5 = {CD → E, E → CD}.

– Step 3: AH and BH are candidate keys of R, and neither of them appear in 13

R_1, R_2, R_5. Create another relation schema

R_6 = {A, H} and F = {}

Then, all relational schemas in the decomposition DECOMP = {R_1, R_2, R_5, R_6} are in 3NF.

Dependency preservation: $\{F_1 \cup F_2 \cup F_5 \cup F_6\}^+ = F^+$

Lossless join property: Try to use algorithm 11.1 to test this decomposition.

Appendix: An Example to Summarize Functional Dependencies and Normal Forms

- Let a relation schema R = {A, B, C} with functional dependency set

$$F = \{A \rightarrow BC, AB \rightarrow C, C \rightarrow B\}$$

– Question#1: Is G = {A → B, AB → C, C → B} equivalent to F ?

(i) Does G |= A → BC ?

Since $A^+_G = ABC \supseteq BC \Rightarrow$ Y es.

(ii) Does F |= A → B ?

Since $A^+_F = ABC \supseteq B \Rightarrow$ Y es.

Both (i) and (ii) return true, then G ≡ F.

– Question#2: Find a minimal cover G of F.

(i) Let G_1 = {A → B, A → C, AB → C, C → B}

(ii) Check for left redundancy.

1) Is A redundant in $AB \rightarrow C$ of G_1 ?

Let $G_2 = \{A \rightarrow B, A \rightarrow C, B \rightarrow C, C \rightarrow B\}$

Does $G_1 \models B \rightarrow C$? Since $B^+_{G_1} = B \not\supseteq C \Rightarrow$ No

Does $G_2 \models AB \rightarrow C$?

Thus, A is needed.

2) Is B redundant in $AB \rightarrow C$ of G_1 ?

Let $G_3 = \{A \rightarrow B, A \rightarrow C, C \rightarrow B\}$

G_1 covers G_3 Since $G_1 \supseteq G_3$

Does $G_3 \models AB \rightarrow C$? Since $AB^+_{G_3} = ABC \supseteq C \Rightarrow$ Yes.

Thus, B is redundant.

(iii) Check for FDs redundancy.

1) Is $A \rightarrow B$ redundant in G_3 ?

Let $G_4 = \{A \rightarrow C, C \rightarrow B\}$

Does $G_4 \models A \rightarrow B$? $A^+_{G_4} = ACB \supseteq B \Rightarrow$ Yes.

Thus, $A \rightarrow B$ is redundant.

2) Is $A \rightarrow C$ redundant in G_4 ?

Let $G_5 = \{C \rightarrow B\}$

Does $G_5 \models A \rightarrow C$? $A^+_{G_5} = A \not\supseteq B \Rightarrow$ No.

Thus, $A \rightarrow C$ is needed.

Conclusion, $G = G_4 = \{A \rightarrow C, C \rightarrow B\}$ is a minimal cover of F. –

Question#3: Find all possible candidate keys of R.

(i) One-attribute keys:

$A^+_F = ABC = R \Rightarrow$ A is a superkey (thus, a candidate key).

$B^+_F = B \neq R$

$C^+_F = CB \neq R$

(ii) Two-attribute keys: We only need to check BC because all other combination will contain A.

$BC^+_F = BC \neq R$

(iii) Three-attribute keys: There is only one possible combination ABC. Since A is a key, ABC is not a key.

Conclusion, There is only one candidate key A.

– Question#4: For the general definition, is R in 2NF ? If not, decompose it to 2NF.

There is only one key with length one. Therefore, no partial dependency exist, i.e., R is in 2NF.

– Question#5: For the general definition, are all relation schemas of the resulting decomposition from Q4 in 3NF ? If not, decompose them to 3NF.

(i) For $A \rightarrow BC$, A is a superkey (ok.)

(ii) For $AB \rightarrow C$, AB is a superkey (ok.)

(iii) For $C \rightarrow B$, C is not a superkey and B is not a prime attribute (not ok.)

Decomposition:

$$R(ABC) \Rightarrow \begin{cases} R_1(AC) & F_1 = \{A \rightarrow C\} \\ R_2(BC) & F_2 = \{C \rightarrow B\} \end{cases}$$

– Question#6: For the general definition, are all relation schemas of the resulting decompositions from Q5 in BCNF ? If not, decompose them to BCNF.

Both R_1 and R_2 are in BCNF.

– Question#7: Does the resulting decompositions from previous questions have dependency-preserving and lossless-join properties?

(i) Check for lossless-join property.

	A	B	C
R_1	a_1	b_{12}	a_3
R_2	b_{21}	a_2	a_3

For $A \rightarrow BC$ No change.

$AB \rightarrow C$ No change.

$C \rightarrow B$ change $b_{12} \rightarrow a_2$.

We have a row with all a_i symbols $\Rightarrow$ It has lossless-join property. (ii) Check for dependency-preserving property.

Does $(F_1 \cup F_2) \equiv F$?

1) Does $F \models A \rightarrow C$? Yes.

2) Does $F_1 \cup F_2 \models A \rightarrow BC, AB \rightarrow C$? Yes.

Therefore, it has the dependency-preserving property.