UNIT 2

Lecture 35

Normalization

Properties of Decomposition

Properties of Decomposition

- **<u>Decomposition -</u>** Decomposition means breaking the relation schema into smaller schema. We may also call this schema refinement.
- Suppose a relation schema R (A, B, C, D, E, F, G) is split into R1 (A, B, C, D) and R2 (E, F, G). We can say that R1 and R2 are decompositions of R.
- We can say a relation R is decomposed into a set of relations R1 and R2 if and only if it satisfies the following properties of decomposition.
  1. Attribute Preservation
  2. Lossless Join Decomposition
  3. Dependency Preservation

Attribute Preservation

- If R1 and R2 are projections of some relation R and R1 and R2 between them include all the attributes of R, then we say that R is decomposed into R1 and R2.
- For e.g., Consider the relation R (A, B, C, D, E, F, G) and the decomposition of this relation are R1 (A, B, C, D) and R2 (E, F, G) are the attribute preserving decompositions.

Loss-Less Join Decomposition

- We say that the decomposition of R into R1, with attribute set X1 and R2 with attribute set X2, is loss-less join decomposition if by joining R1 and R2 we get back the original relation R.
- The word loss-less refers to loss of information and not loss of tuples.
- Let R be a relation with attribute set X and F is the set of FDs that hold over R. The decomposition of R into R1 and R2 with attributes X1 and X2 respectively is loss-less join if and only if F^+ contains either

$$X1 \cap X2 \rightarrow X1$$

Or,

$$X1 \cap X2 \rightarrow X2$$

- In other words, the attributes common to R1 and R2 must contain a key for either R1 or R2.
- If an FD, $X \rightarrow Y$ holds over a relation schema R and $X \cap Y$ is empty, the decomposition of R into

  R –Y and XY is loss-less.
- For E.g. Consider the relation schema R(A, B, C) with FDs $AB \rightarrow C$ and $C \rightarrow B$. The relation is not in BCNF since $C \rightarrow B$ holds in R. We decompose this relation into R1(A, C) (i.e. R –Y) and R2(B, C) (i.e. XY) is a loss-join decomposition because $B \cap C$ is empty.

Dependency Preservation

- If R is decomposed into X, Y and Z, and we enforce the FDs that hold on X, on Y and on Z, then all FDs that were given to hold on R must also hold.
- The decomposition of relation R with FDs F into R1 and R2 is said to be dependency preserving iff $(F_{R1} \cup F_{R2})^+ = F^+$.
- In other words, if we take the dependencies of F_{R1} and F_{R2} and compute the closure of their union, we get back the original FDs in F.
- For e.g., Consider the relation R (A, B, C, D) with FDs $A \rightarrow B$ and $C \rightarrow D$ is decomposed into

  R1(A, B) and R2(C, D) is a dependency preserving decomposition.
- It is always possible to find a dependency preserving decomposition with respect to a F such that the resulting relations are in 3NF.
- In general, there may not be a dependency preserving decomposition that also decomposes relations in BCNF.
- For e.g., Consider a relation R(A, B, C) with FDs, $AB \rightarrow C$ and $C \rightarrow B$ is a relation that cannot be decomposed to satisfy both dependency preserving and BCNF.

Properties of Decomposition

Q. 1 Suppose that we decompose the schema R = (A, B, C, D, E) into

(A, B, C)

(A, D, E)

Show that this decomposition is a loss-less join decomposition if the following set F of functional dependencies holds :

$A \rightarrow BC$

$CD \rightarrow E$

$B \rightarrow D$

$E \rightarrow A$

Sol: A decomposition {R1, R2} is loss-less join decomposition if $R1 \cap R2 \rightarrow R1$ or $R1 \cap R2 \rightarrow R2$.

Let R1 = (A, B, C), and R2 = (A, D, E), So $R1 \cap R2 = A$. Since A is a candidate key of both R1 and R2 therefore $R1 \cap R2 \rightarrow R1$ and $R1 \cap R2 \rightarrow R2$ (i.e. A is a candidate key of both relations R1 and R2). So the decomposition is loss-less join decomposition.

Properties of Decomposition

Q. 2 Suppose that we decompose the schema R = (A, B, C, D, E) into

(A, B, C)

(C, D, E)

Show that this decomposition is not a loss-less join decomposition if the following set F of functional dependencies holds :

$A \rightarrow BC$

$CD \rightarrow E$

$B \rightarrow D$

$E \rightarrow A$

Sol: A decomposition {R1, R2} is loss-less join decomposition if $R1 \cap R2 \rightarrow R1$ or $R1 \cap R2 \rightarrow R2$.

Let R1 = (A, B, C), and R2 = (C, D, E), So $R1 \cap R2$ = C. Since C is not a candidate key of any relations R1 and R2, therefore $R1 \cap R2$ cannot determine either R1 or R2 (i.e. C is not a candidate key of any of the relations R1 and R2). So the decomposition is not a loss-less join decomposition, it is lossy decomposition.

Properties of Decomposition

Q. 3 Give a loss-less join decomposition into BCNF of schema

R = (A, B, C, D, E) with a set of FDs

F = {A → BC,

CD → E,

B → D,

E → A}

Sol : The candidate keys of this relation are A, BC, CD and E. The relation schema R = (A, B, C, D, E) with a set of FDs is not in BCNF because the nontrivial Functional dependency B → D holds in R and B is not a candidate key of R. So the decomposition of R1(A, B, C, E) and R2(B, D) is loss-less join decomposition and in BCNF.

Properties of Decomposition

Q. 4 Give a loss-less join, dependency preserving decomposition into 3NF of schema R = (A, B, C, D, E) with a set of FDs

$F = \{A \rightarrow BC,$

$CD \rightarrow E,$

$B \rightarrow D,$

$E \rightarrow A\}$

Sol: The candidate keys of this relation are A, BC, CD and E.

The dependencies form the canonical cover.

So the decomposition of R1(A, B, C), R2(C, D, E), R3(B, D) and R4(E, A) is a loss-less, dependency preserving decomposition into 3NF.

Properties of Decomposition

Q. 5 Suppose that we decompose the schema R = (A, B, C, D, E) into

(A, B, C)

(A, D, E)

Show that this decomposition is not a dependency preserving decomposition if the following set F of functional dependencies holds :

$A \rightarrow BC$

$CD \rightarrow E$

$B \rightarrow D$

$E \rightarrow A$

Sol: A decomposition {R1, R2} is dependency preserving decomposition iff $(F_{R1} \cup F_{R2})^+ = F^+$.

Properties of Decomposition

Q. 6 Suppose that we decompose the schema R = (A, B, C, D, E) into

(A, B, C)

(C, D, E)

Show that this decomposition is not a dependency preserving decomposition if the following set F of functional dependencies holds :

$A \rightarrow BC$

$CD \rightarrow E$

$B \rightarrow D$

$E \rightarrow A$

Sol: A decomposition {R1, R2} is dependency preserving decomposition iff $(F_{R1} \cup F_{R2})^+ = F^+$.

Properties of Decomposition

Q. 7 Suppose that we decompose the schema R = (A, B, C, D, E) into

(A, B), (B, C) and (C, D)

Show that this decomposition is a dependency preserving decomposition if the following set F of functional dependencies holds :

F = { $A \rightarrow B$

$B \rightarrow C$

$A \rightarrow C$

$C \rightarrow D$ }

Sol: A decomposition {R1, R2, R3} is dependency preserving decomposition iff $(F_{R1} \cup F_{R2} \cup F_{R3})^+ = F^+$.

Properties of Decomposition

Q. 8 Consider R = (A, B, C, D, E) with

F = { $AD \rightarrow C$,

$CD \rightarrow A$,

$B \rightarrow D$,

$D \rightarrow BE$ }.

Determine whether {ABC, BCD, DE} is loss-less. Find all keys for R.

Algorithm 15.3. Testing for Nonadditive Join Property

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

Note: Explanatory comments are given at the end of some of the steps. They follow the format: (**comment**).

1. Create an initial matrix S with one row i for each relation R_i in D, and one column j for each attribute A_j in R.
2. Set $S(i, j) := b_{ij}$ for all matrix entries. (*Each b_{ij} is a distinct symbol associated with indices (i, j)*)
3. For each row i representing relation schema R_i
   {for each column j representing attribute A_j
   {if (relation R_i includes attribute A_j) then set $S(i, j) := a_j$;};}; (*Each a_j is a distinct symbol associated with index (j)*)
4. Repeat the following loop until a *complete loop execution* results in no changes to S
   {for each functional dependency $X \longrightarrow Y$ in F
   {for all rows in S *that have 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 that *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 appears 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 nonadditive join property; otherwise, it does not.

GATE Questions

Consider the schema R(ABCD) and FDs $A \rightarrow B$ and $C \rightarrow D$. Then the decomposition of R into R1(AB) and R2(CD) is

(A) Dependencies preserving and lossless join.

(B) Lossless join but not dependency preserving.

(C) Dependency preserving but not lossless join.

(D) Not dependencies preserving and not lossless join.

[GATE 2001]

GATE Questions

Which one of the following statements about normal forms is FALSE?

(A) BCNF is stricter than 3NF

(B) Lossless, dependency-preserving decomposition into 3NF is always possible

(C) Lossless, dependency-preserving decomposition into BCNF is always possible

(D) Any relation with two attributes is in BCNF

[**GATE 2005]**

GATE Questions

Relation R is decomposed using a set of functional dependencies, F and relation S is decomposed using another set of functional dependencies G. One decomposition is definitely BCNF, the other is definitely 3NF, but it is not known which is which. To make a guaranteed identification, which one of the following tests should be used on the decompositions? (Assume that the closures of F and G are available).

(A) Dependency-preservation
(B) Lossless-join
(C) BCNF definition
(D) 3NF definition

[GATE 2002]

GATE Questions

Let the set of functional dependencies F = {QR → S, R → P, S → Q} hold on a relation schema X = (PQRS). X is not in BCNF. Suppose X is decomposed into two schemas Y and Z, where Y = (PR) and Z = (QRS).

Consider the two statements given below.

I. Both Y and Z are in BCNF

II. Decomposition of X into Y and Z is dependency preserving and lossless

Which of the above statements is/are correct?

(A) Both I and II (B) I only (C) II only (D) Neither I nor II

[GATE 2019]

For Video lecture on this topic please subscribe to my youtube channel.

The link for my youtube channel is

https://www.youtube.com/channel/UCRWGtE76JlTp1iim6aOTRuw?sub_confirmation=1