



**RV College of
Engineering**

Go, change the world

Unit 3(Lossless Join Decomposition)

Original Content:

Ramez Elmasri and Shamkant B. Navathe

Dr. Poonam Ghuli

Associate Professor, Department of CSE

RV College of Engineering, Bengaluru - 59

Relational Decompositions

- Universal Relation Schema:
 - A relation schema $R = \{A_1, A_2, \dots, A_n\}$ that includes all the attributes of the database.
- Universal relation assumption:
 - Every attribute name is unique
- 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 non-additive) join property - to qualify as a good design.

Decomposition

- Decomposition:
 - The process of decomposing the universal relation schema R into a set of relation schemas $D = \{R_1, R_2, \dots, R_m\}$ that will become the relational database schema by using the functional dependencies.
- Goals of Decomposition
- Eliminate redundancy by decomposing a relation into several relations in a higher normal form.
- It is important to check that a decomposition does not lead to bad design

Problems with Decomposition

- There are three potential problems to consider:

- 1) **Lossiness: impossible to reconstruct the original relation!**

Given instances of the decomposed relations, we may not be able to reconstruct the corresponding instance of the original relation – information loss

- 2) Dependency checking may require joins. To Preserve dependency.

- 3) Some queries become more expensive.

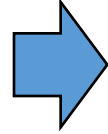
Tradeoff: Must consider these issues vs. redundancy.

Additional Properties

- 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 schema R_i in the decomposition so that no attributes are “lost”.
 - $R_1 \cup R_2 \cup \dots \cup R_n = R$
- Dependency preservation property:
- Lossless (non-additive) join property:

Spurious Tuples Generated

A	B	C
1	2	3
4	5	6
7	2	8



A	B
1	2
4	5
7	2

B	C
2	3
5	6
2	8

$A \rightarrow B; C \rightarrow B$

A	B
1	2
4	5
7	2

|||

B	C
2	3
5	6
2	8

=

A	B	C
1	2	3
4	5	6
7	2	8
1	2	8
7	2	3

Additional Properties

- Decomposition of R into X and Y is lossless-join w.r.t. a set of FDs F if, for every instance r that satisfies F:

$$\pi_X(r) \parallel \pi_Y(r) = r$$

- It is always true that $r \subseteq \pi_X(r) \parallel \pi_Y(r)$
- Definition extended to decomposition into 3 or more relations in a straightforward way.
- *It is essential that all decompositions used to deal with redundancy be lossless!*

Additional Properties

- **Lossless (Non-additive) Join Property of a Decomposition:**

- Definition: Lossless join property: a decomposition $D = \{R_1, R_2, \dots, R_m\}$ of R has the **lossless (nonadditive) 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 natural join of all the relations in D :

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

- Note: The word loss in lossless refers to loss of information, not to loss of tuples. In fact, for “loss of information” a better term is “**addition of spurious information**”
- The decomposition of R into X and Y is **lossless with respect to F** if *and only if* F^+ contains:

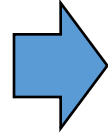
$$X \cap Y \rightarrow X, \text{ or}$$

$$X \cap Y \rightarrow Y$$

\therefore decomposing ABC into AB and BC is lossy, because intersection (i.e., “ B ”) is not a key of either resulting relation.

Additional Properties

A	B	C
1	2	3
4	5	6
7	2	8

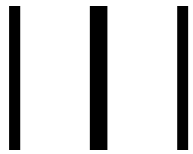


A	C
1	3
4	6
7	8

B	C
2	3
5	6
2	8

$A \rightarrow B; C \rightarrow B$

A	C
1	3
4	6
7	8



B	C
2	3
5	6
2	8

=

A	B	C
1	2	3
4	5	6
7	2	8

But, now we can't check $A \rightarrow B$ without doing a join!

Additional Properties

- **Dependency Preservation Property:**

- A decomposition $D = \{R_1, R_2, \dots, R_m\}$ of R is **dependency-preserving** with respect to F if the union of the projections of F on each R_i in D is equivalent to F ; that is

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

- Projection of set of FDs F : If R is decomposed into X and Y the projection of F on X (denoted F_x) is the set of FDs $U \rightarrow V$ in F^+ (closure of F , not just F) such that all of the attributes U, V are in X . (same holds for Y of course)

Additional Properties

- Decomposition of R into X and Y is dependency preserving if $(F_X \cup F_Y)^+ = F^+$
 - i.e., if we consider only dependencies in the closure F^+ that can be checked in X without considering Y, and in Y without considering X, these imply all dependencies in F^+ .
- Important to consider F^+ in this definition:
 - ABC, $A \rightarrow B$, $B \rightarrow C$, $C \rightarrow A$, decomposed into AB and BC.
 - Is this dependency preserving? Is $C \rightarrow A$ preserved????
 - note: F^+ contains $F \cup \{A \rightarrow C, B \rightarrow A, C \rightarrow B\}$, so...
- F_{AB} contains $A \rightarrow B$ and $B \rightarrow A$; F_{BC} contains $B \rightarrow C$ and $C \rightarrow B$
- So, $(F_{AB} \cup F_{BC})^+$ contains $C \rightarrow A$

Testing for Lossless Join Property

- **Lossless (Non-additive) Join Property of a Decomposition (cont.):**
 - **Algorithm : Testing for Lossless Join Property**
 - **Input:** A universal relation R , a decomposition $D = \{R_1, R_2, \dots, R_m\}$ 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 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) *)

Testing for Lossless Join Property

- **Lossless (Non-additive) Join Property of a Decomposition (cont.):**

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

{for each functional dependency $X \rightarrow Y$ in F

{for all rows in S *which 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 Y, set the other rows to that *same* “a” symbol in the column Y of other rows.

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

};

};

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

Thank YOU