RV College of Engineering

Go, change the world

4^{th} and 5^{th} Normal Forms

Presentation Contents

1. Properties of Lossless Join decomposition
2. Multivalued Dependencies and Fourth Normal Form
3. Join Dependencies and Fifth Normal Form

1. Properties of Relational Decompositions (1)

- Relation Decomposition and Insufficiency of Normal Forms:
  - Universal Relation Schema:
    - A relation schema R = {A1, A2, …, An} that includes all the attributes of the database.
  - Universal relation assumption:
    - Every attribute name is unique.

Properties of Relational Decompositions (2)

- **Relation Decomposition and Insufficiency of Normal Forms (cont.):**
  - Decomposition:
    - The process of decomposing the universal relation schema R into a set of relation schemas D = {R1,R2, …, Rm} that will become the relational database schema by using the functional dependencies.
  - Attribute preservation condition:
    - Each attribute in R will appear in at least one relation schema Ri in the decomposition so that no attributes are "lost".

Properties of Relational Decompositions (2)

- Another goal of decomposition is to have each individual relation Ri in the decomposition D be in BCNF or 3NF.

- Additional properties of decomposition are needed to prevent from generating spurious tuples

Properties of Relational Decompositions (3)

- Dependency Preservation Property of a Decomposition:

  - Definition: Given a set of dependencies F on R, the **projection** of F on R_i, denoted by $p_{Ri}(F)$ where R_i is a subset of R, is the set of dependencies X → Y in F^+ such that the attributes in X ∪ Y are all contained in R_i.

  - Hence, the projection of F on each relation schema R_i in the decomposition D is the set of functional dependencies in F^+, the closure of F, such that all their left- and right-hand-side attributes are in R_i.

Properties of Relational Decompositions (4)

- Dependency Preservation Property of a Decomposition (cont.):
  - Dependency Preservation Property:
    - A decomposition D = {R1, R2, ..., Rm} of R is **dependency-preserving** with respect to F if the union of the projections of F on each Ri in D is equivalent to F; that is
      $((\pi_{R1}(F)) \cup \ldots \cup (\pi_{Rm}(F)))^+ = F^+$

Claim 1:

- It is always possible to find a dependency-preserving decomposition D with respect to F such that each relation Ri in D is in 3nf.

Properties of Relational Decompositions (5)

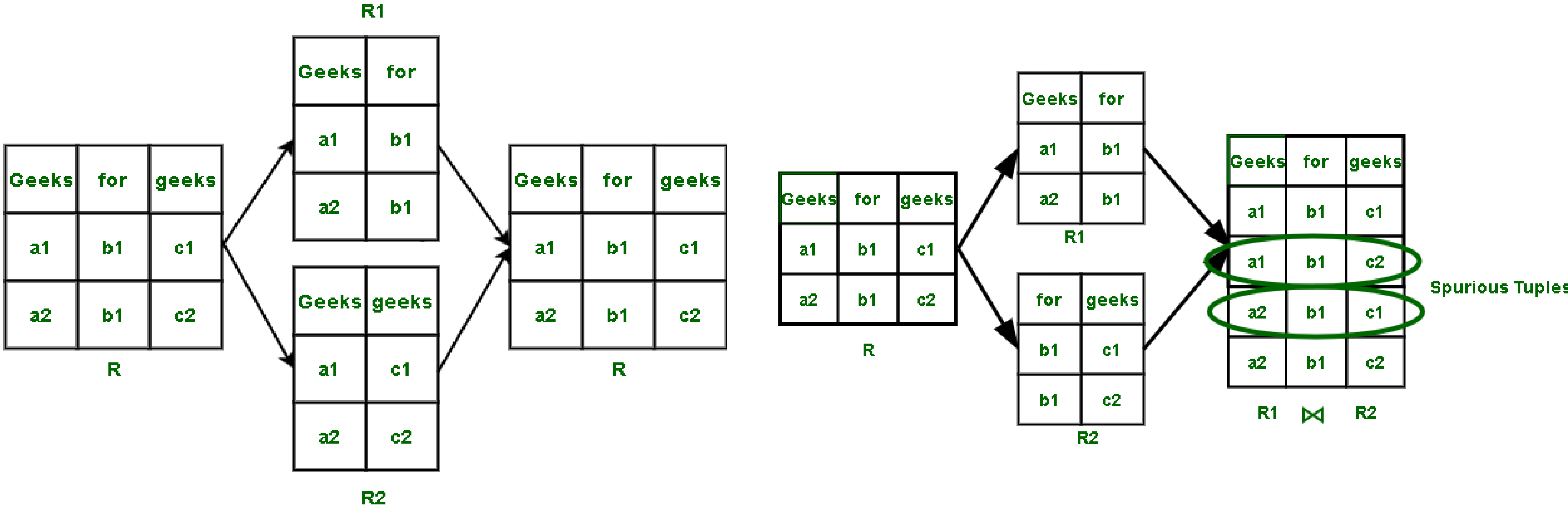
- **Lossless (Non-additive) Join Property of a Decomposition:**

  - Definition: Lossless join property: a decomposition D = {R1, R2, ..., Rm} 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:

    - $(\pi_{R1}(r), ..., \pi_{Rm}(r)) = r$

  - 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**"

Spurious Tuples



Properties of Relational Decompositions (7)

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

  Input: A universal relation R, a decomposition $DECOMP = \{R_1, R_2, \ldots, R_m\}$ 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 $X \rightarrow Y$ 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.

Properties of Relational Decompositions (8)

Lossless (nonadditive) join test for *n*-ary decompositions.
(a) Case 1: Decomposition of EMP_PROJ into EMP_PROJ1 and EMP_LOCS fails test.
(b) A decomposition of EMP_PROJ that has the lossless join property.

(a)

R={SSN, ENAME, PNUMBER, PNAME, PLOCATION, HOURS} $D=\{R_1, R_2\}$
R_1=EMP_LOCS={ENAME, PLOCATION}
R_2=EMP_PROJ1={SSN, PNUMBER, HOURS, PNAME, PLOCATION}

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

	SSN	ENAME	PNUMBER	PNAME	PLOCATION	HOURS
R_1	b_{11}	a_2	b_{13}	b_{14}	a_5	b_{16}
R_2	a_1	b_{22}	a_3	a_4	a_5	a_6

(no changes to matrix after applying functional dependencies)

(b)

EMP

SSN	ENAME

PROJECT

PNUMBER	PNAME	PLOCATION

WORKS_ON

SSN	PNUMBER	HOURS

Properties of Relational Decompositions (8)

Lossless (nonadditive) join test for n-ary decompositions. (c) Case 2: Decomposition of EMP_PROJ into EMP, PROJECT, and WORKS_ON satisfies test.

(c)

R={SSN, ENAME, PNUMBER, PNAME, PLOCATION, HOURS} $D=\{R_1, R_2, R_3\}$

R_1=EMP={SSN, ENAME}

R_2=PROJ={PNUMBER, PNAME, PLOCATION}

R_3=WORKS_ON={SSN, PNUMBER, HOURS}

F={SSN→{ENAME;PNUMBER→{PNAME, PLOCATION} ;{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

(original matrix S at start of algorithm)

	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	$\cancel{b_{32}}$ a_2	a_3	$\cancel{b_{34}}$ a_4	$\cancel{b_{35}}$ a_5	a_6

(matrix S after applying the first two functional dependencies - last row is all "a" symbols, so we stop)

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

$$F = \{A \rightarrow C,\ B \rightarrow C,\ C \rightarrow D,\ DE \rightarrow C,\ CE \rightarrow A\}$$

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

$$R_1 = \{A, D\}$$

$$R_2 = \{A, B\}$$

$$R_3 = \{B, E\}$$

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

$$R_5 = \{A, E\}$$

	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 FDs in F (first loop):

* Try $A \longrightarrow C$:

	A	B	C	D	E
R_1	a_1	b_{12}	b_{13}	a_4	b_{15}
R_2	a_1	a_2	$\cancel{b_{23}}\ b_{13}$	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}	$\cancel{b_{53}}\ b_{13}$	b_{54}	a_5

* Try $B \longrightarrow C$:

	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	$\cancel{b_{33}}\ b_{13}$	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 $C \longrightarrow D$:

	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}	$\cancel{b_{24}}\ a_4$	b_{25}
R_3	b_{31}	a_2	b_{13}	$\cancel{b_{34}}\ a_4$	a_5
R_4	b_{41}	b_{42}	a_3	a_4	a_5
R_5	a_1	b_{52}	b_{13}	$\cancel{b_{54}}\ a_4$	a_5

* Try $DE \longrightarrow C$:

	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	$\cancel{b_{13}}\ a_3$	a_4	a_5
R_4	b_{41}	b_{42}	a_3	a_4	a_5
R_5	a_1	b_{52}	$\cancel{b_{13}}\ a_3$	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	$\cancel{b_{31}}$ a_1	a_2	a_3	a_4	a_5
R_4	$\cancel{b_{41}}$ 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.

3. Multivalued Dependencies and Fourth Normal Form (1)

(a) The EMP relation with two MVDs: ENAME —>> PNAME and ENAME —>> DNAME.
(b) Decomposing the EMP relation into two 4NF relations EMP_PROJECTS and EMP_DEPENDENTS.

(a) **EMP**

ENAME	PNAME	DNAME
Smith	X	John
Smith	Y	Anna
Smith	X	Anna
Smith	Y	John

(b) **EMP_PROJECTS**

ENAME	PNAME
Smith	X
Smith	Y

EMP_DEPENDENTS

ENAME	DNAME
Smith	John
Smith	Anna

Multivalued Dependencies and Fourth Normal Form

Definition:

- A **multivalued dependency** (**MVD**) $X \longrightarrow\!\!> Y$ specified on relation schema R, where X and Y are both subsets of R, specifies the following constraint on any relation state r of R: If two tuples t_1 and t_2 exist in r such that $t_1[X] = t_2[X]$, then two tuples t_3 and t_4 should also exist in r with the following properties, where we use Z to denote (R 2 ($X \cup Y$)):
  - $t_3[X] = t_4[X] = t_1[X] = t_2[X]$.
  - $t_3[Y] = t_1[Y]$ and $t_4[Y] = t_2[Y]$.
  - $t_3[Z] = t_2[Z]$ and $t_4[Z] = t_1[Z]$.
- An MVD $X \longrightarrow\!\!> Y$ in R is called a **trivial MVD** if (a) Y is a subset of X, or (b) $X \cup Y = R$.

4^{th} Normal Form

The database must satisfy the following two things

- The database must meet all the requirement of 3NF, BCNF
- There should be no more than one multi-valued dependencies.
- According to the 4^{th} normal form, a record type should not contain two or more independent multi-valued facts about an entity.

STUDENT	HOBBY	LANGUAGE
1	Cricket	English
1	Poetry	Urdu
2	Cricket	Hindi
2	Poetry	English

The table is Not in 4th Normal Form

Table: The Student table with more than one multi-valued dependency. (For example, one student can have many hobbies and similarly one student can have many languages.)
Instead, they should be represented in the two records

STUDENT	LANGUAGE
1	English
1	Urdu
2	Hindi

STUDENT	HOBBY
1	Cricket
1	Poetry
2	Singing

4. Join Dependencies and Fifth Normal Form

Definition:

- A **join dependency** (**JD**), denoted by JD(R_1, R_2, ..., R_n), specified on relation schema R, specifies a constraint on the states r of R.
  - The constraint states that every legal state r of R should have a non-additive join decomposition into R_1, R_2, ..., R_n; that is, for every such r we have
  - $* (\pi_{R1}(r), \pi_{R2}(r), ..., \pi_{Rn}(r)) = r$

  Note: *an MVD is a special case of a JD where n = 2.*

- A join dependency JD(R_1, R_2, ..., R_n), specified on relation schema R, is a **trivial JD** if one of the relation schemas R_i in JD(R_1, R_2, ..., R_n) is equal to R.

Join Dependencies and Fifth Normal Form (2)

Definition:

- A relation schema R is in **fifth normal form** (**5NF**) (or **Project-Join Normal Form** (**PJNF**)) with respect to a set F of functional, multivalued, and join dependencies if,
  - for every nontrivial join dependency JD(R_1, R_2, ..., R_n) in F^+ (that is, implied by F),
    - every R_i is a superkey of R.

Join Dependency

If a table can be recreated by joining multiple tables and each of this table have a subset of the attributes of the table, then the table is in Join Dependency. It is a generalization of Multivalued Dependency

Join Dependency can be related to 5NF, wherein a relation is in 5NF, only if it is already in 4NF and it cannot be decomposed further.

<Employee>

EmpName	**EmpSkills**	**EmpJob** (Assigned Work)
Tom	Networking	EJ001
Harry	Web Development	EJ002
Katie	Programming	EJ002

<EmployeeSkills>

EmpName	**EmpSkills**
Tom	Networking
Harry	Web Development
Katie	Programming

<EmployeeJob>

EmpName	**EmpJob**
Tom	EJ001
Harry	EJ002
Katie	EJ002

<JobSkills>

EmpSkills	**EmpJob**
Networking	EJ001
Web Development	EJ002
Programming	EJ002

Join Dependency {(EmpName, EmpSkills), (EmpName, EmpJob), (EmpSkills, EmpJob)}

3. Multivalued Dependencies and Fourth Normal Form (1)

(c) The relation SUPPLY with no MVDs is in 4NF but not in 5NF if it has the JD(R1, R2, R3). (d) Decomposing the relation SUPPLY into the 5NF relations R1, R2, and R3.

(c) **SUPPLY**

SNAME	PARTNAME	PROJNAME
Smith	Bolt	ProjX
Smith	Nut	ProjY
Adamsky	Bolt	ProjY
Walton	Nut	ProjZ
Adamsky	Nail	ProjX
Adamsky	Bolt	ProjX
Smith	Bolt	ProjY

(d) **R1**

SNAME	PARTNAME
Smith	Bolt
Smith	Nut
Adamsky	Bolt
Walton	Nut
Adamsky	Nail

R2

SNAME	PROJNAME
Smith	ProjX
Smith	ProjY
Adamsky	ProjY
Walton	ProjZ
Adamsky	ProjX

R3

PARTNAME	PROJNAME
Bolt	ProjX
Nut	ProjY
Bolt	ProjY
Nut	ProjZ
Nail	ProjX

Joint dependency

- Join decomposition is a further generalization of Multivalued dependencies.

- If the join of R1 and R2 over C is equal to relation R then we can say that a join dependency (JD) exists, where R1 and R2 are the decomposition R1(A, B, C) and R2(C, D) of a given relations R (A, B, C, D).

- Alternatively, R1 and R2 are a lossless decomposition of R. A JD ⋈ {R1, R2, ..., Rn} is said to hold over a relation R if R1, R2, ….., Rn is a lossless-join decomposition.

- The *(A, B, C, D), (C, D) will be a JD of R if the join of join's attribute is equal to the relation R. Here, *(R1, R2, R3) is used to indicate that relation R1, R2, R3 and so on are a JD of R.

- Let R is a relation schema R1, R2, R3……..Rn be the decomposition of R. r(R) is said to satisfy join dependency if and only if

$$\bowtie_{i=1}^{n} \Pi_{R_i}(r) = r.$$

Example – Consider the above schema, with a case as "if a company makes a product and an agent is an agent for that company, then he always sells that product for the company". Under these circumstances, the ACP table is shown as:

Table – ACP

AGENT	COMPANY	PRODUCT
A1	PQR	Nut
A1	PQR	Bolt
A1	XYZ	Nut
A1	XYZ	Bolt
A2	PQR	Nut

Table – R1

AGENT	COMPANY
A1	PQR
A1	XYZ
A2	PQR

Table – R2

AGENT	PRODUCT
A1	Nut
A1	Bolt
A2	Nut

Table – R3

COMPANY	PRODUCT
PQR	Nut
PQR	Bolt
XYZ	Nut
XYZ	Bolt

Result of Natural Join of R1 and R3 over 'Company' and then Natural Join of R13 and R2 over 'Agent'and 'Product' will be table **ACP**.

Hence, in this example, all the redundancies are eliminated, and the decomposition of ACP is a lossless join decomposition. Therefore, the relation is in 5NF as it does not violate the property of lossless join.