Lossless Join Decomposition Tutorial

SWEN304/SWEN439

Lecturer: Dr Hui Ma

Engineering and Computer Science

TE WHARE WĀNANGA O TE ŪPOKO O TE IKA A MĀUI
VICTORIA
UNIVERSITY OF WELLINGTON

Outline

- Lossless join decomposition
- Exercises
  - 3NF decomposition
  - BCNF decomposition

FDs and a Relation Schema Key

- Each relation schema **key** is the consequence of a functional dependency from F^+
- Let $R(A_1, \ldots, A_n)$ be a relation schema and F the set of functional dependencies in R
- Set of attributes $X \subseteq R$ is a relation schema **key** if

1° $X \to R \in F^+$ (or $X^+ = R$)

2° $(\forall Y \subset X)(Y \to R \notin F^+)$

- Not null condition still applies to X
- A **prime** attribute is a relation schema attribute that belongs to any of the keys
- Primary key is one of the keys

Lossless Join Decomposition

- A decomposition D = {$R_1, R_2, \ldots, R_m$} of a relation R has the **lossless (nonadditive) join** property wrt the set of dependencies F on R if, for every relation $r(R)$ that satisfies F,

$$* \, (\pi_{R1} r(R), \ldots, \pi_{Rm} r(R)) = r(R)$$

  where * is the natural join of all the relations in D.

- It is proven in the theory of the relational data model that the decomposition of a relation schema R onto R_1 and R_2 is *lossless (non-additive)* if the intersection $R_1 \cap R_2$ contains a **key** of R_1 or a key of R_2

Example 1: Checking Losslessness of D (1)

- Given a set of relation schemas:

$D = \{N_1(\{A, B\}, \{A\}), N_2(\{B, D\}, \{B\}), N_3(\{C, B\}, \{C\})\}$

- How to check whether the whole set of relation schemas represents a lossless join decomposition of the (supposed) universal relation schema?

Example 1: Checking Losslessness of D (2)

- A naïve and generally **wrong** approach:
  - Perform a pair wise checking of the relation schemas
    - for each relation schema you find another one such that the intersection of the two schemas is a schema key of one of the schema

- So, according to that approach:
  - $\{A, B\} \cap \{B, D\} = \{B\}$, and B is the key of N_2
  - $\{B, C\} \cap \{B, D\} = \{B\}$, and B is the key of N_2
  - Conclusion (a wrong one): The set of relation schemas D is a lossless join decomposition (of a universal relation schema)

Example 1: Checking Losslessness of D (2)

$D = \{N_1(\{A, B\}, \{A\}), N_2(\{B, D\}, \{B\}), N_3(\{C, B\}, \{C\})\}$

- A correct approach is to apply this checking iteratively until all the schemas are considered
  - $\{A, B\} \cap \{B, D\} = \{B\}$, and B is the key of $N_{2,}$
  - construct new relation schema N_{12} $(R_{12,}$ Key$(N_{12}))$, with $R_{12} = \{A, B\} \cup \{B, D\} = \{A, B, D\}$ and Key$(N_{12}) = \{A\}$
  - $\{A, B, D\} \cap \{B, C\} = \{B\}$, and check again.
  - B is neither a key of N_{12} nor a key of N_3
- We can conclude the set of relation schemas D is a not a lossless join decomposition (of a universal relation schema).

Example 1: Checking Losslessness of D (3)

Universal Relation Schema

A	*B*	*C*	*D*
1	*0*	*1*	*0*
2	*0*	*2*	*0*

projection

π_{AB} π_{BD} π_{BC}

N_1

<u>*A*</u>	*B*
1	*0*
2	*0*

N_2

<u>*B*</u>	*D*
0	*0*

N_3

B	<u>*C*</u>
0	*1*
0	*2*

Example 1: Checking Losslessness of D (4)

The pair wise approach is wrong since

- It can not ensure the whole decomposition is lossless

N_1

A	B
1	0
2	0

N_2

B	D
0	0

Natural Join $r(N_1)*r(N_2)$

A	B	D
1	0	0
2	0	0

N_3

B	C
0	1
0	2

$(r(N_1)*r(N_2))*r(N_3)$

A	B	C	D
1	0	1	0
1	0	2	0
2	0	1	0
2	0	2	0

One Approach of Checking Losslessness of D

- To check whether a set D of relation schemas is a lossless decomposition is:
  1. Construct a relation schema (U, F), where

$$U = \bigcup_{i=1}^{n} R_i \quad \text{and} \quad F = \bigcup_{i=1}^{n} F_i$$

  2. Find all keys $\{X_i \mid i = 1,\ldots, m\}$ of the constructed "universal" relation schema (U, F)
  3. If there is a relation schema $N(R, K)$ in D that contains a key of the constructed relation schema (U, F), then D is a lossless join decomposition
  4. Otherwise, add a new relation schema that contains only a key X_i of the constructed "universal" relation schema (U, F) to D

$$D = D \cup \{N_X(X_i, X_i)\}$$

Example 2: Checking Losslessness of D

- The universal relation schema key is AC, and decompositions

  $D = \{N_1(\{A, B\}, \{A\}), N_2(\{B, D\}, \{B\}), N_3(\{B, C\}, \{C\})\}$

  Is the decomposition lossless? Why?

- A lossless decomposition

  $D' = \{N_1(\{A, B\}, \{A\}), N_2(\{B, D\}, \{B\}), N_3(\{B, C\}, \{C\}),$
  $N_4(\{A, C\}, \{AC\})\}$

- This can be achieved using the Synthesis Algorithm

Example 2: Checking Losslessness of D (3)

Universal Relation Schema

A	B	C	D
1	0	1	0
2	0	2	0

projection

π_{AB} N_1

A	B
1	0
2	0

π_{BD} N_2

B	D
0	0

π_{BC} N_3

B	C
0	1
0	2

π_{AC} N_4

A	C
1	1
2	2

Example 2: Checking Losslessness of D (4)

N_1

A	B
1	0
2	0

N_2

B	D
0	0

N_3

B	C
0	1
0	2

N_4

A	C
1	1
2	2

Natural Join

$r(N_1)*r(N_2)$

A	B	D
1	0	0
2	0	0

$(r(N_1)*r(N_2))*r(N_3)$

A	B	C	D
1	0	1	0
1	0	2	0
2	0	1	0
2	0	2	0

$((r(N_1)*r(N_2))*r(N_3))*r(N_4)$

A	B	C	D
1	0	1	0
2	0	2	0

Exercise 1: Lossless Join Decomposition

- Consider the following relation schema again

  $Department(\{LecId, LeName, CourId, CoName, DptId\},$
  $\{LecId \rightarrow LeName + CourId, CourId \rightarrow CoName + DptId\})$

- Is Department in 3NF? If not, decompose it into 3NF
- Is Department in BCNF? If not, decompose it into BCNF

Exercise 1: 3NF Decomposition

- Consider the following relation schema again

  $Department(\{LecId, LeName, CourId, CoName, DptId\},$
  $\{LecId \rightarrow LeName + CourId, CourId \rightarrow CoName + DptId\})$

- Compute minimal cover of F

  $G = \{LecId \rightarrow LeName, LecId \rightarrow CourId, CourId \rightarrow CoName,$
  $CourId \rightarrow DptId\})$

1. Group FDs according to LHS and form relation schemas

   $LecId \rightarrow LeName, LecId \rightarrow CourId,$

   $CourId \rightarrow CoName, CourId \rightarrow DptId$

   $Lecturer(\{LecId, LeName, CourId\}, \{LecId\})$
   $Course(\{CourId, CoName, DptId\}, \{CourId\})$

Exercise 1: 3NF Decomposition

2. Compute universal relation keys and check if any of the relation schemas contains one of the keys:
   - *LecId* is the universal relation schema key and is in Lecturer

3. All functional dependencies are preserved

Exercise 1: BCNF Decomposition

- Consider the following relation schema again

  Department ({*LecId*, *LeName*, *CourId*, *CoName*, *DptId*},
  {*LecId* $\rightarrow$ *LeName* + *CourId*, *CourId* $\rightarrow$ *CoName* + *DptId*})

  Is *Department* in BCNF?

- Is decomposed into BCNF using *CourId* $\rightarrow$ *CoName* + *DptId*

  Lecturer ({*LecId*, *LeName*, *CourId*}, {*LecId*})

  Course ({*CourId*, *CoName*, *DptId*}, {*CourId*})

- Both *Lecturer* and *Course* are in BCNF

Exercise 2: Find Keys and Normalization

- Let $R = ABCD$ a relation schema and $F = \{AB \rightarrow C, C \rightarrow D, D \rightarrow A\}$ a set of dependencies for R.

1) Find the candidate keys for R
2) If R is not in BCNF, give a decomposition of R in relations that will be in BCNF

Exercise 2: Find Keys and Normalization

- Let $R = ABCD$ a relation schema and
  $F = \{AB \rightarrow C, C \rightarrow D, D \rightarrow A\}$ a set of dependencies for R.

1) Find the candidate keys for R

$A^+ = A$, $B^+ = B$ $C^+ =$ CDA, $D^+ = DA$,

$AB^+ = ABCD = R = ABC^+ = ABD^+ = ABCD^+$,

$AC^+ = ACD$

$AD^+ = AD$

$BC^+ = BCDA = R = BCD^+$

$BD^+ = BDAC = R$

$CD^+ = CDA$

$ACD^+ = ACD$

AB, *BC*, and *BD* are scheme keys

Exercise 2: Find Keys and Normalization

- Let $R = ABCD$ a relation schema and $F=\{AB \rightarrow C, C \rightarrow D, D \rightarrow A\}$ a set of dependencies for R.

2) If R is not in BCNF, give a decomposition of R in relations that will be in BCNF

R is not in BCNF because there are FDS, $C \rightarrow D$ and $D \rightarrow A$, of which the *LHS* is not a superkey.

Decompose R using $C \rightarrow D$ into $R_1 = ABC$ with $F_1 = \{AB \rightarrow C, C \rightarrow A\}$, $R_2 = CD$ with $F_2 = \{ C \rightarrow D\}$.

R_2 is in BCNF but R_1 is not yet in BCNF because there is FD $C \rightarrow A$, of which the *LHS* is not a super key.

Decompose R_1 along $C \rightarrow A$, *R_1 is decomposed into $R_{11} = BC$ with $F_{11} = \{\}$, and $R_{12} = CA$* with *$F_{12} = \{C \rightarrow A\}$. Both R_{11} and R_{12} are in BCNF*

Exercise 2: Find Keys and Normalization

- Let $R = ABCD$ a relation schema and
  $F = \{AB \rightarrow C,\ C \rightarrow D,\ D \rightarrow A\}$ a set of dependencies for R.

2) If R is not in BCNF, give a decomposition of R in relations that will be in BCNF

R is decomposed into $R_{11}(\{B,C\},\{B+C\})$, $R_{12}(\{C,A\},\{C\})$, $R_{11}(\{D,A\},\{D\})$

$F' = R_{11} \cup R_{12} \cup R_2 = \{C \rightarrow A,\ C \rightarrow D\}$

Functional dependencies $AB \rightarrow C$, $D \rightarrow A$ are lost during the decomposition because based on F'

$D^+ = D$ *and* $A \notin D^+$

$AB^+ = AB$ *and* $C \notin AB^+$

Exercise 3: Find Keys and Normalization

- Let $R = JKL$ a relation and $F = \{JK \rightarrow L, L \rightarrow K\}$ a set of dependencies for R.

- Find two canditate keys in R
- Is R in 3NF? Justify your answer
- If R is not in BCNF, decompose R into BCNF
- Are the functional dependencies preserved during the decomposition?

Exercise 3: Find Keys and Normalization

- Let $R = JKL$ a relation and $F = \{JK \rightarrow L, L \rightarrow K\}$ a set of dependencies for R.

Exercise 3: Find Keys and Normalization

Let R=JKL a relation and F = {JK → L, L → K} a set of dependencies for R.

- Find two candidate keys in R

JK and JL are the keys, since $JK^+ = JKL = R$, and $JL^+ = JLK = R$

- Is R in 3NF? Justify your answer

Yes, it is, since all FDs in F either their *LHS* is a superkey or *RHS* is a prime attribute.

Exercise 3: Find Keys and Normalization

- Let $R=JKL$ a relation and $F = \{JK \rightarrow L,\ L \rightarrow K\}$ a set of dependencies for R.

- If R is not in BCNF, decompose R into BCNF

R is not in BCNF because there is a FD $L \rightarrow K$ of which the LHS is not a super key

Using $L \rightarrow K$, R is decomposed into

$R_1=JL$ with $F_1 = \{\}$ and $R_2=LK$ with $F_2= \{L \rightarrow K\}$

Both R_1 and R_2 is in BCNF since for each of them all FDs having LHS as a super key of R_i

Hence, R is decomposed into $R_1(\{J,L\}, \{J+l\})$, $R_2(\{L,K\}, \{L\})$

- Not all the functional dependencies preserved during the decomposition.

$JK \rightarrow L$ is lost since using $F_1 \cup F_2$, $JK^+ = JK$ and $L \notin JK^+$