BCNF Decomposition

R&G Chapter 19

Announcement

- **Assignment 5 (last assignment) available in Canvas.**
  - Due at 11:59pm on Dec 5.
  - Similar online format as the final exam.

In the Last Lecture

- Normal forms:
  - 1st NF, 2nd NF, 3rd NF, and BCNF (3.5NF)
- If a relation is not in a desired normal form, we need to decompose the relation to eliminate data redundancy.
- Decompositions should be used only when needed, as it can cause potential problems.

Summary of Normal Forms

Normal Form	Constraints
1st normal form (1NF)	Relational database
2nd normal form (2NF)	No partial dependency
3rd normal form (3NF)	No partial dependency No transitive dependency
Boyce-codd normal form (BCNF)	Nothing but superkeys (i.e., for all FDs X->Y, X must be a superkey)

Two short-cut rules:

- **Shortcut Rule 1**: If all candidate keys are singleton keys (i.e., only contain one attribute), then R must satisfy 2NF
- **Shortcut Rule 2**: If all attributes are part of some candidate keys, then R must be 2NF & 3NF

Warm-up Exercise

- **Consider a relation R with five attributes ABCDE. You are given the following dependencies: A $\rightarrow$ B, BC $\rightarrow$ E, and ED $\rightarrow$ A**
  - List all keys for R
  - Is R in 3NF?
  - Is R in BCNF?

Problems with Decompositions

- There are two potential problems of schema decomposition
  1) May be impossible to reconstruct the original relation! (Lossiness)
  2) Dependency checking may require joins.

Features of a Good Decomposition

- **A good decomposition should satisfy two conditions:**
  - Lossless
  - Dependency preserving

BCNF and 3NF Decomposition

- Some rules to remember before we discuss the details of decomposition…

	BCNF Decomposition	3NF Decomposition
Data redundancy	NONE	May still have some
Lossless	Guaranteed	Guaranteed
dependency-preserving	Not guaranteed	Guaranteed

Roadmap of Today's Lecture

- Lossy and lossless decompositions
- Solutions to lossless decompositions
- BCNF decomposition algorithm (guaranteed to be lossless)

- Quiz (15 minutes)

Example of Lossy Decomposition

Lossy decomposition: Join result of the tables after decomposition is NOT the same as the original dataset

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

R

A	B
1	2
4	5
7	2

R1

B	C
2	3
5	6
2	8

R2

A	B
1	2
4	5
7	2

R1

Join

B	C
2	3
5	6
2	8

R2

=

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

R′

R<>R′

Example of Lossless Decomposition

Lossless decomposition: Join result of the tables after decomposition is the same as the original dataset

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

R

A	C
1	3
4	6
7	8

R1

B	C
2	3
5	6
2	8

R2

A	C
1	3
4	6
7	8

R1

Join

B	C
2	3
5	6
2	8

R2

=

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

R′

R=R′

Roadmap of Today's Lecture

- Lossy and lossless decompositions
- Solutions to lossless decompositions
- BCNF decomposition algorithm (guaranteed to be lossless)

- Quiz (15 minutes)

Task #1

- **How to decompose the original relation so that it is lossless?**

Solution to Lossless Decomposition

- Given a table R and a set of FDs F of R
- The decomposition of R into X and Y is lossless with respect to F *if and only if* the $\mathbf{F^+}$ satisfies that:

$$X \cap Y \text{ -> } X \quad \textbf{or} \quad X \cap Y \text{ -> } Y$$

  i.e., the common attributes of X and Y (that X and Y natural joins on) is a superkey of either X or Y.

- If W -> Z holds over R and W ∩ Z is empty, then
  - Decompose R into two tables R1=**R-Z** and R2=**WZ**
  - The decomposition (R-Z, WZ) are guaranteed to be lossless (as R-Z and WZ joins at W, and W->Z)

Revisit Lossy Decomposition Example

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 **->** B; C **->** B

X={A, B}, Y={ B, C}, X ∩ Y = {B}, B $\not\to$ {A, B} and B $\not\to$ {B, C}

Lossy decomposition!

A	B
1	2
4	5
7	2

Join

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

Revisit Lossless Example

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

$\Rightarrow$

A	C
1	3
4	6
7	8

B	C
2	3
5	6
2	8

A -> B; C -> B

X={A, C}, Y={ B, C}, X ∩ Y = {C}, C->{B, C}

Lossless decomposition!

A	C
1	3
4	6
7	8

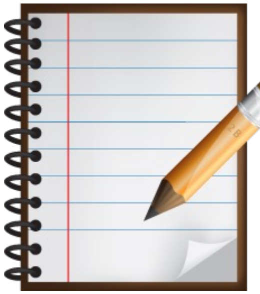
Join

B	C
2	3
5	6
2	8

=

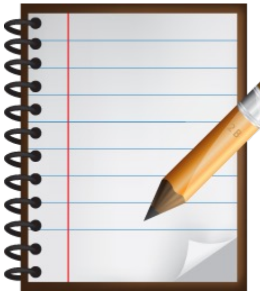
A	B	C
1	2	3
4	5	6
7	2	8

Lossless Decomposition Exercise 1



- **Relational table R=ABCDE**
- **FDs F={AB->C, C->E, B->D, E->A}**
- **R is decomposed into R1=BCD and R2=ACE**
- **Is (R1, R2) a lossless decomposition?**

Lossless Decomposition Exercise



- **Relational table R(ABCDE)**
- **FDs F={AB->C, C->E, B->D, E->A}**
- **R is decomposed into R1(BCD) and R2(ACE)**
- **Is (R1, R2) a lossless decomposition?**
- **Way of thinking:**
  - Step 1: Find common attribute: R1 ∩ R2 = (C);
  - Step 2: Check whether C is a superkey of R1 or R2 (i.e., does C^+ =(BCD) or C^+ = (ACE)?)
    - C^+ = (CEA). So C is a superkey of R2.
    - It is a lossless decomposition.

Roadmap of Today's Lecture

- Lossy and lossless decompositions
- Solutions to lossless decompositions
- BCNF decomposition algorithm (guaranteed to be lossless)
- Quiz (15 minutes)

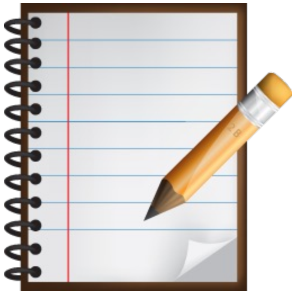
Algorithm of BCNF Decomposition

Consider relation R with FDs F.

- **Step 1**:
  - Ensure all FDs in F contain only single attribute at right-hand side (RHS)
    - For example, if you have AB->CD, spit it into AB->C and AB->D;
- **Step 2**:
  - Identify all FDs F' ⊆ F that hold on R
    - A FD f holds on R if R contains all attributes in f
    - This sub-step is not needed for the first round where R is the original dataset, but required for all the decomposed tables.
  - Check if R satisfies BCNF according to F'.
  - If not, for any X -> Y in F' that violates BCNF (i.e., X is not a superkey of R), decompose R into R1= R - Y and R2= XY.

  Repeat Step 2 on R1 and R2, until all the decomposed tables satisfy BCNF.

- **BCNF decomposition is guaranteed to be lossless!**



Decomposition into BCNF

Consider the relation R={CSJDPQV}:

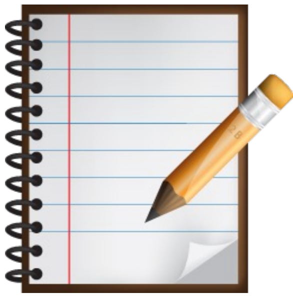
– Its primary key is C;

– It has the following FDs: JP -> C, SD -> P, J -> S.

Question:

- **(1) Does R satisfy BCNF?**
- **(2) If not, decompose R into BCNF tables.**

Decomposition into BCNF



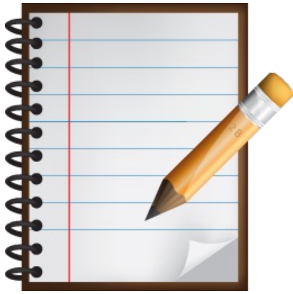
Consider the relation R={CSJDPQV}:
- Its primary key is C;
- It has the following FDs: JP -> C, SD -> P, J -> S.

Question:

- **(1) Does R satisfy BCNF?**
  - FD JP->C: JP^{+}={CSJDPQV}, JP is a superkey. This FD does not violate BCNF.
  - FD SD->P: SD^{+}={SDP}. SD is not a superkey. This FD violates BCNF.
  - FD J->S: J^{+}={JS}. J is not a superkey. This FD violates BCNF.

So R does not satisfy BCNF.

Decomposition into BCNF



Consider the relation R={CSJDPQV}:
- Its primary key is C;
- It has the following FDs: JP -> C, SD -> P, J -> S.

- **Question:**
- **(1) Does R satisfy BCNF?**
- **(2) If not, decompose R into BCNF tables.**
  - Two FD rules violate BCNF: SD->P, J->S on R
  - To deal with SD -> P, decompose R into SDP, CSJDQV.
    - SDP satisfies BCNF (only SD->P holds on SDP)
    - CSJDQV does not satisfy BCNF (J->S violates BCNF)
  - To deal with J -> S, decompose CSJDQV into SJ and CJDQV
    - Both JS and CJDQV satisfy BCNF (SD->P and J->S do not hold on CJDQV)
  - Final decomposition results: *SDP, SJ, and CJDQV*

Decomposition into BCNF

Consider the relation R={CSJDPQV}:

–Its primary key is C;

–It has the following FDs: JP -> C, SD -> P, J -> S.

- **Several FDs may cause violation of BCNF. The order in which we deal with them could lead to very different sets of relations!**

  –We just tried the order of SD -> P, J -> S

  –Now try starting from J -> S, then SD -> P