# BCNF Decomposition

**R&G Chapter 19**

# In 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.

# 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 is**
  - Lossless
  - Dependency preserving

# BCNF and 3NF Decomposition

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

| | **BCNF** | **3NF** |
|---|---|---|
| Data redundancy | NONE | May still have some |
| Lossless-join decomposition | Guaranteed | Guaranteed |
| dependency-preserving decomposition | Not guaranteed | Guaranteed |

# 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 |
|---|---|
| 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 |

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

# Lossless Decomposition (example)

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

# Task #1

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

# Solution to Lossless Decomposition

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

  $X \cap Y \to X$, **or**

  $X \cap Y \to Y$

  In other words, the join attributes of X and Y is the key of either X or Y.
- If W -> Z holds over R and W ∩ Z is empty, then
  - decomposition of R into **R-Z** and **WZ**
  - R-Z and WZ are guaranteed to be loss-less (since R-Z and WZ joins at W)

# 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 $\not\rightarrow${A, B} and B $\not\rightarrow$ {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 |

# Lossless Decomposition (example)

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

# Lossless Decomposition Exercise 1

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

# Lossless Decomposition Exercise 2

- **Table R(A,B,C,D,E)**
- **FDs F=(A->BC, CD->E, B->D, E->A)**
- **R is decomposed into R1(A, B, C) and R2(A, D, E)**
- **Is (R1, R2) a lossless decomposition?**

# Decomposition into BCNF

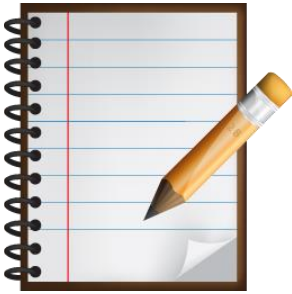
Consider relation R with FDs F.

- **Step 1**:
  - Ensure all FDs in F contain only single attribute on right-hand side (RHS)
    - This is always doable, for example, if you have AB->CD, spit it into AB->C and AB->D;
- **Step 2**:
  - If X -> Y (in F) violates BCNF (i.e., X is not the key of R), decompose R into R - Y and XY (guaranteed to be loss-les).

Repeat Step 1 & 2, until all FDs do not violate BCNF (i.e., the left-hand-side of all FDs are superkeys).

It will give a lossless decomposition that consists of BCNF relations (i.e., data redundancy free).

# Decomposition into BCNF



Consider the relation R={CSJDPQV}:

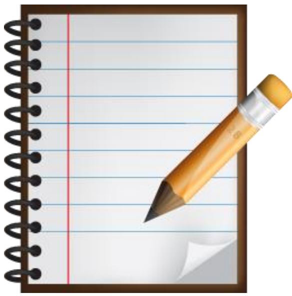
– Its primary key is C;

– 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?**
- **(2) If not, decompose R into BCNF tables.**
  - To deal with SD -> P, decompose into SDP, CSJDQV.
  - To deal with J -> S, decompose CSJDQV into JS and CJDQV
  - So we end up with: SDP, JS, and CJDQV

(note: JP is a candidate key of R, so JP->C does not violate 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.

- **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