# Schema Normalization

**R&G Chapter 19**

# In Last Lecture

- **Normal forms:**
  - 1 NF: No multivalued attributes
  - 2 NF: 1 NF + no partial dependencies
  - 3 NF: 2 NF + no transitive dependencies

# Check Violation of 2NF

- **Given relation R and its FD F, if there exists any FD X->Y in F+ s.t.**

  (1) X is a subset of keys of R, AND

  (2) Y is a non-key attribute (i.e., Y does not appear in any key)

**Then R violates 2NF!**

# Check Violation of 3rd Normal Form

- **Given relation R and its FD F, if there exists any FD X->Y in F+ s.t.**
  1. X is a subset of some key of R (i.e., partial dependency)

  OR

  2. X is not a proper subset of any key (i.e., transitive dependency)

**Then R violates 3NF!**

# 3NF example

- Consider the schema R(A, B, C, D, E)
- Functional dependencies F = {BD->A,  AB->C, C->E}
- Is R in 3NF?
- **Way of thinking**
  – Step 1: find keys of R
    - Key: BD
  – Step 2: find non-key attributes:
    - Non-key attributes: A, C, E

# 3NF example

- Consider the schema R(A, B, C, D, E)
- Functional dependencies F = {BD->A,  AB->C, C->E}
- Is R in 3NF?
- **Way of thinking**
  - Step 3: check violation of partial dependency:
    - Is there any FD x->y s.t. x = B or D alone while y is A, C or E?
  - Step 4: check violation of transitive dependency:
    - Is there any FD x->y s.t. x is not a proper subset of any key and y is A, C, or E?
    - Yes. There is C->E.
  - R is not in 3NF!

# Shortcuts

- **If the relation R satisfies that:**
  - all attributes are part of key
    - R must be 2NF & 3NF (WHY?)
  - Singleton keys
    - R must be at least 2NF

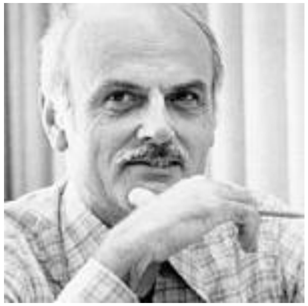
# Normal Form Review

- **Types: 1st, 2nd, 3rd, Boyce-Codd**
- **1st $\supset$ 2nd $\supset$ 3rd $\supset$ Boyce-Codd $\supset$ …**

# Today's Lecture

- **Boyce-Codd Normal Form (BCNF)**
- **Schema decomposition**

# Boyce-Codd Normal Form (BCNF)



- **Also called 3.5NF.**
- **Reln R with FDs *F* is in BCNF if, for all X $\rightarrow$ A in $F^+$**
  - A $\in$ X (called a *trivial* FD), or
  - X is a superkey for R.
- **In other words: "R is in BCNF if the only non-trivial FDs over R are *key constraints*."**

# 3NF VS BCNF

- **Given Reln R and its FDs F, for every FD X -> A in $F^+$,**
  - 3NF requires that at least ONE condition is met
    - (a) X is a superkey for R, or
    - (b) A is a key attribute for R
  - BCNF requires that
    - (a) X is a superkey.

# Example of 3NF VS. BCNF

- **R(ABC)**
- **Key: (A, B)**
- **FD: F={C->B}**
- **Does R satisfy BCNF?**
- **Does R satisfy 3NF?**

# Example of 3NF VS. BCNF

- **R(ABC)**
- **F={AB->C, C->A}**
- **What normal form does R satisfy?**
  - Step 1: Find the key of R:
    - AB, BC.
  - It does not satisfy BCNF
    - In C->A, C is not a key
  - But it satisfies 3NF
    - C->A is OK as A is a key attribute.

# Normal Form Summary

- **Types: 1$^{st}$, 2$^{nd}$, 3$^{rd}$, Boyce-Codd**
- **1$^{st}$ $\supset$ 2$^{nd}$ $\supset$ 3$^{rd}$ $\supset$ Boyce-Codd $\supset$ …**

# Schema Decomposition

# Decomposition of a Relation Schema

- If a relation is not in a desired normal form, it can be *decomposed* into multiple relations
  - Each relation aftre decomposition is in a normal form.
- Consider the relation R(*A1 ... An)*, a *decomposition* of R is to replace R by two or more relations such that:
  - Each new relation scheme contains a subset of the attributes of R, and
  - Every attribute of R must appear in at least one new relation.

# BCNF and Duplicated Values

| S | N | L | R | W | H |
|---|---|---|---|---|---|
| 123-22-3666 | Attishoo | 48 | 8 | 10 | 40 |
| 231-31-5368 | Smiley | 22 | 8 | 10 | 30 |
| 131-24-3650 | Smethurst | 35 | 5 | 7 | 30 |
| 434-26-3751 | Guldu | 35 | 5 | 7 | 32 |
| 612-67-4134 | Madayan | 35 | 8 | 10 | 40 |

Hourly_Emps

- SNLRWH has FDs: S $\rightarrow$ SNLRWH and R $\rightarrow$ W
- Q: Is this relation in BCNF?

No, The second FD causes a violation;
W values repeatedly associated with R values.

# Decomposing a Relation

- Easiest fix is to create a relation RW to store these associations, and to remove W from the main schema:

| S | N | L | R | H |
|---|---|---|---|---|
| 123-22-3666 | Attishoo | 48 | 8 | 40 |
| 231-31-5368 | Smiley | 22 | 8 | 30 |
| 131-24-3650 | Smethurst | 35 | 5 | 30 |
| 434-26-3751 | Guldu | 35 | 5 | 32 |
| 612-67-4134 | Madayan | 35 | 8 | 40 |

Hourly_Emps2

| R | W |
|---|---|
| 8 | 10 |
| 5 | 7 |

Wages

- Q: Are both of these relations are now in BCNF?
- **Decompositions should be used only when needed.**
  - Q: potential problems of decomposition?

# Problems with Decompositions

- Three potential problems:
  1) May be impossible to reconstruct the original relation! (Lossiness)
  2) Dependency checking may require joins.
  3) Some queries become more expensive.
     - e.g., How much does Guldu earn? (in which employees' names are in one table, while salary information in another?)

***Tradeoff*: Must consider these issues vs. redundancy.**

# Task #1

- **How to do lossless decomposition?**

# Lossless Decomposition (example)

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

Decompose table (A, B, C) to Tables (A, C) and (B, C)

| A | C |
|---|---|
| 1 | 3 |
| 4 | 6 |
| 7 | 8 |

| B | C |
|---|---|
| 2 | 3 |
| 5 | 6 |
| 2 | 8 |

| A | C |
|---|---|
| 1 | 3 |
| 4 | 6 |
| 7 | 8 |

$\bowtie$

| B | C |
|---|---|
| 2 | 3 |
| 5 | 6 |
| 2 | 8 |

$=$

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

# Lossy Decomposition (example)

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

Decompose table (A, B, C) to Tables (A, B) and (B, C)

| A | B |
|---|---|
| 1 | 2 |
| 4 | 5 |
| 7 | 2 |

| B | C |
|---|---|
| 2 | 3 |
| 5 | 6 |
| 2 | 8 |

| A | B |
|---|---|
| 1 | 2 |
| 4 | 5 |
| 7 | 2 |

$$\bowtie$$

| 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 |

Tuples that do not exist in the table (A, B, C)

# Lossless Join Decompositions

- 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:

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

- It is always true that $r \subseteq \pi_X(r) \bowtie \pi_Y(r)$

# Solution to Lossless Decomposition

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

$$X \cap Y \rightarrow X, \quad \textbf{or}$$

$$X \cap Y \rightarrow Y$$

i.e., the common attributes of X and Y must be the superkey of either X or Y.

# Lossy Decomposition (example)

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

$\Rightarrow$

| A | B |
|---|---|
| 1 | 2 |
| 4 | 5 |
| 7 | 2 |

| B | C |
|---|---|
| 2 | 3 |
| 5 | 6 |
| 2 | 8 |

$A \to B; C \to B$

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

Lossy decomposition!

| A | B |
|---|---|
| 1 | 2 |
| 4 | 5 |
| 7 | 2 |

$\bowtie$

| 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 |

# Decomposition Principle

- If W $\rightarrow$ Z holds over R and W $\cap$ 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)

# Revisit Decomposition Example

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

$\Rightarrow$

| A | B |
|---|---|
| 1 | 2 |
| 4 | 5 |
| 7 | 2 |

| B | C |
|---|---|
| 2 | 3 |
| 5 | 6 |
| 2 | 8 |

$A \rightarrow B$; $C \rightarrow B$

Lossy decompostition

How can we construct a lossless decomposition of Relation R(A, B, C)?

# Lossless Decomposition (example)

$A \rightarrow B$;
$C \rightarrow B$

| 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 |

$X=\{A, C\}$, $Y=\{B, C\}$, $X \cap Y = \{C\}$, $C \rightarrow \{B, C\}$

Lossless decomposition!

| A | C |
|---|---|
| 1 | 3 |
| 4 | 6 |
| 7 | 8 |

$\bowtie$

| 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!
(Problem #2 of decomposition!)
How to solve this? The next lecture!