Relational Design Theory
Boyce-Codd Normal Form
Relational design by decomposition

- “Mega” relations + properties of the data
- System decomposes based on properties
- Final set of relations satisfies normal form
  - No anomalies, no lost information

- Functional dependencies $\Rightarrow$ Boyce-Codd Normal Form
- Multivalued dependences $\Rightarrow$ Fourth Normal Form
Decomposition of a relational schema

\[ R(A_1, \ldots, A_n) \quad \bar{A} \]
\[ \rightarrow R_1(B_1, \ldots, B_k) \quad \bar{B} \quad \bar{B} \cup \bar{C} = \bar{A} \checkmark \]
\[ R_2(C_1, \ldots, C_m) \quad \bar{C} \quad R_1 \times R_2 = R \checkmark \]

\[ R_1 = TT_{\bar{B}}(R) \checkmark \]
\[ R_2 = TT_{\bar{C}}(R) \checkmark \]
Decomposition Example #1

\[ S_1 (SSN, sName, address, \underline{HScode}, GPA, priority) \]

\[ S_2 (\underline{HScode}, HSname, HScity) \]

\[ \overline{A} \cup \overline{B} = \overline{C} \]

\[ S_1 \times S_2 = \text{Student} \]
Decomposition Example #2

\[ \text{Student}(SSN, sName, address, \text{HScode}, \text{HSname}, \text{HScity}, \text{GPA}, \text{priority}) \]

\[ S_1 (SSN, sName, addr, \text{HScode}, \text{HSname}, \text{HScity}) \]

\[ S_2 (sName, \text{HSname}, \text{GPA}, \text{priority}) \]

\[ \overline{A} \cup \overline{B} = \overline{C} \]

\[ S_1 \cap S_2 = \text{Student} \]
Relational design by decomposition

- “Mega” relations + properties of the data
- System decomposes based on properties
  - “Good” decompositions only
  - Into “good” relations

BCNF
Boyce-Codd Normal Form

Relation $R$ with FDs is in BCNF if:

For each $\bar{A} \rightarrow B$, $\bar{A}$ is a key

BCNF violation

$\bar{A} \rightarrow B$
$\bar{A}$ key
$\bar{A} \rightarrow C D$

"contains a key"
"is a superkey"
BCNF? Example #1

Student(SSN, sName, address, HScode, HSname, HScity, GPA, priority)

\[ \begin{align*}
\text{SSN} & \rightarrow \text{sName}, \text{address}, \text{GPA} \\
\text{GPA} & \rightarrow \text{priority} \\
\text{HScode} & \rightarrow \text{HSname}, \text{HScity}
\end{align*} \]

Keys: \{ssn, HScode\}

Every FD have a key on LHS? No! No
BCNF? Example #2

Apply(SSN, cName, state, date, major)

\[
\text{SSN, cName, state } \rightarrow \text{ date, major}
\]

Key

In BCNF.
Relational design by decomposition

- “Mega” relations + properties of the data
- System decomposes based on properties
  - “Good” decompositions only
  - Into “good” relations

\[ \text{BCNF} \]
BCNF decomposition algorithm

**Input:** relation R + FDs for R

**Output:** decomposition of R into BCNF relations with “lossless join”

Compute keys for R using FDs

Repeat until all relations are in BCNF:

Pick any \( R' \) with \( A \rightarrow B \) that violates BCNF

Decompose \( R' \) into \( R_1(A, B) \) and \( R_2(A, \text{rest}) \)

Compute FDs for \( R_1 \) and \( R_2 \)

Compute keys for \( R_1 \) and \( R_2 \)
BCNF Decomposition Example

Student(SSN, sName, address, HScode, HSname, HScity, GPA, priority)

SSN → sName, address, GPA  GPA → priority

HScode → HSname, HScity

Key: {ssn, HScode}

S1(HScode, HSname, HScity)

S2(SSN, sName, address, HScode, GPA, priority)

S3(GPA, priority)

S4(SSN, sName, address, HScode, GPA)

S5(SSN, sName, address, GPA)

S6(SSN, HScode)
BCNF decomposition algorithm

**Input:** relation R + FDs for R

**Output:** decomposition of R into BCNF relations with “lossless join”

Compute keys for R

Repeat until all relations are in BCNF:

1. Pick any $R'$ with $A \rightarrow B$ that violates BCNF
2. Decompose $R'$ into $R_1(A, B)$ and $R_2(A, \text{rest})$
3. Compute FDs for $R_1$ and $R_2$
4. Compute keys for $R_1$ and $R_2$
Does BCNF guarantee a good decomposition?

- Removes anomalies? ✓
- Can logically reconstruct original relation?
  Too few or too many tuples?

\[
R = \begin{array}{ccc}
A & B & C \\
1 & 2 & 3 \\
4 & 2 & 5 \\
\end{array}
\]

\[
R_1 = \begin{array}{cc}
A & B \\
1 & 2 \\
4 & 2 \\
\end{array},
R_2 = \begin{array}{cc}
B & C \\
2 & 3 \\
2 & 5 \\
\end{array}
\]

\[
R_1 \times R_2 = \begin{array}{ccc}
1 & 2 & 3 \\
1 & 2 & 5 \\
4 & 2 & 3 \\
4 & 2 & 5 \\
\end{array}
\]

Jennifer Widom
Does BCNF guarantee a good decomposition?

- Removes anomalies?
- Can logically reconstruct original relation?
  Too few or too many tuples?
- Some shortcomings discussed in later video