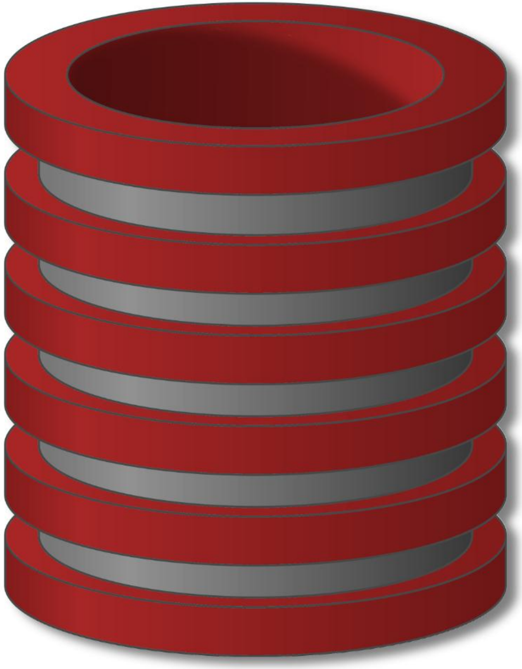


Relational Design Theory

Boyce-Codd Normal Form

Jennifer Widom

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

$R(A_1, \ldots, A_n)$ $\bar{A}$

$\hookrightarrow R_1(B_1, \ldots, B_k)$ $\bar{B}$

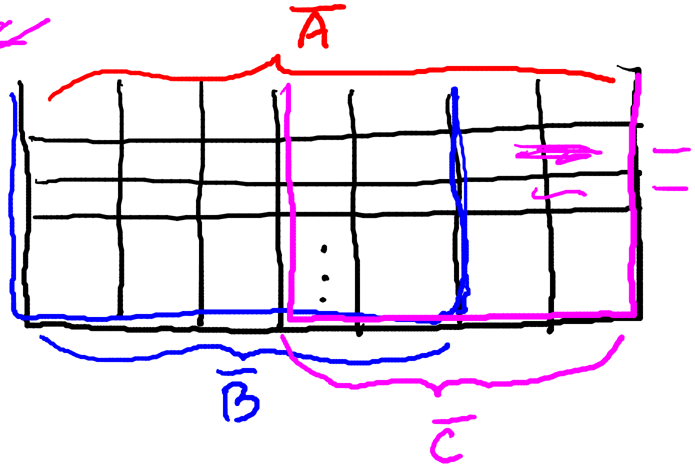
$R_2(C_1, \ldots, C_m)$ $\bar{C}$

$\bar{B} \cup \bar{C} = \bar{A}$ ✓

$R_1 \bowtie R_2 = R$ ✓

$R_1 = \Pi_{\bar{B}}(R)$

$R_2 = \Pi_{\bar{C}}(R)$



Decomposition Example #1

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

S1(SSN, sName, addr, <u>HScode</u>, GPA, priority)

S2(<u>HScode</u>, HSname, HScity)

$\bar{A} \cup \bar{B} = \bar{C}$ $S1 \bowtie S2 = \text{Student}$

Decomposition Example #2

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

S1 (SSN, sName, addr, HScode, HSname, HScity)

S2 (sName, HSname, GPA, priority)

$\bar{A} \cup \bar{B} = \bar{C}$

S1 ⋈ S2 $\stackrel{?}{=}$ Student

Relational design by decomposition

- "Mega" relations + properties of the data
- System decomposes based on properties

❖ "Good" decompositions only ← "reassembly" produces orig.
Lossless join property

❖ 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

$\bar{A}$	B	rest
a	b	—
a	b	—
	⋮	

$\bar{A} \rightarrow B$

key

$\bar{A}$ C D

"contains a key"

"is a superkey"

BCNF? Example #1

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

SSN → sName, address, GPA

GPA → priority

HScode → HSname, HScity

Keys:
{SSN, HScode}

~~Every~~ FD have a Key on LHS?

No! No

BCNF? Example #2

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

$\underbrace{\underline{\text{SSN, cName, state}}}_{\text{key}} \rightarrow \underline{\text{date}}, \underline{\text{major}}$

In BCNF.

Relational design by decomposition

- "Mega" relations + properties of the data
- System decomposes based on properties

❖ "Good" decompositions only — algorithm.

❖ Into "good" relations ✓ 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

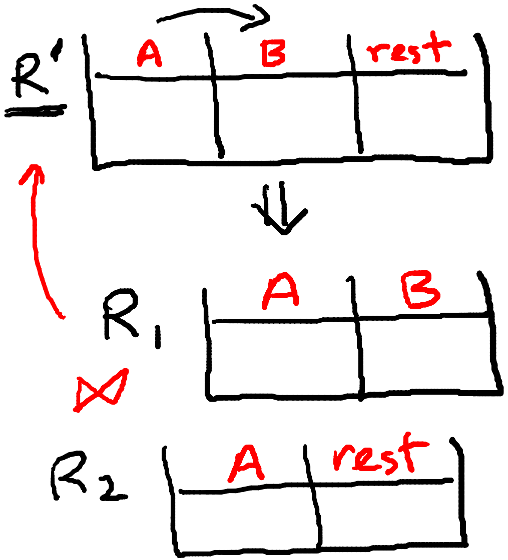
Repeat until all relations are in BCNF:

  Pick any R' with $\bar{A} \rightarrow \bar{B}$ that violates BCNF

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

  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}

$\boxed{S1}$(HScode, HSname, HScity)

~~S2 (SSN, sName, addr, HScode, GPA, priority)~~

$\boxed{S3}$(GPA, priority)

~~S4 (SSN, sName, addr, HScode, GPA)~~

$\boxed{S5}$(SSN, sName, addr, GPA)

$\boxed{S6}$(SSN, HScode)

BCNF decomposition algorithm

Input: relation R + FDs for R

Output: decomposition of R into BCNF relations with "lossless join"

different answer

"Extend"

$A \to B$

$A \to BA^+$

Compute keys for R

Repeat until all relations are in BCNF:

- Pick any R' with $A \to B$ that violates BCNF
- Decompose R' into $R_1(A, B)$ and $R_2(A, \text{rest})$
- Compute FDs for R_1 and R_2 Implied FDs Closure.
- 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?

yes

R $R_1 \bowtie R_2$

R

A	B	C
1	2	3
4	2	5

R1

A	B
1	2
4	2

R2

B	C
2	3
2	5

$R_1 \bowtie R_2$ = 123, 125, 423, 425

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