Introduction to Data Management
CSE 344

Lectures 18: BCNF

Announcements

- Midterm is posted and grades are up
  - Have until 8pm tonight for regrade requests

- WQ6 - 11pm July 31 (Monday)
- HW6 is up due Aug 1 (Tuesday)
  - Database design

- Will drop HW8 - not enough time to get to Spark lecture before it is due.

Review: Relation Decomposition

Break the relation into two:

Name	SSN	PhoneNumber	City
Fred	123-45-6789	206-555-1234	Seattle
Fred	123-45-6789	206-555-6543	Seattle
Joe	987-65-4321	908-555-2121	Westfield

Name	SSN	City
Fred	123-45-6789	Seattle
Joe	987-65-4321	Westfield

SSN	PhoneNumber
123-45-6789	206-555-1234
123-45-6789	206-555-6543
987-65-4321	908-555-2121

Anomalies have gone:

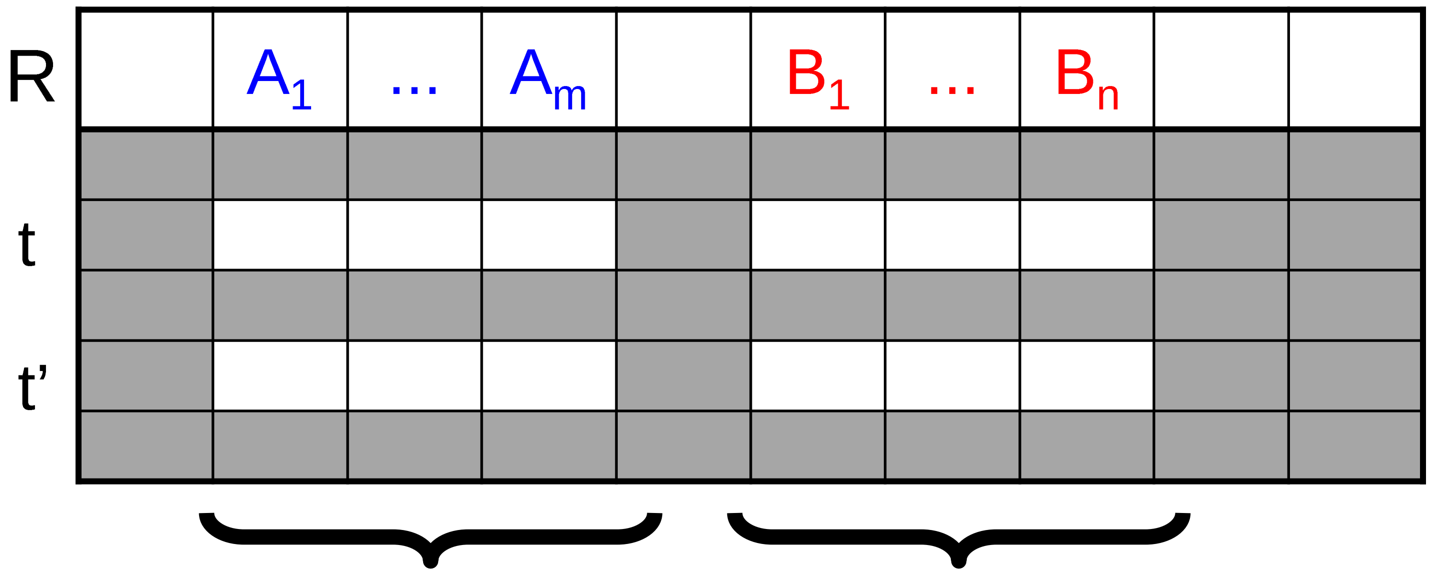
- No more repeated data
- Easy to move Fred to “Bellevue” (how ?)
- Easy to delete all Joe’s phone numbers (how ?)

Review: Functional Dependencies (FDs)

Definition $A_1, \ldots, A_m \rightarrow B_1, \ldots, B_n$ **holds** in R if:

$\forall$ t, t' $\in$ R,

$$(t.A_1 = t'.A_1 \wedge \ldots \wedge t.A_m = t'.A_m \rightarrow t.B_1 = t'.B_1 \wedge \ldots \wedge t.B_n = t'.B_n)$$



if t, t' agree here then t, t' agree here

Closure Algorithm

$X=\{A_1, \ldots, A_n\}$.

Repeat until X doesn't change **do**:
if $B_1, \ldots, B_n \rightarrow C$ is a FD **and**
$B_1, \ldots, B_n$ are all in X
then add C to X.

Example:

1. name → color
2. category → department
3. color, category → price

$\{name, category\}^+ =$
{ name, category, color, department, price }

Hence: $\boxed{\text{name, category} \rightarrow \text{color, department, price}}$

Keys

- A **superkey** is a set of attributes $A_1, ..., A_n$ s.t. for any other attribute B in the same relation, we have $A_1, ..., A_n \rightarrow B$

- A **key** is a minimal superkey
  – A superkey and for which no subset is a superkey

Computing (Super)Keys

- For all sets X, compute X^+
- If X^+ = [all attributes], then X is a superkey
- Try reducing to the minimal X's to get the key

Example

Product(name, price, category, color)

name, category → price
category → color

What is the key ?

Example

Product(name, price, category, color)

name, category $\rightarrow$ price
category $\rightarrow$ color

What is the key ?

(name, category)$^+$ = { name, category, price, color }

Hence (name, category) is a key

How To Eliminate Anomalies

Main idea:

- X → A is OK if X is a (super)key

- X → A is not OK otherwise
  - Need to decompose the table, but how?

Boyce-Codd Normal Form

Boyce-Codd Normal Form

There are no "bad" FDs:

Definition. A relation R is in BCNF if:

Whenever $X \rightarrow B$ is a non-trivial dependency, then X is a superkey.

Equivalently:

Definition. A relation R is in BCNF if:

$\forall$ X in $X \rightarrow B$,
either $X^+ = X$ or $X^+ =$ [all attributes]

BCNF Decomposition Algorithm

Normalize(R)

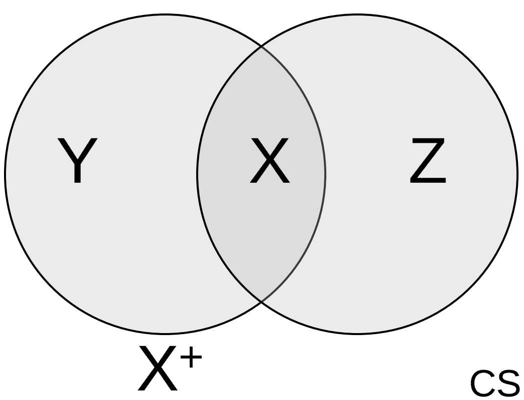
  find X in X→B s.t.: $X \neq X^+$ and $X^+ \neq$ [all attributes]

  <u>if</u> (not found) **<u>then</u>** R is in BCNF

  <u>let</u> $Y = X^+ - X$; $Z =$ [all attributes] $- X^+$

  decompose R into $R1(X \cup Y)$ and $R2(X \cup Z)$

  Normalize(R1); Normalize(R2);

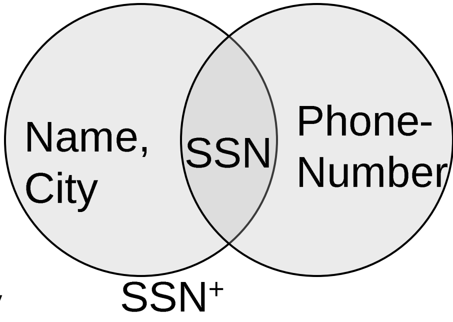


Want X in X→B s.t.: X =X^+ or X^+ = [all attributes]

Example

Name	SSN	PhoneNumber	City
Fred	123-45-6789	206-555-1234	Seattle
Fred	123-45-6789	206-555-6543	Seattle
Joe	987-65-4321	908-555-2121	Westfield
Joe	987-65-4321	908-555-1234	Westfield

SSN → Name, City



The only key is: {SSN, PhoneNumber}

Hence SSN → Name, City is a "bad" dependency

In other words:

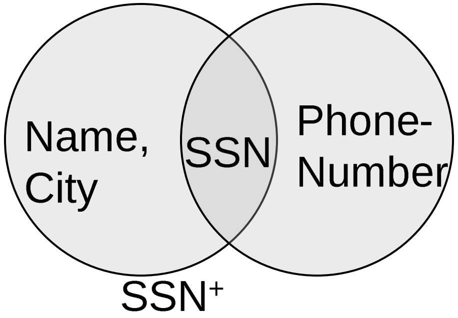
SSN+ = SSN, Name, City and is neither SSN nor All Attributes

Want X in X→B s.t.: X =X^+ or X^+ = [all attributes]

Example BCNF Decomposition

Name	**SSN**	**City**
Fred	123-45-6789	Seattle
Joe	987-65-4321	Westfield

SSN → Name, City



SSN	**PhoneNumber**
123-45-6789	206-555-1234
123-45-6789	206-555-6543
987-65-4321	908-555-2121
987-65-4321	908-555-1234

Let's check anomalies:

- Redundancy ?
- Update ?
- Delete ?

Find X in X→B s.t.: X $\neq X^+$ and $X^+ \neq$ [all attributes]

Example BCNF Decomposition

Person(name, SSN, age, hairColor, phoneNumber)

SSN → name, age

age → hairColor

Find X in X→B s.t.: X $\neq X^+$ and $X^+ \neq$ [all attributes]

Example BCNF Decomposition

Person(name, SSN, age, hairColor, phoneNumber)

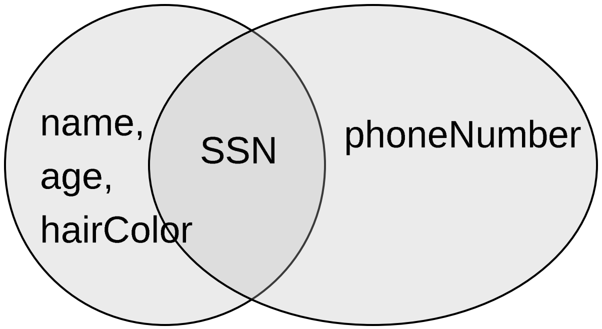
SSN → name, age

age → hairColor

Iteration 1: Person: SSN+ = SSN, name, age, hairColor

Decompose into: P(SSN, name, age, hairColor)

Phone(SSN, phoneNumber)



Find X in X→B s.t.: X ≠ X^+ and X^+ ≠ [all attributes]

Example BCNF Decomposition

Person(name, SSN, age, hairColor, phoneNumber)

SSN → name, age

age → hairColor

What are the keys ?

Iteration 1: Person: SSN+ = SSN, name, age, hairColor

Decompose into: P(SSN, name, age, hairColor)
Phone(SSN, phoneNumber)

Iteration 2: P: age+ = age, hairColor

Decompose: People(SSN, name, age)
Hair(age, hairColor)
Phone(SSN, phoneNumber)

Find X in X→B s.t.: $X \neq X^+$ and $X^+ \neq$ [all attributes]

Example BCNF Decomposition

Person(name, SSN, age, hairColor, phoneNumber)

SSN → name, age

age → hairColor

Note the keys!

Iteration 1: Person: SSN+ = SSN, name, age, hairColor

Decompose into: P(<u>SSN</u>, name, age, hairColor)
Phone(<u>SSN, phoneNumber</u>)

Iteration 2: P: age+ = age, hairColor

Decompose: People(<u>SSN</u>, name, age)
Hair(<u>age</u>, hairColor)
Phone(<u>SSN, phoneNumber</u>)

R(A,B,C,D)

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

Example: BCNF

Recall: find X s.t.
$X \neq X^+ \neq$ [all-attrs]

R(A,B,C,D)

R(A,B,C,D)

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

Example: BCNF

Recall: find X s.t.
$X \neq X^+ \neq$ [all-attrs]

R(A,B,C,D)
$A^+ = ABC \neq ABCD$

R(A,B,C,D)

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

Example: BCNF

Recall: find X s.t.
$X \neq X^+ \neq$ [all-attrs]

R(A,B,C,D)
$A^+ = ABC \neq ABCD$

$R_1(A,B,C)$

$R_2(A,D)$

R(A,B,C,D)

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

Example: BCNF

Recall: find X s.t.
$X \neq X^+ \neq$ [all-attrs]

R(A,B,C,D)
$A^+ = ABC \neq ABCD$

$R_1(A,B,C)$
$B^+ = BC \neq ABC$

$R_2(A,D)$

R(A,B,C,D)

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

Example: BCNF

Recall: find X s.t.
$X \neq X^+ \neq$ [all-attrs]

R(A,B,C,D)
$A^+ = ABC \neq ABCD$

$R_1(A,B,C)$
$B^+ = BC \neq ABC$

$R_2(A,D)$

$R_{11}(B,C)$

$R_{12}(A,B)$

What are the keys ?

What happens if in R we first pick B^+ ? Or AB^+ ?

Decompositions in General

$R(A_1, \ldots, A_n, B_1, \ldots, B_m, C_1, \ldots, C_p)$

$S_1(A_1, \ldots, A_n, B_1, \ldots, B_m)$

$S_2(A_1, \ldots, A_n, C_1, \ldots, C_p)$

S_1 = projection of R on $A_1, \ldots, A_n, B_1, \ldots, B_m$

S_2 = projection of R on $A_1, \ldots, A_n, C_1, \ldots, C_p$

Not all decompositions are good - some are lossy

A Lossless Decomposition

Name	**Price**	**Category**
Gizmo	19.99	Gadget
OneClick	24.99	Camera
Gizmo	19.99	Camera

Name	**Price**
Gizmo	19.99
OneClick	24.99
~~Gizmo~~	~~19.99~~

Name	**Category**
Gizmo	Gadget
OneClick	Camera
Gizmo	Camera

A Lossy Decomposition

What is lossy here?

Name	Price	Category
Gizmo	19.99	Gadget
OneClick	24.99	Camera
Gizmo	19.99	Camera

Name	Category
Gizmo	Gadget
OneClick	Camera
Gizmo	Camera

Price	Category
19.99	Gadget
24.99	Camera
19.99	Camera

Lossy Decomposition

Name	Price	Category
Gizmo	19.99	Gadget
OneClick	24.99	Camera
Gizmo	19.99	Camera

Name	Category
Gizmo	Gadget
OneClick	Camera
Gizmo	Camera

Price	Category
19.99	Gadget
24.99	Camera
19.99	Camera

Decomposition in General

$R(A_1, \ldots, A_n, B_1, \ldots, B_m, C_1, \ldots, C_p)$

$S_1(A_1, \ldots, A_n, B_1, \ldots, B_m)$ $S_2(A_1, \ldots, A_n, C_1, \ldots, C_p)$

Let: S_1 = projection of R on $A_1, \ldots, A_n, B_1, \ldots, B_m$
S_2 = projection of R on $A_1, \ldots, A_n, C_1, \ldots, C_p$

The decomposition is called *lossless* if $R = S_1 \bowtie S_2$

Fact: If $A_1, \ldots, A_n \rightarrow B_1, \ldots, B_m$ then the decomposition is lossless

It follows that every BCNF decomposition is lossless

Schema Refinements = Normal Forms

- 1st Normal Form = all tables are flat
- 2nd Normal Form = obsolete
- Boyce Codd Normal Form = no bad FDs
- 3rd and 4th Normal Form = see book
  - BCNF is lossless but can cause loss of ability to check some FDs (see book 3.4.4)
  - 3NF fixes that (is lossless and dependency-preserving), but some tables might not be in BCNF – i.e., they may have redundancy anomalies

Normalization IRL

- It's good to know about BCNF (and maybe 3NF).
- Should every table always be in BCNF?

R(Item, OrderNo, Customer, Address, Zip Code, State)

What are the FDs:

Normalization IRL

R(Item, OrderNo, Customer, Address, Zip Code, State)

What are the FDs:

OrderNo -> Customer
Customer -> Address, Zip Code, State
Zip Code -> State

What is BCNF:

Normalization IRL

R(Item, OrderNo, Customer, Address, Zip Code, State)

What is BCNF:

R(Item, OrderNo, Customer, Address, Zip Code, State)
OrderNo^{+} = Customer, Address, Zip, Code, State

R_1(Item, OrderNo)

R_2(OrderNo, Customer, Address, Zip Code, State)

R_3(OrderNo, CustomerId)

R_4(CustomerId, Name, Address, Zip Code, State)

Are We Done?

Normalization IRL

R(Item, OrderNo, Customer, Address, Zip Code, State)

What is BCNF:

R(Item, OrderNo, Customer, Address, Zip Code, State)
OrderNo^+ = Customer, Address, Zip, Code, State

R_1(Item, OrderNo)

R_2(OrderNo, Customer, Address, Zip Code, State)

R_3(OrderNo, CustomerId)

R_4(CustomerId, Name, Address, Zip Code)

R_5(Zip Code, State)

Is this a good idea?