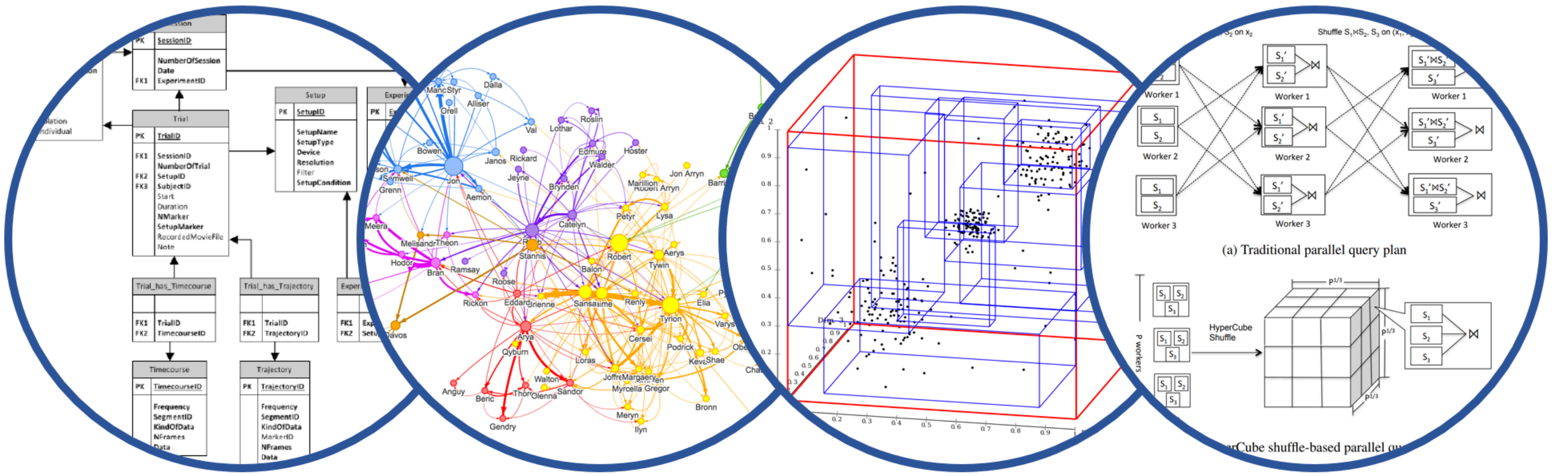


Introduction to Data Management

BCNF Decomposition

Paul G. Allen School of Computer Science and Engineering
University of Washington, Seattle

Recap

Superkey

A **Superkey** is a set of attributes $A_1, \dots, A_n$ s.t. for any single attribute B:

$$A_1, \dots, A_n \rightarrow B$$

In other words, for the set of all attributes C in the relation R, the set $\{A_1, \dots, A_n\}$ is a superkey iff $\{A_1, \dots, A_n\}^+ = C$

Recap

Superkey

A **Superkey** is a set of attributes $A_1, \dots, A_n$ s.t. for any single attribute B:

$$A_1, \dots, A_n \to B$$

In other words, for the set of all attributes C in the relation R, the set $\{A_1, \dots, A_n\}$ is a superkey iff $\{A_1, \dots, A_n\}^+ = C$

Key

A **Key** is a minimal superkey, i.e. no subset of a key is a superkey.

Recap

Superkey

A **Superkey** is a set of attributes $A_1, \ldots, A_n$ s.t. for any single attribute B:

$$A_1, \ldots, A_n \rightarrow B$$

In other words, for the set of all attributes C in the relation R, the set $\{A_1, \ldots, A_n\}$ is a superkey iff $\{A_1, \ldots, A_n\}^+ = C$

Key

A **Key** is a minimal superkey, i.e. no subset of a key is a superkey.

Recap

Restaurants(rid, name, rating, popularity)

rid → name

rid → rating

rating → popularity

	Closure	Superkey?	Key?
{rid, rating}	{rid, name, rating, popularity}		
rid	{rid, name, rating, popularity}		
rating	{rating, popularity}		
popularity	{popularity}		

Recap

Restaurants(rid, name, rating, popularity)
rid → name
rid → rating
rating → popularity

	Closure	Superkey?	Key?
{rid, rating}	{rid, name, rating, popularity}	Yes	
rid	{rid, name, rating, popularity}	Yes	
rating	{rating, popularity}		
popularity	{popularity}		

Recap

Restaurants(rid, name, rating, popularity)

rid → name

rid → rating

rating → popularity

	Closure	Superkey?	Key?
{rid, rating}	{rid, name, rating, popularity}	Yes	
rid	{rid, name, rating, popularity}	Yes	
rating	{rating, popularity}	No	
popularity	{popularity}	No	

Recap

Restaurants(rid, name, rating, popularity)
rid → name
rid → rating
rating → popularity

	Closure	Superkey?	Key?
{rid, rating}	{rid, name, rating, popularity}	Yes	No
rid	{rid, name, rating, popularity}	Yes	
rating	{rating, popularity}	No	
popularity	{popularity}	No	

Recap

Restaurants(rid, name, rating, popularity)
rid → name
rid → rating
rating → popularity

	Closure	Superkey?	Key?
{rid, rating}	{rid, name, rating, popularity}	Yes	No
rid	{rid, name, rating, popularity}	Yes	Yes
rating	{rating, popularity}	No	
popularity	{popularity}	No	

Recap

Restaurants(rid, name, rating, popularity)
rid → name
rid → rating
rating → popularity

	Closure	Superkey?	Key?
{rid, rating}	{rid, name, rating, popularity}	Yes	No
rid	{rid, name, rating, popularity}	Yes	Yes
rating	{rating, popularity}	No	No
popularity	{popularity}	No	No

Usefulness of Keys in Design

What intuitions do we get from data interrelationships?

- FDs that are not superkeys hint at redundancy
  - If a FD antecedent <u>is **not** a superkey</u>, we can remove redundant information, i.e. the FD consequent
- Rephrased
  - $\{A\} \rightarrow \{B\}$ is fine if $\{A\}$ is a superkey
  - Otherwise, we can extract $\{B\}$ into a separate table

Name	SSN	Phone	City
Fred	123-45-6789	206-555-9999	Seattle
Fred	123-45-6789	206-555-8888	Seattle
Joe	987-65-4321	415-555-7777	San Francisco

SSN is not a superkey!

Usefulness of Keys in Design

What intuitions do we get from data interrelationships?

- FDs that are not superkeys hint at redundancy
  - If a FD antecedent <u>is **not** a superkey</u>, we can remove redundant information, i.e. the FD consequent.
- Rephrased
  - $\{A\} \rightarrow \{B\}$ is fine if $\{A\}$ is a superkey
  - Otherwise, we can extract $\{B\}$ into a separate table

Redundancy!

Name	**SSN**	**Phone**	**City**
Fred	123-45-6789	206-555-9999	Seattle
Fred	123-45-6789	206-555-8888	Seattle
Joe	987-65-4321	415-555-7777	San Francisco

SSN is not a superkey!

Think About This

$\{SSN\}^+ = ?$

Previously we converted this

Name	SSN	Phone	City
Fred	123-45-6789	206-555-9999	Seattle
Fred	123-45-6789	206-555-8888	Seattle
Joe	987-65-4321	415-555-7777	San Francisco

into this

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

SSN	Phone
123-45-6789	206-555-9999
123-45-6789	206-555-8888
987-65-4321	415-555-7777

Think About This

$$\{SSN\}^+ = \{SSN, Name, City\}$$

Previously we converted this

Name	SSN	Phone	City
Fred	123-45-6789	206-555-9999	Seattle
Fred	123-45-6789	206-555-8888	Seattle
Joe	987-65-4321	415-555-7777	San Francisco

into this

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

SSN	Phone
123-45-6789	206-555-9999
123-45-6789	206-555-8888
987-65-4321	415-555-7777

Database Design

Database Design is about

(1) characterizing data and (2) organizing data

How to talk about properties
we know or see in the data

Database Design

Database Design is about

(1) characterizing data and (2) organizing data

How to organize data to promote ease of use and efficiency

Normal Forms

Normal Forms

- 1NF → Flat
- 2NF → No partial FDs (obsolete)
- 3NF → Preserve all FDs, but allow anomalies
- BCNF → No transitive FDs, but can lose FDs
- 4NF → Considers multi-valued dependencies
- 5NF → Considers join dependencies (hard to do)

In 414, we only discuss this

Normal Forms

1NF

A relation R is in **First Normal Form** if all attribute values are atomic. Attribute values cannot be multivalued. Nested relations are not allowed.

We call data in 1NF “flat.”

BCNF

BCNF

A relation R is in **Boyce-Codd Normal Form (BCNF)** if for every non-trivial dependency, $X \rightarrow A$, X is a superkey.

Equivalently, a relation R is in BCNF if $\forall X$ either $X^+ = X$ or $X^+ = C$ where C is the set of all attributes in R

BCNF

BCNF

A relation R is in **Boyce-Codd Normal Form (BCNF)** if for every non-trivial dependency, $X \rightarrow A$, X is a superkey.

Equivalently, a relation R is in BCNF if $\forall X$ either $X^+ = X$ or $X^+ = C$ where C is the set of all attributes in R

Trvial FD

Super key

BCNF

BCNF

A relation R is in **Boyce-Codd Normal Form (BCNF)** if for every non-trivial dependency, $X \rightarrow A$, X is a superkey.

Equivalently, a relation R is in BCNF if $\forall X$ either $X^+ = X$ or $X^+ = C$ where C is the set of all attributes in R

Name	SSN	Phone	City
Fred	123-45-6789	206-555-9999	Seattle
Fred	123-45-6789	206-555-8888	Seattle
Joe	987-65-4321	415-555-7777	San Francisco

SSN → SSN, Name, City

We often call these “bad FDs” because they prevent the relation from being in BCNF

BCNF

BCNF

A relation R is in **Boyce-Codd Normal Form (BCNF)** if for every non-trivial dependency, $X \to A$, X is a superkey.

Equivalently, a relation R is in BCNF if $\forall X$ either $X^+ = X$ or $X^+ = C$ where C is the set of all attributes in R

Name	SSN	Phone	City
Fred	123-45-6789	206-555-9999	Seattle
Fred	123-45-6789	206-555-8888	Seattle
Joe	987-65-4321	415-555-7777	San Francisco

SSN → SSN, Name, City

We often call these “bad FDs” because they prevent the relation from being in BCNF

If we remove all the bad FDs, then the relation is in BCNF

Decomposition

- "Extracting" attributes can be done with **decomposition** (split the schema into smaller parts)
- For this class, decomposition means the following:

$$R(A_1, \ldots, A_n, B_1, \ldots, B_m, C_1, \ldots, C_k) \begin{cases} R_1(A_1, \ldots, A_n, B_1, \ldots, B_m) \\ R_2(A_1, \ldots, A_n, C_1, \ldots, C_k) \end{cases}$$

Decomposition

- "Extracting" attributes can be done with **decomposition** (split the schema into smaller parts)
- For this class, decomposition means the following:

$$R(A_1, \dots, A_n, B_1, \dots, B_m, C_1, \dots, C_k) \to \begin{cases} R_1(A_1, \dots, A_n, B_1, \dots, B_m) \\ R_2(A_1, \dots, A_n, C_1, \dots, C_k) \end{cases}$$

Some common attributes are present so we can rejoin data

BCNF Decomposition Algorithm

$Normalize(R)$
  $C \leftarrow$ the set of all attributes in R
  find X **s.t.** $X^+ \neq X$ **and** $X^+ \neq C$
  if X is not found
  then "R is in BCNF"
  else
    decompose R into $R_1(X^+)$ and $R_2\big((C - X^+) \cup X\big)$
    $Normalize(R_1)$
    $Normalize(R_2)$

BCNF Decomposition Algorithm

$Normalize(R)$
  $C \leftarrow$ the set of all attributes in R
  find X **s.t.** $X^+ \neq X$ **and** $X^+ \neq C$
  if X is not found
  then "R is in BCNF"
  else
    decompose R into $R_1(X^+)$ and $R_2\big((C - X^+) \cup X\big)$
    $Normalize(R_1)$
    $Normalize(R_2)$

Determine if R is in BCNF already

BCNF Decomposition Algorithm

$Normalize(R)$
  $C \leftarrow$ the set of all attributes in R
  find X **s.t.** $X^+ \neq X$ **and** $X^+ \neq C$
  if X is not found
  then "R is in BCNF"
  else
    decompose R into $R_1(X^+)$ and $R_2\big((C - X^+) \cup X\big)$
    $Normalize(R_1)$
    $Normalize(R_2)$

Determine if R is in BCNF already

Here we have found a "bad" X^+

BCNF Decomposition Algorithm

$Normalize(R)$
  $C \leftarrow$ the set of all attributes in R
  find X **s.t.** $X^+ \neq X$ **and** $X^+ \neq C$
  if X is not found
  then "R is in BCNF"
  else
    decompose R into $R_1(X^+)$ and $R_2\big((C - X^+) \cup X\big)$
    $Normalize(R_1)$
    $Normalize(R_2)$

Determine if R is in BCNF already

Here we have found a "bad" X^+

Decompose into a relation where X is a superkey

Decompose into a relation with X and attributes X cannot determine

BCNF Decomposition Example

$Normalize(R)$
$C \leftarrow$ the set of all attributes in R
find X **s.t.** $X^+ \neq X$ **and** $X^+ \neq C$
if X is not found
then "R is in BCNF"
else
decompose R into $R_1(X^+)$ and $R_2\big((C - X^+) \cup X\big)$
$Normalize(R_1)$
$Normalize(R_2)$

Restaurants(rid, name, rating, popularity, recommended)

rid → name, rating

rating → popularity

popularity → recommended

Restaurants(rid, name, rating, popularity, recommended)

BCNF Decomposition Example

$Normalize(R)$
  $C \leftarrow$ the set of all attributes in R
  find X **s.t.** $X^+ \neq X$ **and** $X^+ \neq C$
  if X is not found
  then "R is in BCNF"
  else
    decompose R into $R_1(X^+)$ and $R_2((C - X^+) \cup X)$
    $Normalize(R_1)$
    $Normalize(R_2)$

Restaurants(rid, name, rating, popularity, recommended)

rid → name, rating

rating → popularity

popularity → recommended

Restaurants(rid, name, rating, popularity, recommended)

(1) rating → rating, popularity, recommended ("bad" FD)

BCNF Decomposition Example

	$Normalize(R)$ $C \leftarrow$ the set of all attributes in R **find** X **s.t.** $X^+ \neq X$ **and** $X^+ \neq C$ **if** X is not found **then** "R is in BCNF" **else** decompose R into $R_1(X^+)$ and $R_2\big((C - X^+) \cup X\big)$ $Normalize(R_1)$ $Normalize(R_2)$	

Restaurants(rid, name, rating, popularity, recommended)

rid → name, rating

rating → popularity

popularity → recommended

Restaurants(rid, name, rating, popularity, recommended)

(1) rating → rating, popularity, recommended ("bad" FD)

(2) R1 = rating, popularity, recommended

BCNF Decomposition Example

$Normalize(R)$
$C \leftarrow$ the set of all attributes in R
find X **s.t.** $X^+ \neq X$ **and** $X^+ \neq C$
if X is not found
then "R is in BCNF"
else
decompose R into $R_1(X^+)$ and $R_2((C - X^+) \cup X)$
$Normalize(R_1)$
$Normalize(R_2)$

Restaurants(rid, name, rating, popularity, recommended)

rid → name, rating

rating → popularity

popularity → recommended

Restaurants(rid, name, rating, popularity, recommended)

(1) rating → rating, popularity, recommended ("bad" FD)
(2) R1 = rating, popularity, recommended
(3) R2 = rid, name, rating

BCNF Decomposition Example

$Normalize(R)$
$C \leftarrow$ the set of all attributes in R
find X **s.t.** $X^+ \neq X$ **and** $X^+ \neq C$
if X is not found
then "R is in BCNF"
else
decompose R into $R_1(X^+)$ and $R_2\big((C - X^+) \cup X\big)$
$Normalize(R_1)$
$Normalize(R_2)$

Restaurants(rid, name, rating, popularity, recommended)

rid → name, rating

rating → popularity

popularity → recommended

Restaurants(rid, name, rating, popularity, recommended)

(1) rating → rating, popularity, recommended ("bad" FD)

(2) R1 = rating, popularity, recommended

(3) R2 = rid, name, rating

Finished?

BCNF Decomposition Example

$Normalize(R)$
  $C \leftarrow$ the set of all attributes in R
  find X **s.t.** $X^+ \neq X$ **and** $X^+ \neq C$
  if X is not found
  then "R is in BCNF"
  else
    decompose R into $R_1(X^+)$ and $R_2\big((C - X^+) \cup X\big)$
    $Normalize(R_1)$
    $Normalize(R_2)$

Restaurants(rid, name, rating, popularity, recommended)

rid → name, rating

rating → popularity

popularity → recommended

Restaurants(rid, name, rating, popularity, recommended)

(1) rating → rating, popularity, recommended   ("bad" FD)
(2) R1 = rating, popularity, recommended
(3) R2 = rid, name, rating

Finished? NO! (popularity → recommended) is still "bad"

We decompose R1 into R3, R4

BCNF Decomposition Example

$Normalize(R)$
  $C \leftarrow$ the set of all attributes in R
  find X **s.t.** $X^+ \neq X$ **and** $X^+ \neq C$
  if X is not found
  then "R is in BCNF"
  else
    decompose R into $R_1(X^+)$ and $R_2((C - X^+) \cup X)$
    $Normalize(R_1)$
    $Normalize(R_2)$

Restaurants(rid, name, rating, popularity, recommended)

rid → name, rating

rating → popularity

popularity → recommended

Restaurants(rid, name, rating, popularity, recommended)

(1) rating → rating, popularity, recommended ("bad" FD)

(2) R1 = rating, popularity, recommended

(3) R2 = rid, name, rating

Finished? NO! (popularity → recommended) is still "bad"

We decompose R1 into R3, R4

R2 = ride, name, rating R3 = rating, popularity R4 = popularity, recommended

BCNF Decomposition Example

$Normalize(R)$
  $C \leftarrow$ the set of all attributes in R
  find X **s.t.** $X^+ \neq X$ **and** $X^+ \neq C$
  if X is not found
  then "R is in BCNF"
  else
    decompose R into $R_1(X^+)$ and $R_2((C - X^+) \cup X)$
    $Normalize(R_1)$
    $Normalize(R_2)$

Restaurants(rid, name, rating, popularity, recommended)

rid → name, rating

rating → popularity

popularity → recommended

Restaurants(rid, name, rating, popularity, recommended)

(1) rating → rating, popularity, recommended   ("bad" FD)

(2) R1 = rating, popularity, recommended

(3) R2 = rid, name, rating

Finished? NO! (popularity → recommended) is still "bad"

We decompose R1 into R3, R4

These three tables are the final decomp.

R2 = ride, name, rating   R3 = rating, popularity   R4 = popularity, recommended

BCNF Decomposition Order

Restaurants(rid, name, rating, popularity, recommended)

rid → name, rating

rating → popularity

popularity → recommended

Note that we chose to split the tables on (rating → rating, popularity, recommended) first. We could have instead chosen (popularity → recommended) first.

In this case the final tables in BCNF will have the same attributes, but not always.

As long as the end result is in BCNF, the particular distribution of attributes doesn't matter for correctness.

Losslessness

Definition

Lossless Decomposition is a reversible decomposition, i.e. rejoining all decomposed relations will always result exactly with the original data.

This is the opposite of a **Lossy Decomposition**, an irreversible decomposition, where rejoining all decomposed relations may result something other than the original data, specifically with extra tuples.

This concept might be familiar if you have ever encountered lossless data compression (e.g. Huffman encoding or PNG) or lossy data compression (e.g. JPEG).

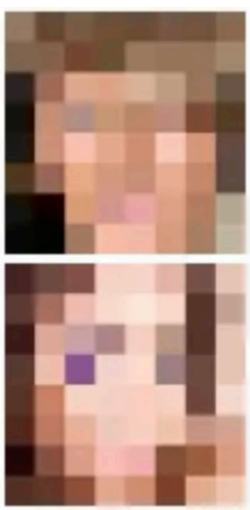
Losslessness

Definition

Lossless Decomposition is a reversible decomposition, i.e. rejoining all decomposed relations will always result exactly with the original data.

This is the opposite of a **Lossy Decomposition**, an irreversible decomposition, where rejoining all decomposed relations may result something other than the original data, specifically with extra tuples.

This concept might be familiar if you have ever encountered lossless data compression (e.g. Huffman encoding or PNG) or lossy data compression (e.g. JPEG).



Neural nets trying to solve the ”Zoom… enhance!” problem

(link from Google Research)

Losslessness

Definition

Lossless Decomposition is a reversible decomposition, i.e. rejoining all decomposed relations will always result exactly with the original data.

This is the opposite of a **Lossy Decomposition**, an irreversible decomposition, where rejoining all decomposed relations may result something other than the original data, specifically with extra tuples.

This concept might be familiar if you have ever encountered lossless data compression (e.g. Huffman encoding or PNG) or lossy data compression (e.g. JPEG).

Neural nets trying to solve the ”Zoom… enhance!” problem

(link from Google Research)

Losslessness

Is BCNF decomposition lossless?

Losslessness

Is BCNF decomposition lossless?

Yes!

In our example:

R2 = ride, name, rating

R3 = rating, popularity

R4 = popularity, recommended

Losslessness

Is BCNF decomposition lossless?

Yes!

In our example:

R2 = rid, name, rating

R3 = rating, popularity

R4 = popularity, recommended

…gives us original R

More examples

Consider this example:

R (A, B, C, D, E, F)

A -> CD
F -> AE
D -> B

More examples

Consider this example:

R (A, B, C, D, E, F)

{A, C, D, B}

A -> CD
F -> AE
D -> B

Good idea to start with closures first:
A+ = {ABCD}
So what's our first decomp?

More examples

Consider this example:

R (A, B, C, D, E, F)

A -> CD

F -> AE

D -> B

Good idea to start with closures first:

A+ = {ABCD}

So what’s our first decomp?

More examples

R (ABCDEF)

A -> CD

F -> AE

D -> B

More examples

R (ABCDEF)

A -> CD

F -> AE

D -> B

R1 (?)

R2 (?)

A+ = {ABCD}

More examples

A -> CD

F -> AE

D -> B

R (ABCDEF)

R1 (ABCD)

R2 (?)

A+ = {ABCD}

More examples

A -> CD
F -> AE
D -> B

R (ABCDEF)

R1 (ABCD)

R2 (AEF)

A+ = {ABCD}

More examples

R (ABCDEF)

R1 (ABCD)

R2 (AEF)

Done?

A -> CD

F -> AE

D -> B

More examples

A -> CD
F -> AE
D -> B

R (ABCDEF)

R1 (ABCD)   R2 (AEF)

Done? No!

More examples

R (ABCDEF)

R1 (ABCD)

R2 (AEF)

DB

ACD

A -> CD

F -> AE

D -> B

Next attribute(s)?