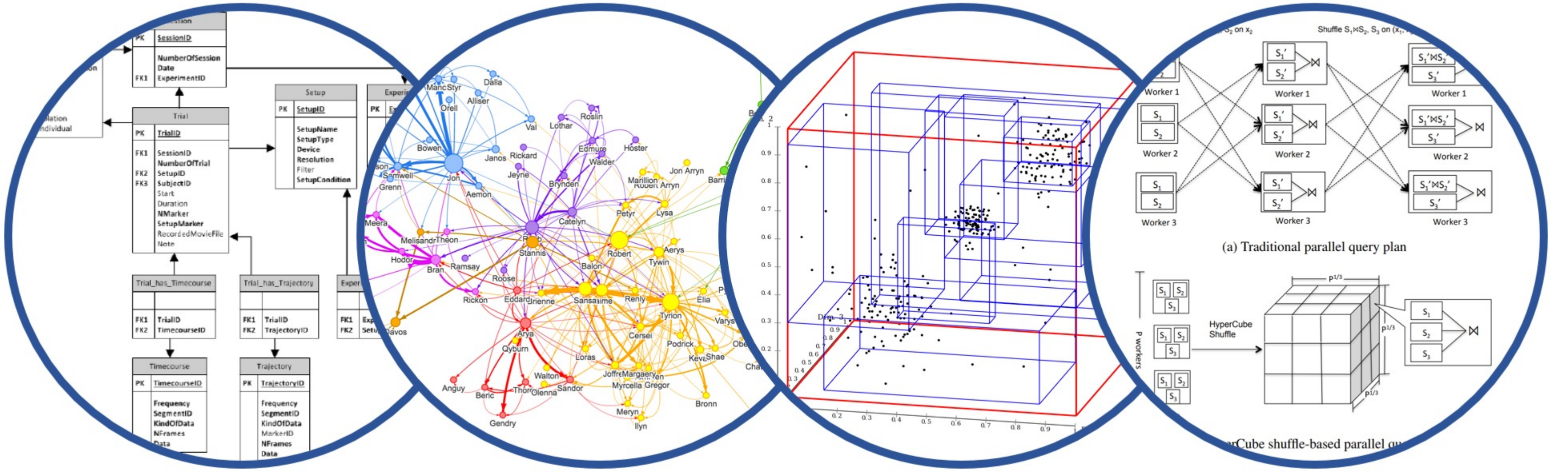


Introduction to Data Management
BCNF Decomposition

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

Announcements

HW4 due on Friday, May 3rd

Midterm:

- This Friday, in class, closed books, no cheat sheet
- Some practice midterms on the course website

longest

- Midterm has four parts:
  - SQL
  - Relational Algebra
  - Entity-Relationship Diagrams (ER)
  - Functional Dependencies

shortest

Inference

An Interesting Observation

If all these FDs are true:

Name → Color
Category → Dept
Color, Dept → Price

Then this FD is also true:

Name, Category → Price

Proof: (see last lecture)

Discussion

Two ways to infer new FDs:

- Armstrong axioms
- The closure operator

Armstrong’s Axioms

Armstrong's Axioms

Reflexivity: if $Y \subseteq X$ then $X \to Y$

Augmentation: if $X \to Y$ then $XZ \to YZ$

Transitivity: if $X \to Y$ and $Y \to Z$ then $X \to Z$

Armstrong's Axioms

Reflexivity: if $Y \subseteq X$ then $X \to Y$

Called a trivial FD

Augmentation: if $X \to Y$ then $XZ \to YZ$

Transitivity: if $X \to Y$ and $Y \to Z$ then $X \to Z$

Using Armstrong's Axioms

Reflexivity: if $Y \subseteq X$ then $X \rightarrow Y$
Augmentation: if $X \rightarrow Y$ then $XZ \rightarrow YZ$
Transitivity: if $X \rightarrow Y$ and $Y \rightarrow Z$ then $X \rightarrow Z$

1. Name → Color
2. Category → Dept
3. Color, Dept → Price

⇒

Name, Category → Price

Using Armstrong's Axioms

Reflexivity: if $Y \subseteq X$ then $X \to Y$
Augmentation: if $X \to Y$ then $XZ \to YZ$
Transitivity: if $X \to Y$ and $Y \to Z$ then $X \to Z$

1. Name → Color
2. Category → Dept
3. Color, Dept → Price

⇒ Name, Category → Price

4. Name, Category → Color, Category (Augmentation of 1)

Using Armstrong's Axioms

Reflexivity: if $Y \subseteq X$ then $X \rightarrow Y$
Augmentation: if $X \rightarrow Y$ then $XZ \rightarrow YZ$
Transitivity: if $X \rightarrow Y$ and $Y \rightarrow Z$ then $X \rightarrow Z$

1. Name → Color
2. Category → Dept
3. Color, Dept → Price

⇒ Name, Category → Price

4. Name, Category → Color, Category (Augmentation of 1)
5. Color, Category → Color, Dept (Augmentation of 2)

Using Armstrong's Axioms

Reflexivity: if $Y \subseteq X$ then $X \rightarrow Y$
Augmentation: if $X \rightarrow Y$ then $XZ \rightarrow YZ$
Transitivity: if $X \rightarrow Y$ and $Y \rightarrow Z$ then $X \rightarrow Z$

1. Name → Color
2. Category → Dept
3. Color, Dept → Price

⇨ Name, Category → Price

4. Name, Category → Color, Category (Augmentation of 1)
5. Color, Category → Color, Dept (Augmentation of 2)
6. Color, Category → Price (Transitivity 5 and 3)

Using Armstrong's Axioms

Reflexivity: if $Y \subseteq X$ then $X \rightarrow Y$
Augmentation: if $X \rightarrow Y$ then $XZ \rightarrow YZ$
Transitivity: if $X \rightarrow Y$ and $Y \rightarrow Z$ then $X \rightarrow Z$

1. Name → Color
2. Category → Dept
3. Color, Dept → Price

⇒ Name, Category → Price

4. Name, Category → Color, Category (Augmentation of 1)
5. Color, Category → Color, Dept (Augmentation of 2)
6. Color, Category → Price (Transitivity 5 and 3)
7. Name, Category → Price (Transitivity 4 and 6)

Discussion

- Armstrong's Axioms were introduced in the 70s, shortly after Codd's relational model
- They are widely known today
- But they are cumbersome to use for inference
- Instead, the efficient inference method uses the **closure operator**: next.

The Closure Operator

The Closure of a set X

Fix a set of Functional Dependencies

The closure X^+ of a
set of attributes X
is the set of attributes A
such that $X \rightarrow A$.

The Closure of a set X

Fix a set of Functional Dependencies

The closure X^+ of a set of attributes X is the set of attributes A such that $X \to A$.

```
Closure(X):
  Repeat:
      find a FD Y → A
      such that Y ⊆ X and A ⊈ X
      X ≔ X ∪ A
  Until "no more change"
```

The Closure of a set X

Fix a set of Functional Dependencies

The closure X^+ of a set of attributes X is the set of attributes A such that $X \rightarrow A$.

```
Closure(X):
  Repeat:
     find a FD Y → A
     such that Y ⊆ X and A ⊈ X
     X ≔ X ∪ A
  Until "no more change"
```

Name → Color
Category → Dept
Color, Dept → Price

$\{\text{Name}, \text{Category}\}^+ =$

The Closure of a set X

Fix a set of Functional Dependencies

The closure X^+ of a set of attributes X is the set of attributes A such that $X \to A$.

```
Closure(X):
  Repeat:
      find a FD Y → A
      such that Y ⊆ X and A ⊈ X
      X ≔ X ∪ A
  Until "no more change"
```

Name → Color
Category → Dept
Color, Dept → Price

$\{\text{Name}, \text{Category}\}^+ =$

$= \{\text{Name}, \text{Category}, \qquad \}$

The Closure of a set X

Fix a set of Functional Dependencies

The closure X^+ of a set of attributes X is the set of attributes A such that $X \rightarrow A$.

```
Closure(X):
  Repeat:
     find a FD Y → A
     such that Y ⊆ X and A ⊄ X
     X := X ∪ A
  Until "no more change"
```

Name → Color
Category → Dept
Color, Dept → Price

$\{Name, Category\}^+ =$

$= \{Name, Category, Color, \qquad \}$

The Closure of a set X

Fix a set of Functional Dependencies

The closure X^+ of a set of attributes X is the set of attributes A such that $X \to A$.

```
Closure(X):
  Repeat:
      find a FD Y → A
      such that Y ⊆ X and A ⊈ X
      X ≔ X ∪ A
  Until "no more change"
```

Name → Color
Category → Dept
Color, Dept → Price

$\{\text{Name}, \text{Category}\}^+ =$

$= \{\text{Name}, \text{Category}, \text{Color}, \text{Dept}, \quad\quad \}$

The Closure of a set X

Fix a set of Functional Dependencies

The closure X^+ of a set of attributes X is the set of attributes A such that $X \to A$.

```
Closure(X):
  Repeat:
      find a FD Y → A
      such that Y ⊆ X and A ⊈ X
      X ≔ X ∪ A
  Until "no more change"
```

Name → Color
Category → Dept
Color, Dept → Price

$$\{\text{Name}, \text{Category}\}^+ = \\ = \{\text{Name}, \text{Category}, \text{Color}, \text{Dept}, \text{Price}\}$$

The Closure of a set X

Fix a set of Functional Dependencies

The closure X^+ of a set of attributes X is the set of attributes A such that $X \rightarrow A$.

```
Closure(X):
  Repeat:
      find a FD Y → A
      such that Y ⊆ X and A ⊄ X
      X := X ∪ A
  Until "no more change"
```

Name → Color
Category → Dept
Color, Dept → Price

$\{\text{Name, Category}\}^+ =$

$= \{\text{Name, Category, Color, Dept, Price}\}$

$\{\text{Color}\}^+ =$

The Closure of a set X

Fix a set of Functional Dependencies

The closure X^+ of a set of attributes X is the set of attributes A such that $X \rightarrow A$.

```
Closure(X):
  Repeat:
     find a FD Y → A
     such that Y ⊆ X and A ⊄ X
     X := X ∪ A
  Until "no more change"
```

Name → Color
Category → Dept
Color, Dept → Price

$$\{\text{Name}, \text{Category}\}^+ =$$
$$= \{\text{Name}, \text{Category}, \text{Color}, \text{Dept}, \text{Price}\}$$

$$\{\text{Color}\}^+ = \{\text{Color}\}$$

Discussion so Far

- Goal is to detect/remove anomalies

- Anomalies are caused by unwanted FDs
  E.g. UID → Name, City; but UID not a key

- Next : **Keys**

Keys

Keys and Superkeys

- Fix a relation $R(A_1, \ldots, A_n)$ and a set of FDs

- A **super-key** is a set X such that $X \rightarrow A_i$ for every attribute A_i

Keys and Superkeys

- Fix a relation $R(A_1, \ldots, A_n)$ and a set of FDs

- A **super-key** is a set X such that $X \to A_i$ for every attribute A_i

Equivalently:
$X^+ = A_1 \ldots A_n$

Keys and Superkeys

- Fix a relation $R(A_1, \ldots, A_n)$ and a set of FDs
- A **super-key** is a set X such that $X \to A_i$ for every attribute A_i

Equivalently:
$X^+ = A_1 \ldots A_n$

- A **key** is a minimal super-key X

Keys and Superkeys

- Fix a relation $R(A_1, \ldots, A_n)$ and a set of FDs

- A **super-key** is a set X such that $X \rightarrow A_i$ for every attribute A_i

Equivalently: $X^+ = A_1 \ldots A_n$

- A **key** is a minimal super-key X

In other words, no super-key $Y \subsetneq X$ exists

Example: Find the Keys

UID	Name	Phone	City
234	Fred	206-555-9999	Seattle
234	Fred	206-555-8888	Seattle
987	Joe	415-555-7777	SF
…	…	…	…

UID → Name, City

UID^+ = UID, Name, City

Example: Find the Keys

UID	Name	Phone	City
234	Fred	206-555-9999	Seattle
234	Fred	206-555-8888	Seattle
987	Joe	415-555-7777	SF
…	…	…	…

UID → Name, City

UID^+ = UID, Name, City

Not a key: missing Phone

Example: Find the Keys

UID	Name	Phone	City
234	Fred	206-555-9999	Seattle
234	Fred	206-555-8888	Seattle
987	Joe	415-555-7777	SF
…	…	…	…

UID → Name, City

UID^+ = UID, Name, City

Not a key: missing Phone

$(UID, Phone)^+$ = **??**

Example: Find the Keys

UID	Name	Phone	City
234	Fred	206-555-9999	Seattle
234	Fred	206-555-8888	Seattle
987	Joe	415-555-7777	SF
…	…	…	…

UID → Name, City

UID^+ = UID, Name, City

Not a key: missing Phone

$(UID, Phone)^+$ = UID, Name, Phone, City

Example: Find the Keys

UID	Name	Phone	City
234	Fred	206-555-9999	Seattle
234	Fred	206-555-8888	Seattle
987	Joe	415-555-7777	SF
…	…	…	…

UID → Name, City

UID^+ = UID, Name, City

Not a key: missing Phone

$(UID, Phone)^+$ = UID, Name, Phone, City

Key

Example: Find the Keys

UID	Name	Phone	City
234	Fred	206-555-9999	Seattle
234	Fred	206-555-8888	Seattle
987	Joe	415-555-7777	SF
…	…	…	…

UID → Name, City

UID^+ = UID, Name, City

Not a key: missing Phone

$(UID, Phone)^+$ = UID, Name, Phone, City

Key

$(UID, Name, Phone)^+$ = **??**

Example: Find the Keys

UID	Name	Phone	City
234	Fred	206-555-9999	Seattle
234	Fred	206-555-8888	Seattle
987	Joe	415-555-7777	SF
…	…	…	…

UID → Name, City

UID^+ = UID, Name, City

Not a key: missing Phone

$(UID, Phone)^+$ = UID, Name, Phone, City

Key

$(UID, Name, Phone)^+$ = UID, Name, Phone, City

Example: Find the Keys

UID	Name	Phone	City
234	Fred	206-555-9999	Seattle
234	Fred	206-555-8888	Seattle
987	Joe	415-555-7777	SF
…	…	…	…

UID → Name, City

UID^+ = UID, Name, City — Not a key: missing Phone

$(UID, Phone)^+$ = UID, Name, Phone, City — Key

$(UID, Name, Phone)^+$ = UID, Name, Phone, City — Super-Key

Example: Find the Keys

UID	Name	Phone	City
234	Fred	206-555-9999	Seattle
234	Fred	206-555-8888	Seattle
987	Joe	415-555-7777	SF
…	…	…	…

UID → Name, City

UID^+ = UID, Name, City — Not a key: missing Phone

$(UID, Phone)^+$ = UID, Name, Phone, City — Key

$(UID, Name, Phone)^+$ = UID, Name, Phone, City — Super-Key

$Phone^+$ = Phone

Example: Find the Keys

UID	Name	Phone	City
234	Fred	206-555-9999	Seattle
234	Fred	206-555-8888	Seattle
987	Joe	415-555-7777	SF
…	…	…	…

UID → Name, City

UID^+ = UID, Name, City — Not a key: missing Phone

$(UID, Phone)^+$ = UID, Name, Phone, City — Key

$(UID, Name, Phone)^+$ = UID, Name, Phone, City — Super-Key

$Phone^+$ = Phone — Not a (Super-)Key

Example: Find the Keys

Compute X^+, for larger and larger sets X, until X^+= [all-attributes]

Name → Color
Category → Dept
Color, Dept → Price

Example: Find the Keys

Compute X^+, for larger and larger sets X, until X^+= [all-attributes]

Name → Color
Category → Dept
Color, Dept → Price

Sets X of size 1

$Name^+$ = Name, Color;
$Color^+$ = Color;

Example: Find the Keys

Compute X^+, for larger and larger sets X, until X^+= [all-attributes]

Name → Color
Category → Dept
Color, Dept → Price

Sets X of size 1

$Name^+$ = Name, Color;
$Color^+$ = Color;
$Category^+$ = Category, Dept;
$Dept^+$ = Dept

Example: Find the Keys

Compute X^+, for larger and larger sets X, until X^+= [all-attributes]

Name → Color
Category → Dept
Color, Dept → Price

$Name^+$ = Name, Color;
$Color^+$ = Color;
$Category^+$ = Category, Dept;
$Dept^+$ = Dept

Sets X of size 1

$(Name, Color)^+$ = Name, Color;

Sets X of size 2

Example: Find the Keys

Compute X^+, for larger and larger sets X, until X^+= [all-attributes]

Name → Color
Category → Dept
Color, Dept → Price

$Name^+$ = Name, Color;
$Color^+$ = Color;

$Category^+$ = Category, Dept;
$Dept^+$ = Dept

Sets X of size 1

$(Name, Color)^+$ = Name, Color;
$(Name, Category)^+$ = Name, Color, Category, Dept, Price;

Sets X of size 2

Example: Find the Keys

Compute X^+, for larger and larger sets X, until X^+= [all-attributes]

Name → Color
Category → Dept
Color, Dept → Price

$Name^+$ = Name, Color;
$Color^+$ = Color;
$Category^+$ = Category, Dept;
$Dept^+$ = Dept

Sets X of size 1

$(Name, Color)^+$ = Name, Color;
$(Name, Category)^+$ = Name, Color, Category, Dept, Price;
// no need to try $(Name, Category, Dept)^+$ why?

Sets X of size 2

Example: Find the Keys

Compute X^+, for larger and larger sets X, until X^+= [all-attributes]

Name → Color
Category → Dept
Color, Dept → Price

Sets X of size 1

$Name^+$ = Name, Color;
$Color^+$ = Color;
$Category^+$ = Category, Dept;
$Dept^+$ = Dept

Sets X of size 2

$(Name, Color)^+$ = Name, Color;
$(Name, Category)^+$ = Name, Color, Category, Dept, Price;
// no need to try $(Name, Category, Dept)^+$ why?
$(Name, Dept)^+$ = … … and so on until we find all keys

Example: Find the Keys

Compute X^+, for larger and larger sets X, until X^+= [all-attributes]

Name → Color
Category → Dept
Color, Dept → Price

Sets X of size 1

$Name^+$ = Name, Color;
$Color^+$ = Color;
$Category^+$ = Category, Dept;
$Dept^+$ = Dept

Sets X of size 2

$(Name, Color)^+$ = Name, Color;
$(Name, Category)^+$ = Name, Color, Category, Dept, Price;
// no need to try $(Name, Category, Dept)^+$ why?
$(Name, Dept)^+$ = … … and so on until we find all keys

A quicker way: any key X must contain Name (why?) and Category (why?)

Keys are Not Unique

R(A,B,C)

A → B,C
B → A,C

$A^+ = B^+ = ABC$

A is a key
B is a key

In SQL
we must choose
either A or B
as primary key

Don't confuse with

A,B → C

$A^+ = A$, $B^+ = B$
$(AB)^+=ABC$

AB is a key

Discussion

- Our redundancies come this FD:

  UID → Name, City

- The problem is that UID is not a key.

- Boyce-Codd Normal Form captures this intuition.

- Next: BCNF

BCNF

BCNF

- Fix a relation $R(A_1, \ldots, A_n)$ and a set of FDs

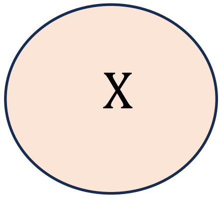
R is in Boyce-Codd Normal Form (BCNF), if every FD $X \to Y$ is either from a superkey X or is trivial: $Y \subseteq X$

- Equivalently: for every set X,
  either $X^+ = X$ or $X^+ = [\text{all}-\text{attributes}]$

Normalization

Algorithm BCNF $R(A_1, \ldots, A_n)$

Find set X s.t. $X \subsetneq X^+ \subsetneq \{A_1, \ldots, A_n\}$

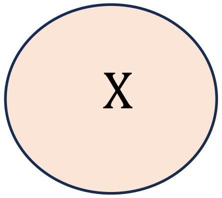


Normalization

Algorithm BCNF $R(A_1, \ldots, A_n)$

Find set X s.t. $X \subsetneq X^+ \subsetneq \{A_1, \ldots, A_n\}$

If not found **then** return $R(A_1, \ldots, A_n)$ // already in BCNF

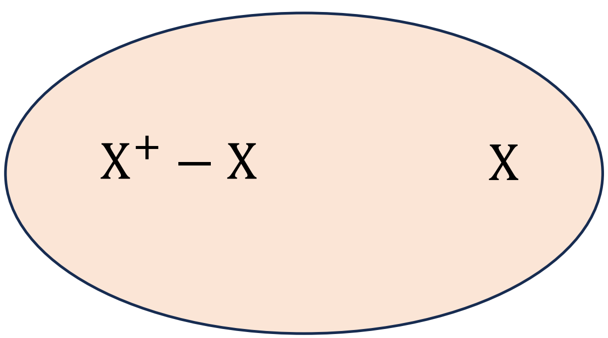


Normalization

Algorithm BCNF $R(A_1, \ldots, A_n)$

Find set X s.t. $X \subsetneq X^+ \subsetneq \{A_1, \ldots, A_n\}$

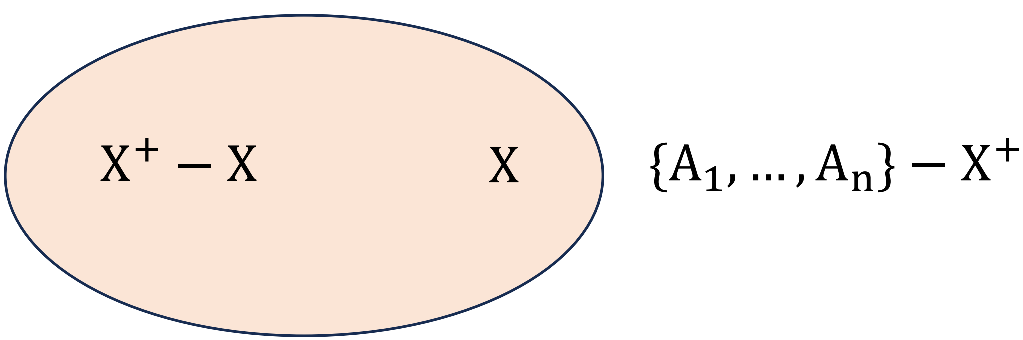
If not found **then** return $R(A_1, \ldots, A_n)$ // already in BCNF



Algorithm BCNF $R(A_1, \ldots, A_n)$

Find set X s.t. $X \subsetneq X^+ \subsetneq \{A_1, \ldots, A_n\}$

If not found **then** return $R(A_1, \ldots, A_n)$ // already in BCNF

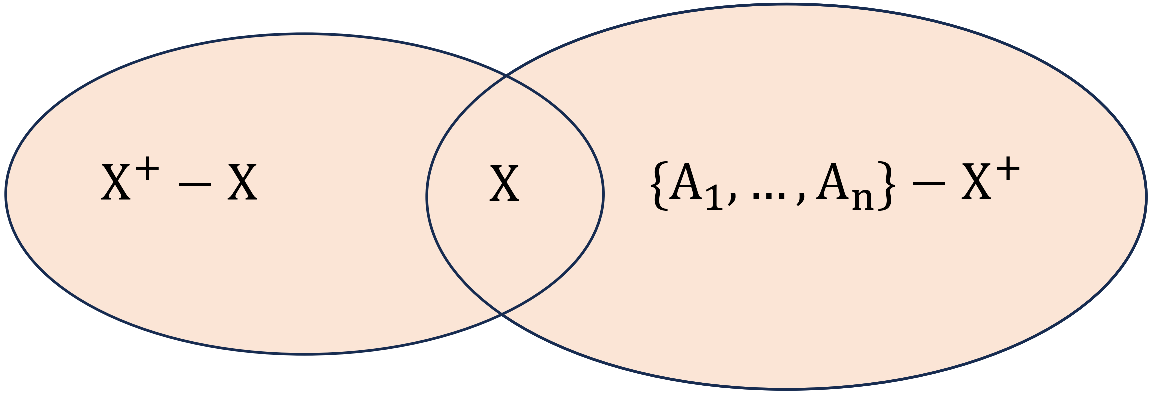


Normalization

Algorithm BCNF $R(A_1, \ldots, A_n)$

Find set X s.t. $X \subsetneq X^+ \subsetneq \{A_1, \ldots, A_n\}$

If not found **then** return $R(A_1, \ldots, A_n)$ // already in BCNF

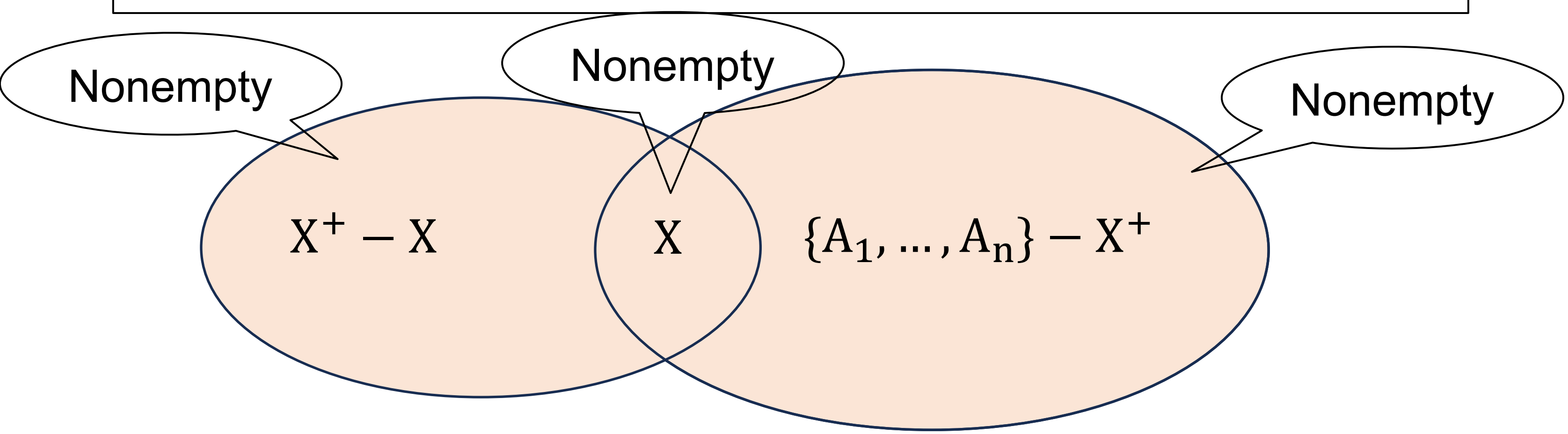


Normalization

Algorithm BCNF $R(A_1, \dots, A_n)$

Find set X s.t. $X \subsetneq X^+ \subsetneq \{A_1, \dots, A_n\}$

If not found **then** return $R(A_1, \dots, A_n)$ // already in BCNF

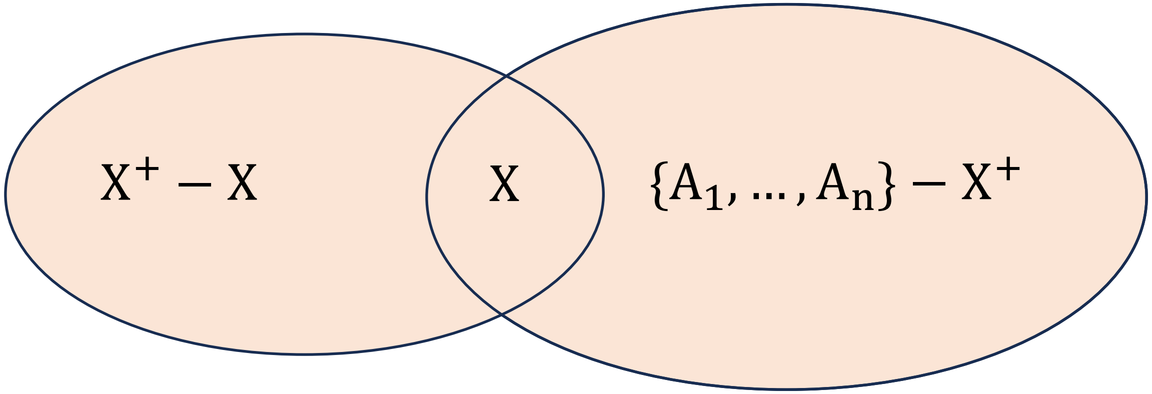


Normalization

Algorithm BCNF $\mathrm{R}(\mathrm{A}_1, \ldots, \mathrm{A}_n)$

Find set X s.t. $\mathrm{X} \subsetneq \mathrm{X}^+ \subsetneq \{\mathrm{A}_1, \ldots, \mathrm{A}_n\}$

If not found **then** return $\mathrm{R}(\mathrm{A}_1, \ldots, \mathrm{A}_n)$ // already in BCNF



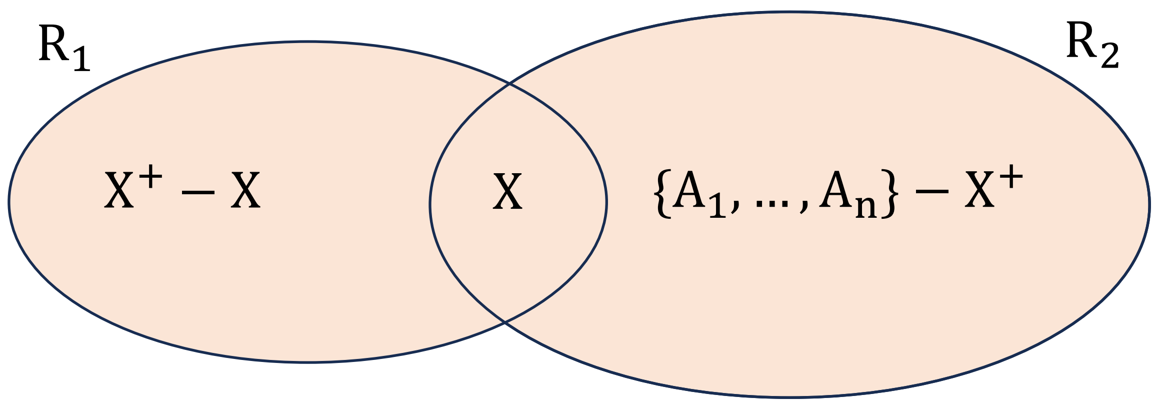
Normalization

Algorithm BCNF $R(A_1, \ldots, A_n)$

Find set X s.t. $X \subsetneq X^+ \subsetneq \{A_1, \ldots, A_n\}$

If not found **then** return $R(A_1, \ldots, A_n)$ // already in BCNF

Decompose: $R(A_1, \ldots, A_n) = R_1(X^+) \bowtie R_2(\{A_1, \ldots, A_n\} - X^+)$



Normalization

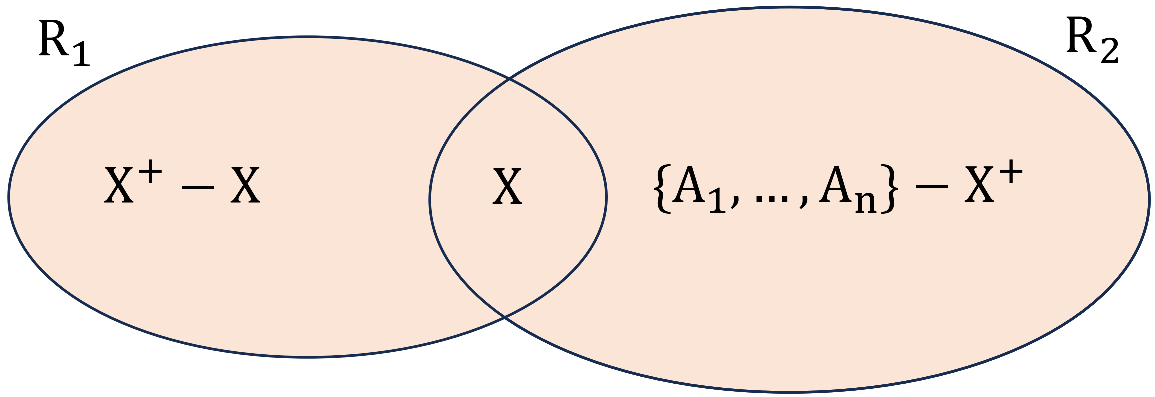
Algorithm BCNF $R(A_1, \ldots, A_n)$

Find set X s.t. $X \subsetneq X^+ \subsetneq \{A_1, \ldots, A_n\}$

If not found **then** return $R(A_1, \ldots, A_n)$ // already in BCNF

Decompose: $R(A_1, \ldots, A_n) = R_1(X^+) \bowtie R_2(\{A_1, \ldots, A_n\} - X^+)$

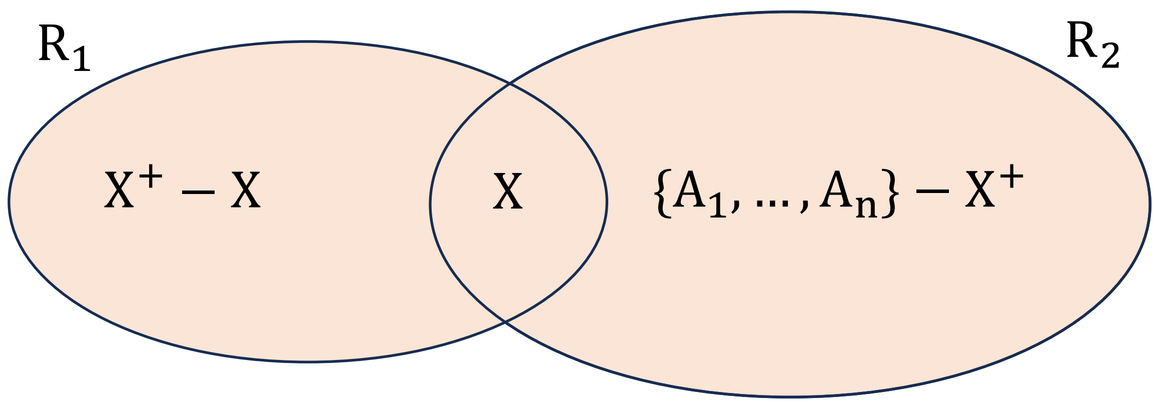
Call recursively BCNF on $R_1(X^+)$
Call recursively BCNF on $R_2(\{A_1, \ldots, A_n\} - X^+)$



Example: Decompose in BCNF

UID	Name	Phone	City
234	Fred	206-555-9999	Seattle
234	Fred	206-555-8888	Seattle
987	Joe	415-555-7777	SF
…	…	…	…

UID → Name, City

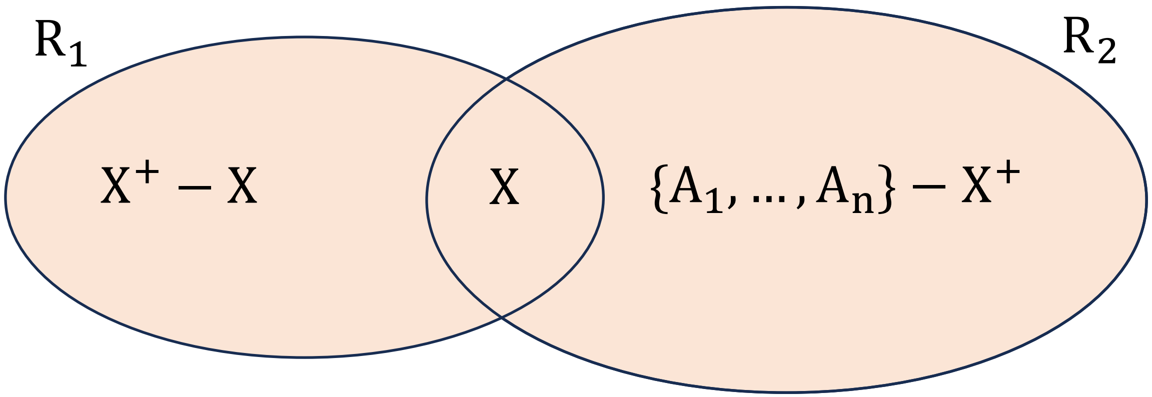


Example: Decompose in BCNF

UID	Name	Phone	City
234	Fred	206-555-9999	Seattle
234	Fred	206-555-8888	Seattle
987	Joe	415-555-7777	SF
…	…	…	…

UID → Name, City

Find set X s.t. $X \subsetneq X^+ \subsetneq \{\text{UID}, \text{Name}, \text{Phone}, \text{City}\}$



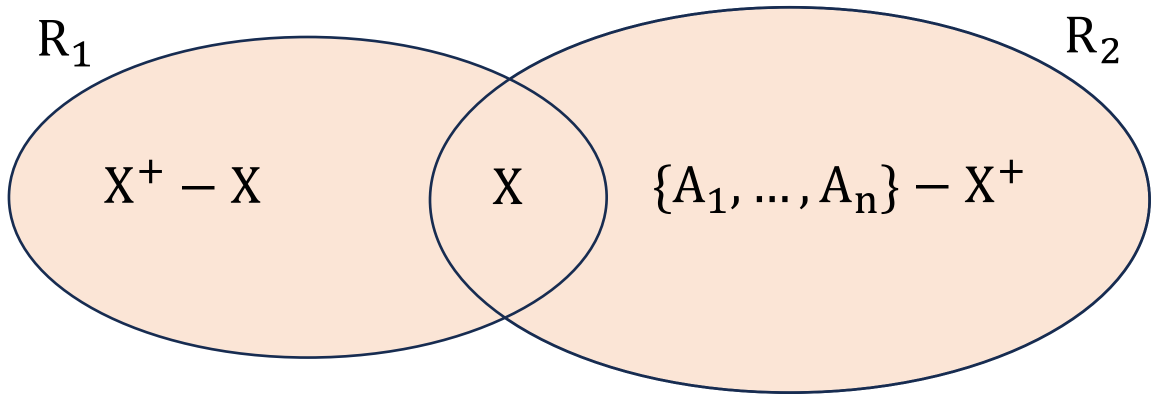
Example: Decompose in BCNF

UID	Name	Phone	City
234	Fred	206-555-9999	Seattle
234	Fred	206-555-8888	Seattle
987	Joe	415-555-7777	SF
…	…	…	…

UID → Name, City

Find set X s.t. $X \subsetneq X^+ \subsetneq \{UID, Name, Phone, City\}$

$X = UID, \quad X^+ = \{UID, Name, City\}$



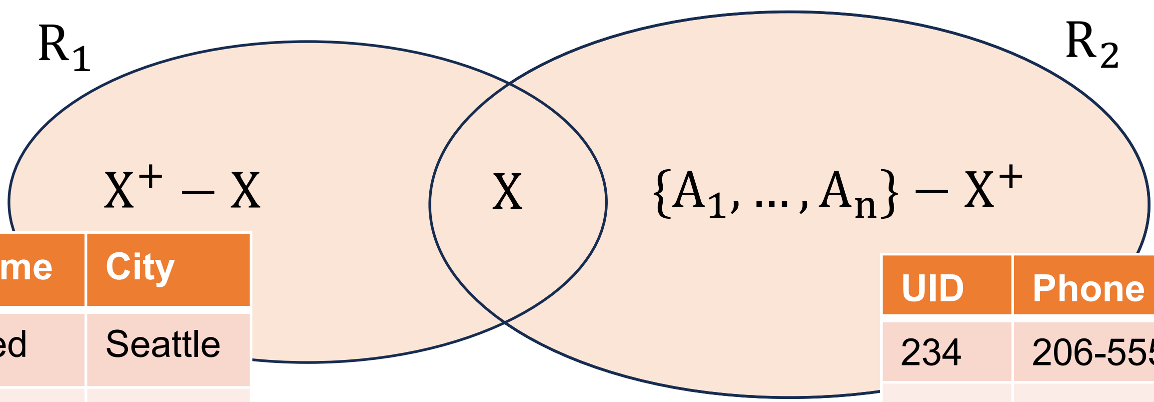
Example: Decompose in BCNF

UID	Name	Phone	City
234	Fred	206-555-9999	Seattle
234	Fred	206-555-8888	Seattle
987	Joe	415-555-7777	SF
…	…	…	…

UID → Name, City

Find set X s.t. $X \subsetneq X^+ \subsetneq \{\mathrm{UID}, \mathrm{Name}, \mathrm{Phone}, \mathrm{City}\}$

$X = \mathrm{UID}, \quad X^+ = \{\mathrm{UID}, \mathrm{Name}, \mathrm{City}\}$



<u>UID</u>	Name	City
234	Fred	Seattle
…		

UID	Phone
234	206-555-9999
…	…

Example: Decompose in BCNF

R(Name, Color, Category, Dept, Price)

Name → Color
Category → Dept
Color, Dept → Price

Example: Decompose in BCNF

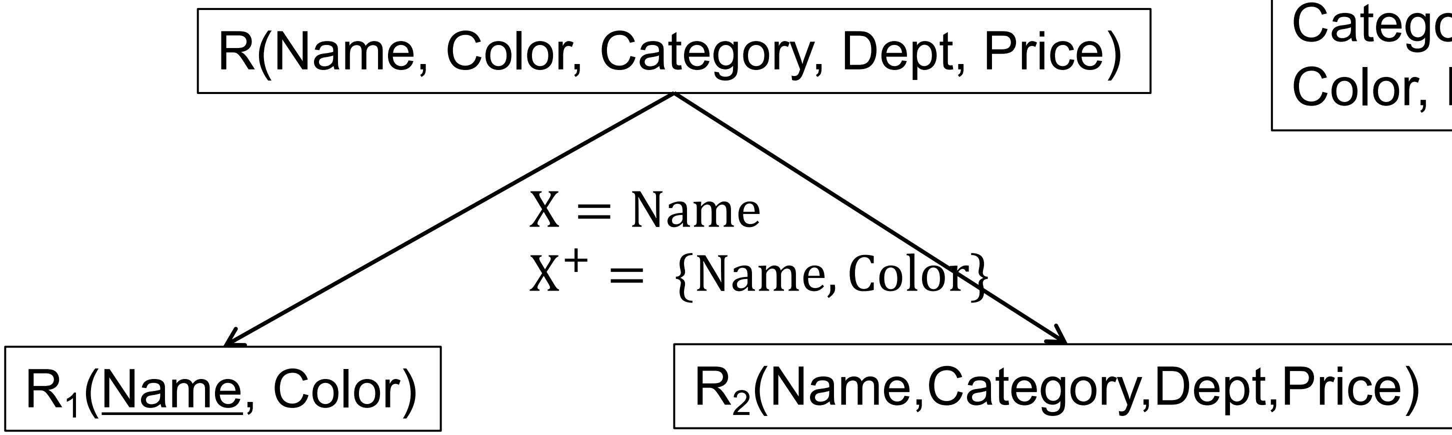
R(Name, Color, Category, Dept, Price)

Name → Color
Category → Dept
Color, Dept → Price

$$X = \text{Name}$$
$$X^{+} = \{\text{Name}, \text{Color}\}$$

Example: Decompose in BCNF

Name → Color
Category → Dept
Color, Dept → Price



Example: Decompose in BCNF

Name → Color
Category → Dept
Color, Dept → Price

R(Name, Color, Category, Dept, Price)

$X = \text{Name}$
$X^+ = \{\text{Name}, \text{Color}\}$

R_1(Name, Color)

R_2(Name,Category,Dept,Price)

BCNF because:
Name+ = Name, Color
Color^+ = Color

Example: Decompose in BCNF

Name → Color
Category → Dept
Color, Dept → Price

R(Name, Color, Category, Dept, Price)

$X = \text{Name}$
$X^{+} = \{\text{Name}, \text{Color}\}$

R_1(<u>Name</u>, Color)

R_2(Name,Category,Dept,Price)

BCNF because:
Name+ = Name, Color
Color^{+} = Color

$X = \text{Category}$
$X^{+} = \{\text{Category}, \text{Dept}\}$

Example: Decompose in BCNF

Name → Color
Category → Dept
Color, Dept → Price

R(Name, Color, Category, Dept, Price)

$X = \text{Name}$
$X^+ = \{\text{Name}, \text{Color}\}$

R_1(Name, Color)

R_2(Name,Category,Dept,Price)

BCNF because:
Name+ = Name, Color
Color^+ = Color

$X = \text{Category}$
$X^+ = \{\text{Category}, \text{Dept}\}$

R_3(Category, Dept)

R_4(Name, Category, Price)

BCNF

Example: Decompose in BCNF

Name → Color
Category → Dept
Color, Dept → Price

R(Name, Color, Category, Dept, Price)

$X = \text{Name}$

$X^+ = \{\text{Name}, \text{Color}\}$

R_1(Name, Color)

R_2(Name,Category,Dept,Price)

BCNF because:
Name+ = Name, Color
Color^+ = Color

$X = \text{Category}$

$X^+ = \{\text{Category}, \text{Dept}\}$

R_3(Category, Dept)

R_4(Name, Category, Price)

BCNF

BCNF
Name,Category is a key because:
$(\text{Name,Category})^+$=…,Price

Example: Decompose in BCNF

Name → Color
Category → Dept
Color, Dept → Price

R(Name, Color, Category, Dept, Price)

$X = \text{Name}$
$X^+ = \{\text{Name}, \text{Color}\}$

R_1(<u>Name</u>, Color)

R_2(Name,Category,Dept,Price)

BCNF because:
Name+ = Name, Color
Color^+ = Color

$X = \text{Category}$
$X^+ = \{\text{Category}, \text{Dept}\}$

R_3(<u>Category</u>, Dept)

BCNF

R_4(<u>Name, Category</u>, Price)

BCNF
Name,Category is a key because:
$(\text{Name},\text{Category})^+$=…,Price

Final Decomposition

Example: Decompose in BCNF

Name → Color
Category → Dept
Color, Dept → Price

We lost the FD
Color, Dept → Price

R(Name, Color, Category, Dept, Price)

$X = \text{Name}$
$X^+ = \{\text{Name}, \text{Color}\}$

R_1(Name, Color)

BCNF because:
Name+ = Name, Color
Color^+ = Color

R_2(Name,Category,Dept,Price)

$X = \text{Category}$
$X^+ = \{\text{Category}, \text{Dept}\}$

R_3(Category, Dept)

BCNF

R_4(Name, Category, Price)

BCNF
Name,Category is a key because:
$(\text{Name,Category})^+$=…,Price

Final Decomposition

Example: Decompose in BCNF

R(Name, Color, Category, Dept, Price)

Name → Color
Category → Dept
Color, Dept → Price

Decomposition is not unique

Example: Decompose in BCNF

Name → Color
Category → Dept
Color, Dept → Price

R(Name, Color, Category, Dept, Price)

$$X = \text{Color, Dept}$$
$$X^{+} = \{\text{Color, Dept, Price}\}$$

Decomposition is not unique

Example: Decompose in BCNF

Name → Color
Category → Dept
Color, Dept → Price

R(Name, Color, Category, Dept, Price)

$X = \text{Color}, \text{Dept}$

$X^{+} = \{\text{Color}, \text{Dept}, \text{Price}\}$

R_1(<u>Color, Dept</u>, Price)

R_2(Name, Category, Color, Dept)

BCNF

Decomposition is not unique

Example: Decompose in BCNF

Name → Color
Category → Dept
Color, Dept → Price

R(Name, Color, Category, Dept, Price)

$X = \text{Color, Dept}$
$X^+ = \{\text{Color, Dept, Price}\}$

R_1(<u>Color, Dept</u>, Price)

BCNF

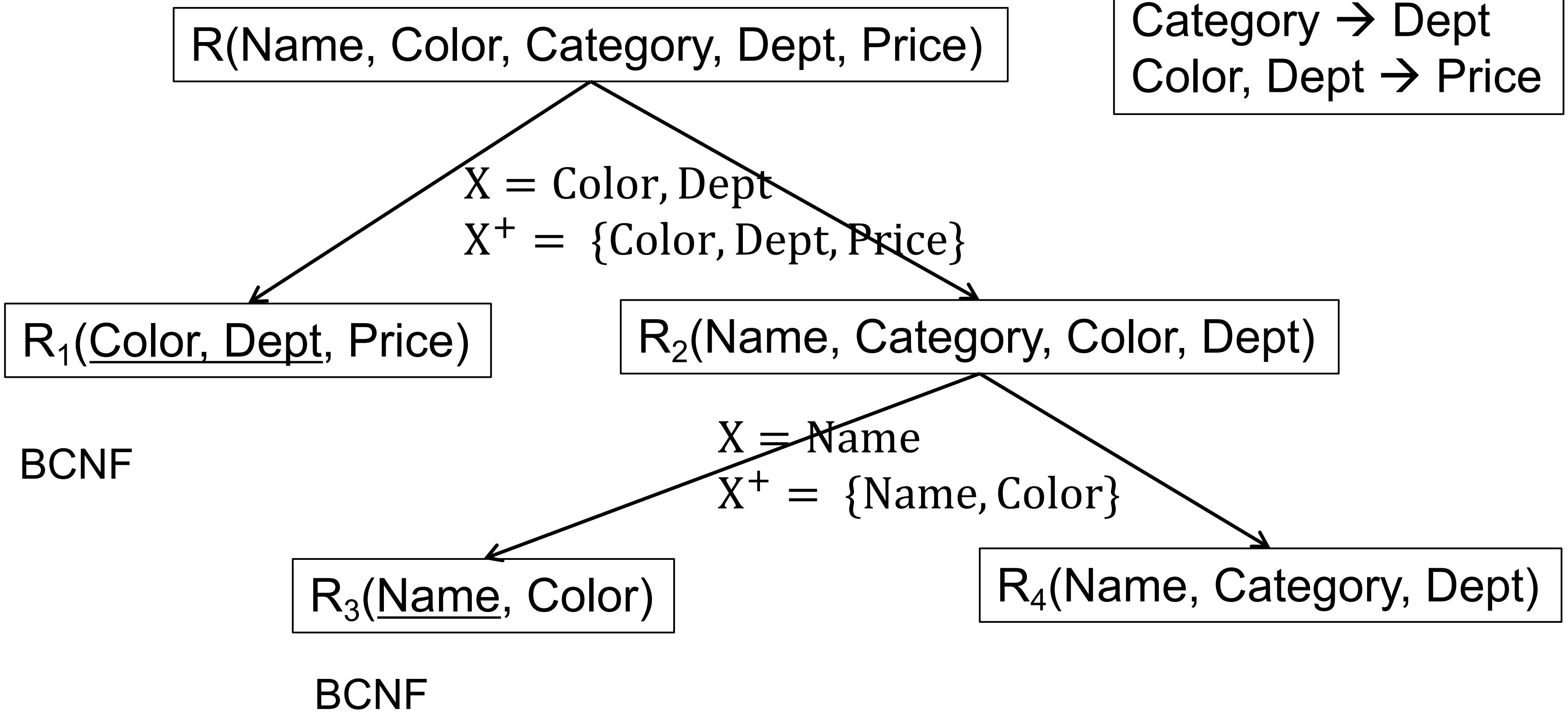
R_2(Name, Category, Color, Dept)

$X = \text{Name}$
$X^+ = \{\text{Name, Color}\}$

Decomposition is not unique

Example: Decompose in BCNF

Name → Color
Category → Dept
Color, Dept → Price



Decomposition is not unique

Example: Decompose in BCNF

Name → Color
Category → Dept
Color, Dept → Price

R(Name, Color, Category, Dept, Price)

$X = \text{Color, Dept}$
$X^+ = \{\text{Color, Dept, Price}\}$

R_1(<u>Color, Dept</u>, Price)

BCNF

R_2(Name, Category, Color, Dept)

$X = \text{Name}$
$X^+ = \{\text{Name, Color}\}$

R_3(<u>Name</u>, Color)

BCNF

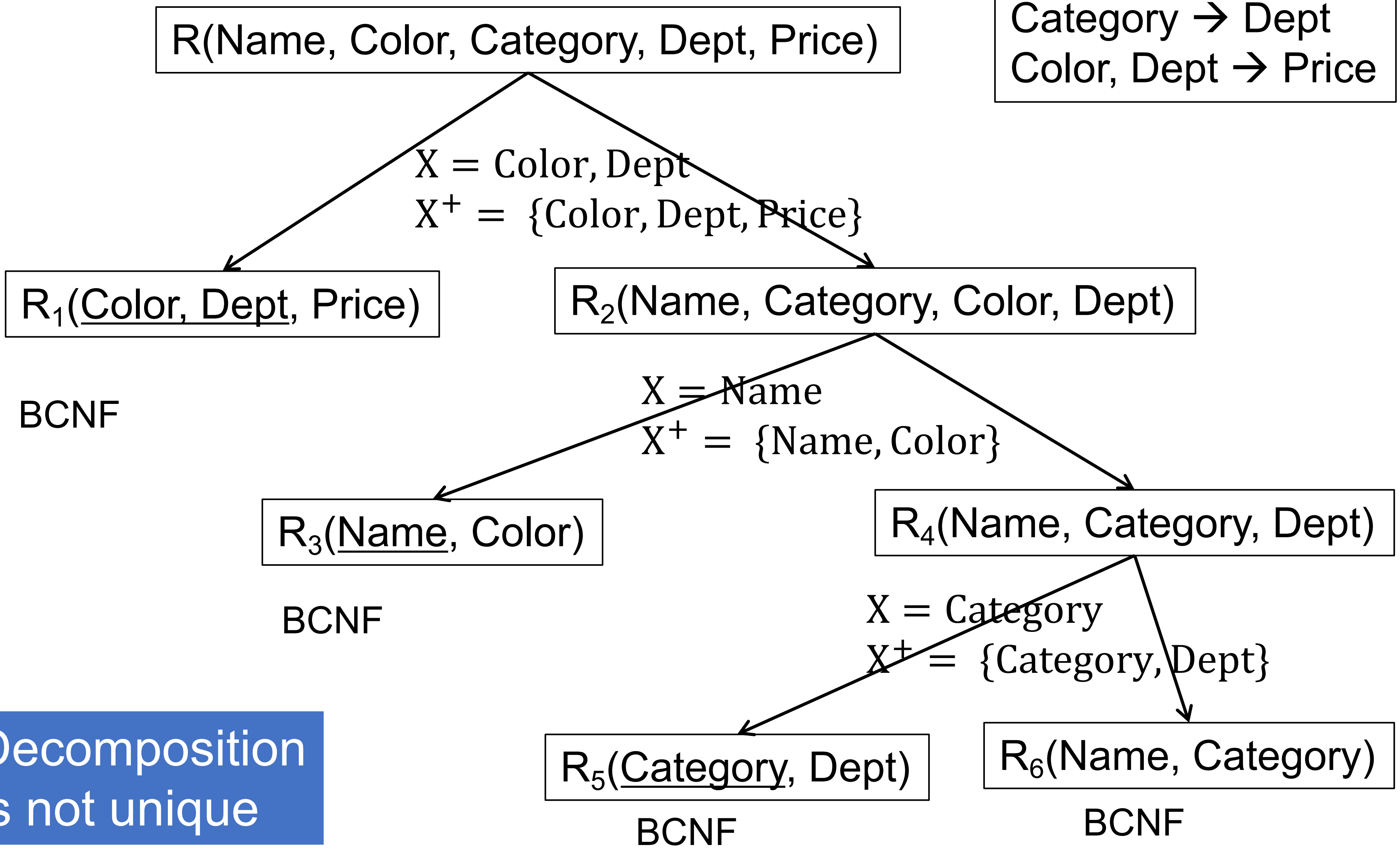
R_4(Name, Category, Dept)

$X = \text{Category}$
$X^+ = \{\text{Category, Dept}\}$

Decomposition is not unique

Example: Decompose in BCNF

Name → Color
Category → Dept
Color, Dept → Price



Decomposition is not unique

Example: Decompose in BCNF

Name → Color
Category → Dept
Color, Dept → Price

All FDs are recovered

R(Name, Color, Category, Dept, Price)

$X = \text{Color, Dept}$
$X^+ = \{\text{Color, Dept, Price}\}$

R_1(<u>Color, Dept</u>, Price) — BCNF

R_2(Name, Category, Color, Dept)

$X = \text{Name}$
$X^+ = \{\text{Name, Color}\}$

R_3(<u>Name</u>, Color) — BCNF

R_4(Name, Category, Dept)

$X = \text{Category}$
$X^+ = \{\text{Category, Dept}\}$

R_5(<u>Category</u>, Dept) — BCNF

R_6(Name, Category) — BCNF

Final decomposition

Decomposition is not unique

Discussion

- The BCNF decomposition eliminates all anomalies
- In general, we may not be able to recover all FDs
- The 3rd Normal Form is another kind of decomposition, which recovers all FDs, but does not eliminate all anomalies
- We won’t discuss 3NF: it is very similar to BCNF but a lot more complicated