CS 564: Database Management Systems

Lecture 9: Normalization I

Xiangyao Yu

2/12/2024

Module A2: Database Design

ER Model

Functional Dependency

Normalization I

Normalization II

Outline of this Lecture

The closure algorithm

Decomposition

Lossless-join decomposition

Boyce-Codd normal form (BCNF)

Normalization

Closure of Attribute Sets

Attribute Closure

If X is an attribute set, the closure X^+ is the set of all attributes B such that:

$$X \longrightarrow B$$

In other words, X^+ includes all attributes that are functionally determined from X

Example

Product(name, category, color, department, price)

- $name \longrightarrow color$
- $category \longrightarrow department$
- $color, category \longrightarrow price$

Attribute Closure:

- $\{name\}^+ =$
- $\{name, category\}^+ =$

Example

Product(name, category, color, department, price)

- $name \longrightarrow color$
- $category \longrightarrow department$
- $color, category \longrightarrow price$

Attribute Closure:

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

Calculate Attribute Closure

Let $X = \{A_1, A_2, \ldots, A_n\}$

UNTIL *X* doesn’t change **REPEAT**:

IF $B_1, B_2, \ldots, B_m \longrightarrow C$ is an FD **AND** $B_1, B_2, \ldots, B_m$ are all in *X*

THEN add *C* to *X*

Output *X*

Attribute Closure – Example

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

- $A, B \longrightarrow C$
- $A, D \longrightarrow E$
- $B \longrightarrow D$
- $A, F \longrightarrow B$

Compute the attribute closures:

- $\{A, B\}^{+} =$
- $\{A, F\}^{+} =$

Attribute Closure – Example

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

- $A, B \longrightarrow C$
- $A, D \longrightarrow E$
- $B \longrightarrow D$
- $A, F \longrightarrow B$

Compute the attribute closures:

- $\{A, B\}^+ = \{A, B, C, D, E\}$
- $\{A, F\}^+ =$

Attribute Closure – Example

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

- $A, B \longrightarrow C$
- $A, D \longrightarrow E$
- $B \longrightarrow D$
- $A, F \longrightarrow B$

Compute the attribute closures:

- $\{A, B\}^{+} = \{A, B, C, D, E\}$
- $\{A, F\}^{+} = \{A, F, B, D, E, C\}$

FD Closure

> **FD Closure**
> If F is a set of FDs, the closure F^+ is the set of all FDs logically implied by F

Armstrong's axioms are:

- **Sound**: any FD generated by an axiom belongs in F^+
- **Complete**: repeated application of the axioms will generate all FDs in F^+

To compute the closure F^+ of FDs

- For each subset of attributes X, compute X^+
- For each subset of attributes $Y \subseteq X^+$, output the FD $X \longrightarrow Y$

Computing Keys and Superkeys

Compute X^+ for all sets of attributes X

If $X^+ = all\ attributes$, then X is a superkey

If no subset of X is a superkey, then X is also a key

Outline of this Lecture

The closure algorithm

Decomposition

Lossless-join decomposition

Boyce-Codd normal form (BCNF)

Normalization

Schema Decomposition

We decompose a relation **R**(A_1, …, A_n) by creating

$\mathbf{R_1}(B_1, .., B_m)$

$\mathbf{R_2}(C_1, \ldots, C_k)$

where $\{B_1, \ldots, B_m\} \cup \{C_1, \ldots, C_k\} = \{A_1, \ldots A_n\}$

The instance of $\mathbf{R_1}$ is the projection of **R** onto $B_1, .., B_m$

The instance of $\mathbf{R_2}$ is the projection of **R** onto $C_1, .., C_l$

In general we can decompose a relation into multiple relations.

Schema Decomposition – Example

R:

<u>**SSN**</u>	**name**	**rating**	**hourly_wages**	**hours_worked**
123-22-3666	Attishoo	8	10	40
231-31-5368	Smiley	8	10	30
131-24-3650	Smethurst	5	7	30
434-26-3751	Guldu	5	7	32
612-67-4134	Madayan	8	10	40

R1:

<u>**SSN**</u>	**name**	**rating**	**hours_worked**
123-22-3666	Attishoo	8	40
231-31-5368	Smiley	8	30
131-24-3650	Smethurst	5	30
434-26-3751	Guldu	5	32
612-67-4134	Madayan	8	40

R2:

rating	**hourly_wages**
8	10
5	7

Properties of Decompositions

What should a good decomposition achieve?

1. Minimize redundancy
2. Avoid information loss (This lecture)
   – Lossless-join
3. Preserve the FDs (Next lecture)
   – Dependency preserving
4. Ensure good query performance

Lossy Decomposition – Example 1

name	age	phoneNumber
Paris	24	608-374-8422
John	24	608-321-1163
Arun	20	206-473-8221

Decompose into:

R_1(name, age)

R_2(age, phoneNumber)

name	age
Paris	24
John	24
Arun	20

age	phoneNumber
24	608-374-8422
24	608-321-1163
20	206-473-8221

We can't figure out which phoneNumber corresponds to which person!

Outline of this Lecture

The closure algorithm

Decomposition

Lossless-join decomposition

Boyce-Codd normal form (BCNF)

Normalization

Lossless-Join Decomposition

$$\begin{tikzcd} & \mathbf{R}(A, B, C) \arrow[dl] \arrow[dr] & \\ \mathbf{R_1}(A, B) \arrow[dr] & & \mathbf{R_2}(B, C) \arrow[dl] \\ & \mathbf{R'}(A, B, C) & \end{tikzcd}$$

decompose (projection)

recover (natural join)

A natural join is a join on the same attribute names

A schema decomposition is **lossless-join** if **R = R’** for any initial instance **R**

Lossy Decomposition – Example 2

R

S	P	D
s1	p1	d1
s2	p2	d2
s3	p1	d3

Decompose

Lossy Decomposition – Example 2

R

S	P	D
s1	p1	d1
s2	p2	d2
s3	p1	d3

Decompose

R1

S	P
s1	p1
s2	p2
s3	p1

R2

P	D
p1	d1
p2	d2
p1	d3

Lossy Decomposition – Example 2

R

S	P	D
s1	p1	d1
s2	p2	d2
s3	p1	d3

Decompose

R1

S	P
s1	p1
s2	p2
s3	p1

R2

P	D
p1	d1
p2	d2
p1	d3

Natural Join

R1 Joins R2

S	P	D
s1	p1	d1
s2	p2	d2
s3	p1	d3
s1	p1	d3
s3	p1	d1

R1 Joins R2 ≠ R

Lossless-Join Decomposition – Example

R:

SSN	**name**	**rating**	**hourly_wages**	**hours_worked**
123-22-3666	Attishoo	8	10	40
231-31-5368	Smiley	8	10	30
131-24-3650	Smethurst	5	7	30
434-26-3751	Guldu	5	7	32
612-67-4134	Madayan	8	10	40

R1:

SSN	**name**	**rating**	**hours_worked**
123-22-3666	Attishoo	8	40
231-31-5368	Smiley	8	30
131-24-3650	Smethurst	5	30
434-26-3751	Guldu	5	32
612-67-4134	Madayan	8	40

R2:

rating	**hourly_wages**
8	10
5	7

R1 Joins R2 = R

Test for Lossless Join

Theorem

Let R be a relation and F be a sets of FDs that hold over R. The decomposition of R into relations with attribute sets $R1$ and $R2$ is lossless **if and only if** F^+ contains either the FD $R1 \cap R2 \longrightarrow R1$ or the FD $R1 \cap R2 \longrightarrow R2$.

The attributes common to R1 and R2 must contain a key for either R1 or R2

Test for Lossless Join

Theorem

Let R be a relation and F be a sets of FDs that hold over R. The decomposition of R into relations with attribute sets $R1$ and $R2$ is lossless **if and only if** F^+ contains either the FD $R1 \cap R2 \longrightarrow R1$ or the FD $R1 \cap R2 \longrightarrow R2$.

The attributes common to R1 and R2 must contain a key for either R1 or R2

If an FD $X \longrightarrow Y$ holds over a relation R and $X \cap Y$ is empty, the decomposition of R into $R - Y$ and XY is lossless

Lossless Join

If an FD $X \longrightarrow Y$ holds over a relation R and $X \cap Y$ is empty, the decomposition of R into $R - Y$ and XY is lossless

R:

SSN	name	rating	hourly_wages	hours_worked
123-22-3666	Attishoo	8	10	40
231-31-5368	Smiley	8	10	30
131-24-3650	Smethurst	5	7	30
434-26-3751	Guldu	5	7	32
612-67-4134	Madayan	8	10	40

R (SSN, name, rating, hourly_wages, hours_worked)

- rating $\longrightarrow$ hourly_wages
- **R1**(SSN, name, rating, hous_worked)
- **R2**(rating, hourly_wages)

Lossless-Join Decomposition – Exercise

R(A, B, C, D)

- FD: A, B $\longrightarrow$ C

Theorem

Let *R* be a relation and *F* be a sets of FDs that hold over *R*. The decomposition of *R* into relations with attribute sets *R1* and *R2* is lossless **if and only if** F^+ contains either the FD $R1 \cap R2 \longrightarrow R1$ or the FD $R1 \cap R2 \longrightarrow R2$.

Are the following decompositions lossless?

- R1(A, B, C), R2(D)
- R1(A, B, D), R2(B, C)
- R1(A, B, D), R2(A, B, C)
- R1(A, B, C), R2(B, C, D)

Lossless-Join Decomposition – Exercise

R(A, B, C, D)

- FD: A, B $\longrightarrow$ C

Theorem

Let *R* be a relation and *F* be a sets of FDs that hold over *R*. The decomposition of *R* into relations with attribute sets *R1* and *R2* is lossless **if and only if** F^+ contains either the FD $R1 \cap R2 \longrightarrow R1$ or the FD $R1 \cap R2 \longrightarrow R2$.

Are the following decompositions lossless?

- R1(A, B, C), R2(D) No
- R1(A, B, D), R2(B, C) No
- R1(A, B, D), R2(A, B, C) Yes
- R1(A, B, C), R2(B, C, D) No

Repeated Decomposition

R(A, B, C, D)

- FD1: $A \longrightarrow B$
- FD2: $C \longrightarrow D$

Repeated Decomposition

R(A, B, C, D)

- FD1: A $\longrightarrow$ B
- FD2: C $\longrightarrow$ D

Decompose **R** into **R1**(A, C, D) and **R2**(A, B)

Repeated Decomposition

R(A, B, C, D)

- FD1: $A \longrightarrow B$
- FD2: $C \longrightarrow D$

Decompose **R** into **R1**(A, C, D) and **R2**(A, B)

Decompose **R1** into **R11**(A, C) and **R12**(C, D)

R1 = **R11** joins **R12**

R = **R1** joins **R2**

Test for Lossless Join (Multiple Relations)

If a table is decomposed into more than two tables, how to test whether it is lossless?

Solution 1: Identify repeated lossless-join decompositions

Solution 2: Chase test

Chase Test Example

Relation $R(A,B,C,D,E)$

FDs: $\{AB \rightarrow C,\ BC \rightarrow D,\ AD \rightarrow E\}$.

Is the decomposition $\{ABC, BCD, ADE\}$ a lossless-join decomposition?

Chase Test Example

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

FDs: {AB → C, BC → D, AD → E}.

Is the decomposition {ABC, BCD, ADE} a lossless-join decomposition?

Solution 1:

{R1=ABCD, R2=ADE} is a lossless-join decomposition of R

Chase Test Example

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

FDs: {$AB \rightarrow C$, $BC \rightarrow D$, $AD \rightarrow E$}.

Is the decomposition {ABC, BCD, ADE} a lossless-join decomposition?

Solution 1:

{R1=ABCD, R2=ADE} is a lossless-join decomposition of R

{ABC, BCD} is a lossless-join decomposition of R1=ABCD

Therefore, {ABC, BCD, ADE} is a lossless-join decomposition of R

Chase Test Example

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

FDs: {AB → C, BC → D, AD → E}.

Is the decomposition {ABC, BCD, ADE} a lossless-join decomposition?

A	B	C	D	E
a	b	c	d1	e1
a2	b	c	d	e2
a	b3	c3	d	e

Construct a tableau; insert one row for each table.

- Use distinguished variable (a,b,c,...) if the attribute is in the table
- Otherwise use a non-distinguished symbol (e1, e2, b3,...)

Chase Test Example

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

FDs: {AB → C, **BC → D**, AD → E}.

Is the decomposition {ABC, BCD, ADE} a lossless-join decomposition?

A	B	C	D	E
a	b	c	~~**d1**~~ **d**	e1
a2	b	c	d	e2
a	b3	c3	d	e

Chase the tableau by applying FDs

- Since first two rows agree on **B and C**, they must agree on **D** as well

Chase Test Example

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

FDs: {AB → C, BC → D, **AD → E**}.

Is the decomposition {ABC, BCD, ADE} a lossless-join decomposition?

A	B	C	D	E
a	b	c	d	**~~e1~~ e**
a2	b	c	d	e2
a	b3	c3	d	e

Chase the tableau by applying FDs

– Since 1st and 3rd rows agree on **A and D**, they must agree on **E** as well

Chase Test Example

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

FDs: {$AB \rightarrow C$, $BC \rightarrow D$, $AD \rightarrow E$}.

Is the decomposition {ABC, BCD, ADE} a lossless-join decomposition?

A	B	C	D	E
a	**b**	**c**	**d**	**e**
a2	b	c	d	e2
a	b3	c3	d	e

Row 1 contains only distinguished symbols, hence the decomposition is lossless

[1] Alfred V. Aho, Catriel Beeri, and Jeffrey D. Ullman: "The Theory of Joins in Relational Databases", ACM Trans. Datab. Syst. 4(3):297-314, 1979.
[2] David Maier, Alberto O. Mendelzon, and Yehoshua Sagiv: "Testing Implications of Data Dependencies". ACM Trans. Datab. Syst. 4(4):455-469, 1979

Chase Test – Exercise

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

FDs: {$AB \rightarrow C$, $BC \rightarrow D$, $AD \rightarrow E$}.

Is the decomposition {ABC, BCD, A~~D~~E} a lossless-join decomposition?

A	B	C	D	E

Outline of this Lecture

The closure algorithm

Decomposition

Lossless-join decomposition

Boyce-Codd normal form (BCNF)

Normalization

Boyce-Codd Normal Form (BCNF)

A relation **R** is in **BCNF** if whenever $X \longrightarrow B$ is a non-trivial FD, then X is a superkey in **R**

Equivalent definition: for every attribute set X

- either $X^+ = X$
- or $X^+ = all\ attributes$

The only nontrivial dependencies are those in which a key determines some attributes.

BCNF Example 1

A relation **R** is in **BCNF** if whenever $X \longrightarrow B$ is a non-trivial FD, then X is a superkey in **R**

SSN	name	age	phoneNumber
934729837	Paris	24	608-374-8422
934729837	Paris	24	603-534-8399
123123645	John	30	608-321-1163
384475687	Arun	20	206-473-8221

$$SSN \longrightarrow name, age$$

key = $\{SSN, phoneNumber\}$

Is this relation in BCNF?

BCNF Example 1

A relation **R** is in **BCNF** if whenever $X \longrightarrow B$ is a non-trivial FD, then X is a superkey in **R**

SSN	name	age	phoneNumber
934729837	Paris	24	608-374-8422
934729837	Paris	24	603-534-8399
123123645	John	30	608-321-1163
384475687	Arun	20	206-473-8221

$$SSN \longrightarrow name, age$$

key = $\{SSN, phoneNumber\}$

$SSN \longrightarrow name, age$ is a "bad" FD

The above relation is **not** in BCNF!

BCNF Example 2

A relation **R** is in **BCNF** if whenever $X \longrightarrow B$ is a non-trivial FD, then X is a superkey in **R**

SSN	name	age
934729837	Paris	24
123123645	John	30
384475687	Arun	20

$$SSN \longrightarrow name, age$$

key = $\{SSN\}$

Is this relation in BCNF?

BCNF Example 2

A relation **R** is in **BCNF** if whenever $X \longrightarrow B$ is a non-trivial FD, then X is a superkey in **R**

SSN	name	age
934729837	Paris	24
123123645	John	30
384475687	Arun	20

$$SSN \longrightarrow name, age$$

key = $\{SSN\}$

The above relation is in BCNF!

BCNF Example 3

A relation **R** is in **BCNF** if whenever $X \longrightarrow B$ is a non-trivial FD, then X is a superkey in **R**

SSN	phoneNumber
934729837	608-374-8422
934729837	603-534-8399
123123645	608-321-1163
384475687	206-473-8221

key = $\{SSN, phoneNumber\}$

Is this relation in BCNF?

BCNF Example 3

A relation **R** is in **<u>BCNF</u>** if whenever $X \longrightarrow B$ is a non-trivial FD, then X is a superkey in **R**

SSN	phoneNumber
934729837	608-374-8422
934729837	603-534-8399
123123645	608-321-1163
384475687	206-473-8221

key = $\{SSN, phoneNumber\}$

The above relation is in BCNF!

Outline of this Lecture

The closure algorithm

Decomposition

Lossless-join decomposition

Boyce-Codd normal form (BCNF)

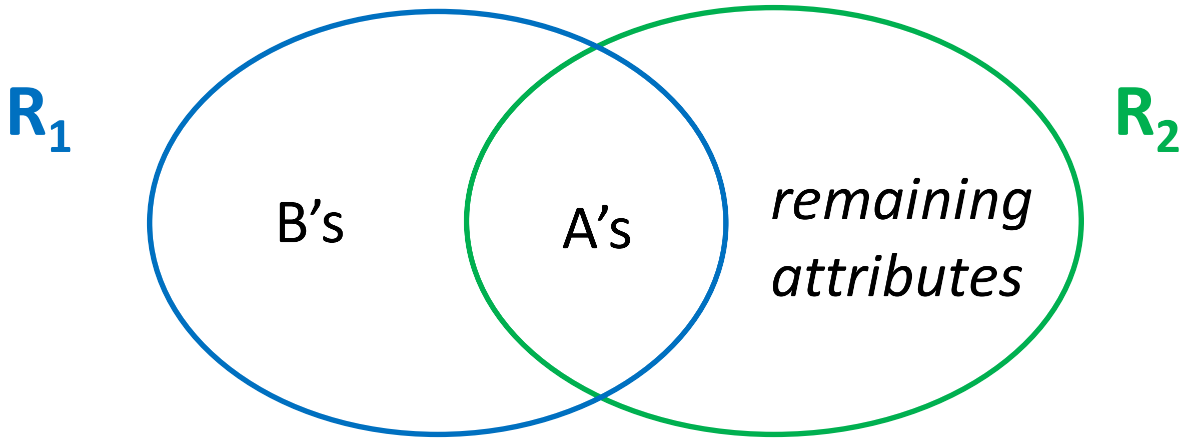
Normalization

Decomposition into BCNF

Find an FD that violates the BCNF condition

$$A_1, A_2, \dots, A_n \longrightarrow B_1, B_2, \dots, B_m$$

Decompose **R** to **R_1** and **R_2**:



Continue until no BCNF violations are left

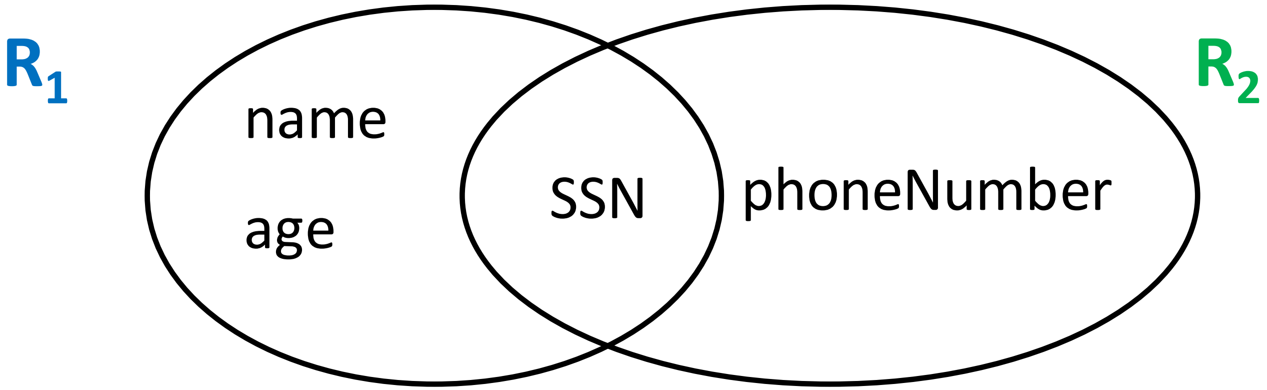
Always possible to obtain a lossless-join decomposition into a collection of BCNF relation schemas

Example 1

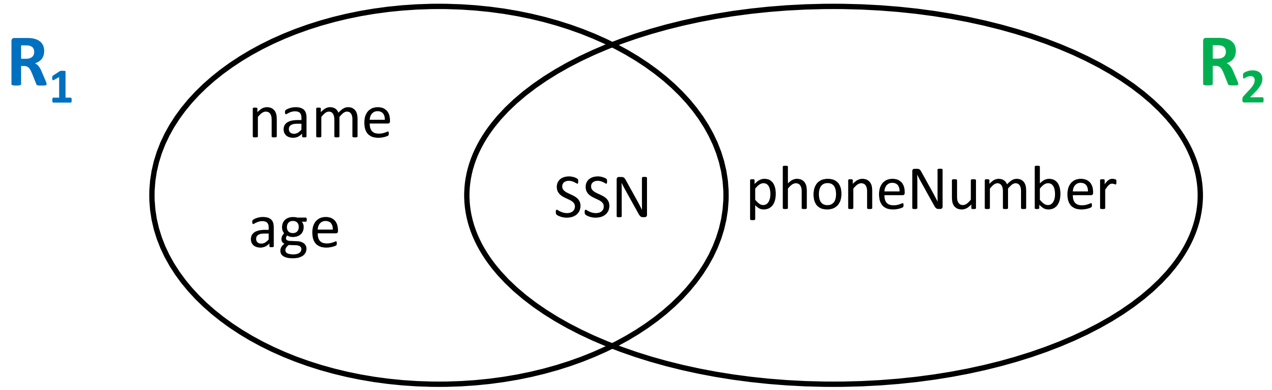
SSN	name	age	phoneNumber
934729837	Paris	24	608-374-8422
934729837	Paris	24	603-534-8399
123123645	John	30	608-321-1163
384475687	Arun	20	206-473-8221

The FD $SSN \longrightarrow name, age$ violates BCNF

Split into two relations R_1, R_2 as follows:



Example 1



$$SSN \longrightarrow name, age$$

SSN	name	age
934729837	Paris	24
123123645	John	30
384475687	Arun	20

SSN	phoneNumber
934729837	608-374-8422
934729837	603-534-8399
123123645	608-321-1163
384475687	206-473-8221

Example 2

Contracts(<u>contractid</u>, supplierid, projectid, deptid, partid, qty, value)

C S J D P Q V

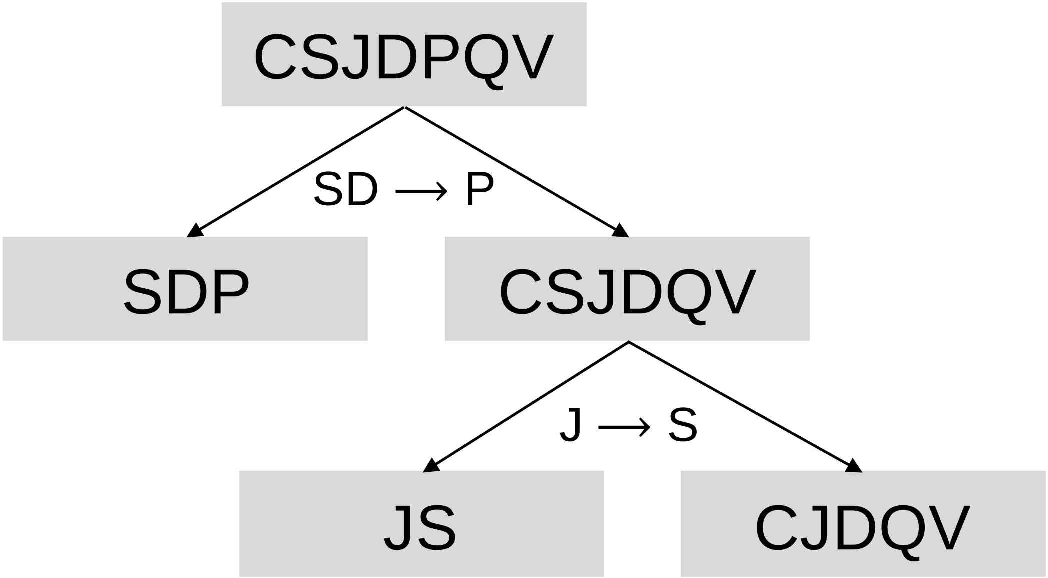
- $C \longrightarrow SJDPQV$ (C is the primary key)
- $J \longrightarrow S$ (each project deals with a single supplier)
- $SD \longrightarrow P$ (a department purchases at most one part from a supplier)

Example 2

Contracts(<u>contractid</u>, supplierid, projectid, deptid, partid, qty, value)

C S J D P Q V

- C ⟶ SJDPQV (C is the primary key)
- J ⟶ S (each project deals with a single supplier)
- SD ⟶ P (a department purchases at most one part from a supplier)

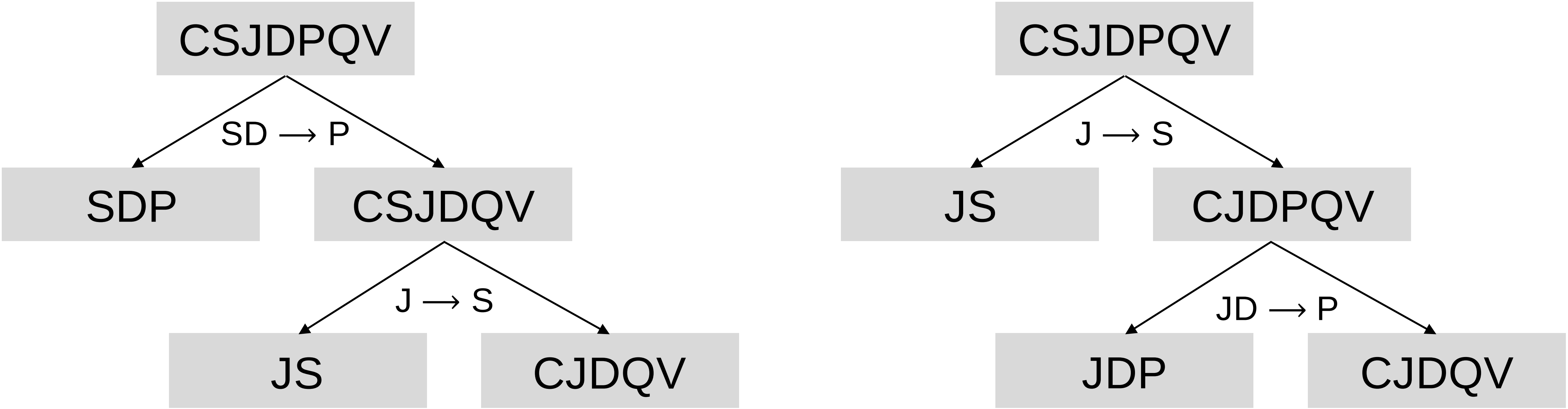


Example 2

Contracts(<u>contractid</u>, supplierid, projectid, deptid, partid, qty, value)

C S J D P Q V

- C $\longrightarrow$ SJDPQV (C is the primary key)
- J $\longrightarrow$ S (each project deals with a single supplier)
- SD $\longrightarrow$ P (a department purchases at most one part from a supplier)



Summary

The closure algorithm

- Attribute closure; FD closure

Decomposition

Lossless-join decomposition

- Chase test

Boyce-Codd normal form (BCNF)

Normalization

- Decompose into BCNF