Kurs Datenbankgrundlagen und Modellierung

Sebastian Maneth, Universität Bremen

maneth@uni-bremen.de

SoSe 2023

07.06.2023

Tutorial 5: Minimal Cover, Third and Boyce-Codd NF

Tutor: Marcel Westenberg

Agenda

1.) Constructing a Minimal Cover

2.) Normalization to Third Normal Form (3NF)

3.) Normalization to Boyce-Codd Normal Form (BCNF)

1.) Constructing a Minimal Cover

Constructing a Minimal Cover

To construct the minimal cover $\mathcal{F}^-$:

1. $\mathcal{F}^- \leftarrow \mathcal{F}$ where all functional dependencies are converted to have only **one attribute on the right side**.
2. **Remove redundant attributes** from the left-hand sides of functional dependencies in $\mathcal{F}^-$:

   foreach $\alpha \rightarrow X \in \mathcal{F}^-$ **do**
       foreach $A \in \alpha$ **do**
           if $X \in (\alpha - A)^+_{\mathcal{F}^-}$ **then** *A redundant in α? Remove it.*
               $\mathcal{F}^- \leftarrow \mathcal{F}^- - \{\alpha \rightarrow X\} \cup \{(\alpha - A) \rightarrow X\}$;

3. **Remove redundant functional dependencies** from $\mathcal{F}^-$:

   foreach $\alpha \rightarrow X \in \mathcal{F}^-$ **do**
       if $(\mathcal{F}^- - \{\alpha \rightarrow X\}) \equiv \mathcal{F}^-$ **then**
           $\mathcal{F}^- \leftarrow \mathcal{F}^- - \{\alpha \rightarrow X\}$;

Constructing a Minimal Cover

✎ **Minimal cover for the following FDs?**

1.) $ABH \rightarrow C$ 2.) $F \rightarrow AD$ 3.) $C \rightarrow E$ 4.) $E \rightarrow F$
5.) $A \rightarrow D$ 6.) $BGH \rightarrow F$ 7.) $BH \rightarrow E$

1. $\mathcal{F}^{-} \leftarrow \mathcal{F}$ where all functional dependencies are converted to have only **one attribute on the right side**.

Constructing a Minimal Cover

✎ **Minimal cover for the following FDs?**

1.) $ABH \rightarrow C$ 2.) $\cancel{F \rightarrow AD}$ 3.) $C \rightarrow E$ 4.) $E \rightarrow F$
5.) $A \rightarrow D$ 6.) $BGH \rightarrow F$ 7.) $BH \rightarrow E$

1 $\mathcal{F}^- \leftarrow \mathcal{F}$ where all functional dependencies are converted to have only **one attribute on the right side**.

F —> A
F —> D

Constructing a Minimal Cover

Minimal cover for the following FDs?

1.) $ABH \rightarrow C$ 2a) F —> A 2b) F —> D 3.) $C \rightarrow E$ 4.) $E \rightarrow F$

5.) $A \rightarrow D$ 6.) $BGH \rightarrow F$ 7.) $BH \rightarrow E$

Minimal cover for the following FDs?

2a) $F \longrightarrow A$

1.) $ABH \rightarrow C$ 2b) $F \longrightarrow D$ 3.) $C \rightarrow E$ 4.) $E \rightarrow F$

5.) $A \rightarrow D$ 6.) $BGH \rightarrow F$ 7.) $BH \rightarrow E$

2 **Remove redundant attributes** from the left-hand sides of functional dependencies in $\mathcal{F}^-$:

foreach $\alpha \rightarrow X \in \mathcal{F}^-$ **do**
  foreach $A \in \alpha$ **do**
    if $X \in (\alpha - A)^+_{\mathcal{F}^-}$ **then** *A redundant in* α*? Remove it.*
      $\mathcal{F}^- \leftarrow \mathcal{F}^- - \{\alpha \rightarrow X\} \cup \{(\alpha - A) \rightarrow X\}$;

✎ **Minimal cover for the following FDs?**

2a) $F \longrightarrow A$

1.) $ABH \rightarrow C$ 2b) $F \longrightarrow D$ 3.) $C \rightarrow E$ 4.) $E \rightarrow F$

5.) $A \rightarrow D$ 6.) $BGH \rightarrow F$ 7.) $BH \rightarrow E$

1.)

a) Is "A" needed?

If we can derive C from BH, then it is **not needed**.

Minimal cover for the following FDs?

1.) $ABH \rightarrow C$ 2a) $F \longrightarrow A$ 2b) $F \longrightarrow D$ 3.) $C \rightarrow E$ 4.) $E \rightarrow F$

5.) $A \rightarrow D$ 6.) $BGH \rightarrow F$ 7.) $BH \rightarrow E$

1.)

a) Is "A" needed?

If we can derive C from BH, then it is **not needed**.

To check this, we compute the attribute closure of BH:

attribute-closure(BH):

BH ===>

Which rules can be applied to BH?

Constructing a Minimal Cover

Minimal cover for the following FDs?

1.) $ABH \rightarrow C$ 2a) $F \longrightarrow A$ 2b) $F \longrightarrow D$ 3.) $C \rightarrow E$ 4.) $E \rightarrow F$

5.) $A \rightarrow D$ 6.) $BGH \rightarrow F$ 7.) $\boxed{BH \rightarrow E}$

1.)

a) Is "A" needed?

If we can derive C from BH, then it is **not needed**.

To check this, we compute the attribute closure of BH:

attribute-closure(BH):

BH ===>

Which rules can be applied to BH?

Only rule: Number 7.)

Minimal cover for the following FDs?

1.) $ABH \to C$ 2a) $F \longrightarrow A$ 2b) $F \longrightarrow D$ 3.) $C \to E$ 4.) $E \to F$

5.) $A \to D$ 6.) $BGH \to F$ 7.) $BH \to E$

1.)

a) Is "A" needed?

If we can derive C from BH, then it is **not needed**.
To check this, we compute the attribute closure of BH:

attribute-closure(BH):

BH ===> BHE
7.)

Apply rule 7.)

✎ **Minimal cover for the following FDs?**

2a) $F \longrightarrow A$

1.) $ABH \rightarrow C$ 2b) $F \longrightarrow D$ 3.) $C \rightarrow E$ 4.) $E \rightarrow F$

5.) $A \rightarrow D$ 6.) $BGH \rightarrow F$ 7.) $BH \rightarrow E$

1.)

a) Is "A" needed?

If we can derive C from BH, then it is **not needed**.
To check this, we compute the attribute closure of BH:

attribute-closure(BH):

BH ===> BHE
7.)

Which rules can be applied to BHE?

Constructing a Minimal Cover

✎ **Minimal cover for the following FDs?**

2a) $F \longrightarrow A$

1.) $ABH \rightarrow C$ 2b) $F \longrightarrow D$ 3.) $C \rightarrow E$ $\boxed{4.)\ E \rightarrow F}$

5.) $A \rightarrow D$ 6.) $BGH \rightarrow F$ 7.) $BH \rightarrow E$

1.)

a) Is "A" needed?

If we can derive C from BH, then it is **not needed**.

To check this, we compute the attribute closure of BH:

attribute-closure(BH):

BH ===> BHE

7.)

Which rules can be applied to BHE?

Rule Nr. 4.)

✎ **Minimal cover for the following FDs?**

1.) $ABH \rightarrow C$ 2a) F —> A 2b) F —> D 3.) $C \rightarrow E$ 4.) $E \rightarrow F$

5.) $A \rightarrow D$ 6.) $BGH \rightarrow F$ 7.) $BH \rightarrow E$

1.)

a) Is "A" needed?

If we can derive C from BH, then it is **not needed**.
To check this, we compute the attribute closure of BH:

attribute-closure(BH):

BH $\overset{7.)}{\Longrightarrow}$ BHE $\overset{4.)}{\Longrightarrow}$ BHEF

Apply rule 4.)

✎ **Minimal cover for the following FDs?**

2a) $F \longrightarrow A$

1.) $\underline{A}BH \rightarrow C$ 2b) $F \longrightarrow D$ 3.) $C \rightarrow E$ 4.) $E \rightarrow F$

5.) $A \rightarrow D$ 6.) $BGH \rightarrow F$ 7.) $BH \rightarrow E$

1.)

a) Is "A" needed?

If we can derive C from BH, then it is **not needed**.

To check this, we compute the attribute closure of BH:

attribute-closure(BH):

BH $\overset{}{\underset{7.)}{===>}}$ BHE $\underset{4.)}{===>}$ BHEF

Which rules can be applied to BHEF?

Minimal cover for the following FDs?

2a) $F \longrightarrow A$

1.) $ABH \rightarrow C$ 2b) $F \longrightarrow D$ 3.) $C \rightarrow E$ 4.) $E \rightarrow F$
5.) $A \rightarrow D$ 6.) $BGH \rightarrow F$ 7.) $BH \rightarrow E$

1.)

a) Is "A" needed?

If we can derive C from BH, then it is **not needed**.
To check this, we compute the attribute closure of BH:

attribute-closure(BH):

BH $\underset{7.)}{===>}$ BHE $\underset{4.)}{===>}$ BHEF

Which rules can be applied to BHEF?
Rule Nr. 2a.)

✎ **Minimal cover for the following FDs?**

2a) $F \longrightarrow A$

1.) $ABH \rightarrow C$ 2b) $F \longrightarrow D$ 3.) $C \rightarrow E$ 4.) $E \rightarrow F$

5.) $A \rightarrow D$ 6.) $BGH \rightarrow F$ 7.) $BH \rightarrow E$

1.)

a) Is "A" needed?

If we can derive C from BH, then it is **not needed**.

To check this, we compute the attribute closure of BH:

attribute-closure(BH):

BH $\xrightarrow{7.)}$ BHE $\xrightarrow{4.)}$ BHEF $\xrightarrow{2a.)}$ BHEFA

Apply rule 2a.)

Minimal cover for the following FDs?

2a) $F \longrightarrow A$

1.) $\underline{A}BH \rightarrow C$ 2b) $F \longrightarrow D$ 3.) $C \rightarrow E$ 4.) $E \rightarrow F$

5.) $A \rightarrow D$ 6.) $BGH \rightarrow F$ 7.) $BH \rightarrow E$

1.)

a) Is "A" needed?

If we can derive C from BH, then it is **not needed**.
To check this, we compute the attribute closure of BH:

attribute-closure(BH):

BH $\underset{7.)}{===>}$ BHE $\underset{4.)}{===>}$ BHEF $\underset{2a.)}{===>}$ BHEFA

Now apply rule 1.)

✎ **Minimal cover for the following FDs?**

2a) $F \longrightarrow A$

1.) $\underline{A}BH \rightarrow C$ 2b) $F \longrightarrow D$ 3.) $C \rightarrow E$ 4.) $E \rightarrow F$

5.) $A \rightarrow D$ 6.) $BGH \rightarrow F$ 7.) $BH \rightarrow E$

1.)

a) Is "A" needed?

If we can derive C from BH, then it is **not needed**.

To check this, we compute the attribute closure of BH:

attribute-closure(BH):

BH ===> BHE ===> BHEF ===> BHEFA ===> C

7.) 4.) 2a.)

Now apply rule 1.)

Minimal cover for the following FDs?

1.) $\cancel{A}\underline{BH} \to C$ 2a) $F \longrightarrow A$ 2b) $F \longrightarrow D$ 3.) $C \to E$ 4.) $E \to F$

5.) $A \to D$ 6.) $BGH \to F$ 7.) $BH \to E$

1.)

a) Is “A” needed? **No!**

b) Is “B” needed?

Minimal cover for the following FDs?

2a) $F \longrightarrow A$

1.) $\cancel{A}\underline{BH} \rightarrow C$ 2b) $F \longrightarrow D$ 3.) $C \rightarrow E$ 4.) $E \rightarrow F$

5.) $A \rightarrow D$ 6.) $BGH \rightarrow F$ 7.) $BH \rightarrow E$

1.)

a) Is “A” needed? **No!**

b) Is “B” needed? **Yes**, because “H” alone cannot derive anything.

Constructing a Minimal Cover

Minimal cover for the following FDs?

1.) $\cancel{A}B\underline{H} \to C$ 2a) $F \longrightarrow A$ 2b) $F \longrightarrow D$ 3.) $C \to E$ 4.) $E \to F$

5.) $A \to D$ 6.) $BGH \to F$ 7.) $BH \to E$

1.)

a) Is “A” needed? **No!**
b) Is “B” needed? **Yes!**
c) Is “H” needed?

Constructing a Minimal Cover

✎ **Minimal cover for the following FDs?**

2a) $F \longrightarrow A$

1.) $\cancel{A}B\underline{H} \rightarrow C$ 2b) $F \longrightarrow D$ 3.) $C \rightarrow E$ 4.) $E \rightarrow F$

5.) $A \rightarrow D$ 6.) $BGH \rightarrow F$ 7.) $BH \rightarrow E$

1.)

a) Is “A” needed? **No!**
b) Is “B” needed? **Yes!**
c) Is “H” needed? Yes, because “B” cannot derive anything.

Constructing a Minimal Cover

Minimal cover for the following FDs?

1.) $\cancel{A}BH \to C$ 2a) $F \longrightarrow A$ 2b) $F \longrightarrow D$ 3.) $C \to E$ 4.) $E \to F$

5.) $A \to D$ 6.) $BGH \to F$ 7.) $BH \to E$

1.)

a) Is “A” needed? **No!**
b) Is “B” needed? **Yes!**
c) Is “H” needed? **Yes!**

Next rule that has more than one attribute on the left?

Minimal cover for the following FDs?

2a) $F \longrightarrow A$

1.) $\cancel{A}BH \rightarrow C$ 2b) $F \longrightarrow D$ 3.) $C \rightarrow E$ 4.) $E \rightarrow F$

5.) $A \rightarrow D$ 6.) $\underline{B}GH \rightarrow F$ 7.) $BH \rightarrow E$

1.)

a) Is “A” needed? **No!**

b) Is “B” needed? **Yes!**

c) Is “H” needed? **Yes!**

Next rule that has more than one attribute on the left?

6.)

Minimal cover for the following FDs?

1.) $\cancel{A}BH \rightarrow C$ 2a) $F \longrightarrow A$ 2b) $F \longrightarrow D$ 3.) $C \rightarrow E$ 4.) $E \rightarrow F$

5.) $A \rightarrow D$ 6.) $\underline{B}GH \rightarrow F$ 7.) $BH \rightarrow E$

1.)

a) Is “A” needed? **No!**
b) Is “B” needed? **Yes!**
c) Is “H” needed? **Yes!**

Next rule that has more than one attribute on the left?

6.)

a) Is “B” needed? **Yes!**

Constructing a Minimal Cover

Minimal cover for the following FDs?

1.) $\cancel{A}BH \rightarrow C$
2a) F —> A
2b) F —> D
3.) $C \rightarrow E$
4.) $E \rightarrow F$
5.) $A \rightarrow D$
6.) $B\underline{G}H \rightarrow F$
7.) $BH \rightarrow E$

1.)

a) Is “A” needed? **No!**
b) Is “B” needed? **Yes!**
c) Is “H” needed? **Yes!**

Next rule that has more than one attribute on the left?

6.)

a) Is “B” needed? **Yes!**
b) Is “G” needed?

Constructing a Minimal Cover

Minimal cover for the following FDs?

1.) $\cancel{A}BH \to C$ 2a) $F \longrightarrow A$ 2b) $F \longrightarrow D$ 3.) $C \to E$ 4.) $E \to F$

5.) $A \to D$ 6.) $B\underline{G}H \to F$ 7.) $BH \to E$

1.)

a) Is "A" needed? **No!**
b) Is "B" needed? **Yes!**
c) Is "H" needed? **Yes!**

Next rule that has more than one attribute on the left?

6.)

a) Is "B" needed? **Yes!**
b) Is "G" needed? **No!** Because $BH \overset{}{\underset{7.)}{===>}} E \underset{4.)}{===>} F$

Constructing a Minimal Cover

Minimal cover for the following FDs?

1.) $\cancel{A}BH \to C$ 2a) $F \longrightarrow A$ 2b) $F \longrightarrow D$ 3.) $C \to E$ 4.) $E \to F$

5.) $A \to D$ 6.) $B\cancel{G}\underline{H} \to F$ 7.) $BH \to E$

1.)

a) Is “A” needed? **No!**
b) Is “B” needed? **Yes!**
c) Is “H” needed? **Yes!**

Next rule that has more than one attribute on the left?

6.)

a) Is “B” needed? **Yes!**
b) Is “G” needed? **No!**
c) Is “H” needed?

Constructing a Minimal Cover

✎ **Minimal cover for the following FDs?**

1.) $\cancel{A}BH \to C$ 2a) $F \longrightarrow A$ 2b) $F \longrightarrow D$ 3.) $C \to E$ 4.) $E \to F$

5.) $A \to D$ 6.) $B\cancel{G}\underline{H} \to F$ 7.) $BH \to E$

1.)

a) Is "A" needed? **No!**

b) Is "B" needed? **Yes!**

c) Is "H" needed? **Yes!**

Next rule that has more than one attribute on the left?

6.)

a) Is "B" needed? **Yes!**

b) Is "G" needed? **No!**

c) Is "H" needed? Yes, because B cannot derive anything.

Constructing a Minimal Cover

Minimal cover for the following FDs?

1.) $\cancel{A}BH \rightarrow C$ 2a) $F \longrightarrow A$ 2b) $F \longrightarrow D$ 3.) $C \rightarrow E$ 4.) $E \rightarrow F$

5.) $A \rightarrow D$ 6.) $B\cancel{G}H \rightarrow F$ 7.) $BH \rightarrow E$

1.)

a) Is "A" needed? **No!**
b) Is "B" needed? **Yes!**
c) Is "H" needed? **Yes!**

Next rule that has more than one attribute on the left?

6.)

a) Is "B" needed? **Yes!**
b) Is "G" needed? **No!**
c) Is "H" needed? **Yes!**

7.)

Neither "H" nor "B" can derive anything (see Rule 1.).
So we are finished with phase 2 of the algorithm.

Constructing a Minimal Cover

✎ **Minimal cover for the following FDs?**

2a) $F \longrightarrow A$

1.) $\cancel{A}BH \rightarrow C$ 2b) $F \longrightarrow D$ 3.) $C \rightarrow E$ 4.) $E \rightarrow F$

5.) $A \rightarrow D$ 6.) $B\cancel{C}H \rightarrow F$ 7.) $BH \rightarrow E$

3 **Remove redundant functional dependencies** from $\mathcal{F}^-$:

foreach $\alpha \rightarrow X \in \mathcal{F}^-$ **do**
  if $(\mathcal{F}^- - \{\alpha \rightarrow X\}) \equiv \mathcal{F}^-$ **then**
    $\mathcal{F}^- \leftarrow \mathcal{F}^- - \{\alpha \rightarrow X\}$;

Minimal cover for the following FDs?

1.) $\cancel{A}BH \rightarrow C$ 2a) $F \longrightarrow A$ 2b) $F \longrightarrow D$ 3.) $C \rightarrow E$ 4.) $E \rightarrow F$

5.) $A \rightarrow D$ 6.) $B\cancel{C}H \rightarrow F$ 7.) $BH \rightarrow E$

1.) Is Rule 1.) needed?
Remove it & check if BH derives C.
If so, then Rule 1.) is **not** needed.

Constructing a Minimal Cover

Minimal cover for the following FDs?

1.) $\cancel{A}BH \rightarrow C$ 2a) $F \rightarrow A$ 2b) $F \rightarrow D$ 3.) $C \rightarrow E$ 4.) $E \rightarrow F$

5.) $A \rightarrow D$ 6.) $B\cancel{C}H \rightarrow F$ 7.) $BH \rightarrow E$

1.) Is Rule 1.) needed?
Remove it & check if BH derives C.
If so, then Rule 1.) is **not** needed.

There is no “C” in the right-hand side of any other rule.
Thus Rule 1.) is needed.

Constructing a Minimal Cover

Minimal cover for the following FDs?

2a) $F \longrightarrow A$

1.) $\cancel{A}BH \rightarrow C$ 2b) $F \longrightarrow D$ 3.) $C \rightarrow E$ 4.) $E \rightarrow F$

5.) $A \rightarrow D$ 6.) $B\cancel{C}H \rightarrow F$ 7.) $BH \rightarrow E$

2a.) Is Rule 2a.) needed?

Constructing a Minimal Cover

Minimal cover for the following FDs?

1.) $\cancel{A}BH \rightarrow C$ 2a) $F \longrightarrow A$ 2b) $F \longrightarrow D$ 3.) $C \rightarrow E$ 4.) $E \rightarrow F$

5.) $A \rightarrow D$ 6.) $B\cancel{C}H \rightarrow F$ 7.) $BH \rightarrow E$

2a.) Is Rule 2a.) needed?

Yes, because A cannot be derived by any other rule.

Constructing a Minimal Cover

✎ **Minimal cover for the following FDs?**

2a) $F \longrightarrow A$

1.) $\cancel{A}BH \rightarrow C$ 2b) $F \longrightarrow D$ 3.) $C \rightarrow E$ 4.) $E \rightarrow F$

5.) $A \rightarrow D$ 6.) $B\cancel{C}H \rightarrow F$ 7.) $BH \rightarrow E$

2b.) Is Rule 2b.) needed?

Minimal cover for the following FDs?

1.) $\cancel{A}BH \to C$ 2a) $F \longrightarrow A$ 2b) $F \longrightarrow D$ 3.) $C \to E$ 4.) $E \to F$

5.) $A \to D$ 6.) $B\cancel{C}H \to F$ 7.) $BH \to E$

2b.) Is Rule 2b.) needed?

No! It is not needed:

F ===> A ===> D

2a.) 5.)

Constructing a Minimal Cover

✎ **Minimal cover for the following FDs?**

1.) $\cancel{A}BH \to C$
2a) $F \longrightarrow A$
~~2b) $F \longrightarrow D$~~
3.) $C \to E$
4.) $E \to F$
5.) $A \to D$
6.) $B\cancel{C}H \to F$
7.) $BH \to E$

3.) Is Rule 3.) needed?

✎ Minimal cover for the following FDs?

2a) $F \longrightarrow A$

1.) $\cancel{A}BH \rightarrow C$ ~~2b) $F \longrightarrow D$~~ 3.) $C \rightarrow E$ 4.) $E \rightarrow F$

5.) $A \rightarrow D$ 6.) $B\cancel{C}H \rightarrow F$ 7.) $BH \rightarrow E$

3.) Is Rule 3.) needed?

Yes. No other rule has “C” on the left.

Constructing a Minimal Cover

✎ **Minimal cover for the following FDs?**

2a) $F \longrightarrow A$

1.) $\cancel{A}BH \to C$ ~~2b) $F \longrightarrow D$~~ 3.) $C \to E$ 4.) $E \to F$

5.) $A \to D$ 6.) $B\cancel{C}H \to F$ 7.) $BH \to E$

4.) Is Rule 4.) needed?

Minimal cover for the following FDs?

2a) $F \longrightarrow A$

1.) $\cancel{A}BH \to C$ ~~2b) $F \longrightarrow D$~~ 3.) $C \to E$ 4.) $E \to F$

5.) $A \to D$ 6.) $B\cancel{C}H \to F$ 7.) $BH \to E$

4.) Is Rule 4.) needed?

Yes. No other rule has "E" on the left.

Constructing a Minimal Cover

Minimal cover for the following FDs?

2a) $F \longrightarrow A$

1.) $\cancel{A}BH \to C$ ~~2b)~~ $\cancel{F \longrightarrow D}$ 3.) $C \to E$ 4.) $E \to F$

5.) $A \to D$ 6.) $B\cancel{C}H \to F$ 7.) $BH \to E$

5.) Is Rule 5.) needed?

Constructing a Minimal Cover

✎ **Minimal cover for the following FDs?**

2a) $F \longrightarrow A$

1.) $\cancel{A}BH \rightarrow C$ ~~2b) $F \longrightarrow D$~~ 3.) $C \rightarrow E$ 4.) $E \rightarrow F$

5.) $A \rightarrow D$ 6.) $B\cancel{C}H \rightarrow F$ 7.) $BH \rightarrow E$

5.) Is Rule 5.) needed?

Yes. No other rule has “A” on the left.

Constructing a Minimal Cover

✎ **Minimal cover for the following FDs?**

2a) $F \longrightarrow A$

1.) $\cancel{A}BH \rightarrow C$ ~~2b) $F \longrightarrow D$~~ 3.) $C \rightarrow E$ 4.) $E \rightarrow F$

5.) $A \rightarrow D$ 6.) $B\cancel{C}H \rightarrow F$ 7.) $BH \rightarrow E$

6.) Is Rule 6.) needed?

Constructing a Minimal Cover

Minimal cover for the following FDs?

2a) $F \longrightarrow A$

1.) $\cancel{A}BH \to C$ ~~2b) $F \longrightarrow D$~~ 3.) $C \to E$ 4.) $E \to F$

5.) $A \to D$ 6.) $B\cancel{C}H \to F$ 7.) $BH \to E$

6.) Is Rule 6.) needed?

No! Because $BH \overset{}{\underset{7.)}{===>}} E \underset{4.)}{===>} F$

✎ Minimal cover for the following FDs?

2a) $F \longrightarrow A$

1.) $\cancel{A}BH \rightarrow C$ ~~2b) $F \longrightarrow D$~~ 3.) $C \rightarrow E$ 4.) $E \rightarrow F$

5.) $A \rightarrow D$ ~~6.) $BGH \rightarrow I$~~ 7.) $BH \rightarrow E$

7.) Is Rule 7.) needed?

✎ **Minimal cover for the following FDs?**

2a) $F \longrightarrow A$

1.) $\cancel{A}BH \rightarrow C$ ~~2b) $F \longrightarrow D$~~ 3.) $C \rightarrow E$ 4.) $E \rightarrow F$

5.) $A \rightarrow D$ ~~6.) $BGH \rightarrow I$~~ 7.) $BH \rightarrow E$

7.) Is Rule 7.) needed?

No!

Because BH $\xrightarrow[1.)]{}$ C $\xrightarrow[3.)]{}$ E

Constructing a Minimal Cover

Minimal cover for the following FDs?

2a) $F \longrightarrow A$

1.) $\cancel{A}BH \rightarrow C$ ~~2b) $F \longrightarrow D$~~ 3.) $C \rightarrow E$ 4.) $E \rightarrow F$

5.) $A \rightarrow D$ ~~6.) $BGH \rightarrow I$~~ ~~7.) $BH \rightarrow E$~~

Finished!

Our Minimal Cover is:

$$\boxed{\begin{aligned} \{\, & BH \longrightarrow C, \\ & F \longrightarrow A, \\ & C \longrightarrow E, \\ & E \longrightarrow F, \\ & A \longrightarrow D \,\} \end{aligned}}$$

Constructing a Minimal Cover

Minimal cover for the following FDs?

2a) F —> A

1.) $\cancel{A}BH \rightarrow C$ ~~2b) F —> D~~ 3.) $C \rightarrow E$ 4.) $E \rightarrow F$

5.) $A \rightarrow D$ ~~6.) $BGH \rightarrow F$~~ ~~7.) $BH \rightarrow E$~~

Finished!

Our Minimal Cover is:

{ BH —> C,
F —> A,
C —> E,
E —> F,
A —> D }

As mentioned, if we number the rules differently at the beginning, we may obtain another, different Minimal Cover.

Constructing a Minimal Cover

Final Remarks:

— steps 2 and 3 may be interchanged.
Still we obtain a Minimal Cover.

— What is a small example that shows that the
Minimal Cover is **not unique**?

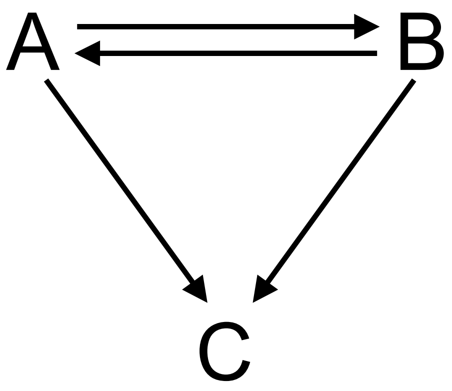
Constructing a Minimal Cover

<u>Final Remarks:</u>

— steps 2 and 3 may be interchanged.
Still we obtain a Minimal Cover.

— What is a small example that shows that the Minimal Cover is **not unique**?

A —> B
B —> A
A —> C
B —> C



Constructing a Minimal Cover

<u>Final Remarks:</u>

— steps 2 and 3 may be interchanged.
Still we obtain a Minimal Cover.

— What is a small example that shows that the Minimal Cover is **not unique**?

A —> B
B —> A
A —> C
B —> C

A —> B
B —> A
B —> C

A —> B
B —> A
A —> C

two possible
Minimal Covers!

2.) Normalization to Third Normal Form (3NF)

2.) Normalization to Third Normal Form (3NF)

Consider the following relation

Shipping(*ShipName*, *ShipType*, *TripId*, *Cargo*, *Port*, *Date*)

a.) Check if **Shipping** is in 3NF
b.) Decompose the relation into 3NF Relations.

2.) Normalization to Third Normal Form (3NF)

Consider the following relation

Shipping(*ShipName*, *ShipType*, *TripId*, *Cargo*, *Port*, *Date*)

a.) Check if **Shipping** is in 3NF

We need to know the functional dependencies of **Shipping**.

2.) Normalization to Third Normal Form (3NF)

Consider the following relation

Shipping(*ShipName*, *ShipType*, *TripId*, *Cargo*, *Port*, *Date*)

a.) Check if **Shipping** is in 3NF

We need to know the functional dependencies (FDs) of **Shipping**.

— let us assume that *ShipNames* are unique!

2.) Normalization to Third Normal Form (3NF)

Consider the following relation

Shipping(*ShipName*, *ShipType*, *TripId*, *Cargo*, *Port*, *Date*)

a.) Check if **Shipping** is in 3NF

We need to know the functional dependencies (FDs) of **Shipping**.

— let us assume that *ShipNames* are unique!

ShipName —> *ShipType*

This is a FD of **Shipping**.

2.) Normalization to Third Normal Form (3NF)

Consider the following relation

Shipping(*ShipName*, *ShipType*, *TripId*, *Cargo*, *Port*, *Date*)

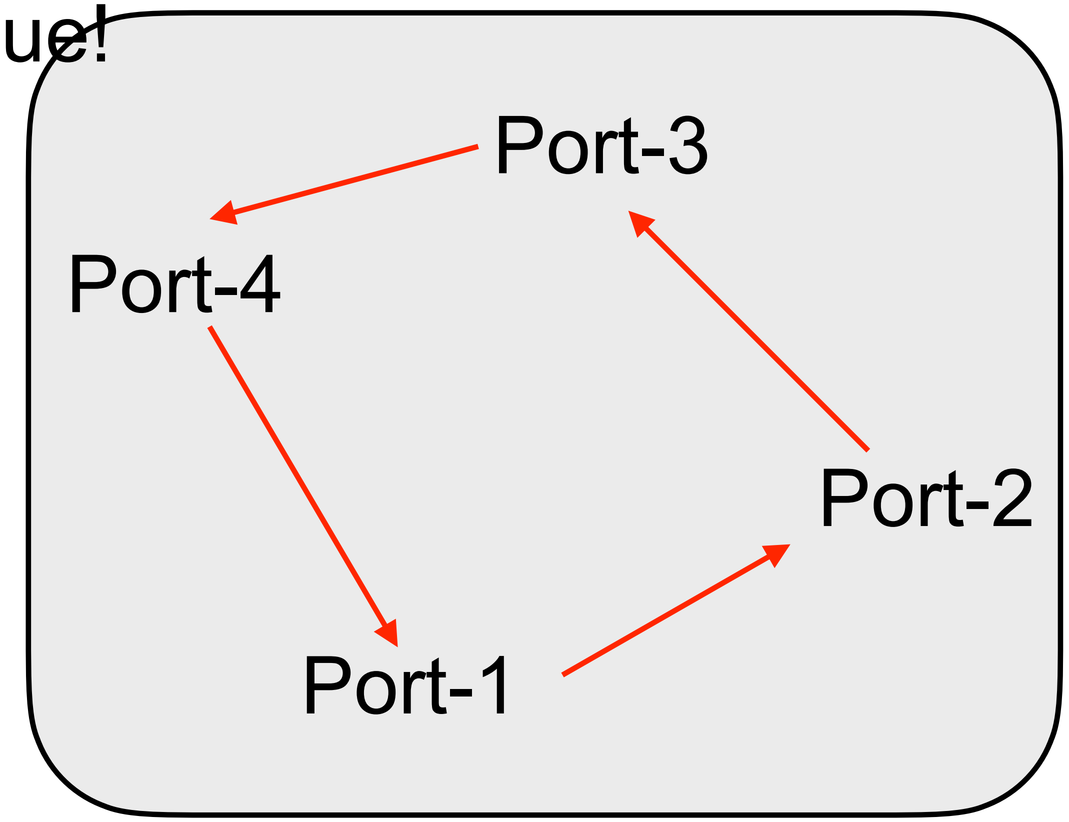
a.) Check if **Shipping** is in 3NF

We need to know the functional dependencies (FDs) of **Shipping**.

— let us assume that *ShipNames* are unique!

ShipName —> *ShipType*

TripId —> *???*



One Trip

2.) Normalization to Third Normal Form (3NF)

Consider the following relation

Shipping(*ShipName*, *ShipType*, *TripId*, *Cargo*, *Port*, *Date*)

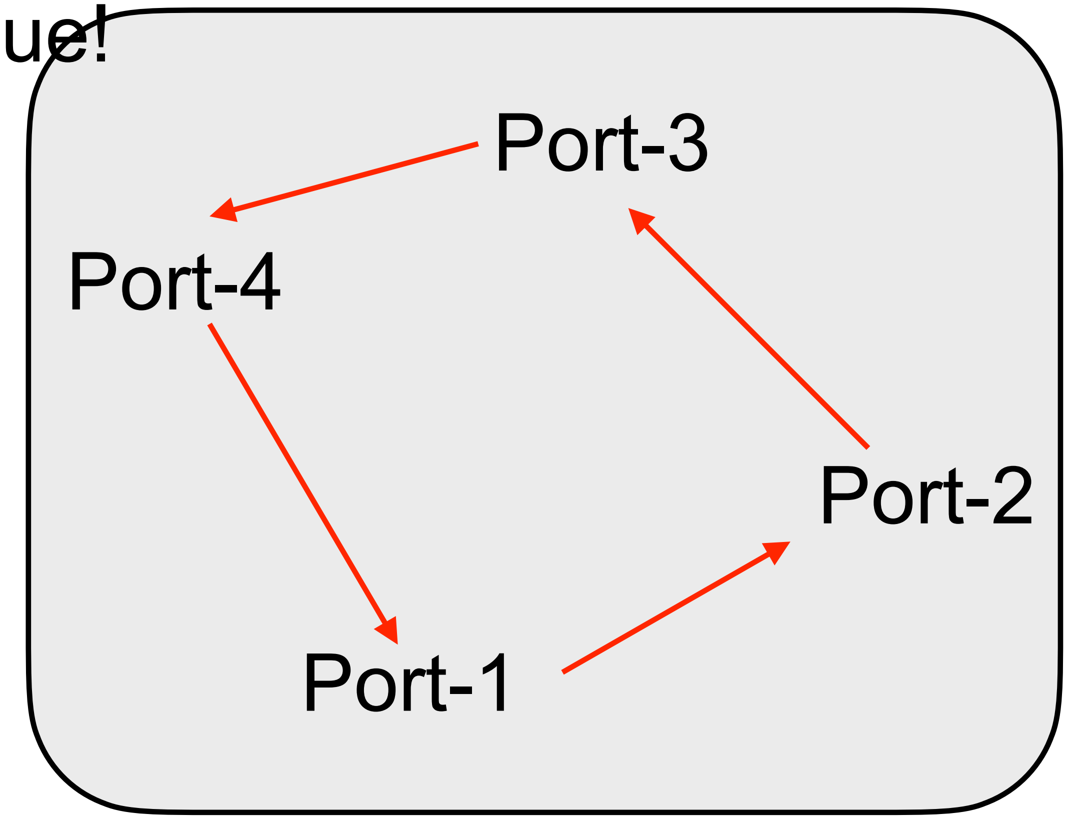
a.) Check if **Shipping** is in 3NF

We need to know the functional dependencies (FDs) of **Shipping**.

— let us assume that *ShipNames* are unique!

ShipName —> *ShipType*

TripId —> *ShipName*, *Cargo*



One Trip

2.) Normalization to Third Normal Form (3NF)

Consider the following relation

Shipping(*ShipName*, *ShipType*, *TripId*, *Cargo*, *Port*, *Date*)

a.) Check if **Shipping** is in 3NF

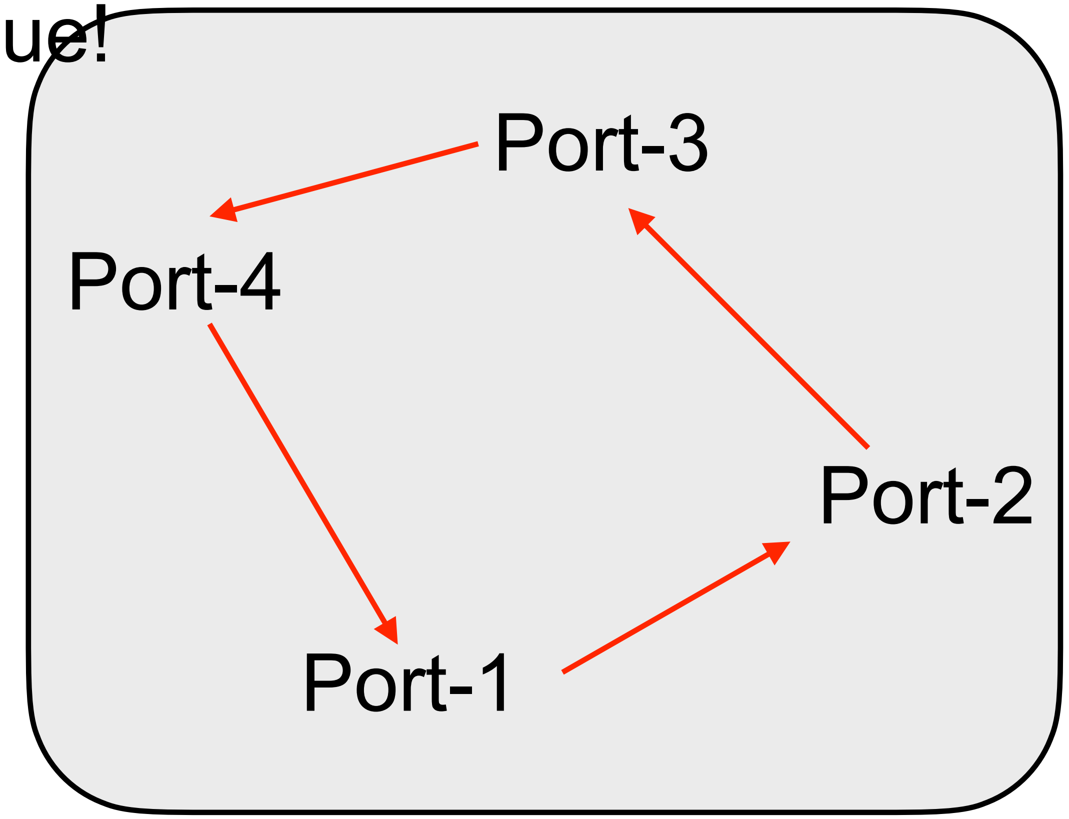
We need to know the functional dependencies (FDs) of **Shipping**.

— let us assume that *ShipNames* are unique!

ShipName —> ShipType

TripId —> ShipName, Cargo

ShipName, Date —> ???



One Trip

2.) Normalization to Third Normal Form (3NF)

Consider the following relation

Shipping(*ShipName*, *ShipType*, *TripId*, *Cargo*, *Port*, *Date*)

a.) Check if **Shipping** is in 3NF

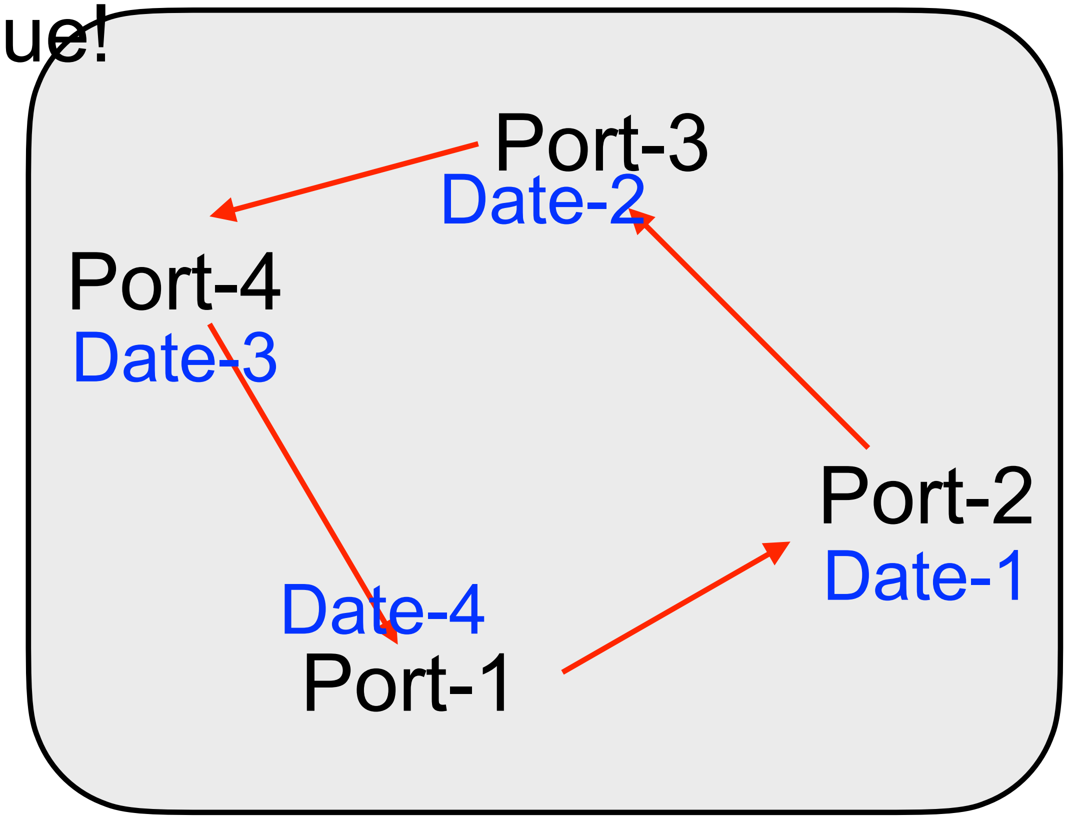
We need to know the functional dependencies (FDs) of **Shipping**.

— let us assume that *ShipNames* are unique!

ShipName —> ShipType

TripId —> ShipName, Cargo

ShipName, Date —> ???



One Trip

2.) Normalization to Third Normal Form (3NF)

Consider the following relation

Shipping(*ShipName*, *ShipType*, *TripId*, *Cargo*, *Port*, *Date*)

a.) Check if **Shipping** is in 3NF

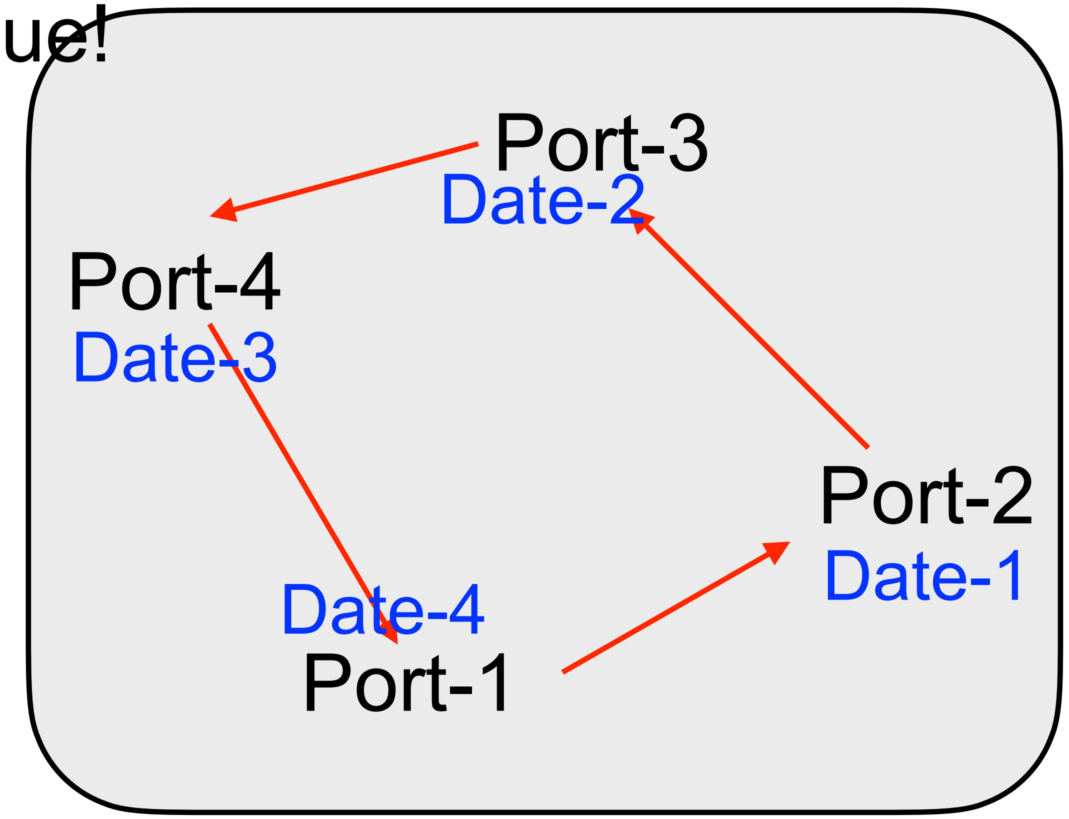
We need to know the functional dependencies (FDs) of **Shipping**.

— let us assume that *ShipNames* are unique!

ShipName —> ShipType

TripId —> ShipName, Cargo

ShipName, Date —> TripId, Port



One Trip

2.) Normalization to Third Normal Form (3NF)

Consider the following relation

Shipping(*ShipName*, *ShipType*, *TripId*, *Cargo*, *Port*, *Date*)

a.) Check if **Shipping** is in 3NF

ShipName —> ShipType
TripId —> ShipName, Cargo
ShipName, Date —> TripId, Port

What are the candidate keys?

2.) Normalization to Third Normal Form (3NF)

Consider the following relation

Shipping(*ShipName*, *ShipType*, *TripId*, *Cargo*, *Port*, *Date*)

a.) Check if **Shipping** is in 3NF

ShipName —> ShipType
TripId —> ShipName, Cargo
ShipName, Date —> TripId, Port

TripId, Port, Date

Is this a candidate key?

2.) Normalization to Third Normal Form (3NF)

Consider the following relation

Shipping(*ShipName*, *ShipType*, *TripId*, *Cargo*, *Port*, *Date*)

a.) Check if **Shipping** is in 3NF

ShipName —> ShipType

2.) *TripId —> ShipName, Cargo*

3.) *ShipName, Date —> TripId, Port*

TripId, Port, Date

Is this a candidate key?

No!

TripId, Date ===> ShipName, Cargo, Date ===> TripId, Port

2.) 3.)

2.) Normalization to Third Normal Form (3NF)

Consider the following relation

Shipping(*ShipName*, *ShipType*, *TripId*, *Cargo*, *Port*, *Date*)

a.) Check if **Shipping** is in 3NF

ShipName —> ShipType

2.) *TripId —> ShipName, Cargo*

3.) *ShipName, Date —> TripId, Port*

TripId, Date ← This is a candidate key!

2.) Normalization to Third Normal Form (3NF)

Consider the following relation

Shipping(*ShipName*, *ShipType*, *TripId*, *Cargo*, *Port*, *Date*)

a.) Check if **Shipping** is in 3NF

ShipName —> ShipType
2.) *TripId —> ShipName, Cargo*
3.) *ShipName, Date —> TripId, Port*

TripId, Date ← This is a candidate key!

Any other candidate key?

2.) Normalization to Third Normal Form (3NF)

Consider the following relation

Shipping(*ShipName*, *ShipType*, *TripId*, *Cargo*, *Port*, *Date*)

a.) Check if **Shipping** is in 3NF

ShipName —> ShipType

2.) *TripId —> ShipName, Cargo*

3.) *ShipName, Date —> TripId, Port*

TripId, Date ← This is a candidate key!

Any other candidate key?

2.) Normalization to Third Normal Form (3NF)

Consider the following relation

Shipping(*ShipName*, *ShipType*, *TripId*, *Cargo*, *Port*, *Date*)

a.) Check if **Shipping** is in 3NF

ShipName —> ShipType
2.) *TripId —> ShipName, Cargo*
3.) *ShipName, Date —> TripId, Port*

TripId, Date ← This is a candidate key!

Any other candidate key?
Yes:

ShipName, Date

2.) Normalization to Third Normal Form (3NF)

Consider the following relation

Shipping(*ShipName*, *ShipType*, *TripId*, *Cargo*, *Port*, *Date*)

a.) Check if **Shipping** is in 3NF

ShipName —> ShipType

2.) *TripId —> ShipName, Cargo*

3.) *ShipName, Date —> TripId, Port*

TripId, Date

ShipName, Date

Shipping(*ShipName*, *ShipType*, *TripId*, *Cargo*, *Port*, *Date*)

ShipName —> ShipType
TripId —> ShipName, Cargo
ShipName, Date —> TripId, Port

b.) Bring it into 3NF.

TripId, Date
ShipName, Date

---

(1) compute a minimal cover **C** of the set **F** of FDs that hold

(2) for each FD X→A in **C**, create a new table and choose X as primary key

(3) if none of the new tables contains a candidate key K of the original table, then add a new table with exactly the columns of K

(4) remove redundant tables (that are contained in others)

Shipping(*ShipName*, *ShipType*, *TripId*, *Cargo*, *Port*, *Date*)

ShipName —> ShipType
TripId —> ShipName, Cargo
ShipName, Date —> TripId, Port

b.) Bring it into 3NF.

TripId, Date
ShipName, Date

---

(1) compute a minimal cover **C** of the set **F** of FDs that hold

ShipName —> ShipType
TripId —> ShipName
TripId —> Cargo
ShipName, Date —> TripId
ShipName, Date —> Port

Shipping(*ShipName*, *ShipType*, *TripId*, *Cargo*, *Port*, *Date*)

ShipName —> ShipType
TripId —> ShipName, Cargo
ShipName, Date —> TripId, Port

b.) Bring it into 3NF.

TripId, Date
ShipName, Date

---

(1) compute a minimal cover **C** of the set **F** of FDs that hold

ShipName —> ShipType
TripId —> ShipName
TripId —> Cargo
ShipName, Date —> TripId
ShipName, Date —> Port

This is already a minimal cover!

Shipping(*ShipName*, *ShipType*, *TripId*, *Cargo*, *Port*, *Date*)

ShipName —> ShipType
TripId —> ShipName, Cargo
ShipName, Date —> TripId, Port

b.) Bring it into 3NF.

TripId, Date
ShipName, Date

---

(1) compute a minimal cover **C** of the set **F** of FDs that hold

ShipName —> ShipType
TripId —> ShipName, Cargo
ShipName, Date —> TripId, Port

This is already a minimal cover!
In fact, we may combine the last two FDs.

Shipping(*ShipName*, *ShipType*, *TripId*, *Cargo*, *Port*, *Date*)

ShipName —> ShipType
TripId —> ShipName, Cargo
ShipName, Date —> TripId, Port

b.) Bring it into 3NF.

TripId, Date
ShipName, Date

ShipName —> ShipType
TripId —> ShipName, Cargo
ShipName, Date —> TripId, Port

(2) for each FD X→A in **C**, create a new table and choose X as primary key

Shipping(*ShipName*, *ShipType*, *TripId*, *Cargo*, *Port*, *Date*)

ShipName —> ShipType
TripId —> ShipName, Cargo
ShipName, Date —> TripId, Port

b.) Bring it into 3NF.

TripId, Date
ShipName, Date

ShipName —> ShipType
TripId —> ShipName, Cargo
ShipName, Date —> TripId, Port

ST(<u>*ShipName*</u>, *ShipType*)
SNC(<u>*TripId*</u>, *ShipName*, *Cargo*)
TIP(<u>*ShipName, Date*</u>, *TripId*, *Port*)

Shipping(*ShipName*, *ShipType*, *TripId*, *Cargo*, *Port*, *Date*)

ShipName —> ShipType
TripId —> ShipName, Cargo
ShipName, Date —> TripId, Port

b.) Bring it into 3NF.

TripId, Date
ShipName, Date

ShipName —> ShipType
TripId —> ShipName, Cargo
ShipName, Date —> TripId, Port

(3) if none of the new tables contains a candidate key K of the original table,
then add a new table with exactly the columns of K

ST(*ShipName*, *ShipType*)
SNC(*TripId*, *ShipName*, *Cargo*)
TIP(*ShipName, Date*, *TripId*, *Port*)

Shipping(*ShipName*, *ShipType*, *TripId*, *Cargo*, *Port*, *Date*)

ShipName —> ShipType
TripId —> ShipName, Cargo
ShipName, Date —> TripId, Port

b.) Bring it into 3NF.

TripId, Date
ShipName, Date

ShipName —> ShipType
TripId —> ShipName, Cargo
ShipName, Date —> TripId, Port

(3) if none of the new tables contains a candidate key K of the original table,
then add a new table with exactly the columns of K

ST(*ShipName*, *ShipType*)
SNC(*TripId*, *ShipName*, *Cargo*)
TIP(*ShipName*, *Date*, *TripId*, *Port*)

Each of the two candidate keys is contained in **TIP**. So, nothing needs to be done.

But: it would be enough if **one** candidate key is contained.

Shipping(*ShipName*, *ShipType*, *TripId*, *Cargo*, *Port*, *Date*)

ShipName —> ShipType
TripId —> ShipName, Cargo
ShipName, Date —> TripId, Port

b.) Bring it into 3NF.

TripId, Date
ShipName, Date

ShipName —> ShipType
TripId —> ShipName, Cargo
ShipName, Date —> TripId, Port

(4) remove redundant tables (that are contained in others)

ST(*<u>ShipName</u>*, *ShipType*)
SNC(*<u>TripId</u>*, *ShipName*, *Cargo*)
TIP(*<u>ShipName, Date</u>*, *TripId*, *Port*)

Not applicable (no table is contained in another)

Shipping(*ShipName*, *ShipType*, *TripId*, *Cargo*, *Port*, *Date*)

ShipName —> ShipType
TripId —> ShipName, Cargo
ShipName, Date —> TripId, Port

b.) Bring it into 3NF.

TripId, Date
ShipName, Date

ShipName —> ShipType
TripId —> ShipName
TripId —> Cargo
ShipName, Date —> TripId
ShipName, Date —> Port

A table T is in 3NF if for every non-trivial FD A —> B of T it holds that:
— either A is a superkey or
— B is prime.

ST(*ShipName*, *ShipType*)
SNC(*TripId*, *ShipName*, *Cargo*)
TIP(*ShipName*, *Date*, *TripId*, *Port*)

Finished!
Let's verify that each table is in 3NF.

Shipping(*ShipName*, *ShipType*, *TripId*, *Cargo*, *Port*, *Date*)

ShipName —> ShipType
TripId —> ShipName, Cargo
ShipName, Date —> TripId, Port

b.) Bring it into 3NF.

TripId, Date
ShipName, Date

ShipName —> ShipType
TripId —> ShipName
TripId —> Cargo
ShipName, Date —> TripId
ShipName, Date —> Port

ST(<u>*ShipName*</u>, *ShipType*)
SNC(<u>*TripId*</u>, *ShipName*, *Cargo*)
TIP(<u>*ShipName, Date*</u>, *TripId*, *Port*)

Finished!
Let's verify that each table is in 3NF.

For every non-trivial FD X —> A
— either X is a superkey or
— A is prime

Shipping(*ShipName*, *ShipType*, *TripId*, *Cargo*, *Port*, *Date*)

ShipName —> ShipType
TripId —> ShipName, Cargo
ShipName, Date —> TripId, Port

TripId, Date
ShipName, Date

ShipName —> ShipType
TripId —> ShipName
TripId —> Cargo
ShipName, Date —> TripId
ShipName, Date —> Port

ST(*<u>ShipName</u>, ShipType*)
SNC(*<u>TripId</u>, ShipName, Cargo*)
TIP(*<u>ShipName, Date</u>, TripId, Port*)

Finished!
Let's verify that each table is in 3NF.

For every non-trivial FD X —> A
— either X is a superkey or
— A is prime

ShipName is prime, therefore OK for 3NF!!

Shipping(*ShipName*, *ShipType*, *TripId*, *Cargo*, *Port*, *Date*)

ShipName —> ShipType
TripId —> ShipName, Cargo
ShipName, Date —> TripId, Port

TripId, Date
ShipName, Date

ShipName —> ShipType
TripId —> ShipName
TripId —> Cargo
ShipName, Date —> TripId
ShipName, Date —> Port

ST(<u>*ShipName*</u>, *ShipType*)
SNC(<u>*TripId*</u>, *ShipName*, *Cargo*)
TIP(<u>*ShipName, Date*</u>, *TripId*, *Port*)

Finished!
Let's verify that each table is in 3NF.

For every non-trivial FD X —> A
— either X is a superkey or
— A is prime

—> what else needs to be checked?

3.) Normalization to Boyce-Codd Normal Form (BCNF)

Shipping(*ShipName*, *ShipType*, *TripId*, *Cargo*, *Port*, *Date*)

ShipName —> ShipType
TripId —> ShipName, Cargo
ShipName, Date —> TripId, Port

d.) Bring the relation into BCNF.

TripId, Date
ShipName, Date

ShipName —> ShipType
TripId —> ShipName
TripId —> Cargo
ShipName, Date —> TripId, Port

ST(*ShipName*, *ShipType*)
SN(*TripId*, *ShipName*)
C(*TripId*, *Cargo*)
TIP(*ShipName, Date*, *TripId*, *Port*)

Algorithm for BCNF Decomposition

BCNF can be obtained by repeatedly **decomposing** a table **along an FD that violates BCNF**:

1 **Algorithm:** BCNFDecomposition

Input : $(\text{sch}(R), \mathcal{F})$

Output: Schema $\{(\text{sch}(R_1), \mathcal{F}_1), \ldots, (\text{sch}(R_n), \mathcal{F}_n)\}$ in BCNF

2 $\mathit{Decomposed} \leftarrow \{(\text{sch}(R), \mathcal{F})\}$;

3 **while** $\exists (\text{sch}(S), \mathcal{F}_S) \in \mathit{Decomposed}$ that is not in BCNF **do**

4   Let $\alpha \rightarrow \beta$ be an FD in $\mathcal{F}_S$ that violates BCNF;

5   Decompose S into $S_1(\alpha\beta)$ and $S_2((\underline{S - \beta}) \cup \alpha)$;

6 **return** $\mathit{Decomposed}$;

In line 5, use the projection mechanism on slide 241 to obtain the $\mathcal{F}_{S_i}$.

Shipping(*ShipName*, *ShipType*, *TripId*, *Cargo*, *Port*, *Date*)

ShipName —> ShipType
TripId —> ShipName, Cargo
ShipName, Date —> TripId, Port

d.) Bring the relation into BCNF.

TripId, Date
ShipName, Date

ShipName —> ShipType
TripId —> ShipName
TripId —> Cargo
ShipName, Date —> TripId, Port

Let $\alpha \to \beta$ be an FD in $\mathcal{F}_S$ that violates BCNF;
Decompose S into $S_1(\alpha\beta)$ and $S_2((S - \beta) \cup \alpha)$;

TripId —> ShipName
α β

ST(<u>*ShipName*</u>, *ShipType*)
SN(<u>*TripId*</u>, *ShipName*)
C(<u>*TripId*</u>, *Cargo*)
TIP(<u>*ShipName, Date*</u>, *TripId, Port*)

Shipping(*ShipName*, *ShipType*, *TripId*, *Cargo*, *Port*, *Date*)

ShipName —> ShipType
TripId —> ShipName, Cargo
ShipName, Date —> TripId, Port

d.) Bring the relation into BCNF.

TripId, Date
ShipName, Date

ShipName —> ShipType
TripId —> ShipName
TripId —> Cargo
ShipName, Date —> TripId, Port

Let $\alpha \rightarrow \beta$ be an FD in $\mathcal{F}_S$ that violates BCNF;
Decompose S into $S_1(\alpha\beta)$ and $S_2((S-\beta)\cup\alpha)$;

TripId —> ShipName
α β

ST(<u>*ShipName*</u>, *ShipType*)
SN(<u>*TripId*</u>, *ShipName*)
C(<u>*TripId*</u>, *Cargo*)
TIP(<u>*ShipName, Date*</u>, *TripId*, *Port*)

TIP-1(<u>*TripId*</u>, *ShipName*)
TIP-2(*Date*, *TripId*, *Port*)

Shipping(*ShipName*, *ShipType*, *TripId*, *Cargo*, *Port*, *Date*)

ShipName —> ShipType
TripId —> ShipName, Cargo
ShipName, Date —> TripId, Port

d.) Bring the relation into BCNF.

TripId, Date
ShipName, Date

ShipName —> ShipType
TripId —> ShipName
TripId —> Cargo
ShipName, Date —> TripId, Port

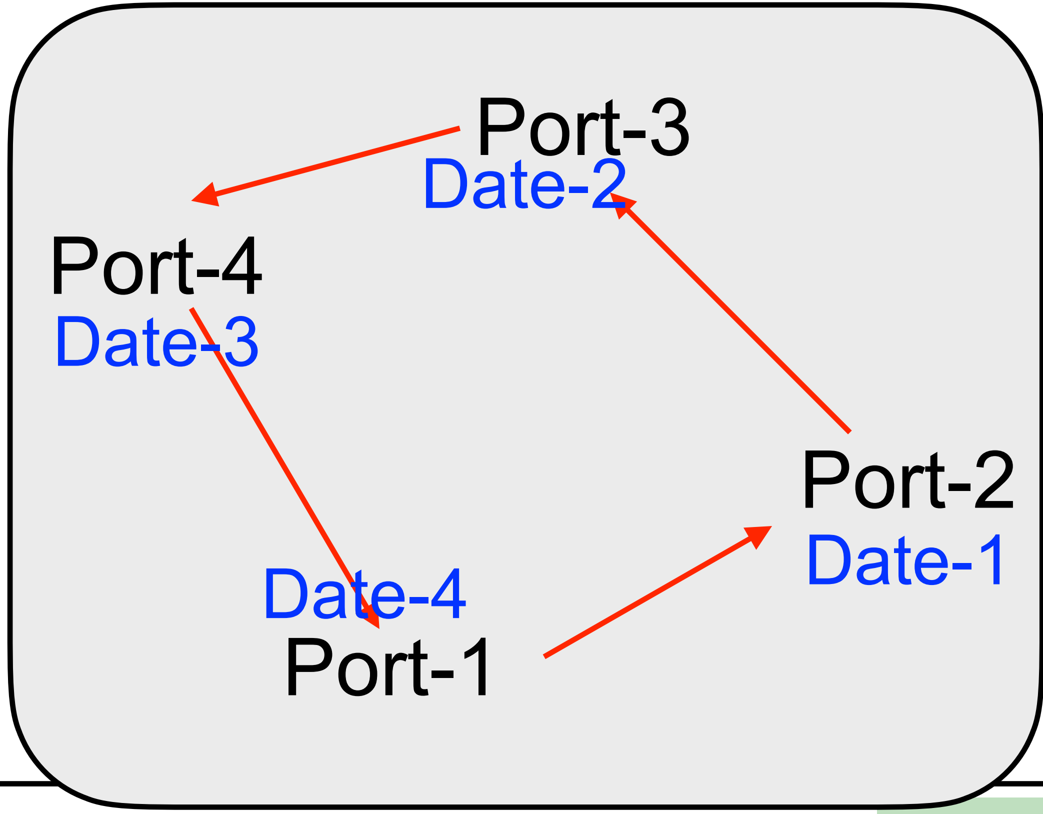
Let $\alpha \to \beta$ be an FD in $\mathcal{F}_S$ that violates BCNF;
Decompose S into $S_1(\alpha\beta)$ and $S_2((S - \beta) \cup \alpha)$;

TripId —> ShipName
α β

ST(*ShipName*, *ShipType*)
SN(*TripId*, *ShipName*)
C(*TripId*, *Cargo*)
TIP(*ShipName, Date*, *TripId*, *Port*)

TIP-1(*TripId*, *ShipName*)
TIP-2(*Date*, *TripId*, *Port*)

What are the FDs for **TIP-2**?



Scenario-1:

Date-1 **can** be equal to Date-2.

ShipName —> ShipType
TripId —> ShipName
TripId —> Cargo
ShipName, Date —> TripId, Port

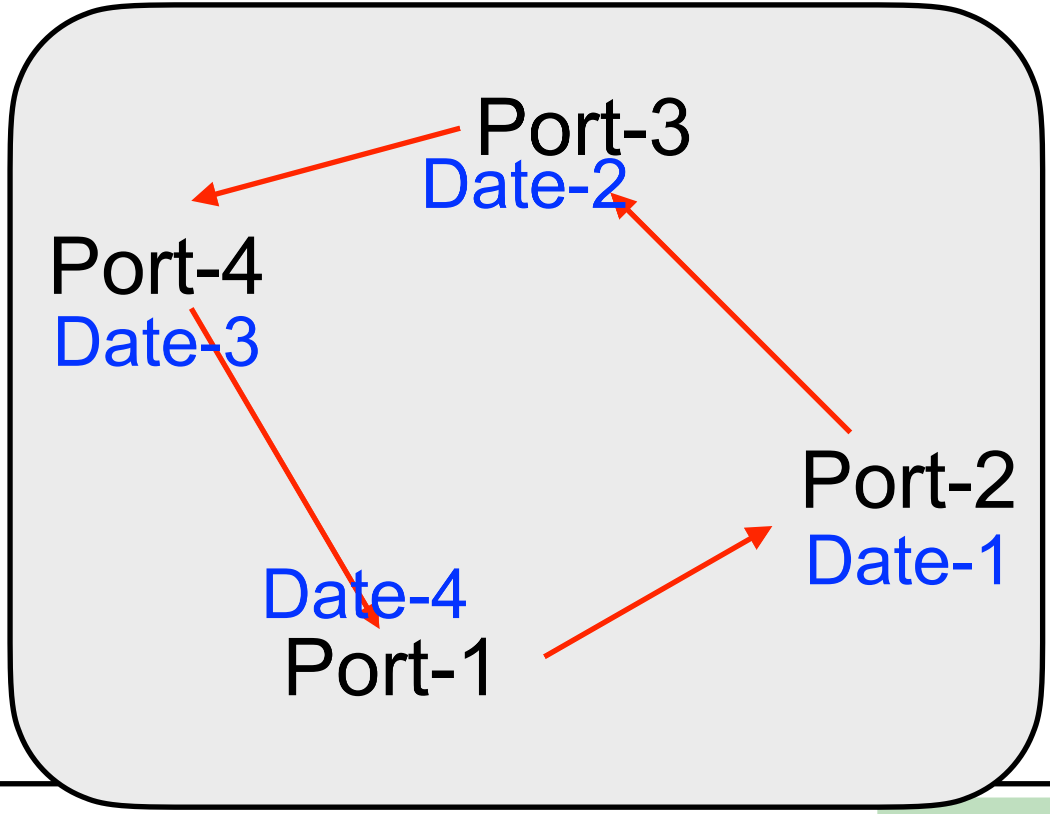
Let $\alpha \rightarrow \beta$ be an FD in $\mathcal{F}_S$ that violates BCNF;
Decompose S into $S_1(\alpha\beta)$ and $S_2((S-\beta) \cup \alpha)$;

TripId —> ShipName
α β

ST(*ShipName, ShipType*)
SN(*TripId, ShipName*)
C(*TripId, Cargo*)
TIP(*ShipName, Date, TripId, Port*)

TIP-1(*TripId, ShipName*)
TIP-2(*Date, TripId, Port*)

What are the FDs for **TIP-2**?



Scenario-1:

Date-1 **can** be equal to Date-2.

ShipName —> ShipType
TripId —> ShipName
TripId —> Cargo
ShipName, Date —> TripId, Port

Let $\alpha \rightarrow \beta$ be an FD in $\mathcal{F}_S$ that violates BCNF;
Decompose S into $S_1(\alpha\beta)$ and $S_2((S - \beta) \cup \alpha)$;

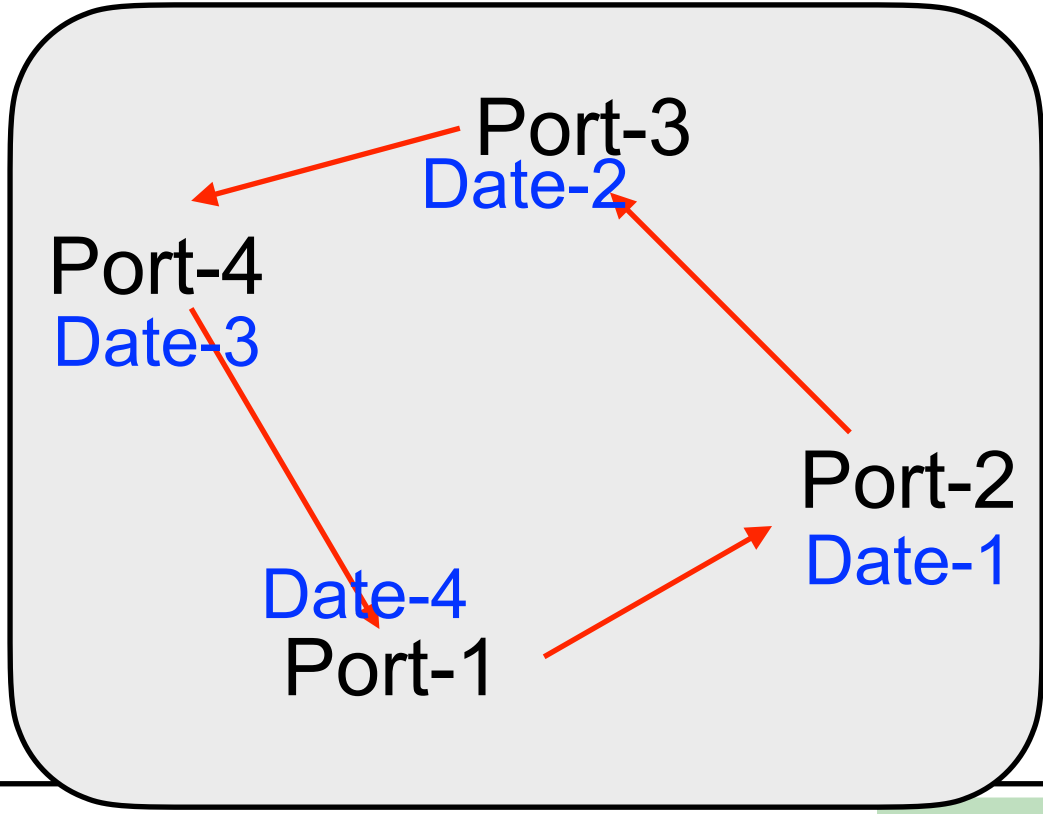
TripId —> ShipName
α β

ST(*ShipName*, *ShipType*)
SN(*TripId*, *ShipName*)
C(*TripId*, *Cargo*)
TIP(*ShipName, Date*, *TripId*, *Port*)

TIP-1(*TripId*, *ShipName*)
TIP-2(*Date*, *TripId*, *Port*)

TripId, Port —> Date

because on one trip a ship does not go to the same port more than once!



Scenario-1:

Date-1 **can** be equal to Date-2.

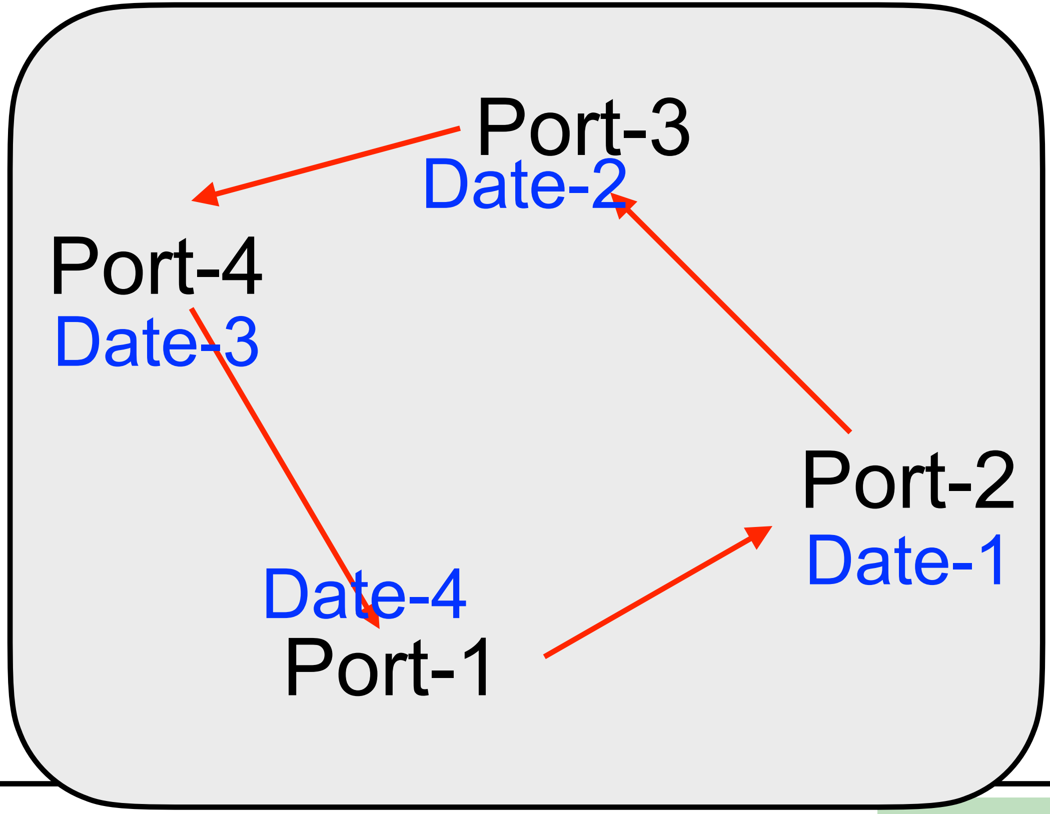
ShipName —> ShipType
TripId —> ShipName
TripId —> Cargo
ShipName, Date —> TripId, Port

Let $\alpha \rightarrow \beta$ be an FD in $\mathcal{F}_S$ that violates BCNF;
Decompose S into $S_1(\alpha\beta)$ and $S_2((S - \beta) \cup \alpha)$;

TripId —> ShipName
α β

ST(*ShipName*, *ShipType*)
SN(*TripId*, *ShipName*)
C(*TripId*, *Cargo*)
TIP(*ShipName, Date*, *TripId*, *Port*)

TIP-1(*TripId*, *ShipName*)
TIP-2(*Date*, *TripId, Port*)



Date's are timestamps, so
Date-1 **cannot** be equal to Date-2.

ShipName —> ShipType
TripId —> ShipName
TripId —> Cargo
ShipName, Date —> TripId, Port

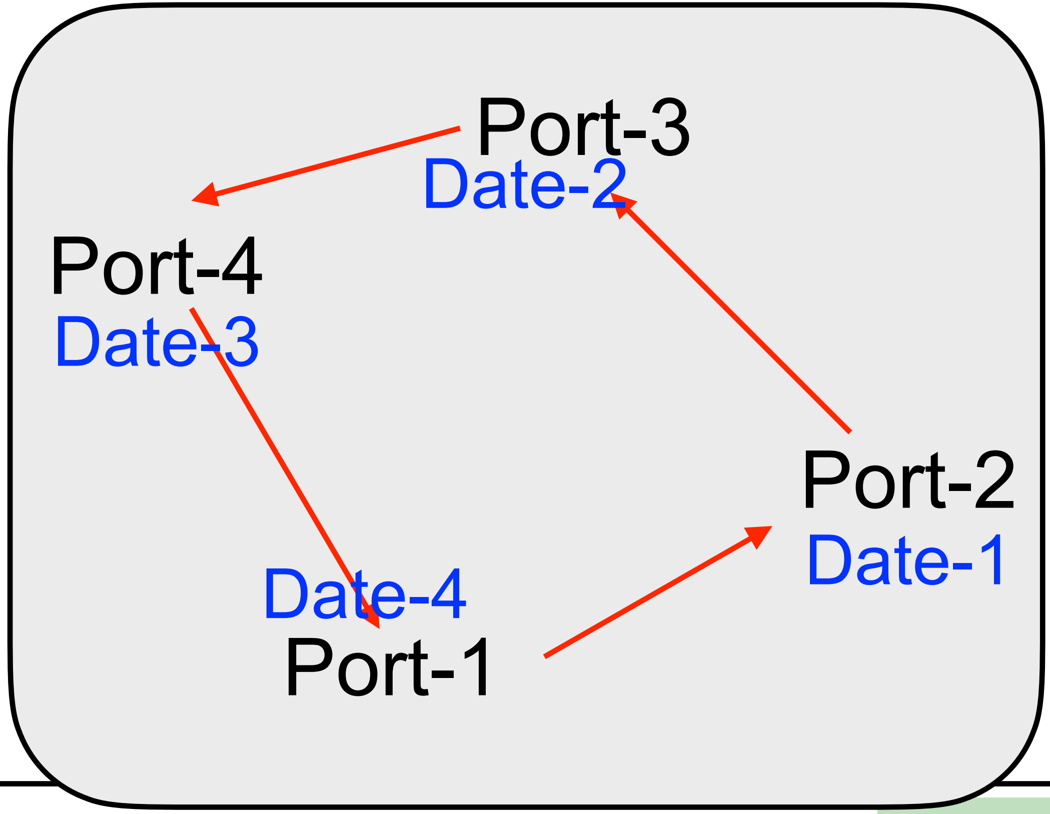
Let $\alpha \to \beta$ be an FD in $\mathcal{F}_S$ that violates BCNF;
Decompose S into $S_1(\alpha\beta)$ and $S_2((S - \beta) \cup \alpha)$;

TripId —> ShipName
α β

ST(*ShipName, ShipType*)
SN(*TripId, ShipName*)
C(*TripId, Cargo*)
TIP(*ShipName, Date, TripId, Port*)

TIP-1(*TripId, ShipName*)
TIP-2(*Date, TripId, Port*)

What are the FDs for **TIP-2**?



Date's are timestamps, so
Date-1 **cannot** be equal to Date-2.

TripId, Port —> Date (as before)
TripId, Date —> Port (new)

ShipName —> ShipType
TripId —> ShipName
TripId —> Cargo
ShipName, Date —> TripId, Port

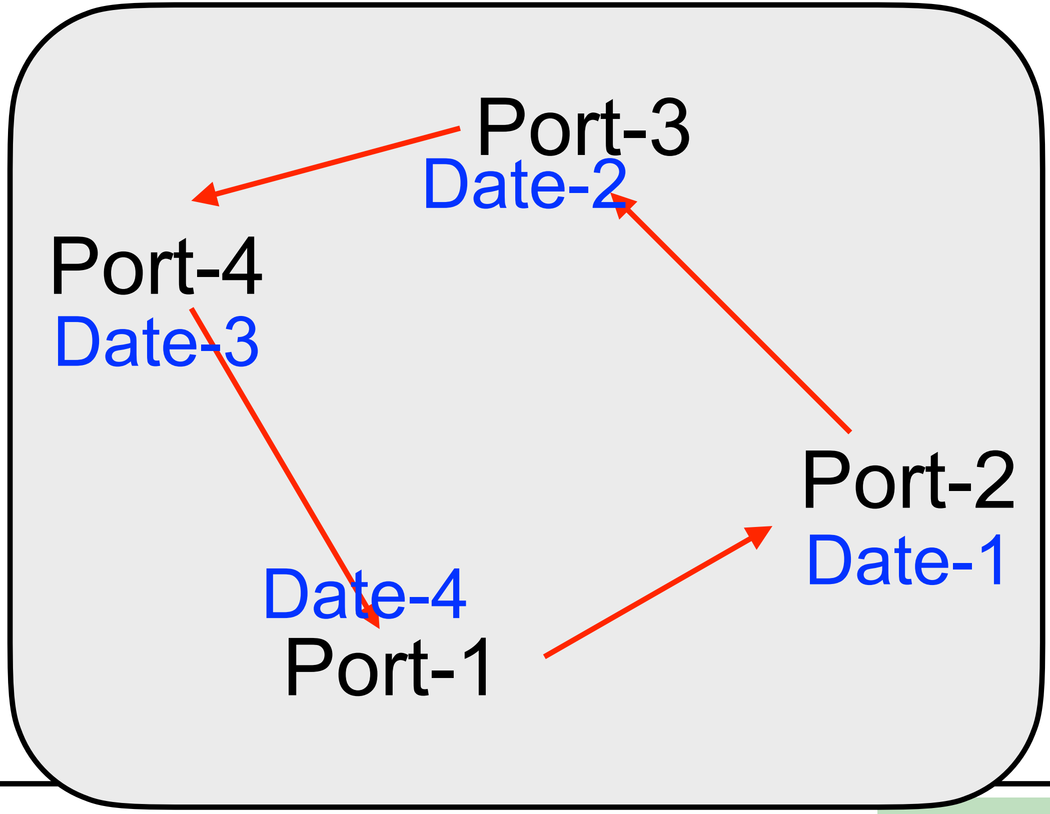
Let $\alpha \to \beta$ be an FD in $\mathcal{F}_S$ that violates BCNF;
Decompose S into $S_1(\alpha\beta)$ and $S_2((S - \beta) \cup \alpha)$;

TripId —> ShipName
α β

ST(*<u>ShipName</u>, ShipType*)
SN(*<u>TripId</u>, ShipName*)
C(*<u>TripId</u>, Cargo*)
TIP(*<u>ShipName, Date</u>, TripId, Port*)

TIP-1(*<u>TripId</u>, ShipName*)
TIP-2(*<u>Date, TripId</u>, Port*)

What are the FDs for **TIP-2**?



Date's are timestamps, so
Date-1 **cannot** be equal to Date-2.

TripId, Port —> *Date*
TripId, Date —> *Port*

Now:
two candidate keys!

ShipName —> *ShipType*
TripId* —> *ShipName
TripId —> *Cargo*
ShipName, Date —> *TripId, Port*

Let $\alpha \to \beta$ be an FD in $\mathcal{F}_S$ that violates BCNF;
Decompose S into $S_1(\alpha\beta)$ and $S_2((S - \beta) \cup \alpha)$;

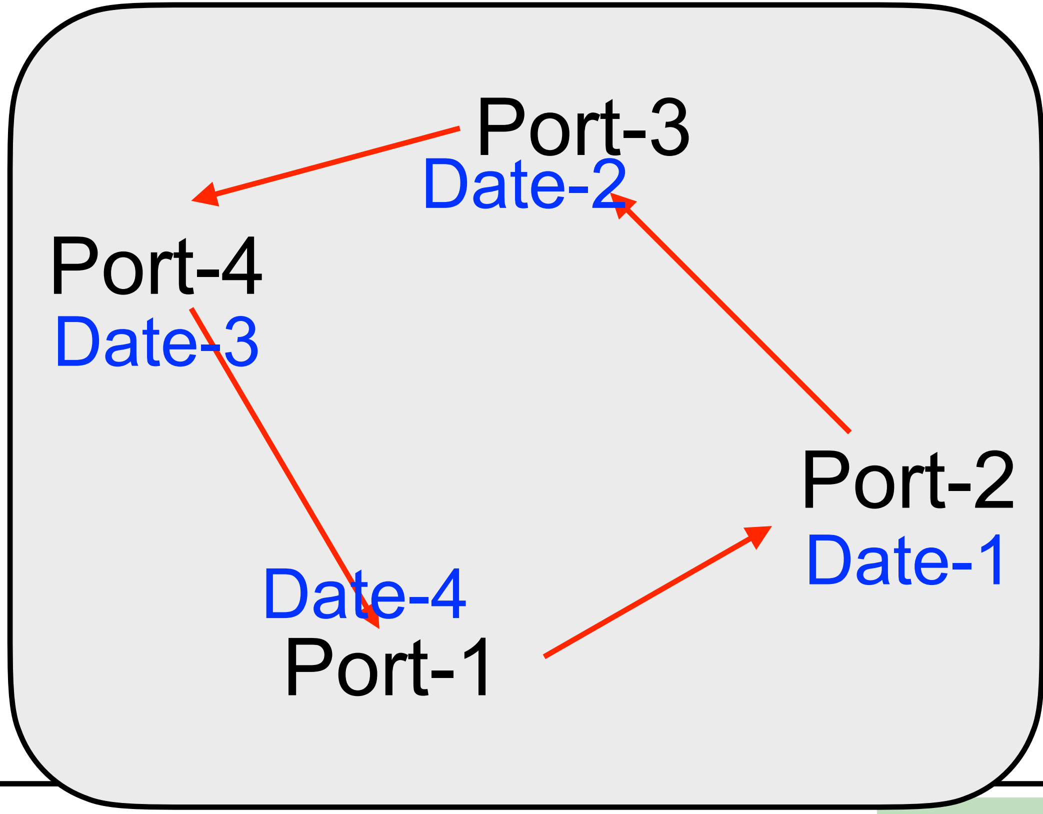
TripId* —> *ShipName
α β

ST(*ShipName*, *ShipType*)
SN(*TripId*, *ShipName*)
C(*TripId*, *Cargo*)
TIP(*ShipName, Date*, *TripId*, *Port*)

~~**TIP-1**(*TripId*, *ShipName*)~~
TIP-2(*Date*, *TripId*, *Port*)

What are the FDs for **TIP-2**?

TIP-1 can be removed, because it is included in SN!



Date's are timestamps, so
Date-1 **cannot** be equal to Date-2.

TripId, Port —> *Date*
TripId, Date —> *Port*

Now:
two candidate keys!

ShipName —> *ShipType*
TripId* —> *ShipName
TripId —> *Cargo*
ShipName, Date —> *TripId, Port*

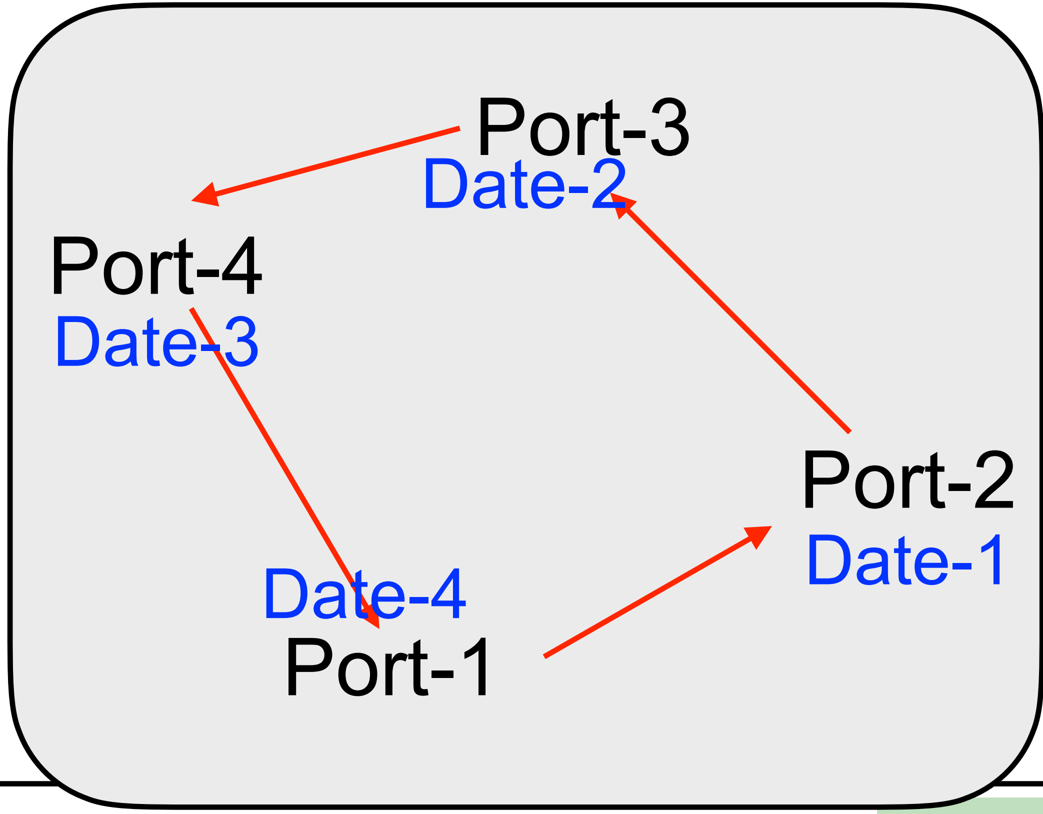
Let $\alpha \to \beta$ be an FD in $\mathcal{F}_S$ that violates BCNF;
Decompose S into $S_1(\alpha\beta)$ and $S_2((S - \beta) \cup \alpha)$;

TripId* —> *ShipName
α β

ST(*ShipName*, *ShipType*)
SN(*TripId*, *ShipName*)
C(*TripId*, *Cargo*)
TIP(*ShipName, Date*, *TripId*, *Port*)

~~**TIP-1**(*TripId*, *ShipName*)~~
TIP-2(*Date*, *TripId*, *Port*)

needed?



Date's are timestamps, so
Date-1 **cannot** be equal to Date-2.

<u>TripId, Port</u> —> Date
<u>TripId, Date</u> —> Port

Now:
two candidate keys!

ShipName —> ShipType
TripId —> ShipName
TripId —> Cargo
ShipName, Date —> TripId, Port

Let $\alpha \rightarrow \beta$ be an FD in $\mathcal{F}_S$ that violates BCNF;
Decompose S into $S_1(\alpha\beta)$ and $S_2((S - \beta) \cup \alpha)$;

TripId —> ShipName
α β

needed?

ST(*<u>ShipName</u>, ShipType*)
SN(*<u>TripId</u>, ShipName*)
C(*<u>TripId</u>, Cargo*)
TIP(*<u>ShipName, Date</u>, TripId, Port*)

~~**TIP-1**(*TripId, ShipName*)~~
TIP-2(*Date, TripId, Port*)

BCNF Decomposition should be formalized
for a *set* of relations, not just a single one!
(as for 3NF Decomposition)