

## 0.14 Canonical Cover

you-said

write this algorithm with notations:  $result := \{FDs\}$  Do Replace All FDs in result of the form  $\alpha \rightarrow \beta_1$  and  $\alpha \rightarrow \beta_2$  by  $\alpha \rightarrow \beta_1 \cup \beta_2$  If  $\alpha \rightarrow \beta$  in result has extraneous attribute  $a$  in the left side: Remove  $a$  from  $\alpha$  If  $\alpha \rightarrow \beta$  in result has extraneous attribute  $b$  in the right side: Remove  $b$  from  $\beta$  While result does not change Return result

ChatGPT said:

Here's the algorithm rewritten with formal notations:

1. **Input:** Set of functional dependencies  $FDs$
2. **Initialize:**  $result := FDs$
3. **Repeat:**
  - **Step 1:** For every pair of functional dependencies  $\alpha \rightarrow \beta_1$  and  $\alpha \rightarrow \beta_2$  in  $result$ :
    - Replace them with a single functional dependency  $\alpha \rightarrow \beta_1 \cup \beta_2$ .
  - **Step 2:** For each functional dependency  $\alpha \rightarrow \beta$  in  $result$ :
    - If there exists an **extraneous attribute**  $a \in \alpha$  such that  $result\{\alpha \rightarrow \beta\} \setminus \{a\}$  implies  $(\alpha \setminus \{a\}) \rightarrow \beta$ , then remove  $a$  from  $\alpha$ .
  - **Step 3:** For each functional dependency  $\alpha \rightarrow \beta$  in  $result$ :
    - If there exists an **extraneous attribute**  $b \in \beta$  such that  $result\{\alpha \rightarrow \beta\} \setminus \{b\}$  implies  $\alpha \rightarrow (\beta \setminus \{b\})$ , then remove  $b$  from  $\beta$ .
4. **Until:**  $result$  does not change.
5. **Output:** Return  $result$

This algorithm iterates through the steps until no further changes can be made, ensuring a minimal and unified set of functional dependencies.

you-said-1

this is the definition of extraneous attributes: Given  $\alpha \rightarrow \beta$ : •  $a$  belongs to  $\alpha$  is extraneous if  $\beta \subseteq (\alpha - a)^+$  under  $F$  •  $a$  belongs to  $\beta$  is extraneous if  $a$  belongs to  $\alpha^+$  under  $F' = (F - \{\alpha \rightarrow \beta\}) \cup \{\alpha \rightarrow (\beta - a)\}$  Rewrite the above algorithm this this

ChatGPT said:

Here is the algorithm rewritten using the formal definition of extraneous attributes:

1. **Input:** Set of functional dependencies  $FDs$

2. **Initialize:**  $result := FDs$

3. **Repeat:**

- **Step 1:** For every pair of functional dependencies  $\alpha \rightarrow \beta_1$  and  $\alpha \rightarrow \beta_2$  in  $result$ :

- Replace them with a single functional dependency  $\alpha \rightarrow \beta_1 \cup \beta_2$ .

- **Step 2:** For each functional dependency  $\alpha \rightarrow \beta$  in  $result$ :

- Check for **extraneous attribute** in the left-hand side ( $\alpha$ ):

- \* If there exists an attribute  $a \in \alpha$  such that  $\beta \subseteq (\alpha - a)^+$  under the functional dependencies  $F$  (i.e., the closure of  $\alpha - a$  still determines  $\beta$ ), then remove  $a$  from  $\alpha$ .

- **Step 3:** For each functional dependency  $\alpha \rightarrow \beta$  in  $result$ :

- Check for **extraneous attribute** in the right-hand side ( $\beta$ ):

- \* If there exists an attribute  $b \in \beta$  such that  $b \in \alpha^+$  under the modified set of functional dependencies  $F' = (F - \{\alpha \rightarrow \beta\}) \cup \{\alpha \rightarrow (\beta - b)\}$  (i.e.,  $\alpha$  still determines  $\beta - b$ ), then remove  $b$  from  $\beta$ .

4. **Until:**  $result$  does not change.

5. **Output:** Return  $result$

This version of the algorithm incorporates the exact definitions of extraneous attributes, ensuring that unnecessary attributes from both the left-hand and right-hand sides of functional dependencies are removed based on the closure conditions.

you-said-2

Using this algorithm find the canonical cover of theses functional dependencies: class (course , title , department, credits, section, semester, year , building, room, capacity ); FD =  $\{\{course\} \Rightarrow \{title, department, credits\}, \{building, room\} \Rightarrow \{capacity\}, \{course, section, semester, year\} \Rightarrow \{building, room\}\}$ . Algorithm: 1. **Input**: Set of functional dependencies  $\setminus$  ( FDs  $\setminus$  ) 2. **Initialize**:  $\setminus$  ( result := FDs  $\setminus$  ) 3. **Repeat**: - **Step 1**: For every pair of functional dependencies  $\setminus$  (  $\alpha \rightarrow \beta_1$  ) and  $\setminus$  (  $\alpha \rightarrow \beta_2$  ) in  $\setminus$  ( result  $\setminus$  ): - Replace them with a single functional dependency  $\setminus$  (  $\alpha \rightarrow \beta_1 \cup \beta_2$  ). - **Step 2**: For each functional dependency  $\setminus$  (  $\alpha \rightarrow \beta$  ) in  $\setminus$  ( result  $\setminus$  ): - Check for **extraneous attribute** in the left-hand side ( $\setminus$  (  $\alpha$  )): - If there exists an attribute  $\setminus$  ( a  $\in$   $\alpha$  ) such that  $\setminus$  (  $\beta \subseteq \alpha \setminus \{a\}$  ) under the functional dependencies  $\setminus$  ( result  $\setminus$  ) (i.e., the closure of  $\setminus$  (  $\alpha \setminus \{a\}$  ) still determines  $\setminus$  (  $\beta$  )), then remove  $\setminus$  ( a ) from  $\setminus$  (  $\alpha$  ). - **Step 3**: For each functional dependency  $\setminus$  (  $\alpha \rightarrow \beta$  ) in  $\setminus$  ( result  $\setminus$  ): - Check for **extraneous attribute** in the right-hand side ( $\setminus$  (  $\beta$  )): - If there exists an attribute  $\setminus$  ( b  $\in$   $\beta$  ) such that  $\setminus$  ( b  $\in$   $\alpha^+$  ) under the modified set of functional dependencies  $\setminus$  ( result' = (result  $\setminus$   $\setminus$  {  $\alpha \rightarrow \beta$  } )  $\cup$  {  $\alpha \rightarrow (\beta \setminus \{b\})$  } ) (i.e.,  $\setminus$  (  $\alpha$  ) still determines  $\setminus$  (  $\beta \setminus \{b\}$  )), then remove  $\setminus$  ( b ) from  $\setminus$  (  $\beta$  ). 4. **Until**:  $\setminus$  ( result  $\setminus$  ) does not change. 5. **Output**: Return  $\setminus$  ( result  $\setminus$  ) This version of the algorithm incorporates the exact definitions of extraneous attributes, ensuring that unnecessary attributes from both the left-hand and right-hand sides of functional dependencies are removed based on the closure conditions.

ChatGPT said:

Memory updated

Let's find the canonical cover for the given functional dependencies using the algorithm.

step-1-initialize-result-with-the-given-functional-dependencies

### Step 1: Initialize result with the given functional dependencies:

$result = \{\{course\} \rightarrow \{title, department, credits\}, \{building, room\} \rightarrow \{capacity\}, \{course, section, semester, year\} \rightarrow \{building, room\}\}$   
step-2-combine-dependencies-with-the-same-left-hand-side

### Step 2: Combine dependencies with the same left-hand side:

There are no pairs of functional dependencies with the same left-hand side, so no changes are made in this step.

step-3-remove-extraneous-attributes

### Step 3: Remove extraneous attributes:

a.-check-for-extraneous-attributes-on-the-left-hand-side

1.  $\{course\} \rightarrow \{title, department, credits\}$   $\{course\} \rightarrow \{title, department, credits\}$ :

- There is only one attribute on the left-hand side (*course*), so there are no extraneous attributes here.
2.  $\{building, room\} \rightarrow \{capacity\}$  {building,room} $\rightarrow$ {capacity}:
- Check if  $capacity \subseteq (building^+)capacity$  (building+) or  $capacity \subseteq (room^+)capacity$  (room+). Since neither attribute alone can determine *capacity*, no extraneous attributes exist in the left-hand side here.
3.  $\{course, section, semester, year\} \rightarrow \{building, room\}$  {course,section,semester,year} $\rightarrow$ {building,room}:
- We check each attribute:
    - For *course*:  $\{section, semester, year\}^+ \{section, semester, year\}^+$  does not include *building* and *room*, so *course* is not extraneous.
    - For *section*: Similarly,  $\{course, semester, year\}^+ \{course, semester, year\}^+$  does not include *building* and *room*, so *section* is not extraneous.
    - The same holds for *semester* and *year*, meaning no attributes on the left-hand side are extraneous.

b.-check-for-extraneous-attributes-on-the-right-hand-side

1.  $\{course\} \rightarrow \{title, department, credits\}$  {course} $\rightarrow$ {title,department,credits}:
- For *title*, *department*, and *credits*, we check if each attribute is extraneous by considering the modified dependency set and checking the closure:
    - None of these attributes is included in  $course^+course^+$  under the modified functional dependencies, so no extraneous attributes on the right-hand side.
2.  $\{building, room\} \rightarrow \{capacity\}$  {building,room} $\rightarrow$ {capacity}:

- There is only one attribute on the right-hand side (*capacity*), so no extraneous attributes here.

3.  $\{course, section, semester, year\} \rightarrow \{building, room\}$   $\{course, section, semester, year\} \rightarrow \{building, room\}$ :

- Neither *building* nor *room* is extraneous, as neither attribute is implied by  $\{course, section, semester, year\}^+ \{course, section, semester, year\}^+$ .

step-4-result-does-not-change-after-these-steps.

**Step 4: Result does not change after these steps.**

step-5-return-the-canonical-cover

**Step 5: Return the canonical cover:**

The canonical cover of the given functional dependencies is:

$\{\{course\} \rightarrow \{title, department, credits\}, \{building, room\} \rightarrow \{capacity\}, \{course, section, semester, year\} \rightarrow \{building, room\}\}$

$\{\{course\} \rightarrow \{title, department, credits\}, \{building, room\} \rightarrow \{capacity\}, \{course, section, semester, year\} \rightarrow \{building, room\}\}$

No extraneous attributes were found, so the original set of functional dependencies is already the canonical cover.

you-said-3

Using same algorithm find canonical cover for books: books(accessionno, isbn, title, author, publisher); FD =  $\{\{accessionno\} \Rightarrow \{isbn\}, \{isbn\} \Rightarrow \{title\}, \{isbn\} \Rightarrow \{publisher\}, \{isbn\} \Rightarrow \{author\}\}$ .

ChatGPT said:

Memory updated

Let's find the canonical cover for the given functional dependencies using the algorithm.

step-1-initialize-result-with-the-given-functional-dependencies-1

**Step 1: Initialize result with the given functional dependencies:**

$result = \{\{accessionno\} \rightarrow \{isbn\}, \{isbn\} \rightarrow \{title\}, \{isbn\} \rightarrow \{publisher\}, \{isbn\} \rightarrow \{author\}\}$

$result = \{\{accessionno\} \rightarrow \{isbn\}, \{isbn\} \rightarrow \{title\}, \{isbn\} \rightarrow \{publisher\}, \{isbn\} \rightarrow \{author\}\}$

step-2-combine-dependencies-with-the-same-left-hand-side-1

**Step 2: Combine dependencies with the same left-hand side:**

1. The dependencies with *isbn* as the left-hand side:

- We have three separate dependencies:  $\{isbn\} \rightarrow \{title\}$ ,  $\{isbn\} \rightarrow \{publisher\}$ ,  $\{isbn\} \rightarrow \{author\}$
- Combine these into a single functional dependency:

$$\{isbn\} \rightarrow \{title, publisher, author\}$$

$$\{isbn\} \rightarrow \{title, publisher, author\}$$

The updated result is:

$$result = \{\{accessionno\} \rightarrow \{isbn\}, \{isbn\} \rightarrow \{title, publisher, author\}\}$$

$$result = \{\{accessionno\} \rightarrow \{isbn\}, \{isbn\} \rightarrow \{title, publisher, author\}\}$$

step-3-remove-extraneous-attributes-1

### Step 3: Remove extraneous attributes:

a.-check-for-extraneous-attributes-on-the-left-hand-side-1

1.  $\{accessionno\} \rightarrow \{isbn\}$   $\{accessionno\} \rightarrow \{isbn\}$ :

- There is only one attribute on the left-hand side (*accessionno*), so no extraneous attributes here.

2.  $\{isbn\} \rightarrow \{title, publisher, author\}$   $\{isbn\} \rightarrow \{title, publisher, author\}$ :

- There is only one attribute on the left-hand side (*isbn*), so no extraneous attributes here.

b.-check-for-extraneous-attributes-on-the-right-hand-side-1

1.  $\{accessionno\} \rightarrow \{isbn\}$   $\{accessionno\} \rightarrow \{isbn\}$ :

- There is only one attribute on the right-hand side (*isbn*), so no extraneous attributes here.

2.  $\{isbn\} \rightarrow \{title, publisher, author\}$   $\{isbn\} \rightarrow \{title, publisher, author\}$ :

- None of the attributes (*title, publisher, author*) are extraneous because removing any of them would make the functional dependency incomplete and *isbn* would no longer imply the removed attributes.

step-4-result-does-not-change-after-these-steps.-1

**Step 4: Result does not change after these steps.**

step-5-return-the-canonical-cover-1

**Step 5: Return the canonical cover:**

The canonical cover of the given functional dependencies is:

$$\{\{accessionno\} \rightarrow \{isbn\}, \{isbn\} \rightarrow \{title, publisher, author\}\}$$

$\{\{accessionno\} \rightarrow \{isbn\}, \{isbn\} \rightarrow \{title, publisher, author\}\}$

This is the minimal and unified set of functional dependencies for the **books** relation.

you-said-4

find canonical cover of this r4 using the same algorithm:  $r4(A, B, C); FD = \{\{A\} \Rightarrow \{B, C\}, \{B\} \Rightarrow \{C\}, \{A\} \Rightarrow \{B\}, \{A, B\} \Rightarrow \{C\}\}$ .

ChatGPT said:

Memory updated

Let's find the canonical cover for the functional dependencies of the relation  $r4(A, B, C)$  using the algorithm.

step-1-initialize-result-with-the-given-functional-dependencies-2

**Step 1: Initialize result with the given functional dependencies:**

$$result = \{\{A\} \rightarrow \{B, C\}, \{B\} \rightarrow \{C\}, \{A\} \rightarrow \{B\}, \{A, B\} \rightarrow \{C\}\}$$

result =  $\{\{A\} \rightarrow \{B, C\}, \{B\} \rightarrow \{C\}, \{A\} \rightarrow \{B\}, \{A, B\} \rightarrow \{C\}\}$

step-2-combine-dependencies-with-the-same-left-hand-side-2

**Step 2: Combine dependencies with the same left-hand side:**

1. The dependencies  $\{A\} \rightarrow \{B, C\}$  and  $\{A\} \rightarrow \{B\}$ :

- Combine them into one functional dependency:

$$\{A\} \rightarrow \{B, C\}$$

$$\{A\} \rightarrow \{B, C\}$$

The updated result is:

$$result = \{\{A\} \rightarrow \{B, C\}, \{B\} \rightarrow \{C\}, \{A, B\} \rightarrow \{C\}\}$$

$$result = \{\{A\} \rightarrow \{B, C\}, \{B\} \rightarrow \{C\}, \{A, B\} \rightarrow \{C\}\}$$

step-3-remove-extraneous-attributes-2

### Step 3: Remove extraneous attributes:

a.-check-for-extraneous-attributes-on-the-left-hand-side-2

$$1. \{A\} \rightarrow \{B, C\} \{A\} \rightarrow \{B, C\}:$$

- There is only one attribute on the left-hand side ( $AA$ ), so no extraneous attributes here.

$$2. \{B\} \rightarrow \{C\} \{B\} \rightarrow \{C\}:$$

- There is only one attribute on the left-hand side ( $BB$ ), so no extraneous attributes here.

$$3. \{A, B\} \rightarrow \{C\} \{A, B\} \rightarrow \{C\}:$$

- Check if  $C \subseteq (A^+)C(A^+)$ :
  - The closure of  $AA$  under the dependencies already includes  $CC$  because  $A \rightarrow \{B, C\} \Rightarrow A \rightarrow \{B, C\}$ .
  - Hence,  $BB$  is an extraneous attribute.
- Remove  $BB$  from the left-hand side, resulting in  $\{A\} \rightarrow \{C\} \{A\} \rightarrow \{C\}$ .

The updated result is:

$$result = \{\{A\} \rightarrow \{B, C\}, \{B\} \rightarrow \{C\}, \{A\} \rightarrow \{C\}\}$$

$$result = \{\{A\} \rightarrow \{B, C\}, \{B\} \rightarrow \{C\}, \{A\} \rightarrow \{C\}\}$$

b.-check-for-extraneous-attributes-on-the-right-hand-side-2



1.  $\{A\} \rightarrow \{B, C\} \{A\} \rightarrow \{B, C\}$ :

- We check if  $CC$  is extraneous. Since  $A \rightarrow C$  is already present as a separate functional dependency,  $CC$  is extraneous here.
- Remove  $CC$  from the right-hand side, resulting in  $\{A\} \rightarrow \{B\} \{A\} \rightarrow \{B\}$ .

2.  $\{B\} \rightarrow \{C\} \{B\} \rightarrow \{C\}$ :

- No extraneous attributes here since  $B^+B^+$  does not contain  $CC$ .

3.  $\{A\} \rightarrow \{C\} \{A\} \rightarrow \{C\}$ :

- There is only one attribute on the right-hand side ( $CC$ ), so no extraneous attributes here.

The updated result is:

$$result = \{\{A\} \rightarrow \{B\}, \{B\} \rightarrow \{C\}, \{A\} \rightarrow \{C\}\}$$

result= $\{\{A\} \rightarrow \{B\}, \{B\} \rightarrow \{C\}, \{A\} \rightarrow \{C\}\}$

step-4-result-does-not-change-after-these-steps.-2

**Step 4: Result does not change after these steps.**

step-5-return-the-canonical-cover-2

**Step 5: Return the canonical cover:**

The canonical cover of the given functional dependencies is:

$$\{\{A\} \rightarrow \{B\}, \{B\} \rightarrow \{C\}, \{A\} \rightarrow \{C\}\}$$

$\{\{A\} \rightarrow \{B\}, \{B\} \rightarrow \{C\}, \{A\} \rightarrow \{C\}\}$

This is the minimal and unified set of functional dependencies for the relation  $r4(A, B, C)$ .  
you-said-5

Here is the algorithm rewritten using the formal definition of extraneous attributes: 1. **\*\*Input\*\***: Set of functional dependencies  $FDs$  2. **\*\*Initialize\*\***:  $result := FDs$  3. **\*\*Repeat\*\***: \* **\*\*Step 1\*\***: For every pair of functional dependencies  $\rightarrow 1 \setminus \alpha \rightarrow \beta_1$  and  $\rightarrow 2 \setminus \alpha \rightarrow \beta_2$  in  $result$  \* Replace them with a single functional dependency  $\rightarrow 12 \setminus \alpha \rightarrow \beta_1 \cup \beta_2$

$\beta \rightarrow 12$ . \* \*\*Step 2\*\*:

For each functional dependency  $\alpha \rightarrow \beta$  in resultresultresult: \* Check for \*\*extraneous attribute\*\* in the left-hand side ( $\alpha$ ): \* If there exists an attribute  $a$  in  $\alpha$  such that  $(\alpha - a) \cup \beta \subseteq \alpha$  under the functional dependencies  $F^+$  (i.e., the closure of  $\alpha - a$  still determines  $\beta$ ), then remove  $a$  from  $\alpha$ . \* \*\*Step 3\*\*:

For each functional dependency  $\alpha \rightarrow \beta$  in resultresultresult: \* Check for \*\*extraneous attribute\*\* in the right-hand side ( $\beta$ ): \* If there exists an attribute  $b$  in  $\beta$  such that  $\alpha \rightarrow \beta - b$  under the modified set of functional dependencies  $F' = (F - \{\alpha \rightarrow \beta\}) \cup \{\alpha \rightarrow \beta - b\}$  (i.e.,  $\alpha$  still determines  $\beta - b$ ), then remove  $b$  from  $\beta$ . 4. \*\*Until\*\*:

resultresultresult does not change. 5. \*\*Output\*\*:

Return resultresultresult This version of the algorithm incorporates the exact definitions of extraneous attributes, ensuring that unnecessary attributes from both the left-hand and right-hand sides of functional dependencies are removed based on the closure conditions.

write it down in a pdf file

ChatGPT said:

I have created the PDF for the canonical cover of the functional dependencies for relation  $r_4$ . You can download it using the link below:

[Download the PDF](#)

you-said-6

for the canonical cover of  $r_4$  the algorithm repeats step 1 to step 3 until there is not change

ChatGPT said:

Yes, you're correct! The algorithm repeats steps 1 through 3 until no further changes occur in the functional dependencies. In this case, I applied the steps and reached the final canonical cover directly because no further combinations or extraneous attributes were found after the first cycle.

Let's clarify how the algorithm works:

algorithm-recap

### Algorithm Recap:

1. **Initialize:** Start with the given set of functional dependencies.
2. **Step 1 (Combining FDs):** For each pair of functional dependencies with the same left-hand side, combine them into a single FD.
3. **Step 2 (Left-side Extraneous Attributes):** Check for extraneous attributes on the left-hand side. If an attribute can be removed without changing the closure,

remove it.

4. **Step 3 (Right-side Extraneous Attributes):** Check for extraneous attributes on the right-hand side. If an attribute can be removed without changing the closure, remove it.
5. **Repeat:** Keep repeating steps 1-3 until the result stabilizes (i.e., no more changes).

In the canonical cover for  $r4r4$ , the algorithm converged after one pass through steps 1-3, so no further changes were necessary. However, in other cases, this process might need to iterate multiple times before reaching a stable result.

Would you like me to make any further clarifications or changes to the explanation or the PDF?

you-said-7

can step 1 be applied on this?  $\{\{A\} \rightarrow \{B\}, \{B\} \rightarrow \{C\}, \{A\} \rightarrow \{C\}\}$

ChatGPT said:

Let's analyze whether **Step 1** (combining functional dependencies with the same left-hand side) can be applied to the set of functional dependencies:

$$FD = \{\{A\} \rightarrow \{B\}, \{B\} \rightarrow \{C\}, \{A\} \rightarrow \{C\}\}$$

$$FD = \{\{A\} \rightarrow \{B\}, \{B\} \rightarrow \{C\}, \{A\} \rightarrow \{C\}\}$$

step-1-combine-dependencies-with-the-same-left-hand-side

### Step 1: Combine dependencies with the same left-hand side

- **Check:** We have two functional dependencies with the same left-hand side:

1.  $\{A\} \rightarrow \{B\}$   $\{A\} \rightarrow \{B\}$
2.  $\{A\} \rightarrow \{C\}$   $\{A\} \rightarrow \{C\}$

- **Action:** We can **combine** them into a single functional dependency:

$$\{A\} \rightarrow \{B, C\}$$

$$\{A\} \rightarrow \{B, C\}$$

- **Result after Step 1:**

$$FD = \{\{A\} \rightarrow \{B, C\}, \{B\} \rightarrow \{C\}\}$$

$$FD = \{\{A\} \rightarrow \{B, C\}, \{B\} \rightarrow \{C\}\}$$

conclusion

## Conclusion

Yes, **Step 1** can be applied here to combine the dependencies  $\{A\} \rightarrow \{B\}$  and  $\{A\} \rightarrow \{C\}$  into  $\{A\} \rightarrow \{B, C\}$ . After this combination, the new set of functional dependencies is simplified and should be used for further steps in the canonical cover algorithm.

Let me know if you'd like to see the complete canonical cover for this set of functional dependencies!

1/2

you-said-8

Using this Algorithm: 1. **Input**: Set of functional dependencies  $\{FDs\}$  2. **Initialize**:  $result := FDs$  3. **Repeat**: - **Step 1**: For every pair of functional dependencies  $\alpha \rightarrow \beta_1$  and  $\alpha \rightarrow \beta_2$  in  $result$ : - Replace them with a single functional dependency  $\alpha \rightarrow \beta_1 \cup \beta_2$ . - **Step 2**: For each functional dependency  $\alpha \rightarrow \beta$  in  $result$ : - Check for **extraneous attribute** in the left-hand side ( $\alpha$ ): - If there exists an attribute  $a \in \alpha$  such that  $\beta \subseteq \alpha \setminus \{a\}$  under the functional dependencies  $result$  (i.e., the closure of  $\alpha \setminus \{a\}$  still determines  $\beta$ ), then remove  $a$  from  $\alpha$ . - **Step 3**: For each functional dependency  $\alpha \rightarrow \beta$  in  $result$ : - Check for **extraneous attribute** in the right-hand side ( $\beta$ ): - If there exists an attribute  $b \in \beta$  such that  $b \in \alpha^+$  under the modified set of functional dependencies  $result' = (result \setminus \{\alpha \rightarrow \beta\}) \cup \{\alpha \rightarrow (\beta \setminus \{b\})\}$  (i.e.,  $\alpha$  still determines  $\beta \setminus \{b\}$ ), then remove  $b$  from  $\beta$ . 4. **Until**:  $result$  does not change. 5. **Output**: Return  $result$  Find canonical cover of  $r_4(A, B, C)$  where  $FDs = \{\{A\} \rightarrow \{B, C\}, \{B\} \rightarrow \{C\}, \{A\} \rightarrow \{B\}, \{A, B\} \rightarrow \{C\}\}$

2/2

ChatGPT said:

Let's use the given algorithm to find the **canonical cover** of the functional dependencies for  $r_4(A, B, C)$ .

functional-dependencies

## Functional Dependencies:

$$FD = \{\{A\} \rightarrow \{B, C\}, \{B\} \rightarrow \{C\}, \{A\} \rightarrow \{B\}, \{A, B\} \rightarrow \{C\}\}$$

$$FD = \{\{A\} \rightarrow \{B, C\}, \{B\} \rightarrow \{C\}, \{A\} \rightarrow \{B\}, \{A, B\} \rightarrow \{C\}\}$$

step-by-step-execution

**Step-by-Step Execution:**

step-1-combine-dependencies-with-the-same-left-hand-side-1

We have:

- $\{A\} \rightarrow \{B, C\}$   $\{A\} \rightarrow \{B, C\}$
- $\{A\} \rightarrow \{B\}$   $\{A\} \rightarrow \{B\}$

Combine these two dependencies into one:

$$\{A\} \rightarrow \{B, C\}$$

$$\{A\} \rightarrow \{B, C\}$$

No other dependencies have the same left-hand side, so after this step, the set becomes:

$$result = \{\{A\} \rightarrow \{B, C\}, \{B\} \rightarrow \{C\}, \{A, B\} \rightarrow \{C\}\}$$

$$result = \{\{A\} \rightarrow \{B, C\}, \{B\} \rightarrow \{C\}, \{A, B\} \rightarrow \{C\}\}$$

step-2-check-for-extraneous-attributes-on-the-left-hand-side

- **For**  $\{A, B\} \rightarrow \{C\}$   **$\{A, B\} \rightarrow \{C\}$ :**

- We need to check if  $BB$  is extraneous.
- Compute the closure of  $\{A\}$  under the current dependencies:

$$\{A\}^+ = \{A, B, C\}$$

$$\{A\}^+ = \{A, B, C\}$$

- Since  $\{A\}^+ \{A\}^+$  already contains  $CC$ ,  $BB$  is **extraneous** in  $\{A, B\} \rightarrow \{C\}$   $\{A, B\} \rightarrow \{C\}$ .
- Remove  $BB$  from the left-hand side, so we replace  $\{A, B\} \rightarrow \{C\}$   $\{A, B\} \rightarrow \{C\}$  with  $\{A\} \rightarrow \{C\}$   $\{A\} \rightarrow \{C\}$ .

After this step:

$$result = \{\{A\} \rightarrow \{B, C\}, \{B\} \rightarrow \{C\}, \{A\} \rightarrow \{C\}\}$$

$$result = \{\{A\} \rightarrow \{B, C\}, \{B\} \rightarrow \{C\}, \{A\} \rightarrow \{C\}\}$$

step-3-check-for-extraneous-attributes-on-the-right-hand-side

• **For**  $\{A\} \rightarrow \{B, C\} \{A\} \rightarrow \{B, C\}$ :

- Check if  $CC$  is extraneous by seeing if  $C \in \{A\}^+C\{A\}^+$  without  $CC$ .
- We already have  $\{A\} \rightarrow \{C\} \{A\} \rightarrow \{C\}$  as a separate dependency, so  $CC$  is **extraneous**.
- Remove  $CC$ , so  $\{A\} \rightarrow \{B, C\} \{A\} \rightarrow \{B, C\}$  becomes  $\{A\} \rightarrow \{B\} \{A\} \rightarrow \{B\}$ .

After this step:

$$result = \{\{A\} \rightarrow \{B\}, \{B\} \rightarrow \{C\}, \{A\} \rightarrow \{C\}\}$$

result= $\{\{A\} \rightarrow \{B\}, \{B\} \rightarrow \{C\}, \{A\} \rightarrow \{C\}\}$   
 step-4-check-if-the-result-changes

At this point, no further changes can be made.  
 final-canonical-cover

**Final Canonical Cover:**

$$\{\{A\} \rightarrow \{B\}, \{B\} \rightarrow \{C\}, \{A\} \rightarrow \{C\}\}$$

$$\{\{A\} \rightarrow \{B\}, \{B\} \rightarrow \{C\}, \{A\} \rightarrow \{C\}\}$$

This is the **canonical cover** of the functional dependencies for  $r_4(A, B, C)r_4(A, B, C)$ .  
 you-said-9

using this algorithm find canonical cover of these functional dependencies:  $\{\{A\} \rightarrow \{B\}, \{B\} \rightarrow \{C\}, \{A\} \rightarrow \{C\}\}$   
 ChatGPT said:

Let's apply the algorithm step-by-step to find the **canonical cover** for the given set of functional dependencies:

$$FD = \{\{A\} \rightarrow \{B\}, \{B\} \rightarrow \{C\}, \{A\} \rightarrow \{C\}\}$$

FD= $\{\{A\} \rightarrow \{B\}, \{B\} \rightarrow \{C\}, \{A\} \rightarrow \{C\}\}$   
 step-by-step-execution-1

**Step-by-Step Execution:**

step-1-combine-dependencies-with-the-same-left-hand-side-2

We have the following functional dependencies:

1.  $\{A\} \rightarrow \{B\} \{A\} \rightarrow \{B\}$
2.  $\{B\} \rightarrow \{C\} \{B\} \rightarrow \{C\}$
3.  $\{A\} \rightarrow \{C\} \{A\} \rightarrow \{C\}$

There are **no** dependencies with the same left-hand side that we can combine. So, after **Step 1**, the functional dependencies remain the same:

$$result = \{\{A\} \rightarrow \{B\}, \{B\} \rightarrow \{C\}, \{A\} \rightarrow \{C\}\}$$

$$result = \{\{A\} \rightarrow \{B\}, \{B\} \rightarrow \{C\}, \{A\} \rightarrow \{C\}\}$$

step-2-check-for-extraneous-attributes-on-the-left-hand-side-1

- **For**  $\{A\} \rightarrow \{B\} \{A\} \rightarrow \{B\}$ :

- The left-hand side  $\{A\}\{A\}$  has only one attribute, so no extraneous attributes are possible here.

- **For**  $\{B\} \rightarrow \{C\} \{B\} \rightarrow \{C\}$ :

- The left-hand side  $\{B\}\{B\}$  has only one attribute, so no extraneous attributes are possible here.

- **For**  $\{A\} \rightarrow \{C\} \{A\} \rightarrow \{C\}$ :

- The left-hand side  $\{A\}\{A\}$  has only one attribute, so no extraneous attributes are possible here.

No attributes on the left-hand side can be removed.

step-3-check-for-extraneous-attributes-on-the-right-hand-side-1

- **For**  $\{A\} \rightarrow \{B\} \{A\} \rightarrow \{B\}$ :

- There are no extraneous attributes on the right-hand side, as it contains only one attribute  $B$ .

- **For**  $\{B\} \rightarrow \{C\} \{B\} \rightarrow \{C\}$ :

- There are no extraneous attributes on the right-hand side, as it contains only one attribute  $C$ .

- **For**  $\{A\} \rightarrow \{C\} \{A\} \rightarrow \{C\}$ :

- The right-hand side contains only  $C$ , and since  $\{A\} \rightarrow \{C\} \{A\} \rightarrow \{C\}$  is not implied by  $\{A\} \rightarrow \{B\} \{A\} \rightarrow \{B\}$  and  $\{B\} \rightarrow \{C\} \{B\} \rightarrow \{C\}$  directly (closure of  $AA$  includes  $CC$ , but this FD is needed explicitly), no attributes are extraneous here.

step-4-check-if-the-result-changes-1

After going through all steps, there are no changes. Therefore, the result does not change further.

final-canonical-cover-1

### Final Canonical Cover:

$$\{\{A\} \rightarrow \{B\}, \{B\} \rightarrow \{C\}, \{A\} \rightarrow \{C\}\}$$

$$\{\{A\} \rightarrow \{B\}, \{B\} \rightarrow \{C\}, \{A\} \rightarrow \{C\}\}$$

In this case, the given functional dependencies are already in their **canonical cover**. No further simplifications were necessary.

you-said-10

There are **no** dependencies with the same left-hand side that we can combine. So, after **Step 1**, the functional dependencies remain the same:

How are there no dependencies with the same left-hand side?

ChatGPT said:

You are right to point out that the previous statement was incorrect. In the given functional dependencies:



$$FD = \{\{A\} \rightarrow \{B\}, \{B\} \rightarrow \{C\}, \{A\} \rightarrow \{C\}\}$$

$$FD = \{\{A\} \rightarrow \{B\}, \{B\} \rightarrow \{C\}, \{A\} \rightarrow \{C\}\}$$

**Step 1** should have identified that two dependencies share the same left-hand side:

1.  $\{A\} \rightarrow \{B\} \{A\} \rightarrow \{B\}$
2.  $\{A\} \rightarrow \{C\} \{A\} \rightarrow \{C\}$

Since both have  $\{A\}$  as the left-hand side, we can **combine** them into a single functional dependency:

$$\{A\} \rightarrow \{B, C\}$$

$$\{A\} \rightarrow \{B, C\}$$

So, after **Step 1**, the dependencies become:

$$result = \{\{A\} \rightarrow \{B, C\}, \{B\} \rightarrow \{C\}\}$$

$$result = \{\{A\} \rightarrow \{B, C\}, \{B\} \rightarrow \{C\}\}$$

Now, the steps for checking extraneous attributes can continue as before.

updated-canonical-cover-process

### Updated Canonical Cover Process:

step-2-check-for-extraneous-attributes-on-the-left-hand-side-2

- **For**  $\{A\} \rightarrow \{B, C\} \{A\} \rightarrow \{B, C\}$ :

- The left-hand side  $\{A\}$  has only one attribute, so no extraneous attributes can be removed.

- **For**  $\{B\} \rightarrow \{C\} \{B\} \rightarrow \{C\}$ :

- The left-hand side  $\{B\}$  has only one attribute, so no extraneous attributes can be removed.

step-3-check-for-extraneous-attributes-on-the-right-hand-side-2

• **For**  $\{A\} \rightarrow \{B, C\} \{A\} \rightarrow \{B, C\}$ :

- Check if  $CC$  is extraneous in  $\{A\} \rightarrow \{B, C\} \{A\} \rightarrow \{B, C\}$ . To do this, we need to see if  $A^+A^+$  without  $CC$  still contains  $CC$ .
- $A^+ = \{A, B, C\} A^+ = \{A, B, C\}$ , but if we remove  $CC$ , we only have  $\{A\} \rightarrow \{B\} \{A\} \rightarrow \{B\}$ .
- However, since  $\{B\} \rightarrow \{C\} \{B\} \rightarrow \{C\}$  is a separate functional dependency,  $CC$  can still be derived from  $BB$ .
- Therefore,  $CC$  is **extraneous**, and we can remove it.
- The functional dependency  $\{A\} \rightarrow \{B, C\} \{A\} \rightarrow \{B, C\}$  simplifies to  $\{A\} \rightarrow \{B\} \{A\} \rightarrow \{B\}$ .

step-4-check-if-the-result-changes-2

After the previous steps, the final set of functional dependencies becomes:

$$result = \{\{A\} \rightarrow \{B\}, \{B\} \rightarrow \{C\}\}$$

result= $\{\{A\} \rightarrow \{B\}, \{B\} \rightarrow \{C\}\}$

No further changes are possible.

final-canonical-cover-2

**Final Canonical Cover:**

$$\{\{A\} \rightarrow \{B\}, \{B\} \rightarrow \{C\}\}$$

$\{\{A\} \rightarrow \{B\}, \{B\} \rightarrow \{C\}\}$

Thus, the canonical cover of the functional dependencies is  $\{\{A\} \rightarrow \{B\}, \{B\} \rightarrow \{C\}\} \{\{A\} \rightarrow \{B\}, \{B\} \rightarrow \{C\}\}$   
you-said-11

Using the same formula find canonical cover of functional dependencies:  $FD = \{ \{A, B, H\} \rightarrow \{C\}, \{A\} \rightarrow \{D, E\}, \{B, G, H\} \rightarrow \{K\}, \{K\} \rightarrow \{A, D, H\}, \{B, H\} \rightarrow \{G, E\} \}$

ChatGPT said:

Let's apply the **canonical cover algorithm** step by step to the functional dependencies:

given-functional-dependencies-fd

### Given Functional Dependencies (FD):

$$FD = \{\{A, B, H\} \rightarrow \{C\}, \{A\} \rightarrow \{D, E\}, \{B, G, H\} \rightarrow \{K\}, \{K\} \rightarrow \{A, D, H\}, \{B, H\} \rightarrow \{G, E\}\}$$

$$FD = \{\{A, B, H\} \rightarrow \{C\}, \{A\} \rightarrow \{D, E\}, \{B, G, H\} \rightarrow \{K\}, \{K\} \rightarrow \{A, D, H\}, \{B, H\} \rightarrow \{G, E\}\}$$

step-by-step-execution-2

### Step-by-Step Execution:

step-1-combine-dependencies-with-the-same-left-hand-side-3

- We inspect the dependencies to see if any share the same left-hand side.
- None of the dependencies have exactly the same left-hand side, so there is **no combination** needed at this stage.

After Step 1:

$$result = \{\{A, B, H\} \rightarrow \{C\}, \{A\} \rightarrow \{D, E\}, \{B, G, H\} \rightarrow \{K\}, \{K\} \rightarrow \{A, D, H\}, \{B, H\} \rightarrow \{G, E\}\}$$

$$result = \{\{A, B, H\} \rightarrow \{C\}, \{A\} \rightarrow \{D, E\}, \{B, G, H\} \rightarrow \{K\}, \{K\} \rightarrow \{A, D, H\}, \{B, H\} \rightarrow \{G, E\}\}$$

step-2-check-for-extraneous-attributes-on-the-left-hand-side-3

- **For**  $\{A, B, H\} \rightarrow \{C\}$   **$\{A, B, H\} \rightarrow \{C\}$ :**
  - We check if any attribute on the left-hand side is extraneous by calculating the closure of the remaining attributes.
  - Check  $(B, H)^+(B, H)^+$  under the current set of functional dependencies.  $(B, H)^+(B, H)^+$  does not include  $CC$ , so  $AA$  is **not extraneous**.
  - Check  $(A, H)^+(A, H)^+$ .  $(A, H)^+(A, H)^+$  also does not include  $CC$ , so  $BB$  is **not extraneous**.
  - Check  $(A, B)^+(A, B)^+$ . Similarly,  $CC$  is not in  $(A, B)^+(A, B)^+$ , so  $HH$  is **not extraneous**.
  - No extraneous attributes are found in  $\{A, B, H\} \rightarrow \{C\}$   **$\{A, B, H\} \rightarrow \{C\}$ .**
- **For**  $\{A\} \rightarrow \{D, E\}$   **$\{A\} \rightarrow \{D, E\}$ :**

- The left-hand side  $AA$  has only one attribute, so no extraneous attributes can be removed.

- **For**  $\{B, G, H\} \rightarrow \{K\} \{B, G, H\} \rightarrow \{K\}$ :

- Check  $(G, H)^+(G, H)^+$ . It does not contain  $KK$ , so  $BB$  is **not extraneous**.
- Check  $(B, H)^+(B, H)^+$ . This does not contain  $KK$ , so  $GG$  is **not extraneous**.
- Check  $(B, G)^+(B, G)^+$ . This does not contain  $KK$ , so  $HH$  is **not extraneous**.
- No extraneous attributes are found in  $\{B, G, H\} \rightarrow \{K\} \{B, G, H\} \rightarrow \{K\}$ .

- **For**  $\{K\} \rightarrow \{A, D, H\} \{K\} \rightarrow \{A, D, H\}$ :

- The left-hand side  $KK$  has only one attribute, so no extraneous attributes can be removed.

- **For**  $\{B, H\} \rightarrow \{G, E\} \{B, H\} \rightarrow \{G, E\}$ :

- Check  $(H)^+(H)^+$ . This does not include  $G, EG, E$ , so  $BB$  is **not extraneous**.
- Check  $(B)^+(B)^+$ . This does not include  $G, EG, E$ , so  $HH$  is **not extraneous**.
- No extraneous attributes are found in  $\{B, H\} \rightarrow \{G, E\} \{B, H\} \rightarrow \{G, E\}$ .

step-3-check-for-extraneous-attributes-on-the-right-hand-side-3

- **For**  $\{A, B, H\} \rightarrow \{C\} \{A, B, H\} \rightarrow \{C\}$ :

- The right-hand side contains only one attribute, so no extraneous attributes can be removed.

- **For**  $\{A\} \rightarrow \{D, E\} \{A\} \rightarrow \{D, E\}$ :

- Check if  $DD$  or  $EE$  is extraneous.
- Check  $A^+A^+$  without  $DD$ .  $A^+A^+$  is  $\{A, D, E\} \{A, D, E\}$ , so removing  $DD$  would not still yield  $DD$ , so it is **not extraneous**.

- Check  $A^+A^+$  without  $EE$ .  $A^+A^+$  is still  $\{A, D, E\}$ , so removing  $EE$  would not yield  $EE$ , so it is **not extraneous**.
- No extraneous attributes on the right-hand side here.
- **For  $\{B, G, H\} \rightarrow \{K\}$   $\{\mathbf{B}, \mathbf{G}, \mathbf{H}\} \rightarrow \{\mathbf{K}\}$ :**
  - The right-hand side contains only one attribute, so no extraneous attributes can be removed.
- **For  $\{K\} \rightarrow \{A, D, H\}$   $\{\mathbf{K}\} \rightarrow \{\mathbf{A}, \mathbf{D}, \mathbf{H}\}$ :**
  - Check if  $AA$ ,  $DD$ , or  $HH$  is extraneous.
  - Check  $K^+K^+$  without  $AA$ . It does not contain  $AA$ , so  $AA$  is **not extraneous**.
  - Check  $K^+K^+$  without  $DD$ . It does not contain  $DD$ , so  $DD$  is **not extraneous**.
  - Check  $K^+K^+$  without  $HH$ . It contains  $HH$ , so  $HH$  is **extraneous** and can be removed from the right-hand side.
  - The dependency becomes:  $\{K\} \rightarrow \{A, D\}$   $\{\mathbf{K}\} \rightarrow \{\mathbf{A}, \mathbf{D}\}$ .
- **For  $\{B, H\} \rightarrow \{G, E\}$   $\{\mathbf{B}, \mathbf{H}\} \rightarrow \{\mathbf{G}, \mathbf{E}\}$ :**
  - Check if  $GG$  or  $EE$  is extraneous.
  - Check  $(B, H)^+(B, H)^+$  without  $GG$ . It does not contain  $GG$ , so  $GG$  is **not extraneous**.
  - Check  $(B, H)^+(B, H)^+$  without  $EE$ . It does not contain  $EE$ , so  $EE$  is **not extraneous**.

step-4-check-if-the-result-changes-3

After removing extraneous attributes, the new set of functional dependencies is:

$$result = \{\{A, B, H\} \rightarrow \{C\}, \{A\} \rightarrow \{D, E\}, \{B, G, H\} \rightarrow \{K\}, \{K\} \rightarrow \{A, D\}, \{B, H\} \rightarrow \{G, E\}\}$$

$$result = \{\{A, B, H\} \rightarrow \{C\}, \{A\} \rightarrow \{D, E\}, \{B, G, H\} \rightarrow \{K\}, \{K\} \rightarrow \{A, D\}, \{B, H\} \rightarrow \{G, E\}\}$$

No further changes are possible.

final-canonical-cover-3

**Final Canonical Cover:**

$$\{\{A, B, H\} \rightarrow \{C\}, \{A\} \rightarrow \{D, E\}, \{B, G, H\} \rightarrow \{K\}, \{K\} \rightarrow \{A, D\}, \{B, H\} \rightarrow \{G, E\}\}$$

$$\{\{A, B, H\} \rightarrow \{C\}, \{A\} \rightarrow \{D, E\}, \{B, G, H\} \rightarrow \{K\}, \{K\} \rightarrow \{A, D\}, \{B, H\} \rightarrow \{G, E\}\}$$

This is the canonical cover for the given set of functional dependencies.

you-said-12

what is closure  $(B, H)^+$  for functional dependencies:  $FD = \{ \{A, B, H\} \rightarrow \{C\}, \{A\} \rightarrow \{D, E\}, \{B, G, H\} \rightarrow \{K\}, \{K\} \rightarrow \{A, D, H\}, \{B, H\} \rightarrow \{G, E\} \}$

ChatGPT said:

To find the closure of  $(B, H)^+$  under the given set of functional dependencies, we proceed step-by-step by using the functional dependencies to infer all attributes that can be determined from  $BB$  and  $HH$ .

given-functional-dependencies-fd-1

**Given Functional Dependencies (FD):**

$$FD = \{\{A, B, H\} \rightarrow \{C\}, \{A\} \rightarrow \{D, E\}, \{B, G, H\} \rightarrow \{K\}, \{K\} \rightarrow \{A, D, H\}, \{B, H\} \rightarrow \{G, E\}\}$$

$$FD = \{\{A, B, H\} \rightarrow \{C\}, \{A\} \rightarrow \{D, E\}, \{B, G, H\} \rightarrow \{K\}, \{K\} \rightarrow \{A, D, H\}, \{B, H\} \rightarrow \{G, E\}\}$$

closure-of-bhbh

**Closure of  $(B, H)^+$  (B,H)+: Closure of  $(B, H)^+ \{+\} (B, H)^+$ :**

- Start with  $\{B, H\} \{B, H\}$ :**

- Initially,  $(B, H)^+ = \{B, H\} (B, H)^+ = \{B, H\}$ .

- Apply functional dependencies:**

- From  $\{B, H\} \rightarrow \{G, E\} \{B, H\} \rightarrow \{G, E\}$ :**

- We can add  $GG$  and  $EE$  to the closure.
- Now,  $(B, H)^+ = \{B, H, G, E\} (B, H)^+ = \{B, H, G, E\}$ .

- From  $\{B, G, H\} \rightarrow \{K\} \{B, G, H\} \rightarrow \{K\}$ :**

- Since  $GG$  is already in the closure, we can apply this dependency and add  $KK$  to the closure.
- Now,  $(B, H)^+ = \{B, H, G, E, K\}$   $(B, H)^+ = \{B, H, G, E, K\}$ .
- **From  $\{K\} \rightarrow \{A, D, H\}$   $\{K\} \rightarrow \{A, D, H\}$ :**
  - Since  $KK$  is now in the closure, we can apply this dependency and add  $AA$  and  $DD$  to the closure (note that  $HH$  is already in the closure).
  - Now,  $(B, H)^+ = \{B, H, G, E, K, A, D\}$   $(B, H)^+ = \{B, H, G, E, K, A, D\}$ .
- **From  $\{A\} \rightarrow \{D, E\}$   $\{A\} \rightarrow \{D, E\}$ :**
  - Since  $AA$  is now in the closure, we can apply this dependency. However, both  $DD$  and  $EE$  are already in the closure, so no new attributes are added.
- **From  $\{A, B, H\} \rightarrow \{C\}$   $\{A, B, H\} \rightarrow \{C\}$ :**
  - Since  $AA$ ,  $BB$ , and  $HH$  are all in the closure, we can apply this dependency and add  $CC$  to the closure.
  - Now,  $(B, H)^+ = \{B, H, G, E, K, A, D, C\}$   $(B, H)^+ = \{B, H, G, E, K, A, D, C\}$ .

final-closure

### Final Closure:

$$(B, H)^+ = \{B, H, G, E, K, A, D, C\}$$

$$(B, H)^+ = \{B, H, G, E, K, A, D, C\}$$

So, the closure of  $(B, H)$   $(B, H)$  under the given functional dependencies is  $\{B, H, G, E, K, A, D, C\}$   $\{B, H, G, E, K, A, D, C\}$

you-said-13  
give closure of  $(A, H)^+$  ,  $(A, B)^+$  ,  $(D)^+$  ,  $(E)^+$  ,  $(B)^+$  ,  $(H)^+$

ChatGPT said:

Let's compute the closures one by one for the given sets of attributes under the same functional dependencies.

given-functional-dependencies-fd-2

### Given Functional Dependencies (FD):

$$FD = \{\{A, B, H\} \rightarrow \{C\}, \{A\} \rightarrow \{D, E\}, \{B, G, H\} \rightarrow \{K\}, \{K\} \rightarrow \{A, D, H\}, \{B, H\} \rightarrow \{G, E\}\}$$

$$FD = \{\{A, B, H\} \rightarrow \{C\}, \{A\} \rightarrow \{D, E\}, \{B, G, H\} \rightarrow \{K\}, \{K\} \rightarrow \{A, D, H\}, \{B, H\} \rightarrow \{G, E\}\}$$

closure-of-ahah

**1. Closure of  $(A, H)^+(\mathbf{A}, \mathbf{H})$ : 1. Closure of  $(\mathbf{A}, \mathbf{H})^{\wedge \{+\}}(\mathbf{A}, \mathbf{H})$ :**

**1. Start with  $\{A, H\}\{\mathbf{A}, \mathbf{H}\}$ :**

- Initially,  $(A, H)^+ = \{A, H\}(\mathbf{A}, \mathbf{H}) = \{\mathbf{A}, \mathbf{H}\}$ .

**2. Apply functional dependencies:**

- From  $\{A\} \rightarrow \{D, E\} \{\mathbf{A}\} \rightarrow \{\mathbf{D}, \mathbf{E}\}$ :**

- Since  $AA$  is in the closure, add  $DD$  and  $EE$  to the closure.
- Now,  $(A, H)^+ = \{A, H, D, E\}(\mathbf{A}, \mathbf{H}) = \{\mathbf{A}, \mathbf{H}, \mathbf{D}, \mathbf{E}\}$ .

- No other dependencies can be applied.

final-closure-of-ahah

**Final closure of  $(A, H)^+(\mathbf{A}, \mathbf{H})$ : Final closure of  $(\mathbf{A}, \mathbf{H})^{\wedge \{+\}}(\mathbf{A}, \mathbf{H})$ :**

$$(A, H)^+ = \{A, H, D, E\}$$

$$(\mathbf{A}, \mathbf{H}) = \{\mathbf{A}, \mathbf{H}, \mathbf{D}, \mathbf{E}\}$$

---

closure-of-abab

**2. Closure of  $(A, B)^+(\mathbf{A}, \mathbf{B})$ : 2. Closure of  $(\mathbf{A}, \mathbf{B})^{\wedge \{+\}}(\mathbf{A}, \mathbf{B})$ :**

**1. Start with  $\{A, B\}\{\mathbf{A}, \mathbf{B}\}$ :**

- Initially,  $(A, B)^+ = \{A, B\}(\mathbf{A}, \mathbf{B}) = \{\mathbf{A}, \mathbf{B}\}$ .

**2. Apply functional dependencies:**

- From  $\{A\} \rightarrow \{D, E\} \{\mathbf{A}\} \rightarrow \{\mathbf{D}, \mathbf{E}\}$ :**



- Since  $AA$  is in the closure, add  $DD$  and  $EE$ .
- Now,  $(A, B)^+ = \{A, B, D, E\}$
- **From  $\{A, B, H\} \rightarrow \{C\}$   $\{A, B, H\} \rightarrow \{C\}$ :**
  - $HH$  is not yet in the closure, so this dependency cannot be applied.
- No other dependencies can be applied.

final-closure-of-abab

**Final closure of  $(A, B)^+$   $\{A, B, D, E\}$ : Final closure of  $(A, B)^+ \{A, B, D, E\}$ :**

$$(A, B)^+ = \{A, B, D, E\}$$

$$(A, B)^+ = \{A, B, D, E\}$$

---

closure-of-dd

**3. Closure of  $(D)^+$   $\{D\}$ : 3. Closure of  $(D)^+ \{D\}$ :**

1. **Start with  $\{D\}$   $\{D\}$ :**

- Initially,  $(D)^+ = \{D\}$

2. **No dependencies can be applied** where  $DD$  is on the left-hand side.

final-closure-of-dd

**Final closure of  $(D)^+$   $\{D\}$ : Final closure of  $(D)^+ \{D\}$ :**

$$(D)^+ = \{D\}$$

$$(D)^+ = \{D\}$$

---

closure-of-ee

#### 4. Closure of $(E)^+(\mathbf{E})$ : 4. Closure of $(\mathbf{E})^{\wedge\{+\}}(\mathbf{E})$ :

##### 1. Start with $\{E\}\{\mathbf{E}\}$ :

- Initially,  $(E)^+ = \{E\}(\mathbf{E}) = \{\mathbf{E}\}$ .

##### 2. No dependencies can be applied where $EE$ is on the left-hand side.

final-closure-of-ee

#### Final closure of $(E)^+(\mathbf{E})$ : Final closure of $(\mathbf{E})^{\wedge\{+\}}(\mathbf{E})$ :

$$(E)^+ = \{E\}$$

$$(\mathbf{E}) = \{\mathbf{E}\}$$

---

closure-of-bb

#### 5. Closure of $(B)^+(\mathbf{B})$ : 5. Closure of $(\mathbf{B})^{\wedge\{+\}}(\mathbf{B})$ :

##### 1. Start with $\{B\}\{\mathbf{B}\}$ :

- Initially,  $(B)^+ = \{B\}(\mathbf{B}) = \{\mathbf{B}\}$ .

##### 2. Apply functional dependencies:

- From  $\{B, H\} \rightarrow \{G, E\}$   $\{\mathbf{B}, \mathbf{H}\} \rightarrow \{\mathbf{G}, \mathbf{E}\}$ :**

–  $HH$  is not yet in the closure, so this dependency cannot be applied.

- From  $\{B, G, H\} \rightarrow \{K\}$   $\{\mathbf{B}, \mathbf{G}, \mathbf{H}\} \rightarrow \{\mathbf{K}\}$ :**

–  $GG$  and  $HH$  are not yet in the closure, so this dependency cannot be applied.

- No other dependencies can be applied.**

final-closure-of-bb

**Final closure of  $(B)^+(B)^+$ : Final closure of  $(B)^{+*}(B)^+$ :**

$$(B)^+ = \{B\}$$

$$(B)^+ = \{B\}$$

---

closure-of-hh

**6. Closure of  $(H)^+(H)^+$ : 6. Closure of  $(H)^{+*}(H)^+$ :**

1. **Start with  $\{H\}^+(H)^+$ :**

- Initially,  $(H)^+ = \{H\}^+(H)^+ = \{H\}$ .

2. **No dependencies can be applied** where  $HH$  is on the left-hand side.

final-closure-of-hh

**Final closure of  $(H)^+(H)^+$ : Final closure of  $(H)^{+*}(H)^+$ :**

$$(H)^+ = \{H\}$$

$$(H)^+ = \{H\}$$

---

summary-of-closures

**Summary of Closures:**

- $(A, H)^+ = \{A, H, D, E\}^+(A, H)^+ = \{A, H, D, E\}$
- $(A, B)^+ = \{A, B, D, E\}^+(A, B)^+ = \{A, B, D, E\}$
- $(D)^+ = \{D\}^+(D)^+ = \{D\}$
- $(E)^+ = \{E\}^+(E)^+ = \{E\}$
- $(B)^+ = \{B\}^+(B)^+ = \{B\}$

- $(H)^+ = \{H\}(\mathbf{H})+=\{\mathbf{H}\}$

You've hit the Free plan limit for GPT-4o.

Responses will use another model until your limit resets after 11:08 PM.

Get Plus  
prompt-textarea

composer-background