0.20 Closure

you-said

for this relation is AG a super key? R = (A, B, C, G, H, I) F = {A → B A → C CG → H CG → I B → H}

ChatGPT said:

To determine whether AGAG is a superkey for the relation $R = (A, B, C, G, H, I)$R=(A,B,C,G,H,I), we need to check if AGAG can functionally determine all the attributes of the relation.

Given the functional dependencies $F = \{A \to B, A \to C, CG \to H, CG \to I, B \to H\}$F={A→B,A→C,CG- let's see what attributes can be derived from AGAG.

starting-with-agag

Starting with AGAG:Starting with AGAG:

1. $A \to B$ **A→B**: From AA, we can determine BB.
2. $A \to C$ **A→C**: From AA, we can determine CC.
3. Now we have AGAG, BB, and CC.
4. $CG \to H$ **CG→H**: Since we now have CC and GG, we can determine HH.
5. $CG \to I$ **CG→I**: Since we now have CC and GG, we can determine II.
6. $B \to H$ **B→H**: Though we already have HH from $CG \to H$ CG→H, this also implies we can determine HH from BB.

conclusion

Conclusion:

Using AGAG, we can determine all the attributes A, B, C, G, H, IA,B,C,G,H,I in the relation. Hence, AGAG is a superkey because it can uniquely determine all the attributes in the relation.

you-said-1

is A a super key or G a super key?

ChatGPT said:

To determine if AA or GG is a superkey, we need to check if either AA or GG can functionally determine all attributes in the relation $R = (A, B, C, G, H, I)$R=(A,B,C,G,H,I) based on the functional dependencies $F = \{A \to B, A \to C, CG \to H, CG \to I, B \to H\}$F={A→B,A→C,CG–

checking-if-aa-is-a-superkey

Checking if AA is a superkey:Checking if AA is a superkey:

Starting with AA:

1. $A \to B$ **A→B**: From AA, we can determine BB.
2. $A \to C$ **A→C**: From AA, we can determine CC.
3. Now we have AA, BB, and CC, but we still need to determine GG, HH, and II.
4. From the functional dependencies given, AA alone doesn't allow us to determine GG, and we need both CC and GG to determine HH and II (i.e., $CG \to H$ CG→H and $CG \to I$ CG→I).

Therefore, AA cannot determine all attributes in the relation, so A**A is not a superkey**.
checking-if-gg-is-a-superkey

Checking if GG is a superkey:Checking if GG is a superkey:

Starting with GG:

1. From the given functional dependencies, there is no direct dependency that allows us to determine any other attributes solely from GG.
2. We need CC and GG together to determine HH and II, but GG alone cannot determine CC, nor can it determine AA, BB, or other attributes.

Therefore, G**G is not a superkey** either.
conclusion-1

Conclusion:

Neither AA nor GG is a superkey for this relation.
you-said-2
which field is A not able to determine?
ChatGPT said:
Let's analyze which attributes AA is unable to determine based on the functional dependencies $F = \{A \to B, A \to C, CG \to H, CG \to I, B \to H\}$ F={A→B,A→C,CG→H,CG→I,B→H} for the relation $R = (A, B, C, G, H, I)$R=(A,B,C,G,H,I).
what-aa-can-determine

What AA can determine:What AA can determine:

- $A \to B$ **A→B**: From AA, we can determine BB.
- $A \to C$ **A→C**: From AA, we can determine CC.

what-aa-cannot-determine

What AA cannot determine:What AA cannot determine:

- G**G**: There is no functional dependency that shows AA can determine GG.
- H**H**: HH can be determined via two paths:
  - $CG \to H$ CG→H: This requires both CC and GG, but AA does not determine GG.
  - $B \to H$ B→H: While AA can determine BB, and hence HH, this route depends on $B \to H$ B→H. However, without knowing GG, the dependency $CG \to H$ CG→H cannot be satisfied.
- I**I**: II can be determined by $CG \to I$ CG→I, but again, this requires both CC and GG, and AA does not determine GG.

conclusion-2

Conclusion:

AA is **not able to determine GG, HH, and II** directly.
you-said-3
what if functional dependencies are F={A→B,A→C,CG→H,CG→I,B→GH}
ChatGPT said:
Let's re-evaluate the scenario with the updated functional dependencies:

- **Relation**: $R = (A, B, C, G, H, I)$R=(A,B,C,G,H,I)
- **Functional dependencies**:
  - $A \to B$ A→B
  - $A \to C$ A→C

- $CG \rightarrow H$ CG→H
- $CG \rightarrow I$ CG→I
- $B \rightarrow GH$ B→GH

what-aa-can-determine-with-these-dependencies

What AA can determine with these dependencies:What AA can determine with these dependencies:

1. $A \rightarrow B$ **A→B**: From AA, we can determine BB.
2. $A \rightarrow C$ **A→C**: From AA, we can determine CC.
3. Now we have AA, BB, and CC.
4. $B \rightarrow GH$ **B→GH**: From BB, we can determine both GG and HH.
5. Once we have GG, we can use $CG \rightarrow I$ **CG→I** to determine II.

what-aa-can-determine-now

What AA can determine now:What AA can determine now:

- $A \rightarrow B$ **A→B**: Determines BB.
- $A \rightarrow C$ **A→C**: Determines CC.
- $B \rightarrow G$ **B→G**: From BB, we determine GG.
- $B \rightarrow H$ **B→H**: From BB, we also determine HH.
- $CG \rightarrow I$ **CG→I**: From CC and GG, we can determine II.

conclusion-3

Conclusion:

With the updated functional dependencies, AA is now able to determine all attributes: A, B, C, G, H, IA,B,C,G,H,I. Therefore, A**A is a superkey** in this case.

you-said-4

For the relation: Test1 (wid, wtitle, tid, aid, tname, aname, aby). Find the closure of Functional Dependencies: FDs: {{aid}=>aby, {aid}=>aname, {aid}=>tid, {aid}=>tname, {aname}=>aby, {aname}=>aid, {aname}=>tid, {aname}=>tname, {tid}=>tname, {tname}=>tid, {wid}=>tid, {wid}=>tname, {wid}=>wtitle, {wtitle}=>tid, {wtitle}=>tname}

ChatGPT said:

To find the **closure** of the functional dependencies (FDs) for the relation `Test1(wid, wtitle, tid, aid, tname, aname, aby)`, we need to calculate the set of all attributes that can be determined from any given set of attributes using the functional dependencies.

given-functional-dependencies-fds

Given Functional Dependencies (FDs):

- $\{aid\} \rightarrow aby$ {aid}→aby
- $\{aid\} \rightarrow aname$ {aid}→aname
- $\{aid\} \rightarrow tid$ {aid}→tid
- $\{aid\} \rightarrow tname$ {aid}→tname
- $\{aname\} \rightarrow aby$ {aname}→aby
- $\{aname\} \rightarrow aid$ {aname}→aid
- $\{aname\} \rightarrow tid$ {aname}→tid
- $\{aname\} \rightarrow tname$ {aname}→tname
- $\{tid\} \rightarrow tname$ {tid}→tname
- $\{tname\} \rightarrow tid$ {tname}→tid
- $\{wid\} \rightarrow tid$ {wid}→tid
- $\{wid\} \rightarrow tname$ {wid}→tname
- $\{wid\} \rightarrow wtitle$ {wid}→wtitle
- $\{wtitle\} \rightarrow tid$ {wtitle}→tid
- $\{wtitle\} \rightarrow tname$ {wtitle}→tname

steps-to-find-closure

Steps to find closure:

Let's calculate the **closure** of various attributes:
closure-of-aid

- Start with $\{aid\}$\{aid\}.
- Using $\{aid\} \to aby$\{aid\}→aby, add abyaby to the closure.
- Using $\{aid\} \to aname$\{aid\}→aname, add $aname$aname to the closure.
- Using $\{aid\} \to tid$\{aid\}→tid, add tidtid to the closure.
- Using $\{aid\} \to tname$\{aid\}→tname, add $tname$tname to the closure.
- Now we have $\{aid, aby, aname, tid, tname\}$\{aid,aby,aname,tid,tname\}.
- Using $\{aname\} \to aid$\{aname\}→aid, nothing new is added since aidaid is already in the closure.
- Using $\{aname\} \to tid$\{aname\}→tid and $\{aname\} \to tname$\{aname\}→tname, nothing new is added since tidtid and $tname$tname are already in the closure.

Thus, Closure($\{aid\}$) = $\{aid, aby, aname, tid, tname\}$Closure(\{aid\})=\{aid,aby,aname,tid,tname\}.
closure-of-aname

- Start with $\{aname\}$\{aname\}.
- Using $\{aname\} \to aby$\{aname\}→aby, add abyaby to the closure.
- Using $\{aname\} \to aid$\{aname\}→aid, add aidaid to the closure.
- Using $\{aname\} \to tid$\{aname\}→tid, add tidtid to the closure.
- Using $\{aname\} \to tname$\{aname\}→tname, add $tname$tname to the closure.
- Now we have $\{aname, aby, aid, tid, tname\}$\{aname,aby,aid,tid,tname\}.
- Using $\{aid\} \to aby, aname, tid, tname$\{aid\}→aby,aname,tid,tname, nothing new is added since all are already in the closure.

Thus, Closure($\{aname\}$) = $\{aname, aby, aid, tid, tname\}$Closure(\{aname\})=\{aname,aby,aid,tid,tname\}.
closure-of-wid

- Start with $\{wid\}$\{wid\}.
- Using $\{wid\} \to tid$\{wid\}→tid, add tidtid to the closure.
- Using $\{wid\} \to tname$\{wid\}→tname, add $tname$tname to the closure.
- Using $\{wid\} \to wtitle$\{wid\}→wtitle, add $wtitle$wtitle to the closure.
- Now we have $\{wid, tid, tname, wtitle\}$\{wid,tid,tname,wtitle\}.
- Using $\{wtitle\} \to tid$\{wtitle\}→tid and $\{wtitle\} \to tname$\{wtitle\}→tname, nothing new is added since tidtid and $tname$tname are already in the closure.

Thus, Closure($\{wid\}$) $= \{wid, tid, tname, wtitle\}$Closure(\{wid\})=\{wid,tid,tname,wtitle\}.
closure-of-wtitle

- Start with $\{wtitle\}$\{wtitle\}.
- Using $\{wtitle\} \to tid$\{wtitle\}→tid, add tidtid to the closure.
- Using $\{wtitle\} \to tname$\{wtitle\}→tname, add $tname$tname to the closure.
- Now we have $\{wtitle, tid, tname\}$\{wtitle,tid,tname\}.

Thus, Closure($\{wtitle\}$) $= \{wtitle, tid, tname\}$Closure(\{wtitle\})=\{wtitle,tid,tname\}.
closure-of-tid

- Start with $\{tid\}$\{tid\}.
- Using $\{tid\} \to tname$\{tid\}→tname, add $tname$tname to the closure.
- Now we have $\{tid, tname\}$\{tid,tname\}.

Thus, Closure($\{tid\}$) $= \{tid, tname\}$Closure(\{tid\})=\{tid,tname\}.
closure-of-tname

- Start with $\{tname\}$\{tname\}.
- Using $\{tname\} \rightarrow tid$\{tname\}→tid, add tidtid to the closure.
- Now we have $\{tname, tid\}$\{tname,tid\}.

Thus, Closure($\{tname\}$) $= \{tname, tid\}$Closure(\{tname\})=\{tname,tid\}.
summary-of-closures

Summary of closures:

- Closure($\{aid\}$) $= \{aid, aby, aname, tid, tname\}$Closure(\{aid\})=\{aid,aby,aname,tid,tname\}
- Closure($\{aname\}$) $= \{aname, aby, aid, tid, tname\}$Closure(\{aname\})=\{aname,aby,aid,tid,tname\}
- Closure($\{wid\}$) $= \{wid, tid, tname, wtitle\}$Closure(\{wid\})=\{wid,tid,tname,wtitle\}
- Closure($\{wtitle\}$) $= \{wtitle, tid, tname\}$Closure(\{wtitle\})=\{wtitle,tid,tname\}
- Closure($\{tid\}$) $= \{tid, tname\}$Closure(\{tid\})=\{tid,tname\}
- Closure($\{tname\}$) $= \{tname, tid\}$Closure(\{tname\})=\{tname,tid\}

These closures represent the attributes that can be determined from the given set of attributes using the functional dependencies.
you-said-5

what is the closure of aby?
ChatGPT said:
To find the **closure** of the attribute {`aby`} using the given functional dependencies, we need to see if any attribute can be determined from {`aby`}.
given-functional-dependencies-fds-1

Given Functional Dependencies (FDs):

- $\{aid\} \rightarrow aby$\{aid\}→aby
- $\{aid\} \rightarrow aname$\{aid\}→aname
- $\{aid\} \rightarrow tid$\{aid\}→tid
- $\{aid\} \rightarrow tname$\{aid\}→tname

- $\{aname\} \rightarrow aby$ {aname}→aby
- $\{aname\} \rightarrow aid$ {aname}→aid
- $\{aname\} \rightarrow tid$ {aname}→tid
- $\{aname\} \rightarrow tname$ {aname}→tname
- $\{tid\} \rightarrow tname$ {tid}→tname
- $\{tname\} \rightarrow tid$ {tname}→tid
- $\{wid\} \rightarrow tid$ {wid}→tid
- $\{wid\} \rightarrow tname$ {wid}→tname
- $\{wid\} \rightarrow wtitle$ {wid}→wtitle
- $\{wtitle\} \rightarrow tid$ {wtitle}→tid
- $\{wtitle\} \rightarrow tname$ {wtitle}→tname

closure-of-aby

Closure of {`aby`}:Closure of {aby}:

Start with the set $\{aby\}$ {aby}.

- Looking at all the functional dependencies, none of the given FDs have abyaby on the left-hand side. This means abyaby alone does not determine any other attributes.

conclusion-4

Conclusion:

The closure of {`aby`} is simply $\{aby\}$ {aby}, as no other attributes can be derived from it.

Thus, $\text{Closure}(\{aby\}) = \{aby\}$Closure({aby})={aby}.

you-said-6

Discover candidate keys for Relation r1 and functional dependencies F: r1 (E,D ,C, B, A); F = {{A,B, C}=>{D}, {A, B,C} => {D,E}, {D }=>{ A , B}, {E}=>{ A , C }}.

ChatGPT said:

To discover the candidate keys for the relation $r1(E, D, C, B, A)$r1(E,D,C,B,A) with the given functional dependencies FF, we need to determine the minimal set of attributes that can uniquely identify all the attributes in the relation.

step-1-list-the-functional-dependencies-fds

Step 1: List the functional dependencies (FDs)

We are given the following functional dependencies:

1. $\{A, B, C\} \to D$ {A,B,C}→D
2. $\{A, B, C\} \to D, E$ {A,B,C}→D,E
3. $\{D\} \to A, B$ {D}→A,B
4. $\{E\} \to A, C$ {E}→A,C

step-2-find-the-closure-of-the-given-attributes

Step 2: Find the closure of the given attributes

We will compute the closures of various attribute sets to identify the candidate keys. A candidate key is a minimal set of attributes whose closure contains all attributes of the relation $r1(E, D, C, B, A)$r1(E,D,C,B,A).

step-3-check-potential-candidate-keys

Step 3: Check potential candidate keys

Let's test different combinations to see if their closures include all attributes of the relation.

closure-of-abcabc

- Start with $\{A, B, C\}$ {A,B,C}.
- Using FD $\{A, B, C\} \to D$ {A,B,C}→D, add DD to the closure.
- Using FD $\{A, B, C\} \to D, E$ {A,B,C}→D,E, add EE to the closure.
- Now we have $\{A, B, C, D, E\}$ {A,B,C,D,E}.
- Using FD $\{D\} \to A, B$ {D}→A,B, no new attributes are added since AA and BB are already in the closure.
- Using FD $\{E\} \to A, C$ {E}→A,C, no new attributes are added since AA and CC are already in the closure.

Thus, $\text{Closure}(\{A, B, C\}) = \{A, B, C, D, E\}$Closure({A,B,C})={A,B,C,D,E}, which contains all attributes in $r1(E, D, C, B, A)$r1(E,D,C,B,A).

Therefore, $\{A, B, C\}$ {A,B,C} is a **superkey**. However, we need to check if it's minimal or if a smaller set can also be a key.

closure-of-dede

- Start with $\{D, E\}$ {D,E}.
- Using FD $\{D\} \to A, B$ {D}→A,B, add AA and BB to the closure.
- Now we have $\{D, E, A, B\}$ {D,E,A,B}.
- Using FD $\{E\} \to A, C$ {E}→A,C, add CC to the closure.
- Now we have $\{D, E, A, B, C\}$ {D,E,A,B,C}.
- Using FD $\{A, B, C\} \to D, E$ {A,B,C}→D,E, no new attributes are added since DD and EE are already in the closure.

Thus, Closure($\{D, E\}$) $= \{A, B, C, D, E\}$Closure({D,E})={A,B,C,D,E}, which contains all attributes in $r1(E, D, C, B, A)$r1(E,D,C,B,A).

Therefore, $\{D, E\}$ {D,E} is also a **superkey**.

closure-of-ecec

- Start with $\{E, C\}$ {E,C}.
- Using FD $\{E\} \to A, C$ {E}→A,C, add AA to the closure (note CC is already included).
- Now we have $\{E, C, A\}$ {E,C,A}.
- We cannot apply any other FDs because none of the remaining FDs are satisfied by this set of attributes.

Thus, Closure($\{E, C\}$) $= \{E, C, A\}$Closure({E,C})={E,C,A}, which does **not** contain all attributes. Therefore, $\{E, C\}$ {E,C} is **not** a superkey.

step-4-identify-candidate-keys

Step 4: Identify candidate keys

From the closures we calculated, we found that:

- $\{A, B, C\}$ {A,B,C} is a superkey.
- $\{D, E\}$ {D,E} is a superkey.

Both sets are minimal and cannot be reduced further, so these are the **candidate keys**.

conclusion-5

Conclusion:

The candidate keys for the relation $r1(E, D, C, B, A)$r1(E,D,C,B,A) are:

- $\{A, B, C\}$\{A,B,C\}
- $\{D, E\}$\{D,E\}

you-said-7

are these 2 the only possible candidate keys?
ChatGPT said:
Yes, based on the given functional dependencies and the analysis of closures, the candidate keys for the relation $r1(E, D, C, B, A)$r1(E,D,C,B,A) are:

1. $\{A, B, C\}$\{A,B,C\}
2. $\{D, E\}$\{D,E\}

These are the **only possible candidate keys** for the following reasons:
step-by-step-justification

Step-by-step justification:

1. **From the functional dependencies**:

   - $\{A, B, C\} \to D$\{A,B,C\}→D and $\{A, B, C\} \to D, E$\{A,B,C\}→D,E: This shows that $\{A, B, C\}$\{A,B,C\} can uniquely determine DD and EE, making it a superkey.
   - $\{D\} \to A, B$\{D\}→A,B and $\{E\} \to A, C$\{E\}→A,C: These dependencies show that DD and EE combined can determine all other attributes, making $\{D, E\}$\{D,E\} a superkey.

2. **Minimality**:

   - $\{A, B, C\}$\{A,B,C\} is minimal because removing any attribute from this set would prevent it from determining all attributes. For example:

     – Removing AA or BB would prevent you from determining DD and EE.

- Removing CC would prevent you from determining DD and EE as well.

- $\{D, E\}$\{D,E\} is minimal because removing either DD or EE would prevent it from determining all attributes. For example:
  - Removing DD means you cannot determine AA and BB.
  - Removing EE means you cannot determine AA and CC.

3. **Testing other combinations**:

- Testing subsets like $\{A, B\}, \{A, C\}, \{B, C\}, \{D, C\}, \{E, C\}$,\{A,B\},\{A,C\},\{B,C\},\{D,C\},\{E,C\}, etc., do not yield closures that include all attributes of the relation.
- Any combination not involving both A, B, CA,B,C or D, ED,E would not have enough determining power based on the given FDs.

conclusion-6

Conclusion:

The only minimal sets of attributes whose closure includes all attributes in the relation are $\{A, B, C\}$\{A,B,C\} and $\{D, E\}$\{D,E\}. Therefore, these two are the only **candidate keys**.

you-said-8

what is the closure of CD

ChatGPT said:

To find the **closure** of the attribute set $\{C, D\}$\{C,D\} for the given relation $r1(E, D, C, B, A)$r1(E,D,C,B,A) and functional dependencies FF, we will calculate all the attributes that can be determined from $\{C, D\}$\{C,D\} using the given functional dependencies.

given-functional-dependencies-fds-2

Given Functional Dependencies (FDs):

1. $\{A, B, C\} \rightarrow D$\{A,B,C\}→D
2. $\{A, B, C\} \rightarrow D, E$\{A,B,C\}→D,E
3. $\{D\} \rightarrow A, B$\{D\}→A,B
4. $\{E\} \rightarrow A, C$\{E\}→A,C

step-by-step-closure-calculation-for-cdcd

Step-by-step closure calculation for $\{C, D\}$ {C,D}:Step-by-step closure calculation for \{ C,D\}{C,D}:

1. **Start with** $\{C, D\}${C,D}.

   - Initial closure: $\{C, D\}${C,D}

2. Using FD $\{D\} \to A, B${D}→A,B:

   - From DD, we can determine AA and BB.
   - Add AA and BB to the closure.
   - New closure: $\{C, D, A, B\}${C,D,A,B}

3. Using FD $\{A, B, C\} \to D${A,B,C}→D:

   - From $\{A, B, C\}${A,B,C}, we can determine DD, but DD is already in the closure.
   - No new attributes are added.

4. Using FD $\{A, B, C\} \to D, E${A,B,C}→D,E:

   - From $\{A, B, C\}${A,B,C}, we can determine EE.
   - Add EE to the closure.
   - New closure: $\{C, D, A, B, E\}${C,D,A,B,E}

5. Using FD $\{E\} \to A, C${E}→A,C:

   - From EE, we can determine AA and CC, but both AA and CC are already in the closure.
   - No new attributes are added.

final-closure

Final closure:

The closure of $\{C, D\}$\{C,D\} is $\{C, D, A, B, E\}$\{C,D,A,B,E\}.

Thus, Closure($\{C, D\}$) = $\{C, D, A, B, E\}$Closure(\{C,D\})=\{C,D,A,B,E\}.

you-said-9

so is this a candidate key?

ChatGPT said:

No, $\{C, D\}$\{C,D\} is **not** a candidate key because its closure does not contain all attributes in the relation $r1(E, D, C, B, A)$r1(E,D,C,B,A).

To be a candidate key, the closure of $\{C, D\}$\{C,D\} must include all the attributes in the relation $r1(E, D, C, B, A)$r1(E,D,C,B,A), which are $\{A, B, C, D, E\}$\{A,B,C,D,E\}. However, we notice that the closure of $\{C, D\}$\{C,D\} is:

$$\{C, D, A, B, E\}$$

\{C,D,A,B,E\}

While this does contain all attributes, **minimality** must also be checked to confirm whether this is a candidate key.

minimality-check

Minimality Check:

To be a candidate key, $\{C, D\}$\{C,D\} must not contain any smaller subset that can determine all attributes.

Let's check smaller subsets:

1. **Closure of** $\{D\}$**\{D\}**:

   - $\{D\} \to A, B$\{D\}→A,B
   - So, Closure($\{D\}$) = $\{D, A, B\}$Closure(\{D\})=\{D,A,B\}, which does not include all attributes.

2. **Closure of** $\{C\}$**\{C\}**:

   - None of the FDs allow CC alone to determine all attributes.
   - Closure($\{C\}$) = $\{C\}$Closure(\{C\})=\{C\}, which does not include all attributes.

Since no smaller subset of $\{C, D\}$\{C,D\} can determine all the attributes, $\{C, D\}$\{C,D\} is minimal.

conclusion-7

Conclusion:

Yes, $\{C, D\}${C,D} **is a candidate key** because:

- Its closure contains all attributes in the relation $r1(E, D, C, B, A)$r1(E,D,C,B,A).
- It is minimal, meaning no proper subset of $\{C, D\}${C,D} can be a candidate key.

Thus, $\{C, D\}${C,D} is a valid candidate key along with $\{A, B, C\}${A,B,C} and $\{D, E\}${D,E}.
you-said-10

what are all the candidate keys for this relation and functional dependencies?
ChatGPT said:
Let's summarize the process and determine **all the candidate keys** for the relation $r1(E, D, C, B, A)$r1(E,D,C,B,A) with the functional dependencies FF.
step-1-functional-dependencies-fds

Step 1: Functional Dependencies (FDs)

The given FDs are:

1. $\{A, B, C\} \rightarrow D${A,B,C}→D
2. $\{A, B, C\} \rightarrow D, E${A,B,C}→D,E
3. $\{D\} \rightarrow A, B${D}→A,B
4. $\{E\} \rightarrow A, C${E}→A,C

step-2-determine-the-superkeys

Step 2: Determine the Superkeys

A **superkey** is a set of attributes that can uniquely identify all the attributes in the relation. To determine the superkeys, we compute the closures of different attribute sets and check if they include all attributes $\{A, B, C, D, E\}${A,B,C,D,E}.
closure-of-abcabc-1

- From FD $\{A, B, C\} \rightarrow D, E${A,B,C}→D,E, we can determine DD and EE.
- Therefore, Closure$(\{A, B, C\}) = \{A, B, C, D, E\}$Closure({A,B,C})={A,B,C,D,E}.
- $\{A, B, C\}${A,B,C} is a **superkey**.

closure-of-dede-1

- From FD $\{D\} \to A, B$ {D}→A,B, we can determine AA and BB.
- From FD $\{E\} \to A, C$ {E}→A,C, we can determine AA and CC, but AA is already included.
- Therefore, Closure$(\{D, E\}) = \{A, B, C, D, E\}$Closure({D,E})={A,B,C,D,E}.
- $\{D, E\}${D,E} is a **superkey**.

closure-of-cdcd

- From FD $\{D\} \to A, B$ {D}→A,B, we can determine AA and BB.
- From FD $\{A, B, C\} \to D, E$ {A,B,C}→D,E, we can determine EE.
- Therefore, Closure$(\{C, D\}) = \{A, B, C, D, E\}$Closure({C,D})={A,B,C,D,E}.
- $\{C, D\}${C,D} is a **superkey**.

step-3-minimality-check-for-candidate-keys

Step 3: Minimality Check for Candidate Keys

Now, we check if these superkeys are **minimal** (i.e., no proper subset of them is a superkey):

1. **For** $\{A, B, C\}$**{A,B,C}**:

   - Any proper subset of $\{A, B, C\}${A,B,C} does not determine all attributes, so $\{A, B, C\}${A,B,C} is **minimal** and is a candidate key.

2. **For** $\{D, E\}$**{D,E}**:

   - Any proper subset of $\{D, E\}${D,E} does not determine all attributes, so $\{D, E\}${D,E} is **minimal** and is a candidate key.

3. **For** $\{C, D\}$**{C,D}**:

   - Any proper subset of $\{C, D\}${C,D} does not determine all attributes, so $\{C, D\}${C,D} is **minimal** and is a candidate key.

step-4-conclusion-list-of-candidate-keys

Step 4: Conclusion — List of Candidate Keys

The candidate keys for the relation $r1(E, D, C, B, A)$r1(E,D,C,B,A) are:

1. $\{A, B, C\}$\{A,B,C\}
2. $\{D, E\}$\{D,E\}
3. $\{C, D\}$\{C,D\}

These are the minimal sets of attributes that can uniquely identify all the attributes in the relation, so they are the **candidate keys**.

you-said-11

what about BE

ChatGPT said:

Let's calculate the closure of $\{B, E\}$\{B,E\} to determine if it's a candidate key for the relation $r1(E, D, C, B, A)$r1(E,D,C,B,A) with the functional dependencies FF.

step-1-given-functional-dependencies-fds

Step 1: Given Functional Dependencies (FDs)

1. $\{A, B, C\} \to D$\{A,B,C\}→D
2. $\{A, B, C\} \to D, E$\{A,B,C\}→D,E
3. $\{D\} \to A, B$\{D\}→A,B
4. $\{E\} \to A, C$\{E\}→A,C

step-2-closure-of-bebe

Step 2: Closure of $\{B, E\}$**\{B,E\}Step 2: Closure of \\{ B,E\\}\{B,E\}**

Let's calculate the closure of $\{B, E\}$\{B,E\} step-by-step.

- Start with $\{B, E\}$\{B,E\}.
- Using FD $\{E\} \to A, C$\{E\}→A,C, we can determine AA and CC.
    - New closure: $\{B, E, A, C\}$\{B,E,A,C\}

- Now we have $\{A, B, C\}${A,B,C}, and from FD $\{A, B, C\} \to D, E${A,B,C}→D,E, we can determine DD and EE.

  - New closure: $\{B, E, A, C, D\}${B,E,A,C,D}

step-3-does-the-closure-include-all-attributes

Step 3: Does the Closure Include All Attributes?

The closure of $\{B, E\}${B,E} is:

$$\{B, E, A, C, D\}$$

{B,E,A,C,D}

This contains all attributes in the relation $r1(E, D, C, B, A)$r1(E,D,C,B,A).

step-4-minimality-check-for-bebe

Step 4: Minimality Check for $\{B, E\}${B,E}Step 4: Minimality Check for \{ B,E\}{B,E}

To confirm whether $\{B, E\}${B,E} is a candidate key, we need to check if it's minimal:

- **Remove** B**B**: The closure of $\{E\}${E} is $\{A, C\}${A,C}, which does not include all attributes, so $\{E\}${E} alone is not a superkey.
- **Remove** E**E**: The closure of $\{B\}${B} does not determine all attributes, so $\{B\}${B} alone is not a superkey.

Since neither $\{B\}${B} nor $\{E\}${E} alone can determine all attributes, $\{B, E\}${B,E} is minimal.

conclusion-8

Conclusion

Yes, $\{B, E\}${B,E} is a **candidate key** for the relation $r1(E, D, C, B, A)$r1(E,D,C,B,A), along with $\{A, B, C\}${A,B,C}, $\{D, E\}${D,E}, and $\{C, D\}${C,D}.

final-list-of-candidate-keys

Final List of Candidate Keys:

1. $\{A, B, C\}${A,B,C}
2. $\{D, E\}${D,E}
3. $\{C, D\}${C,D}

4. $\{B, E\}$ {B,E}

These are all the candidate keys for this relation.
you-said-12

Use this algorithm to find candidate keys: Given a set of functional dependencies F over a relation r: 1. IfanattributeaofrisnotinF,thenaisineverykey 2. If an attribute a occurs on the right-hand side of a functional dependency in F but never occurs on the left-hand side, then a is never in a key 3. If an attribute a occurs on the left-hand side of a functional dependency in F but never occurs on the right-hand side, then a is in every key 4. If an attribute a occurs on both sides of functional dependencies in F, then one cannot say anything about such attribute 1. Compute the core (cases 1 and 3 before) 2. If the closure of the core contains all attributes, break (the core is the single candidate key) 3. Let's gamma be all attributes that are not exterior (case 4) 1. Compute all possible combinations of gamma in ascending order by size (one attribute, two attributes, three attributes, ...) 2. For each combination of size x: add the core to the current combination, named delta, and check delta+; delta is candidate if it fulfills the definition 3. If all combinations of size x are superkeys; break • beta is minimal if there does not exist another beta' such that beta' subset beta

ChatGPT said:

Let's apply the algorithm step-by-step to find the **candidate keys** for the relation $r1(E, D, C, B, A)$ r1(E,D,C,B,A) with the functional dependencies (FDs) provided:

functional-dependencies-fds

Functional Dependencies (FDs):

1. $\{A, B, C\} \rightarrow D$ {A,B,C}→D
2. $\{A, B, C\} \rightarrow D, E$ {A,B,C}→D,E
3. $\{D\} \rightarrow A, B$ {D}→A,B
4. $\{E\} \rightarrow A, C$ {E}→A,C

step-1-compute-the-core-cases-1-and-3

Step 1: Compute the core (Cases 1 and 3)

- **Case 1**: If an attribute aa of rr is not in F, then aa is in every key.

  - All attributes $\{A, B, C, D, E\}${A,B,C,D,E} appear in F, so no attribute can be immediately marked as part of every key based on this case.

- **Case 3**: If an attribute aa occurs on the left-hand side of an FD in F but never occurs on the right-hand side, then aa is in every key.

  - Let's check the attributes:

    * AA: Appears on both sides (so Case 3 doesn't apply).
    * BB: Appears on both sides (so Case 3 doesn't apply).
    * CC: Appears on both sides (so Case 3 doesn't apply).
    * DD: Appears on the right-hand side (so Case 3 doesn't apply).
    * EE: Appears on both sides (so Case 3 doesn't apply).

Conclusion from Step 1: No attribute belongs to the core (based on cases 1 and 3). step-2-check-if-the-closure-of-the-core-contains-all-attributes

Step 2: Check if the closure of the core contains all attributes

Since no core is determined from Step 1, the core is empty (i.e., no attributes are included by default). The closure of the empty set does not contain all attributes, so we need to move on to the next step.

step-3-identify-attributes-not-in-cases-1-or-3-gamma

Step 3: Identify attributes not in cases 1 or 3 (Gamma)

Now, we identify the set of attributes that are **not exterior**:

- **Case 2**: If an attribute occurs only on the right-hand side of an FD and never on the left-hand side, it is not part of a key.

  - Let's check the attributes:

    * DD: Appears only on the right-hand side of FDs $\{A, B, C\} \to D${A,B,C}→D, so DD cannot be part of any candidate key.

- **Gamma** (the remaining attributes, which are ambiguous and need further checking):

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

step-4-compute-combinations-of-gamma-in-ascending-order-and-check-the-closure

Step 4: Compute combinations of Gamma (in ascending order) and check the closure

Now, we compute all combinations of $\{A, B, C, E\}${A,B,C,E} (Gamma) in ascending order by size and check their closure along with any core (which is empty in this case).

combination-of-size-1

1. **Closure of** $\{A\}$**{A}**:
   - There is no FD that allows AA alone to determine all attributes.
   - So, Closure$(\{A\}) = \{A\}$Closure({A})={A}.
   - $\{A\}${A} is **not a candidate key**.
2. **Closure of** $\{B\}$**{B}**:
   - There is no FD that allows BB alone to determine all attributes.
   - So, Closure$(\{B\}) = \{B\}$Closure({B})={B}.
   - $\{B\}${B} is **not a candidate key**.
3. **Closure of** $\{C\}$**{C}**:
   - There is no FD that allows CC alone to determine all attributes.
   - So, Closure$(\{C\}) = \{C\}$Closure({C})={C}.
   - $\{C\}${C} is **not a candidate key**.
4. **Closure of** $\{E\}$**{E}**:
   - From FD $\{E\} \to A, C${E}→A,C, we get AA and CC.
   - The closure is Closure$(\{E\}) = \{A, C, E\}$Closure({E})={A,C,E}, which is still not all attributes.
   - So, $\{E\}${E} is **not a candidate key**.

combination-of-size-2

1. **Closure of** $\{A, B\}$**{A,B}**:

   - There is no FD that allows $\{A, B\}${A,B} to determine all attributes.
   - So, Closure($\{A, B\}$) = $\{A, B\}$Closure({A,B})={A,B}.
   - $\{A, B\}${A,B} is **not a candidate key**.

2. **Closure of** $\{A, C\}$**{A,C}**:

   - There is no FD that allows $\{A, C\}${A,C} to determine all attributes.
   - So, Closure($\{A, C\}$) = $\{A, C\}$Closure({A,C})={A,C}.
   - $\{A, C\}${A,C} is **not a candidate key**.

3. **Closure of** $\{A, E\}$**{A,E}**:

   - From FD $\{E\} \to A, C$ {E}→A,C, we get CC.
   - So, Closure($\{A, E\}$) = $\{A, C, E\}$Closure({A,E})={A,C,E}, which is still not all attributes.
   - $\{A, E\}${A,E} is **not a candidate key**.

4. **Closure of** $\{B, C\}$**{B,C}**:

   - There is no FD that allows $\{B, C\}${B,C} to determine all attributes.
   - So, Closure($\{B, C\}$) = $\{B, C\}$Closure({B,C})={B,C}.
   - $\{B, C\}${B,C} is **not a candidate key**.

5. **Closure of** $\{B, E\}$**{B,E}**:

   - From FD $\{E\} \to A, C$ {E}→A,C, we get AA and CC.
   - So, Closure($\{B, E\}$) = $\{A, B, C, E\}$Closure({B,E})={A,B,C,E}, which is still not all attributes.
   - $\{B, E\}${B,E} is **not a candidate key**.

6. **Closure of** $\{C, E\}$**{C,E}**:

- From FD $\{E\} \rightarrow A, C$ {E}→A,C, we get AA.
- So, Closure($\{C, E\}$) $= \{A, C, E\}$Closure({C,E})={A,C,E}, which is still not all attributes.
- $\{C, E\}${C,E} is **not a candidate key**.

combination-of-size-3

1. **Closure of** $\{A, B, C\}$**{A,B,C}**:

    - From FD $\{A, B, C\} \rightarrow D, E$ {A,B,C}→D,E, we get DD and EE.
    - So, Closure($\{A, B, C\}$) $= \{A, B, C, D, E\}$Closure({A,B,C})={A,B,C,D,E}, which includes all attributes.
    - $\{A, B, C\}${A,B,C} is a **candidate key**.

2. **Closure of** $\{B, C, E\}$**{B,C,E}**:

    - From FD $\{E\} \rightarrow A, C$ {E}→A,C, we get AA.
    - So, Closure($\{B, C, E\}$) $= \{A, B, C, E\}$Closure({B,C,E})={A,B,C,E}, which does not include all attributes.
    - $\{B, C, E\}${B,C,E} is **not a candidate key**.

combination-of-size-4

1. **Closure of** $\{A, B, C, E\}$**{A,B,C,E}**:

    - From FD $\{A, B, C\} \rightarrow D, E$ {A,B,C}→D,E, we get DD.
    - So, Closure($\{A, B, C, E\}$) $= \{A, B, C, D, E\}$Closure({A,B,C,E})={A,B,C,D,E}, which includes all attributes.
    - $\{A, B, C, E\}${A,B,C,E} is a **superkey**, but it's not minimal since $\{A, B, C\}${A,B,C} is already a candidate key.

final-list-of-candidate-keys-1

Final List of Candidate Keys:

The only candidate key found using this algorithm is $\{A, B, C\}$ {A,B,C}.

you-said-13

D occurs on the left side in Functional dependencies

ChatGPT said:

You're correct! DD does appear on the left-hand side in the functional dependency $D \to A, B$ D→A,B, so we should reconsider that step. Let's go through the process again, correcting the oversight.

step-by-step-with-correct-considerations

Step-by-Step with Correct Considerations:

functional-dependencies-fds-1

Functional Dependencies (FDs):

1. $\{A, B, C\} \to D$ {A,B,C}→D
2. $\{A, B, C\} \to D, E$ {A,B,C}→D,E
3. $\{D\} \to A, B$ {D}→A,B
4. $\{E\} \to A, C$ {E}→A,C

step-1-compute-the-core-cases-1-and-3-1

Step 1: Compute the core (Cases 1 and 3)

- **Case 1**: If an attribute of rr is not in F, then it is in every key.

  - All attributes $\{A, B, C, D, E\}$ {A,B,C,D,E} are involved in F, so no attribute can immediately be marked as part of every key based on this case.

- **Case 3**: If an attribute occurs on the left-hand side of an FD but never on the right-hand side, it is in every key.

  - Attributes that occur on the left but **never on the right**:

    * DD: Appears only on the left-hand side of FD $D \to A, B$ D→A,B, and never on the right-hand side of any dependency.

- **Conclusion**: DD is **in every key**.

step-2-check-if-the-closure-of-the-core-contains-all-attributes-1

Step 2: Check if the closure of the core contains all attributes

The core we derived is DD. Let's compute the closure of $\{D\}${D}:

- $D \to A, B$D→A,B allows us to get A, BA,B.
- Now we have $\{A, B, D\}${A,B,D}.
- $\{A, B, C\} \to D, E${A,B,C}→D,E: If we had CC, we could get EE, but we don't have CC in this closure yet.

So, the closure of $\{D\}${D} is $\{A, B, D\}${A,B,D}, which does **not** include all attributes. Therefore, DD alone is **not a candidate key**.

step-3-identify-attributes-not-in-cases-1-or-3-gamma-1

Step 3: Identify attributes not in cases 1 or 3 (Gamma)

Now, we check for the remaining attributes (Gamma), which do not fit into cases 1 or 3:

- A, B, C, EA,B,C,E need further analysis since they do not fit into cases 1 or 3.

step-4-compute-combinations-of-gamma-in-ascending-order-by-size-and-check-closure

Step 4: Compute combinations of Gamma (in ascending order by size) and check closure

We compute all combinations of $\{A, B, C, E\}${A,B,C,E} and check their closure along with the core $\{D\}${D}.

combination-of-size-1-1

1. **Closure of $\{D, A\}${D,A}**:

    - From FD $D \to A, B$D→A,B, we get A, BA,B.
    - $\{D, A\}${D,A} doesn't give us all attributes (missing CC and EE).
    - So, $\{D, A\}${D,A} is **not a candidate key**.

2. **Closure of** $\{D, B\}$**{D,B}**:

   - Similarly, from $D \to A, B$D→A,B, we only get A, BA,B.
   - Missing CC and EE.
   - So, $\{D, B\}${D,B} is **not a candidate key**.

3. **Closure of** $\{D, C\}$**{D,C}**:

   - From FD $D \to A, B$D→A,B, we get A, BA,B.
   - Missing EE.
   - So, $\{D, C\}${D,C} is **not a candidate key**.

4. **Closure of** $\{D, E\}$**{D,E}**:

   - From FD $D \to A, B$D→A,B, we get A, BA,B, and from FD $E \to A, C$E→A,C, we get A, CA,C.
   - So, $\text{Closure}(\{D, E\}) = \{A, B, C, D, E\}$Closure({D,E})={A,B,C,D,E}, which contains all attributes.
   - $\{D, E\}$**{D,E} is a candidate key**.

combination-of-size-2-no-need-to-compute

Since we already found a candidate key $\{D, E\}${D,E}, and it's minimal, there is no need to compute larger combinations.
conclusion-9

Conclusion:

The candidate keys for this relation are:

- $\{D, E\}${D,E}

Thus, $\{D, E\}${D,E} is the **only candidate key** in this case.
you-said-14

For this is BH a candidate key? r2 (A,B ,C, D,E ,G ,H ,K) ; F = {{A, B,H}=>{ C}, {A}=>{D,E }, {B,G,H} => {K}, {K}=> {A,D,H}, {B,H}=>{G, E }.

ChatGPT said:

Let's go through the steps to determine if $\{B, H\}$\{B,H\} is a candidate key for relation $r2(A, B, C, D, E, G, H, K)$r2(A,B,C,D,E,G,H,K) and the functional dependencies (FDs) given:

relation-r2abcdeghkr2abcdeghk-and-fds

Relation $r2(A, B, C, D, E, G, H, K)$r2(A,B,C,D,E,G,H,K) and FDs:Relation r2(A,B,C,D,E,G,H,K)r and FDs:

1. $\{A, B, H\} \to C$ {A,B,H}→C
2. $\{A\} \to D, E$ {A}→D,E
3. $\{B, G, H\} \to K$ {B,G,H}→K
4. $\{K\} \to A, D, H$ {K}→A,D,H
5. $\{B, H\} \to G, E$ {B,H}→G,E

step-by-step-process

Step-by-Step Process:

step-1-compute-the-closure-of-bhbh

We need to calculate what attributes $\{B, H\}$\{B,H\} can determine by using the given functional dependencies.

- From FD $\{B, H\} \to G, E$ {B,H}→G,E, we get GG and EE.
- Now we have $\{B, H, G, E\}$\{B,H,G,E\}.
- From FD $\{B, G, H\} \to K$ {B,G,H}→K, and since we now have B, G, HB,G,H, we can determine KK.
- Now we have $\{B, H, G, E, K\}$\{B,H,G,E,K\}.
- From FD $\{K\} \to A, D, H$ {K}→A,D,H, and since we now have KK, we can determine AA and DD.
- Now we have $\{B, H, G, E, K, A, D\}$\{B,H,G,E,K,A,D\}.

- From FD $\{A\} \to D, E$ {A}→D,E, and since we already have AA, we can determine DD and EE (but we already have both).
- From FD $\{A, B, H\} \to C$ {A,B,H}→C, and since we have A, B, HA,B,H, we can determine CC.
- Now we have $\{A, B, C, D, E, G, H, K\}$ {A,B,C,D,E,G,H,K}, which includes **all attributes** of the relation.

step-2-is-bhbh-a-candidate-key

Since the closure of $\{B, H\}$ {B,H} contains all attributes of the relation, $\{B, H\}$ {B,H} is a **superkey**.

Now we need to check if it is **minimal**, i.e., there should not be a smaller subset of $\{B, H\}$ {B,H} that is also a superkey. In this case, neither BB nor HH alone can determine all attributes, so $\{B, H\}$ {B,H} is minimal.

conclusion-10

Conclusion:

Yes, $\{B, H\}$ **{B,H} is a candidate key** for the relation $r2(A, B, C, D, E, G, H, K)$ r2(A,B,C,D,E,G,H,K) with the given functional dependencies.

you-said-15

is AB a candidate key?

ChatGPT said:

Let's determine if $\{A, B\}$ {A,B} is a candidate key for relation $r2(A, B, C, D, E, G, H, K)$ r2(A,B,C,D,E,G,H, with the given functional dependencies.

relation-r2abcdeghkr2abcdeghk-and-fds-1

Relation $r2(A, B, C, D, E, G, H, K)$ r2(A,B,C,D,E,G,H,K) and FDs:Relation r2(A,B,C,D,E,G,H,K)r and FDs:

1. $\{A, B, H\} \to C$ {A,B,H}→C
2. $\{A\} \to D, E$ {A}→D,E
3. $\{B, G, H\} \to K$ {B,G,H}→K
4. $\{K\} \to A, D, H$ {K}→A,D,H
5. $\{B, H\} \to G, E$ {B,H}→G,E

step-1-compute-the-closure-of-abab

Step 1: Compute the closure of $\{A, B\}$**{A,B}Step 1: Compute the closure of \{ A,B\}{A,B}**

We need to determine what attributes $\{A, B\}${A,B} can determine using the functional dependencies.

- From FD $\{A\} \to D, E${A}→D,E, and since we have AA, we can determine DD and EE.
- Now we have $\{A, B, D, E\}${A,B,D,E}.

At this point, we can't determine any further attributes directly from the functional dependencies using only AA and BB. Specifically:

- FD $\{A, B, H\} \to C${A,B,H}→C requires HH.
- FD $\{B, G, H\} \to K${B,G,H}→K requires both GG and HH, which we don't have.
- FD $\{K\} \to A, D, H${K}→A,D,H requires KK, which we also don't have.
- FD $\{B, H\} \to G, E${B,H}→G,E requires HH.

Thus, the closure of $\{A, B\}${A,B} is $\{A, B, D, E\}${A,B,D,E}, which does **not** include all attributes.

step-2-conclusion

Step 2: Conclusion

Since the closure of $\{A, B\}${A,B} does not contain all attributes of the relation, $\{A, B\}$**{A,B} is not a candidate key** for the given relation $r2$r2 with the provided functional dependencies.

you-said-16

is GBK a candidate key?

ChatGPT said:

Let's determine if $\{G, B, K\}${G,B,K} is a candidate key for relation $r2(A, B, C, D, E, G, H, K)$r2(A,B,C,D,E with the given functional dependencies.

relation-r2abcdeghkr2abcdeghk-and-fds-2

Relation $r2(A, B, C, D, E, G, H, K)$**r2(A,B,C,D,E,G,H,K) and FDs:Relation r2(A,B,C,D,E,G,H,K)r and FDs:**

1. $\{A, B, H\} \to C${A,B,H}→C

2. $\{A\} \to D, E$ {A}→D,E
3. $\{B, G, H\} \to K$ {B,G,H}→K
4. $\{K\} \to A, D, H$ {K}→A,D,H
5. $\{B, H\} \to G, E$ {B,H}→G,E

step-1-compute-the-closure-of-gbkgbk

Step 1: Compute the closure of $\{G, B, K\}$**{G,B,K}**Step 1: Compute the closure of \{ G,B,K\}{G,B,K}

We need to calculate what attributes $\{G, B, K\}${G,B,K} can determine using the functional dependencies.

- From FD $\{K\} \to A, D, H$ {K}→A,D,H, and since we have KK, we can determine A, D, HA,D,H.
- Now we have $\{G, B, K, A, D, H\}${G,B,K,A,D,H}.
- From FD $\{A\} \to D, E$ {A}→D,E, and since we already have AA, we can determine DD and EE. However, DD is already in our closure, so we add EE.
- Now we have $\{G, B, K, A, D, H, E\}${G,B,K,A,D,H,E}.
- From FD $\{B, H\} \to G, E$ {B,H}→G,E, and since we have both BB and HH, we can determine GG and EE, but we already have both GG and EE in the closure.
- From FD $\{A, B, H\} \to C$ {A,B,H}→C, and since we now have A, B, HA,B,H, we can determine CC.
- Now we have $\{G, B, K, A, D, H, E, C\}${G,B,K,A,D,H,E,C}, which includes **all attributes** of the relation.

step-2-is-gbkgbk-a-candidate-key

Step 2: Is $\{G, B, K\}$**{G,B,K}** a candidate key?Step 2: Is \{ G,B,K\}{G,B,K} a candidate key?

Since the closure of $\{G, B, K\}${G,B,K} contains all attributes of the relation, $\{G, B, K\}${G,B,K} is a **superkey**.

Next, we need to check if it is **minimal**, i.e., there should not be a smaller subset of $\{G, B, K\}${G,B,K} that is also a superkey.

- If we remove GG, the closure of $\{B, K\}${B,K} wouldn't be able to determine GG, so $\{B, K\}${B,K} isn't a superkey.

- Similarly, removing BB or KK would also make it impossible to determine all attributes.

Thus, $\{G, B, K\}$\{G,B,K\} is minimal.
conclusion-11

Conclusion:

Yes, $\{G, B, K\}$**\{G,B,K\} is a candidate key** for the relation $r2(A, B, C, D, E, G, H, K)$r2(A,B,C,D,E,G,H
with the given functional dependencies.
you-said-17
compute the canonical cover of these: r1 (E,D ,C, B, A); F = \{\{A,B, C\}=>\{D\}, \{A, B,C\} => \{D,E\}, \{D \}=>\{ A , B\}, \{E\}=>\{ A , C \}\}.
ChatGPT said:
To compute the **canonical cover** of a set of functional dependencies, we need to follow a step-by-step process that ensures the dependencies are in a simplified and minimal form. The canonical cover should eliminate extraneous attributes on the left-hand side of the dependencies, combine redundant dependencies, and ensure each functional dependency has a single attribute on the right-hand side.
given-functional-dependencies-for-r1edcbar1edcba

Given Functional Dependencies for $r1(E, D, C, B, A)$r1(E,D,C,B,A):Given Functional Dependencies for r1(E,D,C,B,A)r1(E,D,C,B,A):

1. $\{A, B, C\} \rightarrow D$\{A,B,C\}→D
2. $\{A, B, C\} \rightarrow D, E$\{A,B,C\}→D,E
3. $\{D\} \rightarrow A, B$\{D\}→A,B
4. $\{E\} \rightarrow A, C$\{E\}→A,C

steps-to-compute-the-canonical-cover

Steps to Compute the Canonical Cover

step-1-split-the-right-hand-sides
We start by ensuring that every functional dependency has a single attribute on the right-hand side.

- $\{A, B, C\} \rightarrow D$\{A,B,C\}→D remains as $\{A, B, C\} \rightarrow D$\{A,B,C\}→D.

- $\{A, B, C\} \rightarrow D, E$ {A,B,C}→D,E is split into:

  - $\{A, B, C\} \rightarrow D$ {A,B,C}→D
  - $\{A, B, C\} \rightarrow E$ {A,B,C}→E

- $\{D\} \rightarrow A, B$ {D}→A,B is split into:

  - $\{D\} \rightarrow A$ {D}→A
  - $\{D\} \rightarrow B$ {D}→B

- $\{E\} \rightarrow A, C$ {E}→A,C is split into:

  - $\{E\} \rightarrow A$ {E}→A
  - $\{E\} \rightarrow C$ {E}→C

So now the set of FDs is:

1. $\{A, B, C\} \rightarrow D$ {A,B,C}→D
2. $\{A, B, C\} \rightarrow D$ {A,B,C}→D (duplicate)
3. $\{A, B, C\} \rightarrow E$ {A,B,C}→E
4. $\{D\} \rightarrow A$ {D}→A
5. $\{D\} \rightarrow B$ {D}→B
6. $\{E\} \rightarrow A$ {E}→A
7. $\{E\} \rightarrow C$ {E}→C

step-2-remove-extraneous-attributes-from-the-left-hand-sides

Now, we check for any extraneous attributes on the left-hand side of each functional dependency.

1. **For** $\{A, B, C\} \rightarrow D$ **{A,B,C}→D**:

- Compute the closure of $\{A, B\}${A,B} (without CC) and see if we can still determine DD:

  - $\{A, B\}${A,B} does not imply DD, so CC is not extraneous.
- Therefore, $\{A, B, C\} \to D${A,B,C}→D remains unchanged.

2. **For** $\{A, B, C\} \to E$**{A,B,C}→E**:

   - Compute the closure of $\{A, B\}${A,B} (without CC) and see if we can still determine EE:

     - $\{A, B\}${A,B} does not imply EE, so CC is not extraneous.
   - Therefore, $\{A, B, C\} \to E${A,B,C}→E remains unchanged.

3. **For** $\{D\} \to A$**{D}→A**:

   - $D \to A$D→A has a single attribute on the left-hand side, so no attributes to remove.

4. **For** $\{D\} \to B$**{D}→B**:

   - Same as above, $D \to B$D→B has no extraneous attributes.

5. **For** $\{E\} \to A$**{E}→A**:

   - $E \to A$E→A has a single attribute on the left-hand side, so no attributes to remove.

6. **For** $\{E\} \to C$**{E}→C**:

   - Same as above, $E \to C$E→C has no extraneous attributes.

step-3-remove-redundant-functional-dependencies

Now we check for any redundant functional dependencies, i.e., dependencies that can be derived from others.

1. **For** $\{A, B, C\} \rightarrow D$**{A,B,C}→D**:

   - This dependency appears twice, so we can remove the duplicate. We keep only one occurrence of $\{A, B, C\} \rightarrow D${A,B,C}→D.

2. **For** $\{A, B, C\} \rightarrow E$**{A,B,C}→E**:

   - This is not redundant, as it provides new information.

3. **For** $\{D\} \rightarrow A$**{D}→A**:

   - This is not redundant.

4. **For** $\{D\} \rightarrow B$**{D}→B**:

   - This is not redundant.

5. **For** $\{E\} \rightarrow A$**{E}→A**:

   - This is not redundant.

6. **For** $\{E\} \rightarrow C$**{E}→C**:

   - This is not redundant.

final-canonical-cover

Final Canonical Cover:

After applying the steps, the canonical cover is:

1. $\{A, B, C\} \rightarrow D${A,B,C}→D
2. $\{A, B, C\} \rightarrow E${A,B,C}→E

3. $\{D\} \rightarrow A$ {D}→A

4. $\{D\} \rightarrow B$ {D}→B

5. $\{E\} \rightarrow A$ {E}→A

6. $\{E\} \rightarrow C$ {E}→C

This is the minimal and simplified set of functional dependencies for the relation $r1(E, D, C, B, A)$r1(E,D,C,B,A).
you-said-18

compute canonical cover of r2 (A,B ,C, D,E ,G ,H ,K) ; F = {{A, B,H}=>{ C}, {A}=>{D,E }, {B,G,H} => {K}, {K}=> {A,D,H}, {B,H}=>{G, E }
ChatGPT said:
To compute the **canonical cover** of a set of functional dependencies, we need to follow a series of steps:

1. **Split the right-hand sides** so that each functional dependency has only one attribute on the right-hand side.

2. **Remove extraneous attributes** from the left-hand side of functional dependencies.

3. **Remove redundant functional dependencies** by checking if a functional dependency can be derived from others.

given-relation-r2abcdeghkr2abcdeghk-and-functional-dependencies-fds

Given Relation $r2(A, B, C, D, E, G, H, K)$r2(A,B,C,D,E,G,H,K) and Functional Dependencies (FDs):Given Relation r2(A,B,C,D,E,G,H,K)r2(A,B,C,D,E,G,H,K) and Functional Dependencies (FDs):

1. $\{A, B, H\} \rightarrow C$ {A,B,H}→C

2. $\{A\} \rightarrow D, E$ {A}→D,E

3. $\{B, G, H\} \rightarrow K$ {B,G,H}→K

4. $\{K\} \rightarrow A, D, H$ {K}→A,D,H

5. $\{B, H\} \rightarrow G, E$ {B,H}→G,E

step-1-split-the-right-hand-sides-1

Step 1: Split the right-hand sides

We split the dependencies that have more than one attribute on the right-hand side.

- $\{A, B, H\} \to C$ {A,B,H}→C stays as it is.
- $\{A\} \to D, E$ {A}→D,E is split into:
  - $\{A\} \to D$ {A}→D
  - $\{A\} \to E$ {A}→E
- $\{B, G, H\} \to K$ {B,G,H}→K stays as it is.
- $\{K\} \to A, D, H$ {K}→A,D,H is split into:
  - $\{K\} \to A$ {K}→A
  - $\{K\} \to D$ {K}→D
  - $\{K\} \to H$ {K}→H
- $\{B, H\} \to G, E$ {B,H}→G,E is split into:
  - $\{B, H\} \to G$ {B,H}→G
  - $\{B, H\} \to E$ {B,H}→E

So after splitting, we have the following set of functional dependencies:

1. $\{A, B, H\} \to C$ {A,B,H}→C
2. $\{A\} \to D$ {A}→D
3. $\{A\} \to E$ {A}→E
4. $\{B, G, H\} \to K$ {B,G,H}→K
5. $\{K\} \to A$ {K}→A
6. $\{K\} \to D$ {K}→D
7. $\{K\} \to H$ {K}→H
8. $\{B, H\} \to G$ {B,H}→G
9. $\{B, H\} \to E$ {B,H}→E

step-2-remove-extraneous-attributes-from-the-left-hand-sides-1

Step 2: Remove extraneous attributes from the left-hand sides

We check each functional dependency to see if any attributes on the left-hand side are extraneous. An attribute is extraneous if removing it still allows the functional dependency to hold.

1. **For** $\{A, B, H\} \to C$ **{A,B,H}→C**:

   - Check if $\{B, H\} \to C$ {B,H}→C holds. The closure of $\{B, H\}$ {B,H} does not determine CC, so AA is not extraneous.
   - Check if $\{A, H\} \to C$ {A,H}→C holds. The closure of $\{A, H\}$ {A,H} does not determine CC, so BB is not extraneous.
   - Check if $\{A, B\} \to C$ {A,B}→C holds. The closure of $\{A, B\}$ {A,B} does not determine CC, so HH is not extraneous.
   - Thus, $\{A, B, H\} \to C$ {A,B,H}→C remains unchanged.

2. **For** $\{A\} \to D$ **{A}→D**:

   - Since AA is the only attribute on the left-hand side, it cannot be reduced further.

3. **For** $\{A\} \to E$ **{A}→E**:

   - Same as above, it cannot be reduced.

4. **For** $\{B, G, H\} \to K$ **{B,G,H}→K**:

   - Check if $\{G, H\} \to K$ {G,H}→K holds. The closure of $\{G, H\}$ {G,H} does not determine KK, so BB is not extraneous.
   - Check if $\{B, H\} \to K$ {B,H}→K holds. The closure of $\{B, H\}$ {B,H} does not determine KK, so GG is not extraneous.
   - Check if $\{B, G\} \to K$ {B,G}→K holds. The closure of $\{B, G\}$ {B,G} does not determine KK, so HH is not extraneous.
   - Thus, $\{B, G, H\} \to K$ {B,G,H}→K remains unchanged.

5. **For** $\{K\} \to A$ **{K}→A**:

- KK is the only attribute on the left-hand side, so it cannot be reduced further.

6. **For** $\{K\} \to D$ **{K}→D**:

   - Same as above.

7. **For** $\{K\} \to H$ **{K}→H**:

   - Same as above.

8. **For** $\{B, H\} \to G$ **{B,H}→G**:

   - Check if $\{H\} \to G$ {H}→G holds. The closure of $\{H\}$ {H} does not determine GG, so BB is not extraneous.
   - Check if $\{B\} \to G$ {B}→G holds. The closure of $\{B\}$ {B} does not determine GG, so HH is not extraneous.
   - Thus, $\{B, H\} \to G$ {B,H}→G remains unchanged.

9. **For** $\{B, H\} \to E$ **{B,H}→E**:

   - Same reasoning as above, no extraneous attributes can be removed.

step-3-remove-redundant-functional-dependencies-1

Step 3: Remove redundant functional dependencies

We check if any functional dependency can be derived from others, making it redundant.

- None of the functional dependencies are redundant. Each provides unique information that cannot be derived from the others.

final-canonical-cover-1

Final Canonical Cover

The canonical cover is the set of functional dependencies after splitting, removing extraneous attributes, and eliminating redundant dependencies:

1. $\{A, B, H\} \to C$ {A,B,H}→C

2. $\{A\} \to D$ {A}→D
3. $\{A\} \to E$ {A}→E
4. $\{B, G, H\} \to K$ {B,G,H}→K
5. $\{K\} \to A$ {K}→A
6. $\{K\} \to D$ {K}→D
7. $\{K\} \to H$ {K}→H
8. $\{B, H\} \to G$ {B,H}→G
9. $\{B, H\} \to E$ {B,H}→E

This is the canonical cover of the functional dependencies for relation $r2(A, B, C, D, E, G, H, K)$r2(A,B,C,D,
you-said-19

for r2 (A,B ,C, D,E ,G ,H ,K) ; F = {{A, B,H}=>{ C}, {A}=>{D,E }, {B,G,H} => {K}, {K}=> {A,D,H}, {B,H}=>{G, E }. Is B an extraneous attribute in ABH -> C?

ChatGPT said:

To determine if BB is an **extraneous attribute** in the functional dependency $\{A, B, H\} \to C$ {A,B,H}→C we need to check whether the functional dependency $\{A, H\} \to C$ {A,H}→C holds by computing the **closure** of $\{A, H\}$ {A,H} under the given set of functional dependencies.

given-functional-dependencies

Given Functional Dependencies:

1. $\{A, B, H\} \to C$ {A,B,H}→C
2. $\{A\} \to D, E$ {A}→D,E
3. $\{B, G, H\} \to K$ {B,G,H}→K
4. $\{K\} \to A, D, H$ {K}→A,D,H
5. $\{B, H\} \to G, E$ {B,H}→G,E

step-1-compute-the-closure-of-ahah

Step 1: Compute the closure of $\{A, H\}$\{A,H\}Step 1: Compute the closure of \\{ A,H\\}\{A,H\}

Start with $\{A, H\}$\{A,H\} and apply the functional dependencies iteratively to find the attributes that can be determined by $\{A, H\}$\{A,H\}.

1. **Initially**: $\{A, H\} = \{A, H\}$\{A,H\}=\{A,H\}.
2. Using $\{A\} \to D$\{A\}→D and $\{A\} \to E$\{A\}→E, we can add DD and EE to the closure:

   - $\{A, H\} \to \{A, H, D, E\}$\{A,H\}→\{A,H,D,E\}.
3. No other functional dependencies apply directly to $\{A, H, D, E\}$\{A,H,D,E\}, especially none that can give us CC.

Since $\{A, H\}$\{A,H\} does **not** determine CC, the closure of $\{A, H\}$\{A,H\} does **not** contain CC, which means that B**B is not extraneous** in the functional dependency $\{A, B, H\} \to C$\{A,B,H\}→C.

conclusion-12

Conclusion:

BB is **not** extraneous in $\{A, B, H\} \to C$\{A,B,H\}→C. Removing BB would result in the functional dependency $\{A, H\} \to C$\{A,H\}→C, which does not hold under the given functional dependencies.

you-said-20

is D extraneous in K -> ADH?

ChatGPT said:

To determine if DD is **extraneous** in the functional dependency $K \to A, D, H$K→A,D,H, we need to check whether the functional dependency $K \to A, H$K→A,H (without DD) still holds by computing the **closure** of KK under the given set of functional dependencies.

given-functional-dependencies-1

Given Functional Dependencies:

1. $\{A, B, H\} \to C$\{A,B,H\}→C
2. $\{A\} \to D, E$\{A\}→D,E
3. $\{B, G, H\} \to K$\{B,G,H\}→K

4. $\{K\} \to A, D, H$ {K}→A,D,H
5. $\{B, H\} \to G, E$ {B,H}→G,E

step-1-compute-the-closure-of-kk

Step 1: Compute the closure of KKStep 1: Compute the closure of KK

We need to compute the closure of KK and see if it includes AA and HH without relying on DD.

1. **Initially**: $\{K\} = \{K\}$ {K}={K}.
2. From $\{K\} \to A$ {K}→A, we can add AA to the closure:
   - $\{K\} \to \{K, A\}$ {K}→{K,A}.
3. From $\{K\} \to H$ {K}→H, we can add HH to the closure:
   - $\{K, A\} \to \{K, A, H\}$ {K,A}→{K,A,H}.

step-2-check-if-the-closure-of-kk-includes-dd

Step 2: Check if the closure of KK includes DDStep 2: Check if the closure of KK includes DD

Now, let's compute the closure of KK without DD in the right-hand side of $K \to A, D, H$ K→A,D,H:

- The closure of KK under the functional dependencies is $\{K, A, H\}$ {K,A,H}, but it does not include DD.
- DD is not implied by the closure of KK without DD in the functional dependency.

conclusion-13

Conclusion:

Since $K \to A, H$ K→A,H holds, but the closure of KK does not include DD without it being explicitly included in the right-hand side, D**D is not extraneous** in the functional dependency $K \to A, D, H$ K→A,D,H.

you-said-21

conditions for extraneous attributes are: Given alpha → beta: • a belongs to alpha is extraneous if beta subset (alpha - a)+ under F •a belongs to beta is extraneous if a belongs to alpha+ underF'=(F – {alpha → beta}) union {alpha → (beta - a)}

ChatGPT said:

You're correct in stating the conditions for determining extraneous attributes in a functional dependency. Let's apply these conditions step by step to see if DD is extraneous in $K \rightarrow A, D, H$ K→A,D,H.

conditions-for-extraneous-attributes

Conditions for Extraneous Attributes:

1. **Left-hand side**: An attribute aa in the left-hand side (α) is extraneous if:
   - $\beta \subseteq (\alpha - a)^+$(a)+ under the set of functional dependencies FF.
2. **Right-hand side**: An attribute aa in the right-hand side (β) is extraneous if:
   - $a \in \alpha^+$a+ under a modified set of functional dependencies F'F, where $F' = (F - \{\alpha \rightarrow \beta\}) \cup \{\alpha \rightarrow ($

lets-apply-this-to-left.-krightarrow-adh-right.kadh

Let's apply this to $K \rightarrow A, D, H$ K→A,D,H:Let's apply this to \left. K\rightarrow A,D,H \right.K→A,D,H:

step-1-check-if-dd-is-extraneous-on-the-right-hand-side-in-left.-krightarrow-adh-right.kadh.

We will check if removing DD from β (right-hand side) results in a dependency that still implies DD under a modified set of functional dependencies.

1. **Modify the functional dependency**:
   - We remove DD from the right-hand side of $K \rightarrow A, D, H$ K→A,D,H.
   - So, we replace $K \rightarrow A, D, H$ K→A,D,H with $K \rightarrow A, H$ K→A,H in the set of functional dependencies.
2. **Compute the closure of KK under the modified set of FDs**:

- We now have the modified FDs:

  (a) $\{A, B, H\} \rightarrow C$ {A,B,H}→C
  (b) $\{A\} \rightarrow D, E$ {A}→D,E
  (c) $\{B, G, H\} \rightarrow K$ {B,G,H}→K
  (d) $\{K\} \rightarrow A, H$ {K}→A,H (modified version of the original dependency)
  (e) $\{B, H\} \rightarrow G, E$ {B,H}→G,E

3. **Closure of KK under this modified set**:

- Starting with KK, we compute the closure:

  - $\{K\} \rightarrow A$ {K}→A (from $K \rightarrow A$ K→A).
  - $\{K\} \rightarrow H$ {K}→H (from $K \rightarrow H$ K→H).
- The closure of KK is $\{K, A, H\}$ {K,A,H}.
- Since DD is not in the closure of KK under this modified set, D**D is not extraneous** on the right-hand side because it is not implied by $K \rightarrow A, H$ K→A,H.

conclusion-14

Conclusion:

Based on the second condition for extraneous attributes (for the right-hand side), D**D is not extraneous** in $K \rightarrow A, D, H$ K→A,D,H. Removing DD from the right-hand side results in a closure that does not include DD, meaning DD is necessary in this functional dependency.

you-said-22

K -> KA -> KAH -> KAHDE

ChatGPT said:

You're applying the closure computation for the functional dependency $K \rightarrow A, D, H$ K→A,D,H, so let's verify step by step.

closure-of-kk-under-the-set-of-functional-dependencies

Closure of KK under the set of functional dependencies:Closure of KK under the set of functional dependencies:

given-fds

1. $\{A, B, H\} \rightarrow C$ {A,B,H}→C

2. $\{A\} \to D, E$ {A}→D,E
3. $\{B, G, H\} \to K$ {B,G,H}→K
4. $\{K\} \to A, D, H$ {K}→A,D,H
5. $\{B, H\} \to G, E$ {B,H}→G,E

We are calculating the closure of KK:

1. **Start with KK**: Initially, $\{K\} = \{K\}$\{K\}=\{K\}.
2. From the functional dependency $\{K\} \to A$ {K}→A, we add AA to the closure:
   - $\{K\} \to \{K, A\}$ {K}→{K,A}.
3. From the same functional dependency $\{K\} \to D$ {K}→D, we add DD to the closure:
   - $\{K, A\} \to \{K, A, D\}$ {K,A}→{K,A,D}.
4. From the same functional dependency $\{K\} \to H$ {K}→H, we add HH to the closure:
   - $\{K, A, D\} \to \{K, A, D, H\}$ {K,A,D}→{K,A,D,H}.
5. Now that we have AA, from the functional dependency $A \to D, E$ A→D,E, we can add EE to the closure:
   - $\{K, A, D, H\} \to \{K, A, D, H, E\}$ {K,A,D,H}→{K,A,D,H,E}.

So, the **closure of KK** is $\{K, A, D, H, E\}$\{K,A,D,H,E\}.
conclusion-15

Conclusion:

- The closure of KK is $\{K, A, D, H, E\}$\{K,A,D,H,E\}, which includes all the attributes in the right-hand side of $K \to A, D, H$ K→A,D,H.
- This shows that DD is **not extraneous** in the functional dependency $K \to A, D, H$ K→A,D,H, because removing DD would prevent us from determining all the attributes originally in the right-hand side.

you-said-23

using the update functional dependencies: {A,B,H}→C {}→, {A}→D,E {,,}→ {B,G,H}→K {}→, {K}→A,H (modified version of the original dependency) {,}→, {B,H}→G,E Now from K -> AH we get closure of k as KAH. Then from A -> DE we get KAHDE. So does that prove that D is extraneous in K -> ADH?

2/2

ChatGPT said:

To determine if DD is extraneous in the functional dependency $K \rightarrow A, D, H$ K→A,D,H using the provided functional dependencies, we will apply the condition for extraneous attributes as you described.

given-functional-dependencies-2

Given Functional Dependencies:

1. $\{A, B, H\} \rightarrow C$ {A,B,H}→C
2. $\{A\} \rightarrow D, E$ {A}→D,E
3. $\{B, G, H\} \rightarrow K$ {B,G,H}→K
4. $\{K\} \rightarrow A, H$ {K}→A,H
5. $\{B, H\} \rightarrow G, E$ {B,H}→G,E

step-1-check-if-dd-is-extraneous-on-the-right-hand-side-of-left.-krightarrow-adh-right.kadh

Step 1: Check if DD is extraneous on the right-hand side of $K \rightarrow A, D, H$ K→A,D,HStep 1: Check if DD is extraneous on the right-hand side of \left. K\rightarrow A,D,H \right.K→A,D,H

To test if DD is extraneous, we will analyze the functional dependency without DD on the right side, i.e., checking $K \rightarrow A, H$ K→A,H.

modified-functional-dependencies

- Remove DD from the original functional dependency to check if $K \rightarrow A, H$ K→A,H holds.

step-2-compute-the-closure-of-kk-with-the-modified-fds

Step 2: Compute the Closure of KK with the Modified FDsStep 2: Compute the Closure of KK with the Modified FDs

1. **Start with KK**: Initially, $\{K\} = \{K\}$ {K}={K}.

2. Using $K \to A$ K→A:

   - $\{K\} \to \{K, A\}$ {K}→{K,A}.

3. Using $K \to H$ K→H:

   - $\{K, A\} \to \{K, A, H\}$ {K,A}→{K,A,H}.

Now, the closure of KK is $\{K, A, H\}${K,A,H}.
step-3-check-for-dd

Step 3: Check for DDStep 3: Check for DD

Next, we need to see if we can derive DD from $K \to A, H$ K→A,H with the existing functional dependencies:

1. Using $A \to D, E$ A→D,E:

   - Since KK gives us AA, we can add DD and EE:
   - $\{K, A, H\} \to \{K, A, H, D, E\}$ {K,A,H}→{K,A,H,D,E}.

Thus, the closure of KK under these modified dependencies can be represented as:

- $\{K, A, D, H, E\}${K,A,D,H,E}.

conclusion-16

Conclusion

- Since you can still derive DD through AA from the closure of KK under the modified dependencies, this means that DD is **not** extraneous in the dependency $K \to A, D, H$ K→A,D,H.

In summary, although you derived $K \to A, D, H$ K→A,D,H using $K \to A, H$ K→A,H and $A \to D, E$ A→D,E, it indicates that removing DD does not affect the functional dependency because KK leads to AA, which then leads to DD. Thus, DD is indeed needed to preserve the original dependency structure, so it is not extraneous.
4o mini
you-said-24

For R(A, B,C); FD = {A -> BC, B -> C, A -> B, AB. -> C}}. is A extraneous in the left hand side of AB -> C

ChatGPT said:

To determine if AA is extraneous in the left-hand side of the functional dependency $AB \to C$ AB→C given the functional dependencies (FDs) for the relation $R(A, B, C)$R(A,B,C):

given-functional-dependencies-3

Given Functional Dependencies:

1. $A \to BC$ A→BC
2. $B \to C$ B→C
3. $A \to B$ A→B
4. $AB \to C$ AB→C

condition-for-extraneous-attributes

Condition for Extraneous Attributes:

An attribute aa in the left-hand side (α) of a functional dependency $\alpha \to \beta$→ is extraneous if:

- $\beta \subseteq (\alpha - a)^+$(a)+ under the set of functional dependencies F'F, where $F' = F - \{\alpha \to \beta\}$F=F{→} and $\alpha - a$a means the set of attributes in α without aa.

step-1-remove-aa-from-left.-abrightarrow-c-right.abc

Step 1: Remove A**A** from $AB \to C$ **AB→CStep 1: Remove AA from \left. AB\rightarrow C \right.AB→C**

- This means we need to check if $B \to C$ B→C holds under the remaining functional dependencies without AA.

step-2-check-the-closure-of-bb

Step 2: Check the closure of B**BStep 2: Check the closure of BB**

Now we need to compute the closure of BB using the remaining functional dependencies:

- Start with BB:

- $\{B\}^+ = \{B\}$ {B}+={B}

Using the functional dependencies:

1. From $B \to C$ B→C:
   - $\{B\} \to \{B, C\}$ {B}→{B,C}
2. Now, check if we can reach CC:
   - Since we can derive CC from BB using $B \to C$ B→C, we can conclude that $C \in \{B\}^+$C{B}+.

step-3-verify-the-condition

Step 3: Verify the condition

- Since $C \subseteq \{B\}^+$C{B}+, we find that AA is indeed extraneous in the left-hand side of the functional dependency $AB \to C$ AB→C.

conclusion-17

Conclusion:

Yes, A**A is extraneous** in the left-hand side of the functional dependency $AB \to C$ AB→C because BB alone is sufficient to determine CC based on the functional dependencies given.

4o mini

you-said-25

find canonical form of For R(A, B,C); FD = {A -> BC, B -> C, A -> B, AB. -> C}}.

ChatGPT said:

To find the canonical cover (also known as the minimal cover) for the set of functional dependencies given for the relation $R(A, B, C)$R(A,B,C), we follow a systematic process that includes removing extraneous attributes and redundant functional dependencies.

given-functional-dependencies-4

Given Functional Dependencies:

1. $A \to BC$ A→BC

2. $B \to C$ B→C

3. $A \to B$ A→B

4. $AB \to C$ AB→C

step-1-decompose-the-functional-dependencies

Step 1: Decompose the Functional Dependencies

We start by breaking down the functional dependencies to ensure they are in the form where the right side has only one attribute.

- **From** $A \to BC$ A→BC, we can rewrite this as:
  - $A \to B$ A→B
  - $A \to C$ A→C

Now our set of functional dependencies is:

1. $A \to B$ A→B
2. $A \to C$ A→C
3. $B \to C$ B→C
4. $AB \to C$ AB→C

step-2-remove-extraneous-attributes

Step 2: Remove Extraneous Attributes

Next, we check for extraneous attributes in the left-hand sides of the functional dependencies:

1. **Check** $AB \to C$ **AB→C**:
   - To see if AA is extraneous, compute the closure of BB using the remaining FDs (excluding $AB \to C$ AB→C):

- Start with BB:
  - $B^+ = \{B\}$B+={B}
  - Using $B \to C$B→C:
    - $B^+ = \{B, C\}$B+={B,C}
  - AA is not needed to determine CC using BB. Thus, AA is **not** extraneous here.

- Next, check if BB is extraneous in $AB \to C$AB→C:
  - Compute the closure of AA:
    - Start with AA:
      - $A^+ = \{A\}$A+={A}
      - Using $A \to B$A→B:
      - $A^+ = \{A, B\}$A+={A,B}
      - Using $A \to C$A→C:
      - $A^+ = \{A, B, C\}$A+={A,B,C}
    - Since AA alone can derive CC, BB is **extraneous** in $AB \to C$AB→C.
- So we remove BB from $AB \to C$AB→C, yielding $A \to C$A→C as a standalone dependency.

The revised set of functional dependencies is:

1. $A \to B$A→B
2. $A \to C$A→C
3. $B \to C$B→C

step-3-remove-redundant-functional-dependencies-2

Step 3: Remove Redundant Functional Dependencies

Now, we check if any of the remaining dependencies can be derived from others:

1. **Check if $A \to C$A→C is redundant**:

- Compute the closure of AA using the remaining FDs ($A \rightarrow B$ A→B and $B \rightarrow C$ B→C):

  - Start with AA:

    * $A^+ = \{A\}$A+={A}
    * Using $A \rightarrow B$ A→B:

      · $A^+ = \{A, B\}$A+={A,B}
    * Using $B \rightarrow C$ B→C:

      · $A^+ = \{A, B, C\}$A+={A,B,C}
  - Thus, $A \rightarrow C$ A→C can be derived from $A \rightarrow B$ A→B and $B \rightarrow C$ B→C, making it redundant.

2. **Check if $B \rightarrow C$ B→C is redundant**:

- Compute the closure of BB using the remaining FDs:

  - Start with BB:

    * $B^+ = \{B\}$B+={B}
    * Using $B \rightarrow C$ B→C:

      · $B^+ = \{B, C\}$B+={B,C}
  - $B \rightarrow C$ B→C cannot be derived from other dependencies in the remaining set.

step-4-final-set-of-functional-dependencies

Step 4: Final Set of Functional Dependencies

After removing the extraneous attributes and redundant dependencies, we have the canonical cover:

Canonical Cover:

1. $A \to B$ A→B
2. $B \to C$ B→C

This is the minimal set of functional dependencies that preserves the original dependencies for the relation $R(A, B, C)$R(A,B,C).

4o mini

you-said-26

write the canonical cover of R(A,B ,C ,D ,E,F ,G ,H); F = {A -> CGH, AD -> C, DE -> F, G -> H}.

ChatGPT said:

To find the canonical cover (or minimal cover) for the set of functional dependencies given for the relation $R(A, B, C, D, E, F, G, H)$R(A,B,C,D,E,F,G,H), we will follow a systematic process of decomposing, removing extraneous attributes, and eliminating redundant functional dependencies.

given-functional-dependencies-5

Given Functional Dependencies:

1. $A \to CGH$ A→CGH
2. $AD \to C$ AD→C
3. $DE \to F$ DE→F
4. $G \to H$ G→H

step-1-decompose-functional-dependencies

Step 1: Decompose Functional Dependencies

First, we need to ensure that all functional dependencies have a single attribute on the right-hand side.

- From $A \to CGH$ A→CGH, we can break this down into:

- $A \to C$ A→C
- $A \to G$ A→G
- $A \to H$ A→H

Now our set of functional dependencies is:

1. $A \to C$ A→C
2. $A \to G$ A→G
3. $A \to H$ A→H
4. $AD \to C$ AD→C
5. $DE \to F$ DE→F
6. $G \to H$ G→H

step-2-remove-extraneous-attributes-1

Step 2: Remove Extraneous Attributes

Next, we check for extraneous attributes in the left-hand sides of the functional dependencies:

1. **Check** $AD \to C$ **AD→C**:

   - To see if AA is extraneous, we compute the closure of DD using the remaining functional dependencies (excluding $AD \to C$ AD→C):

     - Start with DD:

       * $D^+ = \{D\}$D+={D}
       * None of the remaining dependencies allow us to derive CC, so AA is not extraneous.

   - Next, check if DD is extraneous:

- Compute the closure of AA:

  - Start with AA:

    - $A^+ = \{A\}$A+={A}
    - Using $A \to C$ A→C:
    - $A^+ = \{A, C\}$A+={A,C}
    - Using $A \to G$ A→G:
    - $A^+ = \{A, C, G\}$A+={A,C,G}
    - Using $G \to H$ G→H:
    - $A^+ = \{A, C, G, H\}$A+={A,C,G,H}

  - Since CC can be derived without DD, DD is **extraneous** in $AD \to C$ AD→C.

- Remove DD from $AD \to C$ AD→C, yielding $A \to C$ A→C.

The revised set of functional dependencies is now:

1. $A \to C$ A→C
2. $A \to G$ A→G
3. $A \to H$ A→H
4. $DE \to F$ DE→F
5. $G \to H$ G→H

step-3-remove-redundant-functional-dependencies-3

Step 3: Remove Redundant Functional Dependencies

Now, we check if any of the remaining dependencies can be derived from others:

1. **Check if $A \to G$ A→G is redundant**:

- Compute the closure of AA using the remaining FDs:

- Start with AA:
  - $A^+ = \{A\}$A+={A}
  - Using $A \to C$ A→C:
    - $A^+ = \{A, C\}$A+={A,C}
  - Using $A \to H$ A→H:
    - $A^+ = \{A, C, H\}$A+={A,C,H}
  - Since GG cannot be derived from these, $A \to G$ A→G is **not** redundant.

2. **Check if $A \to H$ A→H is redundant**:

   - Again, use the closure of AA:
     - $A^+ = \{A, C, G\}$A+={A,C,G} using $A \to G$ A→G.
     - Since HH cannot be derived without $A \to H$ A→H, it is **not** redundant.

3. **Check if $G \to H$ G→H is redundant**:

   - Compute the closure of GG:
     - Start with GG:
       - $G^+ = \{G\}$G+={G}

* Using $G \to H$ G→H:

  · $G^+ = \{G, H\}$G+={G,H}

- Since HH cannot be derived from the other dependencies, $G \to H$ G→H is **not** redundant.

4. **Check if $DE \to F$ DE→F is redundant**:

- Compute the closure of DEDE:

  - Start with DEDE:

    * $DE^+ = \{D, E\}$DE+={D,E}
    * Using $DE \to F$ DE→F:

      · $DE^+ = \{D, E, F\}$DE+={D,E,F}

  - Since FF cannot be derived from the others, $DE \to F$ DE→F is **not** redundant.

final-set-of-functional-dependencies

Final Set of Functional Dependencies

After removing extraneous attributes and redundant dependencies, we have the canonical cover:

Canonical Cover:

1. $A \to C$ A→C
2. $A \to G$ A→G
3. $A \to H$ A→H
4. $DE \to F$ DE→F

5. $G \to H$ G→H

This set of functional dependencies is the minimal cover for the relation $R(A, B, C, D, E, F, G, H)$R(A,B,C,D
4o mini
you-said-27

show me how this: A -> C, A -> G, A -> H, DE -> F, G -> H is NOT the canonical cover of : R(A,B ,C ,D ,E,F ,G ,H); F = {A -> CGH, AD -> C, DE -> F, G -> H}
ChatGPT said:
To demonstrate how the functional dependencies

$$\{A \to C, A \to G, A \to H, DE \to F, G \to H\}$$

{A→C,A→G,A→H,DE→F,G→H}
is **not** the canonical cover for

$$F = \{A \to CGH, AD \to C, DE \to F, G \to H\}$$

F={A→CGH,AD→C,DE→F,G→H}
we need to show that this set of functional dependencies can be simplified further or that it is redundant.
step-1-check-if-all-dependencies-are-covered

Step 1: Check if All Dependencies are Covered

1. **Reconstruct Original Dependencies from Canonical Form:**

   - **From** $A \to C$ A→C, $A \to G$ A→G, and $A \to H$ A→H:

     - We can combine these as:

       $$A \to CGH$$

       A→CGH
     - This means that $A \to CGH$ A→CGH can be derived from the canonical form.

2. **From** $DE \to F$ DE→F:

- This dependency remains unchanged as it is not derivable from any other dependencies in the canonical form.

3. **From** $G \to H$ G→H:

   - This is also present in the original set and is not derived from the other dependencies.

step-2-check-for-redundant-functional-dependencies

Step 2: Check for Redundant Functional Dependencies

Now we need to check whether any of the dependencies in the proposed canonical cover are redundant:

1. **Redundancy Check for** $A \to H$ **A→H**:

   - Compute the closure of AA:

     - Start with AA: $A^+ = \{A\}$A+={A}
     - From $A \to C$ A→C: $A^+ = \{A, C\}$A+={A,C}
     - From $A \to G$ A→G: $A^+ = \{A, C, G\}$A+={A,C,G}
     - From $G \to H$ G→H: $A^+ = \{A, C, G, H\}$A+={A,C,G,H}

   - Thus, $A \to H$ A→H is **not** redundant, as it is derived from the closure.

2. **Redundancy Check for** $A \to G$ **A→G**:

   - Since $A \to G$ A→G is also essential for deriving other dependencies, it is **not** redundant.

3. **Redundancy Check for** $DE \to F$ **DE→F**:

- Compute the closure of DEDE:

  - Start with DEDE: $DE^+ = \{D, E\}$DE+={D,E}
  - From $DE \to F$DE→F: $DE^+ = \{D, E, F\}$DE+={D,E,F}

- Therefore, $DE \to F$DE→F is **not** redundant.

4. **Redundancy Check for $G \to H$G→H**:

- Compute the closure of GG:

  - Start with GG: $G^+ = \{G\}$G+={G}
  - From $G \to H$G→H: $G^+ = \{G, H\}$G+={G,H}

- Thus, $G \to H$G→H is also essential.

step-3-identify-potential-simplifications

Step 3: Identify Potential Simplifications

The canonical cover cannot be reduced further without losing essential dependencies, but we can check:

1. **Check if $A \to C$A→C and $A \to G$A→G are needed together**:

- Since $A \to CGH$A→CGH captures both $A \to C$A→C and $A \to G$A→G in a single dependency, it implies that $A \to C$A→C and $A \to G$A→G are not needed separately.

conclusion-18

Conclusion

Since the functional dependencies

$$\{A \to C, A \to G, A \to H, DE \to F, G \to H\}$$

{A→C,A→G,A→H,DE→F,G→H}
can be summarized into

$$\{A \to CGH, DE \to F, G \to H\}$$

{A→CGH,DE→F,G→H}
the original set FF has dependencies that can be merged into fewer rules. Hence, the proposed set is not minimal because it can be simplified further, which means that

$$\{A \to C, A \to G, A \to H, DE \to F, G \to H\}$$

{A→C,A→G,A→H,DE→F,G→H}
is **not** a canonical cover for FF since the essential dependency $A \to CGH$ A→CGH can express the same information in a more compact way.
correct-canonical-cover

Correct Canonical Cover

A better canonical cover for

$$F = \{A \to CGH, AD \to C, DE \to F, G \to H\}$$

F={A→CGH,AD→C,DE→F,G→H}
would be:

$$\{A \to CGH, DE \to F, G \to H\}$$

{A→CGH,DE→F,G→H}
This set covers all original dependencies with minimal redundancy.
4o mini
you-said-28
prove this {A→CGH, DE→F, G→H} is not the correct canonical cover of R(A,B ,C ,D ,E,F ,G ,H); F = {A -> CGH, AD -> C, DE -> F, G -> H}
ChatGPT said:
To prove that the set of functional dependencies $\{A \to CGH, DE \to F, G \to H\}$ {A→CGH,DE→F,G→H} is **not** the correct canonical cover for the relation $R(A, B, C, D, E, F, G, H)$R(A,B,C,D,E,F,G,H) with the original functional dependencies

$$F = \{A \to CGH, AD \to C, DE \to F, G \to H\},$$

F={A→CGH,AD→C,DE→F,G→H},

we will need to analyze whether the proposed set can derive all the original dependencies, and whether it is minimal and free of extraneous attributes.

step-1-derive-the-original-functional-dependencies

Step 1: Derive the Original Functional Dependencies

1. **Check if $AD \to C$ AD→C can be derived**:

   - Using the proposed canonical cover $\{A \to CGH, DE \to F, G \to H\}$ {A→CGH,DE→F,G→H}:

     - Start with ADAD:

       * $A^+ = \{A, C, G, H\}$A+={A,C,G,H} from $A \to CGH$ A→CGH.
       * However, DD is not included in this closure. Thus, we cannot derive CC using just AA and DD from the proposed set, which means we **cannot derive $AD \to C$ AD→C**.

Since $AD \to C$ AD→C is one of the original functional dependencies but cannot be derived from the proposed canonical cover, we can conclude that:

conclusion-1-not-all-original-dependencies-are-covered

Conclusion 1: Not All Original Dependencies are Covered

Since the dependency $AD \to C$ AD→C is missing from the proposed set, it indicates that the proposed set does not capture the same information as the original set FF.

step-2-assess-minimality-and-extraneous-attributes

Step 2: Assess Minimality and Extraneous Attributes

To ensure that a canonical cover is valid, it must also be minimal. This means that no functional dependency can be removed without losing the ability to derive all original dependencies.

1. **Assess Minimality**:

- Each dependency in $\{A \to CGH, DE \to F, G \to H\}$ {A→CGH,DE→F,G→H} needs to be checked if it is necessary:

  - **For** $A \to CGH$ **A→CGH**: It covers C, G, HC,G,H. Removing it would lose CC.
  - **For** $DE \to F$ **DE→F**: This is not immediately reducible; it is required to derive FF.
  - **For** $G \to H$ **G→H**: This dependency can be derived from $A \to CGH$ A→CGH because HH is already included in that closure.

conclusion-2-not-a-canonical-cover-due-to-missing-dependencies

Conclusion 2: Not a Canonical Cover Due to Missing Dependencies

Since the proposed canonical cover $\{A \to CGH, DE \to F, G \to H\}$ {A→CGH,DE→F,G→H} does not include all original functional dependencies from FF (specifically $AD \to C$ AD→C), it cannot be considered a correct canonical cover.

final-conclusion

Final Conclusion

The proposed set of functional dependencies $\{A \to CGH, DE \to F, G \to H\}$ {A→CGH,DE→F,G→H} is not a valid canonical cover for the relation $R(A, B, C, D, E, F, G, H)$R(A,B,C,D,E,F,G,H) because it fails to derive $AD \to C$ AD→C, which is essential in the original set of functional dependencies FF.

Thus, the canonical cover must include all original dependencies, and the proposed set does not fulfill that requirement. A correct canonical cover would need to ensure all dependencies in FF can be derived from it, which this set fails to do.

4o mini

you-said-29

is {A→CGH, DE→F, G→H} the correct canonical cover for R(A,B ,C ,D ,E,F ,G ,H); F = {A -> CGH, AD -> C, DE -> F, G -> H}

ChatGPT said:

To determine whether $\{A \to CGH, DE \to F, G \to H\}$ {A→CGH,DE→F,G→H} is the correct canonical cover for $R(A, B, C, D, E, F, G, H)$R(A,B,C,D,E,F,G,H) given the set of functional dependencies

$$F = \{A \to CGH, AD \to C, DE \to F, G \to H\},$$

F={A→CGH,AD→C,DE→F,G→H},

we need to check two main criteria:

1. **Coverage**: The proposed canonical cover must derive all the original dependencies.
2. **Minimality**: Each dependency in the proposed cover must be essential and cannot be removed without losing derivation capability.

step-1-check-coverage

Step 1: Check Coverage

1. **Derive $A \to CGH$ A→CGH**:
   - This dependency is included in both sets, so it can be derived.
2. **Derive $DE \to F$ DE→F**:
   - This dependency is also present in both sets, so it can be derived.
3. **Derive $G \to H$ G→H**:
   - This dependency is included in both sets, so it can be derived.
4. **Check $AD \to C$ AD→C**:
   - **Closure of ADAD** using the proposed set:
     - Start with ADAD: $AD^+ = \{A, D\}$AD+={A,D}.
     - From $A \to CGH$ A→CGH: $AD^+ = \{A, C, G, H, D\}$AD+={A,C,G,H,D}.
     - **C is in the closure**, so we can derive $AD \to C$ AD→C.

step-2-check-minimality

Step 2: Check Minimality

Next, we need to check if any of the dependencies in the proposed cover can be removed without losing the ability to derive the original dependencies.

1. **Check** $A \to CGH$ **A→CGH**:

   - This dependency cannot be removed as it is essential to derive C, G, H C,G,H.

2. **Check** $DE \to F$ **DE→F**:

   - This dependency cannot be removed as it is required to derive FF.

3. **Check** $G \to H$ **G→H**:

   - This dependency can be derived from $A \to CGH$ A→CGH since HH is included in that closure.
   - Thus, we can remove $G \to H$ G→H without losing derivation capability.

conclusion-19

Conclusion

The proposed set $\{A \to CGH, DE \to F, G \to H\}$ {A→CGH,DE→F,G→H} does cover all the original functional dependencies, but it is not minimal because the dependency $G \to H$ G→H is redundant.

correct-canonical-cover-1

Correct Canonical Cover

The correct canonical cover for the given functional dependencies would be:

$$\{A \to CGH, DE \to F\}$$

{A→CGH,DE→F}

This set includes all essential dependencies and does not have any redundancy.

final-answer

Final Answer

$\{A \to CGH, DE \to F, G \to H\}$ {A→CGH,DE→F,G→H} is **not** the correct canonical cover because it includes a redundant dependency $G \to H$ G→H. The correct canonical cover should be $\{A \to CGH, DE \to F\}$ {A→CGH,DE→F}.

4o mini

you-said-30

it should be CC: {{A}=>{C, G}, {D, E}=>{F}, {G}=>{H}}

ChatGPT said:

To verify that the canonical cover for the relation $R(A, B, C, D, E, F, G, H)$R(A,B,C,D,E,F,G,H) with functional dependencies

$$F = \{A \to CGH, AD \to C, DE \to F, G \to H\}$$

F={A→CGH,AD→C,DE→F,G→H}

is indeed

$$CC = \{A \to C, A \to G, DE \to F, G \to H\},$$

CC={A→C,A→G,DE→F,G→H},

we will follow the process of computing the canonical cover step by step.

step-1-decompose-the-functional-dependencies-1

Step 1: Decompose the Functional Dependencies

The first step in finding a canonical cover is to ensure that all functional dependencies are in a single attribute on the right-hand side.

1. From $A \to CGH$ A→CGH:

    - We can decompose this into:

        - $A \to C$ A→C
        - $A \to G$ A→G
        - $A \to H$ A→H

Thus, we have the following dependencies:

- $A \to C$ A→C

- $A \to G$ A→G
- $A \to H$ A→H
- $AD \to C$ AD→C
- $DE \to F$ DE→F
- $G \to H$ G→H

step-2-remove-extraneous-attributes-2

Step 2: Remove Extraneous Attributes

Next, we need to check for extraneous attributes in the left-hand side of the dependencies.

1. **Check if DD is extraneous in $AD \to C$ AD→C**:

    - The closure of AA under FF (without DD):

        - Start with AA: $A^+ = \{A, C, G, H\}$A+={A,C,G,H}.
        - Since CC is included in A^+A+, DD is extraneous in $AD \to C$ AD→C.

2. The dependency can thus be simplified to:

    - $A \to C$ A→C

3. **Remove redundant dependencies**:

    - The $G \to H$ G→H dependency can be checked for redundancy by considering $A \to G$ A→G (which indirectly leads to $G \to H$ G→H).
    - Since HH can be derived from $A \to G$ A→G and $G \to H$ G→H is not strictly needed in the context of FF after simplification.

resulting-dependencies

Resulting Dependencies

After decomposing and removing extraneous attributes, we get:

- $A \to C$ A→C
- $A \to G$ A→G
- $DE \to F$ DE→F
- $G \to H$ G→H

This means the correct canonical cover is:

$$CC = \{A \to C, A \to G, DE \to F, G \to H\}$$

CC={A→C,A→G,DE→F,G→H}
conclusion-20

Conclusion

The proposed canonical cover you provided, $\{A \to C, A \to G, DE \to F, G \to H\}$ {A→C,A→G,DE→F,G→H} is correct and minimal, satisfying the properties of a canonical cover.

So indeed, the canonical cover can be represented as:

$$CC = \{A \to C, A \to G, DE \to F, G \to H\}$$

CC={A→C,A→G,DE→F,G→H}

This confirms that your statement is accurate. Thank you for your patience while we clarified the derivation process!

4o mini

you-said-31

now what is the canonical cover of R(A,B ,C ,D ,E,F ,G ,H); F = {A -> CGH, AD -> C, DE -> F, G -> H}

ChatGPT said:

To compute the canonical cover for the relation $R(A, B, C, D, E, F, G, H)$R(A,B,C,D,E,F,G,H) given the set of functional dependencies

$$F = \{A \to CGH, AD \to C, DE \to F, G \to H\},$$

F={A→CGH,AD→C,DE→F,G→H},

we will follow the steps of decomposing the functional dependencies, removing extraneous attributes, and ensuring minimality.

step-1-decompose-the-functional-dependencies-2

Step 1: Decompose the Functional Dependencies

The first step is to ensure that all functional dependencies have a single attribute on the right-hand side.

1. From $A \to CGH$ A→CGH, we can decompose this into:

    - $A \to C$ A→C
    - $A \to G$ A→G
    - $A \to H$ A→H

Now we have the following dependencies:

- $A \to C$ A→C
- $A \to G$ A→G
- $A \to H$ A→H
- $AD \to C$ AD→C
- $DE \to F$ DE→F
- $G \to H$ G→H

step-2-remove-extraneous-attributes-3

Step 2: Remove Extraneous Attributes

Next, we need to check for extraneous attributes in the left-hand side of the dependencies.

1. **Check if DD is extraneous in $AD \to C$ AD→C**:

    - Compute the closure of AA under the set of functional dependencies:

        - Start with AA: $A^+ = \{A\}$A+={A}.
        - Using $A \to C$ A→C and $A \to G$ A→G, we get $A^+ = \{A, C, G\}$A+={A,C,G}.

- Since CC is derived from AA alone, DD is extraneous in $AD \rightarrow C$ AD→C.

Thus, we can simplify this dependency to:

- $A \rightarrow C$ A→C

step-3-final-dependencies

Step 3: Final Dependencies

Now we check for redundancy in the remaining dependencies:

- We still have:
  - $A \rightarrow C$ A→C
  - $A \rightarrow G$ A→G
  - $DE \rightarrow F$ DE→F
  - $G \rightarrow H$ G→H

Check for any redundancy:

- $G \rightarrow H$ G→H is required to derive HH from GG, and it cannot be derived from other dependencies, so it remains.
- $DE \rightarrow F$ DE→F cannot be derived from the other dependencies as well.

resulting-canonical-cover

Resulting Canonical Cover

After decomposition and removal of extraneous attributes, the canonical cover is:

$$CC = \{A \rightarrow C, A \rightarrow G, DE \rightarrow F, G \rightarrow H\}$$

CC={A→C,A→G,DE→F,G→H}

summary

Summary

The canonical cover for the relation $R(A, B, C, D, E, F, G, H)$R(A,B,C,D,E,F,G,H) given the functional dependencies $F = \{A \rightarrow CGH, AD \rightarrow C, DE \rightarrow F, G \rightarrow H\}$ F={A→CGH,AD→C,DE→F,G→H} is:

$$CC = \{A \to C, A \to G, DE \to F, G \to H\}$$

CC={A→C,A→G,DE→F,G→H}

This set is minimal, as each functional dependency is necessary and cannot be further simplified.

4o mini

you-said-32

what is the canonical cover of r3 (A,B ,C ,D ,E,F ,G ,H); F = {{A}=>{C , G, H}, { A , D}=>{C}, {D,E}=>{F},{G}=>{H}}.

ChatGPT said:

To find the canonical cover for the relation $R(A, B, C, D, E, F, G, H)$R(A,B,C,D,E,F,G,H) with the set of functional dependencies

$$F = \{A \to CGH, AD \to C, DE \to F, G \to H\},$$

F={A→CGH,AD→C,DE→F,G→H},

we'll follow the steps of decomposing functional dependencies, removing extraneous attributes, and ensuring minimality.

step-1-decompose-the-functional-dependencies-3

Step 1: Decompose the Functional Dependencies

First, we need to ensure that all functional dependencies have a single attribute on the right-hand side.

1. From $A \to CGH$ A→CGH, we can decompose this into:

    - $A \to C$ A→C
    - $A \to G$ A→G
    - $A \to H$ A→H

Now, the set of functional dependencies is:

- $A \to C$ A→C
- $A \to G$ A→G
- $A \to H$ A→H

- $AD \to C$ AD→C
- $DE \to F$ DE→F
- $G \to H$ G→H

step-2-remove-extraneous-attributes-4

Step 2: Remove Extraneous Attributes

Next, we check for extraneous attributes in the left-hand side of the dependencies.

1. **Check if DD is extraneous in $AD \to C$ AD→C**:

   - Compute the closure of AA under the set of functional dependencies:

     - Start with AA: $A^+ = \{A\}$A+={A}.
     - Using $A \to C$ A→C, we get $A^+ = \{A, C\}$A+={A,C}.
     - Since CC can be derived from AA alone, DD is extraneous in $AD \to C$ AD→C.

Thus, we can simplify this dependency to:

- $A \to C$ A→C

step-3-check-for-redundancy

Step 3: Check for Redundancy

Now we check for redundancy in the remaining dependencies:

1. **Check $A \to G$ A→G**:

   - This cannot be derived from other dependencies.

2. **Check $A \to H$ A→H**:

- This also cannot be derived from other dependencies.

3. **Check** $DE \to F$ **DE→F**:

   - This dependency cannot be derived from the other dependencies.

4. **Check** $G \to H$ **G→H**:

   - This dependency cannot be derived from the others.

resulting-canonical-cover-1

Resulting Canonical Cover

After decomposition and removal of extraneous attributes, the canonical cover is:

$$CC = \{A \to C, A \to G, A \to H, DE \to F, G \to H\}$$

CC={A→C,A→G,A→H,DE→F,G→H}
summary-1

Summary

The canonical cover for the relation $R(A, B, C, D, E, F, G, H)$R(A,B,C,D,E,F,G,H) given the functional dependencies

$$F = \{A \to CGH, AD \to C, DE \to F, G \to H\}$$

F={A→CGH,AD→C,DE→F,G→H}
is:

$$CC = \{A \to C, A \to G, A \to H, DE \to F, G \to H\}$$

CC={A→C,A→G,A→H,DE→F,G→H}
This set is minimal, as each functional dependency is necessary and cannot be further simplified.
4o mini
prompt-textarea

composer-background