Decomposition algorithm - Exercises

SAPIENZA
UNIVERSITÀ DI ROMA

The content of the slides is authored by Prof. Maria de Marsico, Sapienza University of Rome
Part of the material is taken from the book: Atzeni, Ceri, Paraboschi, Torlone Basi di dati - Modelli e linguaggi di interrogazione , McGraw-Hill, 2002.

Example 1

given the following schema:

$$R = (A, B, C, D, E, H)$$

and the following set of functional dependencies:

$$F = \{ AB \rightarrow CD, C \rightarrow E, AB \rightarrow E, ABC \rightarrow D \}$$

- verify that ABH is a key of R
- knowing that ABH is the only key of R, verify that R is not in 3NF
- find a minimal cover G of F
- find a decomposition of R such that it preserves G and every schema in is in 3NF
- find a decomposition of R such that it preserves G, has a lossless join and every schema in is in 3NF

Example 1: key

SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

$R = (A, B, C, D, E, H)$

$F = \{ AB \rightarrow CD, C \rightarrow E, AB \rightarrow E, ABC \rightarrow D \}$

verifying that ABH is a key means checking two conditions:

- ABH must **functionally determine the entire** schema
- **no subset of ABH must functionally determine the entire schema**

- To check the first condition we check if the closure of the attribute set ABH contains all the attributes of the schema. Applying the algorithm for calculating the closure of a set of attributes, at the first step we insert in the closure of ABH the attributes C, D and E thanks to the functional dependencies in which the left part is contained in ABH, i.e. AB CD and AB *E*, so we can also stop because we have inserted all the attributes of the schema, i.e. we have verified $(ABH)^+ = \{A,B,C,D,E,H\}$.

Example 1: key

SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

R = (A, B, C, D, E, H)

F = { AB → CD, C → E, AB → E, ABC → D }

Verifying that ABH is a key means verifying two conditions

- ABH must **functionally determine the entire** schema
- **No subset of ABH must functionally determine the entire schema**

- To check the second condition, we need to check **that the closure of no subset of ABH** contains all attributes of the schema. Here we note that **H must appear in any case in a key of** the schema, because it is not determined by other attributes. Thus we can avoid computing the closures of the single attributes A and B, which could not determine H even by transitivity. Moreover, H never appears to the left of the dependencies, so by itself it does not determine any attribute. It remains to check the closures of AH and BH, but it is trivial to verify that in both cases the algorithm would end immediately because there are no dependencies with subsets of AH or BH on the left.
- We can therefore conclude that **ABH is key to the given schema**.

Example 1: key

Sapienza Università di Roma
Dipartimento di Informatica

$\boxed{R = (A, B, C, D, E, H)}$ $\boxed{F = \{ AB \rightarrow CD, C \rightarrow E, AB \rightarrow E, ABC \rightarrow D \}}$

- To verify that the schema is not in third normal form, it is enough to observe the presence of the partial dependencies AB CD and AB *E*.

Example 1: minimal cover

SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

$F = \{ AB \rightarrow CD, C \rightarrow E, AB \rightarrow E, ABC \rightarrow D \}$

- (step 1) to find a minimal cover, we first reduce the dependents to singleton

  $F = \{AB \rightarrow C, AB \rightarrow D, C \rightarrow E, ABC \rightarrow D\}$

- (step 2) now we need to check whether there are redundancies in the dependencies in the dependents:
- $AB \rightarrow C$: $A \rightarrow C \in F^+$, i.e., $C \in (A)^+_F$? $(A)^+_F = A$; so, $C \notin (A)^+_F$
- $AB \rightarrow C$: $B \rightarrow C \in F^+$, i.e., $C \in (B)^+_F$? $(B)^+_F = B$; so, $C \notin (B)^+_F$
- $AB \rightarrow E$: $A \rightarrow E \in F^+$, i.e., $E \in (A)^+_F$? $(A)^+_F = A$; so, $E \notin (A)^+_F$
- $AB \rightarrow E$: $B \rightarrow E \in F^+$, i.e., $E \in (B)^+_F$? $(B)^+_F = B$; so, $E \notin (B)^+_F$
- $ABC \rightarrow D$: as $AB \rightarrow D$ exists in the dependency set, not only can we eliminate the attribute C, but we can eliminate the entire dependency $ABC \rightarrow D$!

Example 1: minimal cover

SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

$G = \{ AB \rightarrow C, AB \rightarrow D, C \rightarrow E, AB \rightarrow E \}$

- (step 3) let us now see if this set contains redundant dependencies
- we observe that C is determined by AB only, so by eliminating $AB \rightarrow C$, we cannot determine C anymore, so the dependency cannot be deleted
- the same applies to D
- $C \rightarrow E$: we check if $E \in (C)^+_G$, where $G = \{ AB \rightarrow C, AB \rightarrow D, AB \rightarrow E\}$; but $(C)^+_G = \{C\}$
- $AB \rightarrow E$: we check if $E \in (AB)^+_G$, where $G = \{ AB \rightarrow C, AB \rightarrow D, AB \rightarrow E \}$; $(AB)^+_G = \{A,B,C,D,E\}$, so the dependency can be deleted
- the minimal cover is: $G = \{AB \rightarrow C, AB \rightarrow D, C \rightarrow E\}$

Example 1: decomposition

SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

$R = (A, B, C, D, E, H)$ $F = \{AB \rightarrow C, AB \rightarrow D, C \rightarrow E\}$

- (step 1) attribute H does not appear in any dependency, so S = {H}, ρ = {H} and R becomes (A,B,C,D,E)
- (step 2) we check if there are dependencies involving all the attributes of the schema, but there are not
- (step 3) the final ρ = {H, ABC, ABD, CE}

- in order to have a decomposition with lossless join, we add a subschema that contains the key ABH:

  ρ = { {H}, {A,B,C}, {A,B,D}, {C,E}, {A,B,H} } = { H, ABC, ABD, CE, ABH }

Example 2

given the following schema:

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

and the following set of functional dependencies:

$$F = \{ AB \rightarrow C, B \rightarrow D, D \rightarrow C \}$$

- check if R is in 3NF
- if not, find a decomposition, such that:
  - every schema of the decomposition is in 3NF,
  - the decomposition preserves F,
  - the decomposition has a lossless join

Example 2: key

SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

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

F = { AB → C, B → D, D → C }

- **attribute E must be part of any key**, as it does not appear in the dependencies and thus can only be functionally determined by reflexivity
- **C never appears in a determinant**, neither alone nor with other attributes, so **it won't be part of any key**
- $(AB)^+_F = \{A,B,C,D\}$
- only E is missing, so we check $(ABE)^+_F = \{A,B,C,D,E\} = R$
  - let's check if ABE is minimal: $(AE)^+_F = \{A, E\}$, $(BE)^+_F = \{B,C,D,E\}$, so ABE is minimal
- let's calculate the closures of the other determinants, to check if we can find another key
- It is useless to try B… as E must always be in any key, and we have already checked BE; so, let's try with DE: $(DE)^+_F = \{D, E, C\}$
- conclusion: ABE is the only key
- in F there are some partial dependencies (AB→C, B→D) and a transitive dependency (D→C), so, **F is not in 3NF**

Example 2: minimal cover

SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

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

F = { AB → C, B → D, D → C }

- (step 1) all dependents are already singleton
- (step 2) let's try to reduce AB→C: $(A)^+_F = \{A\}$, $(B)^+_F = \{B,D,C\}$, which contains C, so, the dependency can be reduced to B→C

G = { B → C, B → D, D → C }

- (step 3) we check if there are redundant dependencies: we start by B → C with G = {B → *D*, D → C }, we obtain $(B)^+_G = \{B,D,C\}$, so, the dependency can be deleted

G = { B → D, D → C }

- it is useless to try the other one, as the dependents appear only once, so we found the minimal cover:

G = { B → D, D → C }

Example 2: decomposition

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

$G = \{ B \rightarrow D, D \rightarrow C \}$

- (step 1) attributes A and E do not appear in any dependency, so S = {AE} and ρ = { {AE} }, while R becomes (B,C,D)
- (step 2) there **are no** dependencies involving all the attributes of R
- (step 3) ρ = {AE, BD, DC}

- to obtain a decomposition with a lossless join, we add the key ABE:

ρ = { {A, E}, {B,D}, {D, C}, {A, B, E} } = {AE, BD, DC, ABE}

Example 3

given the relational schema **R = ABCDEH** and the set of functional dependencies:

$$\mathbf{F = \{ D \rightarrow H, B \rightarrow AC, CD \rightarrow H, C \rightarrow AD \}}$$

- determine the unique key of R
- say why R with the set of functional dependencies F is not in 3NF
- find a decomposition of **R** such that:
  - each schema in is in 3NF
  - preserves F
  - has a lossless join

Example 3: key

SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

R = ABCDEH

F = { D → H, B → AC, CD → H, C → AD }

- to check if the schema is 3NF we must first identify the key
- **attribute E must be part of the key**, as it does not appear in any dependency
- **A never appears in a determinant**, neither alone nor with other attributes, so **it will not be part of the key**
- we check $(D)^+_F = \{D, H\}$; even by adding E, we do not obtain the entire schema
- we check $(B)^+_F = \{B, A, C, D, H\}$; by adding E, we obtain R, so, BE is a super key
  - BE is also minimal, as the closure of B does not contain E and the closure of E contains only E
- BE is a key, and we know that it is unique
- B → A and B → C in F^+ are partial dependencies, C → A and C → D in F^+ are transitive dependencies, so, **F is not in 3NF**

Example 3: minimal cover

SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

R = ABCDEH

F = { D → H, B → AC, CD → H, C → AD }

- (step 1) dependents are reduced to singletons:

G = { D → H, B → A, B → C, CD → H, C → A, C → D }

- (step 2) CD → H : we can eliminate it, as D → H is in F

G = { D → H, B → A, B → C, C → A, C → D }

- (step 3) it is useless to try to eliminate D → H, B → C and C → D, as the dependets appear in these dependencies only
- B → A: by considering G = {D → H, B → C, C → A, C → D } we obtain $(B)^+_G$ = {B, C, A, D, H}, so, the dependency can be deleted and we obtain:

G = { D → H, B → C, C → A, C → D }

- which is a minimal cover

Example 3: decomposition

SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

R = ABCDEH **G = { D → H, B → C, C → A, C → D }**

- (step 1) attribute E does not appear in any dependency, so ρ = {S} = { {E} } and the schema R becomes (A, B, C, D, H)
- (step 2) there are no dependencies involving all the attributes in the schema
- (step 3) the output is: ρ = {E, DH, BC, CA, CD}

- we add the key BE:

  ρ = {E, DH, BC, CA, CD, BE}

Example 4

given the relational schema **R = ABCDEHI**
and the set of functional dependencies **F = { A → C , C → D , BI → H , H → I }**

- find the two keys of the **R schema** and explain why they are keys
- say if **R** is in 3NF, and why
- provide a decomposition ρ of R such that:
  - each schema in ρ is in 3NF
  - ρ preserves F
  - ρ has a lossless join

Example 4: keys

SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

R = ABCDEHI

F = { A → C , C → D , BI → H , H → I }

- **attribute E must be part of any key**, as it does not appear in any dependency, so it can be functionally determined by reflexivity only
- **A and B must** also be **part of any key**, as they are not in any dependent
- **D never appears** in a determinant, so **it will not be part of any key**
- $(AB)^+_F$ = {A, B, C, D}; even by adding E, we do not obtain the schema R
- $(C)^+_F$ = {C, D}; even by adding E, we do not obtain the schema R
- $(BI)^+_F$ = {B, I, H}; even by adding E, we do not obtain the schema R
- $(H)^+_F$ = {H, I}; even by adding E, we do not obtain the schema R
- looking at the above closures (what is missing and how it could be inserted), we notice that we could try ABEI and ABEH: $(ABEI)^+_F$ = {A, B, E, I, C, D, H} = R $(ABEH)^+_F$ = {A, B, E, H, C, D, I} = R
- minimality is implicit in the evidence from the above closures
- C → D is transitive, the other ones are partial, so, **F is not in 3NF**

Example 4: minimal cover

SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

R = ABCDEHI

F = { A → C , C → D , BI → H , H → I }

- (step 1) dependents are already singleton
- (step 2) we try to reduce BI→H : $(B)^+_F = \{B\}$ and $(I)^+_F = \{I\}$, so, it cannot be reduced
- (step 3) we check for redundant dependencies:
- it is useless to try to eliminate them all, as all the dependents appear only once
- F is already minimal:

F = { A → C , C → D , BI → H , H → I }

Example 4: decomposition

SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

R = ABCDEHI **F = { A → C , C → D , BI → H , H → I }**

- (step 1) E does not appear in any dependency, so ρ = { S } = { {E} } and the schema R becomes (A, B, C, D, H, I)
- (step 2) there are no dependencies involving all the attributes of the schema
- (step 3) ρ = {E, AC, CD, BIH, HI}
- we add **one of the keys** (ABEI or ABEH)
- there are **two possible** decompositions:

  1 ρ = {E, AC, CD, BIH, HI, ABEI}

  2 ρ = {E, AC, CD, BIH, HI, ABEH}

Example 5

SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

given the relation schema **R = ABCD** and the set of functional dependencies

$F = \{ AB \rightarrow C, AD \rightarrow BC, AC \rightarrow B, B \rightarrow D \}$

- find the keys of **R**, and explain why they are keys
- say if **R** is in 3NF, and justify your answer
- provide a decomposition of R such that:
  - each schema in is in 3NF
  - it preserve F
  - it has a lossless join

Example 5: keys

R = ABCD

F = {AB → C, AD → BC, AC → B, B → D}

- A is not determined by any attribute, so it must be in each key
- $(AB)^+_F$ = {A, B, C, D}, and AB is minimal, as $(A)^+_F$ = {A} and $(B)^+_F$ = {B, D}
- $(AD)^+_F$ = {A, D, B, C}, and AD is minimal, as $(D)^+_F$ = {D}
- $(AC)^+_F$ = {A, C, B, D}, and AC is minimal, as $(C)^+_F$ = {C}

- all the determinants are super keys, and B → D doesn't create any issue, as D is prime
- even if the schema is already in 3NF, let us suppose that we want to decompose it anyway

Example 5: minimal cover

SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

R = ABCD

F = {AB → C, AD → BC, AC → B, B → D}

- we have already determined a minimal cover of this set; indeed, we have found two of them:

F1 = {AD → C, AC → B, B → D}

F2 = {AB → C, AD → B, AC → B, B → D}

Example 5: decomposition 1

SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

R = ABCD

$F1 = \{AD \to C, AC \to B, B \to D\}$

- (step 1) there are no attributes to be added to S
- (step 2) there **are no** dependencies involving all the attributes of the schema
- (step 3) ρ_1 = {ADC, ACB, BD}
- we add **one of the keys** (AB, AD or AC), if they are not already contained in one or more of the subschemas; in fact, all of them are!
- so, ρ_1 = {ADC, ACB, BD}

Example 5: Decomposition/2

R = ABCD

F2 = {AB → C, AD → B, AC → B, B → D}

- (step 1) there are no attributes to be added to S
- (step 2) there **are no** dependencies involving all the attributes of the schema
- (step 3) ρ_2 = {ABC, ADB, ACB, BD}
- we add **one of the keys** (AB, AD or AC), if they are not already contained in one or more of the subschemas; in fact, all of them are!
- so, ρ_2 = {ABC, ADB, ACB, BD}