COMP 353 Databases
Assignment no.2

Duc Nguyen

Gina Cody School of Computer Science and Software Engineering
Concordia University, Montreal, QC, Canada

Summer 2020

Contents

1 Question 1: [10 marks] 3

1.1 Problem Description: . . . 3
1.2 A. . . . 3
1.3 B. . . . 3
1.4 C. . . . 3

2 Question 2: [10 marks] 4

2.1 Problem Description . . . 4
2.2 {B -> C, D -> A}: decompose into BC and AD. . . . 5
2.2.1 Find the candidate key(s): . . . 5
2.2.2 Evaluate the decomposition . . . 5
2.3 {AB -> C, C-> A, C -> D}: decompose into ACD and BC. . . . 6
2.3.1 Find the candidate key(s): . . . 6
2.3.2 Evaluate the decomposition . . . 6
2.4 {A -> BC, C -> AD}: decompose into ABC and AD. . . . 7
2.4.1 Find the candidate key(s): . . . 7
2.4.2 Evaluate the decomposition . . . 7
2.5 {A -> B, B -> C, C -> D}: decompose into AB and ACD. . . . 7
2.5.1 Find the candidate key(s): . . . 7
2.5.2 Evaluate the decomposition . . . 8
2.6 {A -> B, B -> C, C -> D}: decompose into AB, AD and CD. . . 9
2.6.1 Find the candidate key(s): . . . 9
2.6.2 Evaluate the decomposition . . . 9

3 Question 3: [10 marks] 10

3.1 Problem Description: . . . 10
3.2 Find all the candidate keys of F. Prove that these are the only keys 11
3.3 Derive a canonical cover for F in a systematic manner. . . . 12
3.3.1 Make every function dependency X in G determine only 1 attribute: . . . 12
3.4 Minimize the left hand side X of every FD . . . 12
3.5 Remove redundant FD, if any . . . 12
3.6 Does a set of FDs have a unique canonical cover? Why? . . . 13

1 Question 1: [10 marks]

1.1 Problem Description:

Prove or disprove the following statements. To prove a statement is true, give a formal argument (in cases involving implications among FDs, use Armstrongs Axioms). To disprove falsity, give a counterexample.

1.2 A.

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

$$C \rightarrow AB \Rightarrow C \rightarrow AANDC \rightarrow B(Decomposition/Splittingrule)$$

With:

- A -> B
- C -> A

=> C -> B
Conclusion: The statement is true

1.3 B.

$\{KL \rightarrow M, L \rightarrow N\} \Rightarrow \{KN \rightarrow M\}$
Based on pseudo transitivity rule => **Conclusion**: The statement is true

1.4 C.

$\{A \rightarrow C, BD \rightarrow A, C \rightarrow D\} \Rightarrow \{AB \rightarrow CD\}$

- A -> C => AB -> C
- AB -> C, C -> D => AB -> D
- AB -> C, AB -> D => AB -> CD

Conclusion: The statement is true

2 Question 2: [10 marks]

2.1 Problem Description

Suppose you are given a relation scheme R = A, B, C, D. For each of the following sets of functional dependencies, assuming those are the only dependencies that hold for R, do the following:

- Identify the candidate key(s) for R.
- State whether or not the proposed decomposition of R into smaller relations is a good decomposition, briefly explaining why or why not.

The super key with all they keys in it: {A,B,C,D} $\{A, B, C, D\}^+ = \{A, B, C, D\}$

2.2 {B -> C, D -> A}: decompose into BC and AD.

2.2.1 Find the candidate key(s):

From the set of function dependencies:

- C is redundant if there is B in the key
- A is redundant if there is D in the key

The keys left are: {B,D}
$\{B, D\}^+ = \{A, B, C, D\}$ => {B,D} is a super key. Proper subsets of {B,D} are:

- {B}. $\{B\}^+ = \{B, C\}$ => not a super key.
- {D}. $\{D\}^+ = \{D, A\}$ => not a super key.

No proper subsets of {B, D} are super keys => {B,D} is a candidate key. Moreover, there is no function dependencies in F that determine the keys B or D (prime keys). hence, it is the only candidate key of the relation.

Conclusion: Candidate key: { B, D }

2.2.2 Evaluate the decomposition

- $R_1(BC) \cup R_2(AD) = R(ABCD)$
- $R_1(BC) \cap R_2(AD) = \emptyset$

=> **The decomposition is lossy.**
=> **The decomposition is not good.**

2.3 {AB -> C, C-> A, C -> D}: decompose into ACD and BC.

2.3.1 Find the candidate key(s):

From the set of function dependencies:

- A is redundant if there is C in the key
- D is redundant if there is C in the key

The keys left are: {B,C}
$\{B, C\}^+ = \{A, B, C, D\}$ => {B,C} is a super key. Proper subsets of {B,C} are:

- {B}. $\{B\}^+ = \{B\}$ => not a super key.
- {C}. $\{C\}^+ = \{C, D, A\}$ => not a super key.

No proper subsets of {B, C} are super keys => {B,C} is a candidate key.
Prime keys: B, C

The prime key C can be determined by AB in F => try replacing C with AB => Key: {A,B}
$\{A, B\}^+ = \{A, B, C, D\}$ => {A,B} is a super key. Proper subsets of {A,B} are:

- {A}. $\{A\}^+ = \{A\}$ => not a super key.
- {B}. $\{B\}^+ = \{B\}$ => not a super key.

No proper subsets of {A, B} are super keys => {A,B} is a candidate key.
Prime keys: A, B, C

Finished checking all the FD's in which the right hand side is a prime key
Conclusion: Candidate keys: {B,C}, {A,B}

2.3.2 Evaluate the decomposition

- $R_1(ACD) \cup R_2(BC) = R(ABCD)$
- $R_1(ACD) \cap R_2(BC) = C \neq \emptyset$
- $C^+ = \{C, A, D\}$ => C is super key of R_1

=> **The decomposition is lossless.**
=> **The decomposition is good.**

2.4 {A -> BC, C -> AD}: decompose into ABC and AD.

2.4.1 Find the candidate key(s):

Splitting the FD's in F:

- {A->BC} => {A->B} AND {A->C} (Splitting rule)
- {C->AD} => {C->A} AND {C->D} (Splitting rule)
- {A->C}, {C->D} => {A->D} (Transitivity)

F would be transformed to: {A->B, A->C, A->D, C->A, C->D}
From the set of function dependencies:

- B,D is redundant if there is A in the key
- D is redundant if there is C in the key

The key left is {A}
$\{A\}^+ = \{A, B, C, D\}$ => {A} is a super key.
There are no proper subsets of {A} => {A} is a candidate key. **Prime keys:** A
The prime key A can be determined by C in F => try replacing A with C => Key: {C}
$\{C\}^+ = \{A, B, C, D\}$ => {C} is a super key.
There are no proper subsets of {C} => {C} is a candidate key.
Prime keys: A, C
Finished checking all the FD's in which the right hand side is a prime key
Conclusion: Candidate keys: {A}, {C}

2.4.2 Evaluate the decomposition

- $R_1(ABC) \cup R_2(AD) = R(ABCD)$
- $R_1(ABC) \cap R_2(AD) = A \neq \emptyset$
- $A^+ = \{A, B, C, D\}$ => A is super key of both R_1 and R_2

=> **The decomposition is lossless.**
=> **The decomposition is good.**

2.5 {A -> B, B -> C, C -> D}: decompose into AB and ACD.

2.5.1 Find the candidate key(s):

Extend the FD's in F:

- {A->B}, {B->C} => {A->C} (Transitivity)
- {A->C}, {C->D} => {A->D} (Transitivity)

F would be transformed to: {A->B, B->C, C->D, A->C, A->D}
From the set of function dependencies:

- B,C, D is redundant if there is A in the key

The key left is {A}
$\{A\}^+ = \{A, B, C, D\}$ => {A} is a super key.
There are no proper subsets of {A} => {A} is a candidate key. Finished checking all the FD's in which the right hand side is a prime key
Conclusion: Candidate key: {A}

2.5.2 Evaluate the decomposition

- $R_1(AB) \cup R_2(ACD) = R(ABCD)$
- $R_1(AB) \cap R_2(ACD) = A \neq \emptyset$
- $A^+ = \{A, B, C, D\}$ => A is super key of both R_1 and R_2

=> **The decomposition is lossless.**
=> **The decomposition is good.**

2.6 {A -> B, B -> C, C -> D}: decompose into AB, AD and CD.

2.6.1 Find the candidate key(s):

The relation and the function dependencies are exactly the same as in question d => **Candidate key: {A}**

2.6.2 Evaluate the decomposition

	A	B	C	D
R1(AB)	α	α	β	β
R2(AD)	α	β	α	β
R3(CD)	β	β	α	α

Use function dependencies to transform the table

	A	B	C	D
R1(AB)	α	α	β	β
R2(AD)	α	α	α	α
R3(CD)	β	β	α	α

The second row of R2 is full of α
=> **The decomposition is lossless.**
=> **The decomposition is good.**

3 Question 3: [10 marks]

3.1 Problem Description:

You are given a relation scheme R = B, N, S, T, A, R, C where B = Building, N = Door Number, S = Street, T = Type, A = Architect, R = Subcontractor and C = Class. Constraints between the attributes can be expressed in the form of the following functional dependencies:
F = {AB -> T, A -> B, R -> C, NS -> BT}.

3.2 Find all the candidate keys of F. Prove that these are the only keys

From the set F:
=>

- {NS -> BT} => NS -> B AND NS -> T (Splitting rule)

The first super key contains all the attributes:
$\{B, N, S, T, A, R, C\}^+$ = B, N, S, T, A, R, C

From the set of function dependencies F:

- B is redundant if there are N,S in the key
- T is redundant if there are N,S in the key
- C is redundant if there is R in the key

=> The keys left are N, S, A, R
$\{N, S, A, R\}^+ = \{N, S, A, R, B, T, C\}$ => {N,S,A,R} is a super key
Proper subsets of N,S,A,R are:

- {N}. $\{N\}^+ = \{N\}$ => not a super key
- {S}. $\{S\}^+ = \{S\}$ => not a super key
- {A}. $\{A\}^+ = \{A, B, T\}$ => not a super key
- {R}. $\{R\}^+ = \{R, C\}$ => not a super key
- {N, S}. $\{N, S\}^+ = \{N, S, B, T\}$ => not a super key
- {N, A}. $\{N, A\}^+ = \{N, A, B, T\}$ => not a super key
- {N, R}. $\{N, R\}^+ = \{N, R, C\}$ => not a super key
- {S, A}. $\{S, A\}^+ = \{S, A, B, T\}$ => not a super key
- {S, R}. $\{S, R\}^+ = \{S, R, C\}$ => not a super key
- {A, R}. $\{A, R\}^+ = \{A, R, B, T, C\}$ => not a super key
- {N, S, A}. $\{N, S, A\}^+ = \{N, S, A, B, T\}$ => not a super key
- {N, S, R}. $\{N, S, R\}^+ = \{N, S, R, B, T, C\}$ => not a super key
- {N, A, R}. $\{N, A, R\}^+ = \{N, R, A, B, C, T\}$ => not a super key
- {S, A, R}. $\{S, A, R\}^+ = \{S, A, R, B, T\}$ => not a super key

No proper subsets of {N,S,A,R} are super keys => It is candidate key.
Moreover, there is no function dependencies in F that determine any key inside {N,S,A,R}. Therefore, no key could possibly replace a key in the candidate key {N,S,A,R}; hence it is the only candidate key.

3.3 Derive a canonical cover for F in a systematic manner.

Let G be the minimal/canonical cover of F (the set of function dependencies)
G = {AB->T, A->B, R->C, NS->BT}

3.3.1 Make every function dependency X in G determine only 1 attribute:

The only dependency needed to be splitted is: NS -> BT

- {NS->BT} => NS->B AND NS->T

  G = {AB->T, A->B, R->C, NS->B, NS->T}

3.4 Minimize the left hand side X of every FD

- {A->B, AB->T} => A->T AND B->T

=> G = {A->T, B->T, A->B, R->C, NS->B, NS->T}

3.5 Remove redundant FD, if any

G = {A->T, B->T, A->B, R->C, NS->B, NS->T}

- A->T. $A^+ = \{A, B\}$. There're no T in the closure => the FD is not redundant.
- B->T. $B^+ = \{B\}$. There're no T in the closure => the FD is not redundant.
- A->B. $A^+ = \{A, B\}$. There is B in the closure => the FD is redundant => Remove this FD
- R->C. $R^+ = \{R\}$. There're no C in the closure => the FD is not redundant.
- NS->B. $\{N, S\}^+ = \{N, S, T\}$. There're no B in the closure => the FD is not redundant.
- NS->T. $\{N, S\}^+ = \{N, S, B, T\}$. There is T in the closure => the FD is redundant => Remove this FD

The canonical cover of F is G = {A->T, B->T, R->C, NS->B}

3.6 Does a set of FDs have a unique canonical cover? Why?

The answer is no. A set of FD's might have more than one unique canonical cover. Depending on the order of simplifying, the canonical cover might be different each time.