Bilkent University

Department of Computer Engineering

CS - 353 Database Systems

Homework 4

Functional Dependencies

Name : Osman Burak İntişah
Id : 2160243
Section: 3
Date : 13/04/2020

Question 1)

Which of the following dependencies may hold in relation R?

a) and g) and h)

Dependencies cannot hold - which tuples cause violation?

b) does not hold → tuple 2 and 4 & tuple 1 and 3 cause the violation

c) does not hold → tuple 1 and 2 cause the violation

d) does not hold → tuple 2 and 4 & tuple 1 and 3 cause the violation

e) does not hold → tuple 1 and 2 & tuple 3 and 4 cause the violation

f) does not hold → tuple 3 and 4 cause the violation

i) does not hold → tuple 3 and 4 cause the violation

Question 2)

- **a)** All candidate keys in P { A , B }
- **b)** All candidate keys in Q { BC, CD }

Question 3)

Relation: S(A, B, C, D, E, F, G)

Functional Dependencies: F = {BCD → A, BC → E, A → F, F → G, C→ D, A→ G}

First show as standard form (skip this since all right hand side is one attribute)

First canonical cover is computed: {BC → EA, A → F, F → G, C→ D}

Decompose:

R1 = (B,C,E,A)

R2 = (A,F)

R3 = (F,G)

R4 = (C,D)

These decomposition is dependency preserving since all of the dependencies can still be obtained. And also it is lossless-join since it holds each 3 conditions that I provided below.

Condition 1 = Union of both the sub relations must contain all the attributes that are present in the original relation R.

Condition 2 = Intersection of both the sub relations must not be null.

Condition 3 = Intersection of both the sub relations must be a super key of either R1 or R2 or both

Condition 1 - We can obtain all S(A, B, C, D, E, F, G) from this decomposition.

Condition 2- Intersections are not null.

R1 = { B,C,E,A } ∩ R2 = { A,F } = A

R2 = { A,F } ∩ R3 = { F,G } = F

R1 = { B,C,E,A } ∩ R4 = { C,D } = C

Condition 3 - All of these are super key in one of the decomposition.

A super key in R2

F super key in R3

C super key in R4.

Therefore, it is also lossless join.

Question 4)

Relation: Q(A,B,C,D,E)

Functional Dependencies: AB → E and D→ C

a) superkeys = { ABD, ABCD,ABDE,ABCDE }

   candidate keys = { ABD }

b) Decomposition into BCNF

First checking all functional dependencies A → B in F+, check if A is a superkey

Since ABD is the candidate key, both dependencies does not hold.

Then, decompose.

Q(A,B,C,D,E)

F = { AB → E ,D→ C}

Candidate keys = {ABD}

BCNF? = NO. AB → E violates.

Q1 = (A,B,E)

F1 = { AB→ E}

Candidate Keys: {AB}

BCNF = True

Q2 = (A,B,C,D)

F2 = { D → C }

Candidate Keys: {D}

BCNF? = NO. D→ C violates

Q21 = (A,B,D)

F21 = none

BCNF = True

Q22 = (C,D)

F22 = {D → C}

BCNF = True

Finally;

Q1 = (A,B,E) & **Q21 = (A,B,D)** & **Q22 = (C,D)**

Question 5)

Condition 1 = Union of both the sub relations must contain all the attributes that are present in the original relation R.

Condition 2 = Intersection of both the sub relations must not be null.

Condition 3 = Intersection of both the sub relations must be a super key of either R1 or R2 or both.

a) It is **lossy**. To find this,

First find these compositions are dependency preserving or not.

S1(B, C, D) = F1 = { BD → C}

S2(A, B, D) = F2 = { BD → A}

S3(A, E) = F3 = None

This decomposition is not dependency preserving. So to check lossless-join we will use the F1 union F2 union F3 = {BD → C,BD → A}

Then we need to check whether or not we obtain the conditions that I have provided above.

For condition 1) S1(B, C, D) ∪ S2(A, B, D) ∪ S3(A, E) = S (A,B,C,D,E)

Thus condition 1 satisfies.

For condition 2) S1(B, C, D) ∩ S2(A, B, D) = B, D

S2(A, B, D) ∩ S3(A, E) = A

Clearly, intersections of the sub relations are not all null. We can natural join S1 and S2 then S2 with S3.

For condition 3),

With using B we can determine all the attributes of sub relation S1(B, C, D).

Thus, it is a super key of the sub relation S1(B, C, D).

A is not a key for both S2 and S3.

So, condition 3 fails. Thus, we conclude that the decomposition is **lossy**.

b) It is **lossy.** To find this,

First we need to check whether these compositions are dependency preserving or not.

S1(A, B, C) = F1 = { A→ B,A→ C, B→ C }

S2(B, C, D) = F2 = { B→ C , CD → B}

This decomposition is not dependency preserving. So to check lossless-join we will use the F1 union F2 = F = { A→ B, A→ C, B→ C }

For condition 1) S1(A, B, C) ∪ S2(B, C, D) = S (A,B,C,D)

Thus condition 1 satisfies.

For condition 2) S1(A, B, C) ∩ S2(B, C, D) = B, C

Clearly, intersections of the sub relations are not null. We can natural join S1 and S2 according to B and C.

For condition 3),

With using B and C we can not determine attribute ‘A’ of sub relation S1(A, B, C).

Thus, it is not a super key of the sub relation S1(A, B, C).

With using B and C we can not determine attribute ‘D’ of sub relation S2(B, C, D).

Thus, it is not a super key of the sub relation S2(B, C, D).

So, condition 3 fails. Thus, we conclude that the decomposition is **lossy**.

c) It is **lossless**. To find this,

First we need to check whether these compositions are dependency preserving or not.

S1(A, B, D) = F1 = { A→ B,A→ D}

S2(B, C)= F2 = { B→ C }

This decomposition is not dependency preserving. So to check lossless-join we will use the

F1 union F2 = F = { A→ B,A→ D, B→ C }

For condition 1) S1(A, B, D) ∪ S2(B, C) = S (A,B,C,D)

Thus condition 1 satisfies.

For condition 2) S1(A, B, C) ∩ S2(B, C) = B

Clearly, intersections of the sub relations are not null. We can natural join S1 and S2 according to B.

For condition 3),

With using B we can not determine attribute ‘A’ of sub relation S1(A, B, C).

Thus, it is not a super key of the sub relation S1(A, B, C).

With using B we can determine all the attributes of sub relation S2(B, C).

Thus, it is a super key of the sub relation S2(B, C).

So, condition 3 satisfies. Thus, we conclude that the decomposition is **lossless**.

Question 6)

S(A, B, C, D)

F = {A → BCD, B→ C, CD→ A }.

S1(A, B, C) and S2(B, C, D)

It is **NOT** dependency preserving. To achieve this, first we need to write the normal forms of the dependencies.

F = { A→ B, A→ C, A→ D, B→ C, CD→ A, CD → B}

Then, we need to find which dependencies will be belonging to which decomposition.

S1(A, B, C) → { A→ B,A→ C, B→ C } = F1

S2(B, C, D) → { B→ C , CD → B} = F2

We find the F1 union F2. It is { A→ B,A→ C, B→ C, CD → B}.

Finally we compare result of the union and F. See that these two are not equal to each other. (CD → A) is missing. This implies that this decomposition is **not dependency preserving**.