HW6
Functional Dependencies and Normalization

Ivan Martinovic

Due date: March 12, 2021 @12pm
Submission: Upload your typed PDF document to Canvas
Maximum: 100pts
Note: This homework is to be done by each student individually. No help besides the textbook should be taken. *Copying any answers or part of answers from other sources (including your colleagues) will earn you a grade of zero.* All homework is to be typed, no handwritten diagrams or pictures of diagrams are accepted. Handwritten submissions will result in a 20% grade penalty.

Problem 1 (35 Points)

For the relational schema given below and the corresponding functional dependencies (FDs)
R(A, B, C, D, E) S = { AB → E, B → C, B → D, CE → A },
answer the following questions:

a. (10 Points) find all candidate keys of the relation R through an exhaustive set of attribute closures. Note when an attribute set closure is trivial.
$\{A\}^+ = \{A\}$ (trivial)
$\{B\}^+ = \{B\ C\ D\}$
$\{C\}^+ = \{C\}$ (trivial)
$\{D\}^+ = \{D\}$ (trivial)
$\{E\}^+ = \{E\}$ (trivial)

$\{AB\}^+ = \{A\ B\ C\ D\ E\}$ (candidate key)
$\{AC\}^+ = \{A\ C\}$ (trivial)
$\{AD\}^+ = \{A\ D\}$ (trivial)
$\{AE\}^+ = \{A\ E\}$ (trivial)
$\{BC\}^+ = \{B\ C\ D\}$
$\{BD\}^+ = \{B\ C\ D\}$
$\{BE\}^+ = \{A\ B\ C\ D\ E\}$ (candidate key)
$\{CD\}^+ = \{C\ D\}$ (trivial)
$\{CE\}^+ = \{A\ C\ E\}$
$\{DE\}^+ = \{D\ E\}$ (trivial)

$\{ABC\}^+ = \{A\ B\ C\ D\ E\}$ (super key)
$\{ABD\}^+ = \{A\ B\ C\ D\ E\}$ (super key)
$\{ABE\}^+ = \{A\ B\ C\ D\ E\}$ {super key)
$\{ACD\}^+ = \{A\ C\ D\}$ (trivial)
$\{ACE\}^+ = \{A\ C\ E\}$ (trivial)
$\{ADE\}^+ = \{A\ D\ E\}$ (trivial)
$\{BCD\}^+ = \{B\ C\ D\}$ (trivial)
$\{BCE\}^+ = \{A\ B\ C\ D\ E\}$ (super key)
$\{BDE\}^+ = \{A\ B\ C\ D\ E\}$ (super key)
$\{CDE\}^+ = \{A\ C\ D\ E\}$

$\{ABCD\}^+ = \{A\ B\ C\ D\ E\}$ (super key)
$\{ABCE\}^+ = \{A\ B\ C\ D\ E\}$ (super key)

$\{ABDE\}^+ = \{A\ B\ C\ D\ E\}$ {super key)
$\{ACDE\}^+ = \{A\ C\ D\ E\}$ (trivial)
$\{BCDE\}^+ = \{A\ B\ C\ D\ E\}$ (super key)

$\{ABCDE\}^+ = \{A\ B\ C\ D\ E\}$ (trivial) (super key)

b. (5 Points) Assume that S is a minimal basis for R. List the dependencies that violate **3NF**, if any.
S={AB → E, B → C, B → D, CE → A}, Candidate Keys <A,B> <B,E>

B -> C violates 3NF because
1) it is not trivial
2) B is not a super key
3) C is not part of a candidate key

B -> D violates 3NF because
1) it is not trivial
2) B is not a super key
3) D is not part of a candidate key

c. (5 points) If R is NOT in **3NF**, decompose it into multiple relations that are in **3NF**.

S={AB → E, B → C, B → D, CE → A}
R (<u>A</u>, <u>B</u>, C, D, E)

Use B->C to decompose R
R1(<u>B</u>, C), Foreign key <B> in R2
R2(<u>A</u>, <u>B</u>, D, E)

Use B->D to decompose R2
R3(<u>B</u>, D) Foreign Key <B> in R4
R4(<u>A</u>, <u>B</u>, E)

Note: R1 Foreign key <B> is now in R4

Combine R1 and R2
R5(<u>B</u>, C, D) Foreign key <B> in R4

If we look at the FD's, we can see that we are missing CE->A. We create a new relation R6
R6(<u>C</u>, <u>E</u>, A)

Finally:
R4(<u>A</u>, <u>B</u>, E)
R5(<u>B</u>, C, D) Foreign key <B> in R4
R6(<u>C</u>, <u>E</u>, A) Foreign Key<C> in R5 and <E> in R4

d. (5 points) List the dependencies, in the order given in S, that violate **BCNF**.
B -> C (not trivial, B not super key)
B -> D (not trivial, B not super key)
CE -> A (not trivial, CE not super key)

e. (10 points) If R is not in **BCNF**, provide decomposition into multiple relations where each one is in BCNF. For each decomposition step, use the first FD violation following the FD order given in S. For example, if AB → E and B → C are in BCNF but the other two FDs are in violation, then you would use B → D for the decomposition. Make sure to specify which FD is used to make the decomposition.

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

Step 1 (use B->C to decompose R):
R1(B, C)
R2(A, B, D, E)
Step 2 (use B->D to decompose R2):
R3(B, D)
R4(A, B, E)

FInally we have 3 relations:
R1(B, C) Foreign key B in R4
R3(B, D) Foreign key B in R4
R4(A, B, E)

We may merge relations R1 and R3:
R5(B, C, D), foreign key B in R4
R4(A, B, E)

Problem 2 (45 Points)

For the relational schema given below and its corresponding functional dependencies (FDs)
R(A, B, C, D, E) S = { B → A, B → E, CE → D, D → B }
answer the following questions:

a. (5 Points) find all candidate keys of the relation through an exhaustive set of attribute closures. Note when an attribute set closure is trivial.

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

$\{AB\}^+ = \{A\ B\ E\}$
$\{AC\}^+ = \{A\ C\}$ (trivial)
$\{AD\}^+ = \{A\ B\ D\ E\}$

$\{AE\}^+ = \{A\ E\}$ (trivial)
$\{BC\}^+ = \{A\ B\ C\ D\ E\}$ (candidate key)
$\{BD\}^+ = \{A\ B\ D\ E\}$
$\{BE\}^+ = \{A\ B\ E\}$
$\{CD\}^+ = \{A\ B\ C\ D\ E\}$ (candidate key)
$\{CE\}^+ = \{A\ B\ C\ D\ E\}$ (candidate key)
$\{DE\}^+ = \{A\ B\ D\ E\}$

$\{ABC\}^+ = \{A\ B\ C\ D\ E\}$ (super key)
$\{ABD\}^+ = \{A\ B\ D\ E\}$
$\{ABE\}^+ = \{A\ B\ E\}$ {trivial)
$\{ACD\}^+ = \{A\ B\ C\ D\ E\}$ (super key)
$\{ACE\}^+ = \{A\ B\ C\ D\ E\}$ (super key)
$\{ADE\}^+ = \{A\ B\ D\ E\}$
$\{BCD\}^+ = \{A\ B\ C\ D\ E\}$ (super key)
$\{BCE\}^+ = \{A\ B\ C\ D\ E\}$ (super key)
$\{BDE\}^+ = \{A\ B\ D\ E\}$
$\{CDE\}^+ = \{A\ B\ C\ D\ E\}$ (super key)

$\{ABCD\}^+ = \{A\ B\ C\ D\ E\}$ (super key)
$\{ABCE\}^+ = \{A\ B\ C\ D\ E\}$ (super key)
$\{ABDE\}^+ = \{A\ B\ D\ E\}$ {trivial)
$\{ACDE\}^+ = \{A\ B\ C\ D\ E\}$ (super key)
$\{BCDE\}^+ = \{A\ B\ C\ D\ E\}$ (super key)

$\{ABCDE\}^+ = \{A\ B\ C\ D\ E\}$ (trivial) (super key)

b. (5 Points) Given the keys you defined in step 1, find the FDs (from the given ones) that violate BCNF.

S = { B → A, B → E, CE → D, D → B }

B->A (not trivial, B not super key)
B->E (not trivial, B not super key)
D->B (not trivial, D not super key)

c. (15 Points) Decompose the relations to satisfy BCNF. Specify which FD is used to make the decomposition. If there is multi-step decomposition, then indicate each step along with which FD is used for the decomposition.

Candidate Keys <B,C>, <C,D> and <C,E>
R(A,B,C,D,E)
Step 1: Use B-> A to decompose R:
R1(B, A) foreign key <B> in R2
R2(B, C, D, E)

Step 2: Use B->E to decompose R2:
R3(B, E) foreign key <B> in R4

R4(B, C, D)

Note: R1(B, A) foreign key <B> in R4

Combine R1 and R3:
R5(B, A, E) foreign key <B> in R4

Use D->B to decompose R4:
R6(D, B) Foreign key <D> in R7
R7(C, D)

Note: R5(B, A, E) foreign key <B> in R6

Finally:
R7(C, D)
R6(D, B) Foreign key <D> in R7
R5(B, A, E) foreign key <B> in R6

d. (20 Points) If the FDs are not in 3NF, calculate a minimal basis for the FDs and decompose the relations to satisfy 3NF.

S = { B → A, B → E, CE → D, D → B }
Candidate Keys <B,C>, <C,D> and <C,E>
Relation R(A,B,C,D,E)

- B->A violates 3NF (it is not trivial, B not a super key, A is not part of a candidate key)

Minimal basis:
B -> AE
CE -> D
D->B

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

Step 1) Use B -> A to decompose R:
R1(B, A) foreign key <B> in R2
R2(B, C, D, E)

Step 2) Use D -> B to decompose R2:
R3(D,B) Foreign Key <D> in R4
R4(C, E, D)

Note: R1(B, A) foreign key <B> in R3

Now we have:
R4(C, E, D)
R3(D,B) Foreign Key <D> in R4
R1(B, A) foreign key <B> in R2

If we look at the FD's, we see that we're missing B->E. We can just add it to R1:
R1(B, A, E) foreign key <B> in R2

Finally:
R4(C, E, D)
R3(D,B) Foreign Key <D> in R4
R1(B, A, E) foreign key <B> in R2

Problem 3 (20 Points)
Answer the questions using the table below:

Artist	Gallery	Address	Artwork
Haring	Miami Beach	818 Lincoln Rd	Radiant Baby
Britto	Miami Beach	818 Lincoln Rd	Garden of Eden
Warhol	Chicago	100 Michigan Ave	Campbell Soup Cans
Warhol	Boston	100 Michigan Ave	Marilyn Monroe

1. (10 Points) Indicate whether each of the following decompositions is Lossy or Lossless and state why?
    a. Artist and Artwork are in one relation. Gallery, Address, and Artwork are in the other relation.
    This is a **lossless** decomposition -> a natural join of the 2 relations would produce the original relation.
    Another way to look at this is the following:
    The common attributes of the two relations is just the Artwork.
    Since the artwork is a key for both relations (i.e. Artwork -> Artist and Artwork -> Gallery,Address), the decomposition is lossless.

    b. Gallery, Address, and Artwork are in one relation. Artist and Gallery are in one relation.
    This is a **lossy** decomposition -> a natural join would create extra records for the Miami Beach Gallery
    Another way to look at this:
    The common attributes of the two relations is just the Gallery.
    Since Gallery is not a key in either of them (i.e. Gallery –x> Artist and Gallery –x> Arist,Address), the decomposition is lossy.

2. (10 Points) Identify and list the set of functional dependencies from the data in the table above. Then, specify which of the following decompositions preserve those dependencies, and state why.

    Artist -> Address
    Gallery -> Address
    Artwork -> Address
    Artwork -> Artist
    Artwork -> Gallery
    Artist,Gallery -> Artwork

    Note: it may help to draw the relations to visualize what may have changed through decomposition.

a. Gallery, Address, and Artist are in one relation. Artwork and Artist are in the other relation.

Gallery	Address	Artist
Miami Beach	818 Lincoln Rd	Harring
Miami Beach	818 Lincoln Rd	Britto
Chicago	100 Michig Ave	Warhol
Boston	100 Michig Ave	Warhol

Artwork	Artist
Radiant Baby	Harring
Garden of Eden	Britto
Campbell Soup Cans	Warhol
Marilyin Monroe	Warhol

For the two tables individually we can see the following:

Preserves: Gallery -> Address
Artwork -> Artist
Artist -> Address

Doesn't preserve: Artwork -> Gallery
Artist,Gallery -> Artwork
Artwork -> Address

A natural Join of the Two Tables Would Looks as Follows

Gallery	Address	Artist	Artwork
Miami Beach	818 Lincoln Rd	Harring	Radiant Baby
Miami Beach	818 Lincoln Rd	Britto	Garden of Eden
Chicago	100 Michig Ave	Warhol	Campbell Soup Cans
Chicago	100 Michig Ave	Warhol	Marilyin Monroe
Boston	100 Michig Ave	Warhol	Marilyin Monroe
Boston	100 Michig Ave	Warhol	Campbell Soup Cans

We can see from the natural join the following:

Preserves: Gallery -> Address
Artwork -> Artist
Artist -> Address
Artwork -> Address

Doesn't preserve: Artwork -> Gallery
Artist,Gallery -> Artwork

Since we can only join the tables on the Artist, we have no way of assigning a unique Gallery or Address to an Artwork. We've also lost the ability to imply the Artwork using the Artist and the Gallery.

b. Gallery, Artist are in one relation. Artwork, Artist, and Address are in the other relation.

Gallery	Artist
Miami Beach	Harring
Miami Beach	Britto
Chicago	Warhol
Boston	Warhol

Artwork	Artist	Address
Radiant Baby	Harring	818 Lincoln Rd
Garden of Eden	Britto	818 Lincoln Rd
Campbell Soup Cans	Warhol	100 Michig Ave
Marilyin Monroe	Warhol	100 Michig Ave

From the two tables individually we can see the following:

Preserves: Artwork -> Artist
Artwork -> Address
Artist -> Address

Doesn't Preserve: Gallery -> Address
Artwork -> Gallery
Artist,Gallery -> Artwork

A natural join of the two tables would look as follows:

Gallery	Address	Artist	Artwork
Miami Beach	818 Lincoln Rd	Harring	Radiant Baby
Miami Beach	818 Lincoln Rd	Britto	Garden of Eden
Chicago	100 Michig Ave	Warhol	Campbell Soup Cans
Chicago	100 Michig Ave	Warhol	Marilyin Monroe
Boston	100 Michig Ave	Warhol	Marilyin Monroe
Boston	100 Michig Ave	Warhol	Campbell Soup Cans

From the natural join we can see the following:

Preserves: Gallery -> Address
Artwork -> Artist
Artist -> Address
Artwork -> Address

Doesn't preserve: Artwork -> Gallery
Artist,Gallery -> Artwork

Since we can only join the tables on the Artist, we have no way of assigning a unique Gallery or Address to an Artwork. We've also lost the ability to imply the Artwork using the Artist and the Gallery.

Deliverables:
Students should submit a .pdf file containing their appropriately numbered responses to questions. The file MUST be a .pdf file (other file types need to be converted to .pdf for submission).

Submission:
Submit via Canvas. Late submissions will not be accepted.

Submission notes:

1) Include your name on the sheet
2) Be sure that your submission is in .pdf format.
3) Handwritten solutions or pictures of handwritten solutions will not be accepted.