HW5

1. Suppose we have relation R (A, B, C, D, E), which we have decomposed into relations with sets of attributes S1 = {A, B, C}, S2 = {B, C, D}, and S3 = {A, C, E}. Given the set of FDs are {A→D, B→D, CD→E}.
   1. Use the chase test to tell whether the decomposition of R is lossless.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E |
| R1 | a | b | c | d1 | e1 |
| R2 | a2 | b | c | d | e2 |
| R3 | a | b3 | c | d3 | e |

A→D

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E |
| R1 | a | b | c | d1 | e1 |
| R2 | a2 | b | c | d | e2 |
| R3 | a | b3 | c | ~~d3~~  d1 | e |

B→D

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E |
| R1 | a | b | c | ~~d1~~ d | e1 |
| R2 | a2 | b | c | d | e2 |
| R3 | a | b3 | c | ~~d1~~ d | e |

CD→E

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E |
| R1 | a | b | c | d | ~~e1~~ e |
| R2 | a2 | b | c | d | ~~e2~~ e |
| R3 | a | b3 | c | d | e |

The decomposition is lossless.

* 1. Is the decomposition dependency preserving?

First we will calculate the projection of FDs onto the decomposed tables.

For S1=ABC:

(A)+=A~~D~~

(B)+=B~~D~~

(C)+=C

(AB)+=AB~~D~~

(AC)+=AC~~DE~~

(BC)+=BC~~DE~~

So projected FDs for S1 is an empty set, F1={}

For S2=BCD:

(B)+=BD

(C)+=C

(D)+=D

(BC)+=BCD~~E~~

(BD)+=BD

(CD)+=CD~~E~~

So projected FDs for S2 are, F2={B→D, BC→D}

For S3=ACE:

(A)+=A~~D~~

(C)+=C

(E)+=E

(AC)+=AC~~D~~E

(AE)+=AE~~D~~

(CE)+=CE

So projected FDs for S3 are, F3={AC→E}.

So all the FDs for S1, S2, S3 are { B→D, BC→D, AC→E }.

Now we will find if the original FDs {A→D, B→D, CD→E} are preserved.

First we will check if A→D is preserved by finding the (A)+ by using the dependencies of the projected FDs.

(A)+=A

So A cannot functionally determine D from the projected FDs, so the dependency A→D is lost.

Then B→D is preserved. This FD exists in the projected FDs as well.

Finally, we will check if CD→E is preserved.

(CD)+=CD

Again CD cannot functionally determine E from the projected FDs, so the dependency CD→E is lost.

All in all, FDs A→D and CD→E are lost. So the decomposition is not dependency preserving.

* 1. Is the decomposition in BCNF? If not, indicate all BCNF violations and try to decompose R into BCNF.

In order to find if the decomposed tables are in BCNF we need to find if the FDs that hold for every decomposed table violate the BCNF. (We have already calculated the projected FDs on the decomposed tables from the previous b) sub question, so we will just use them).

R1(A, B, C) is in BCNF. It doesn’t have any projected FDs on R1.

R2(B, C, D) is not in BCNF because B→D violates the BCNF. B is not a superkey for the relation R2.

R3(A, C, E) is in BCNF because the only dependency that holds here is AC→E and AC is a key.

The relation R itself is not in BCNF as well, since none of the left hand sides of the set of FDs is a superkey. The violations are A→D, B→D and CD→E.

We could use any of the violating FDs in order to decompose our relation. We will use the A→D FD, but similarly you can decompose the relation based on the other FDs.

If we decompose based on A→D:

R1={AD}, F1={A→D}

R2={ABCE}, F2={AC→E,BC→E, ABC→E}

The projected FDs on R1={AD} are:

(A)+=AD

(D)+=D

So F1={A→D}

The projected FDs on R2={ABCE} are:

(A)+=A~~D~~

(B)+=B~~D~~

(C)+=C

(E)+=E

(AB)+=AB~~D~~

(AC)+=AC~~D~~E

(AE)+=AE~~D~~

(BC)+=BC~~D~~E

(BE)+=BE~~D~~

(CE)+=CE

(ABC)+=ABC~~D~~E

(ABE)+=ABE~~D~~

(BCE)+=BCE~~D~~

So F2={AC→E, BC→E, ABC→E}.

R1 is BCNF since A is a key but R2 is not. Both AC→E and BC→E violate BCNF for R2. So we decompose R2 further by using AC→E (You could either use BC→E):

R21={ACE}

R22={ABC}

The projected FDs on R21={ACE} based on F2 are:

(A)+=A

(C)+=C

(E)+=E

(AC)+=ACE

(AE)+=AE

(CE)+=CE

So F21={AC→E}

The projected FDs on R22={ABC} based on F2 are:

(A)+=A

(B)+=B

(C)+=C

(AB)+=AB

(AC)+=AC~~E~~

(BC)+=BC~~E~~

So F22={}

Now both R21 and R22 are in BCNF. AC is a key for R21 and R22 doesn’t have any non-trivial FDs.

So final decomposition is R1{AD}, R21{ACE} and R22{ABC}.

**OTHER WAYS**

If we decompose based on B→D:

R1={BD}, F1={B→D}

R2={ABCE}, F2={AC→E,BC→E, ABC→E}

R1 is BCNF but R2 is not. AC→E and BC→E violate BCNF for R2. So we decompose further by using AC→E:

R21={ACE}, F21={AC->E}

R22={ABC}, F22={}

So final decomposition is R1={BD}, R21={ACE}, R22={ABC}.

If we decompose based on CD→E:

R1={CDE}, F1={ CD→E}

R2={ABCD}, F2={A→D, B→D, AB→D, AC→D, BC→D, ABC→D}

R1 is BCNF but R2 is not. A→D, B→D, AB→D, AC→D and BC→D violate BCNF for R2. So we can decompose R2 further by using A→D:

R21={AD}, F21={A→D}

R22={ABC}, F22={}

So the final decomposition is R1={CDE}, R21={AD}, R22={ABC}.

1. Let R(A,B,C,D,E), the set of FDs {A→BC, C→D} and the decomposition: R1(A,B,C) and R2(A,D,E).
   1. Is the decomposition lossless join?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E |
| R1 | a | b | c | d1 | e1 |
| R2 | a | b2 | c2 | d | e |

A→BC

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E |
| R1 | a | b | c | d1 | e1 |
| R2 | a | ~~b2~~ b | ~~c2~~ c | d | e |

C→D

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E |
| R1 | a | b | c | ~~d1~~ d | e1 |
| R2 | a | b | c | d | e |

The decomposition is lossless.

* 1. Is the decomposition dependency preserving?

First we will calculate the projection of FDs onto the decomposed tables.

For R1=ABC:

(A)+=ABC

(B)+=B

(C)+=C~~D~~

(BC)+=BC~~D~~

So projected FDs for R1 are F1={ A→BC }

For S2=ADE:

(A)+=A~~BC~~D

(D)+=D

(E)+=E

(AD)+=AD

(AE)+=A~~BC~~DE

(DE)+=DE

So projected FDs for R2 are, F2={A→D, AE→D }.

So all the FDs for R1, R2 are { A→BC, A→D, AE→D }.

Now we will find if the original FDs { A→BC, C→D} are preserved.

First we will check if A→ BC is preserved. It is preserved as it is, so need to find the closure of the A attribute.

Then we will check if C→D by finding the (C)+ by using the dependencies of the projected FDs.

(C)+=C

So C cannot functionally determine D from the projected FDs, so the dependency C→D is lost.

So the decomposition is not dependency preserving.

* 1. Is the decomposition BCNF?

R1 is in BCNF because F1 = {A→BC} and A is a key.

R2 is not in BCNF because of A→D. A is not a superkey.

* 1. Can you give a BCNF, dependency preserving decomposition?

The candidate key of R is AE. None of the left hand sides of the set of FDs is a superkey. Both A→BC and C→D violate BCNF. When we start decompose R by using A→BC, we get R1={ABC} and R2={ADE}. However, this decomposition is not dependency preserving as shown in 2). So we try to decompose R using C→D:

R1={CD}, F1={C→D}

R2={ABCE}, F2={A→BC, AB→C, AC→B, ABE→C, ACE→B}

R1 is BCNF but R2 is not. A→BC, AB→C and AC→B violates BCNF for R2. So we decompose further by using A→BC:

R21={ABC}, F21={A→BC, AB→C, AC→B}

R22={AE}, F22={}

So the final dependency preserving BCNF decomposition is R1(CD), R21(ABC) and R22(AE).

1. Let R(A, B, C, D, E) and F={A→B, BC→E, and ED →A}. Decompose R in BCNF.

(A)+ = AB

(BC)+ = BCE

(ED)+ = ABDE

None of the left hand sides of the FDs is a superkey of R. All of them are BCNF violations.

If we decompose based on A→B:

R1={AB}

R2={ACDE}

For R1={AB}:

(A)+=AB

(B)+=B

So projected FDs for R1 are F1={ A→B }

For R2={ACDE}:

(A)+=A~~B~~

(C)+=C

(D)+=D

(E)+=E

(AC)+=AC~~B~~E

(AD)+=AD~~B~~

(AE)+=AE~~B~~

(CD)+=CD

(CE)+=CE

(DE)+=DEA~~B~~

(ACD)+=ACD~~B~~E

(ACE)+=ACE~~B~~

(ADE)+=ADE~~B~~

(CDE)+=CDEA~~B~~

So projected FDs for R2 are F2={ AC→E, ED→A, (ACD→E, CDE→A) }

R1 is in BCNF but R2 is not. AC→E and ED→A violate BCNF for R2. So we decompose further by using ED→A (we could also use AC→E):

R21={ADE}, F21={ED→A}

R22={CDE}, F22={}

The projected FDs for R21={ADE} by using F2 are:

(A)+=A

(D)+=D

(E)+=E

(AD)+=AD

(AE)+=AE

(ED)+=EDA

So projected FDs for R21 are F21={ ED→A}

The projected FDs for R22={CDE} by using F2 are:

(C)+=C

(D)+=D

(E)+=E

(CD)+=CD

(CE)+=CE

(ED)+=ED~~A~~

So projected FDs for R22 are F22={}

Now both R21 and R22 are in BCNF.

**So the final decomposition is R1(AB), R21(ADE) and R22(CDE).**

OTHER WAYS

Similarly if we decompose based on the other initial FDs your results will be:

If we decompose based on BC→E:

R1={BCE}, F1={BC→E}

R2={ABCD}, F2={A→B, AC→B, AD→B, ACD→B, BCD→A}

R1 is in BCNF but R2 is not. A→B, AC→B and AD→B violate BCNF for R2. So we decompose further by using A→B:

R21={AB}, F21={A→B}

R22={ACD}, F22={}

So the final decomposition is R1(BCE), R21(AB) and R22(ACD).

If we decompose based on ED →A:

R1={ADE}, F1={ED →A}

R2={BCDE}, F2={BC→E, DE→B, BCD→E, CDE→B}

R1 is in BCNF but R2 is not. BC→E and DE→B violate BCNF for R2. So we decompose further by using BC→E:

R21={BCE}, F21={ BC→E}

R22={BCD}, F22={}

So the final decomposition is R1(ADE), R21(BCE) and R22(BCD).

1. Assume the relation schema R(A,B,C,D,E,F,G,H) and the following set of functional dependencies: {AB→E, C→D, D→E, FG→A}. Decompose R in BCNF. Is your decomposition dependency preserving?
   1. AB→E causes a violation in R because (AB)+=ABE, so AB is not a superkey. Break in R1(ABE) and R2(ABCDFGH) with obvious FDs be:

F1 = {AB→E}

F2 = {C→D, FG→A}

* 1. C→D causes a violation in R2 since C+=CD. Break R2 in R21(CD) and R22(ABCFGH) with

F21 = {C→D}

F22 = {FG→A}

* 1. R21 is in BCNF but R22 is not because (FG)+=FGA, so FG is not a superkey. Break R22 in R221(FGA) and R222(BCFGH) with

F221 = {FG→A}

F222 = {}

Final decomposition: R1(ABE), R21(CD), R221(FGA) and R222(BCFGH).

(F1 U F21 U F221 U F222) = {AB→E, C→D, FG→A}

Now we will find if the original FDs { AB→E, C→D, D→E, FG→A} are preserved. Obviously, AB→E, C→D, FG→A are preserved. We will check if D→E is preserved by finding the (D)+ by using the dependencies of the final decomposition FDs.

(D)+=D so D→E is not preserved in this decomposition. Thus, the decomposition is not dependency preserving.