Logan Murray

March 4, 2019

CSC 355 – 502

Assignment 5 - Solution

1.

a.

{a} = {a, b, c, d}

{b} = {b}

{c} = {c, b}

{d} = {d, b}

{a, b} = {a, b, c, d}

{a, c} = {a, b, c, d}

{a, d} = {a, b, c, d}

{b, c} = {c, b}

{b, d} = {b, d}

{c, d} = {b, c, d}

{a, b, c} = {a, b, c, d}

{b, c, d} = {b, c, d}

{c, d, a} = {a, b, c, d}

{d, a, b} = {a, b, c, d}

b.

{a} is a superkey of R because, {a🡪d} and {d🡪b} therefore {a, b🡪 c}.

The pair {a, b} is a superkey of R because {a, b🡪c} and {a🡪d}.

c.

{a, b} is a candidate key pair because just {b} by itself cannot determine anything but itself.

2.

a.

The candidate key is {VNumber, DNumber} because together they can identify all attributes but separately they cannot. {VNumber} = {VDate, PNumber, PAge, PZip} while {DNumber} = {DSpec} and together {VNumber, DNumber} = {Diagnosis} which encompasses all attributes.

b.

For this decomposition or any further use of this relation I am using substitutes for clarity and efficiency. So now VNumber = a, VDate = b, PNumber = c, PAge = d, PZip = e, DNumber = f, DSpec = g, Diagnosis = h, which means ENCOUNTER(a, b, c, d, e, f, g, h) and the functional dependencies are as follows.

F = {a🡪b, c, d, e ; c🡪d, e ; f🡪g ; a, f🡪h} where {a, f} is a minimal superkey that determines all attributes.

S = {a, b, c, d, e, f, g, h) // initialization S = {R}

S = {a, b, c, f, g, h} S2 = { c, d, e} // (c🡪d, e) is a functional dependency that violates BCNF, therefore is decomposed into its own relation that contains ‘c’ in both relations as a foreign key.

3.

a.

projection of F on R1(A, B) = {B🡪A} and nothing else

b.

projection of F on R2(B, C, D) = {B🡪 C ; D🡪 B}

c.

No, it does preserve this property because as you can see above the union of both projections would be as such {B🡪A,} **U** {B🡪 C ; D🡪 B} = { B 🡪 A ; B 🡪 C ; D🡪B }. And the original F is = {A🡪D ; B🡪A, C ; D🡪 A, B}. Therefore, the dependencies such as A🡪D ; D 🡪A are lost after decomposition.

4.

a.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Initial State | **A1** | **A2** | **A3** | **A4** |
| **R1** | A1 | A2 | B13 | B14 |
| **R2** | B12 | A2 | A3 | A4 |
| **R3** | A1 | B32 | B33 | A4 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A4🡪A3 | **A1** | **A2** | **A3** | **A4** |
| **R1** | A1 | A2 | B13 | B14 |
| **R2** | B12 | A2 | A3 | A4 |
| **R3** | A1 | B32 | ~~B33~~ >A3 | A4 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A1, A3🡪A2 | **A1** | **A2** | **A3** | **A4** |
| **R1** | A1 | A2 | B13 | B14 |
| **R2** | B12 | A2 | A3 | A4 |
| **R3** | A1 | ~~B32~~ >A2 | ~~B33~~ >A3 | A4 |

\*Next state of the matrix below\*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A2🡪A1 | **A1** | **A2** | **A3** | **A4** |
| **R1** | A1 | A2 | B13 | B14 |
| **R2** | ~~B12~~ >A1 | A2 | A3 | A4 |
| **R3** | A1 | ~~B32~~ >A2 | ~~B33~~ >A3 | A4 |

b.

Based on the final state of the matrix S, this decomposition does preserve lossless join. R3 was able to be filled completely.

5.

W🡪U stays because we don’t need to pair U, W to get X we just need Y.

U, W🡪X U, W🡪X

W🡪U, Y W🡪U

OR

W, Z🡪Y W🡪Y

W, Z🡪Y stays because we can’t derive Z from any other functional dependency.

Y🡪X W, Z🡪Y

Y🡪X

Y🡪X stays because we can’t derive X from any other functional dependency left over.

6. As before I will use substitutes in place of the values given

STUDENT(StuID = a, SSNum = b, FName = c, LName = d, Major = e, Dept = f, Group = g}

F = {a🡪b, e ; b🡪c, d, a ; a🡪f ; e🡪f ; c🡪g}

a.

candidate keys for F are {StuID} and {SSNum}

b.

a🡪b

a🡪e

b🡪c

b🡪d

b🡪a

~~a🡪f~~  < This gets removed and the rest is a minimal basis of F.

e🡪f

c🡪g

Joining all the independent dependencies together that share the same left hand side results in three relations.

MAJOR(StuID, SSNum, Major, Dept),

STUDENT(StuID, SSNum, FName, LName),

GROUP(FName, Group)