SJSU Students
CS-157A Section: 04
October 21, 2019

Homework 3

1. **Exercise 3.3.2:**

   We have a relation R(A, B, C, D, E) and the set of FD's: {A→B, A→C, A→D}.

   We're going to use the BCNF Decomposition Algorithm recursively on R.
   Let's take the FD: A→B.
   We get relations R1(A,B) and R2(A, C, D, E). R1 is already in BCNF as it has the superkey A→B.

   We're going to use the BCNF Decomposition Algorithm recursively on R2.
   Let's take the FD: A→C.
   We get relations R3(A,C) and R4(A, D, E). R3 is already in BCNF as it has the superkey A→C.

   We're going to use the BCNF Decomposition Algorithm recursively on R4.
   Let's take the FD: A→D.
   We get relations R5(A,D) and R6(A, E). R5 is already in BCNF as it has the superkey A→D. R6 is already in BCNF as two column relations are trivially in BCNF. Therefore, we get R1, R3, R5, R6 from our BCNF decomposition.

   If we take the BCNF violation A→BCD first, we get,
   S1(A, B, C, D) and S2(A, E). S1 is already in BCNF as A is the superkey. S2 is trivially in BCNF.

   Therefore, we don't get the same result with the two decompositions.

2. **Exercise 3.4.1**

   c) We have relation R(A, B, C, D, E) with FD's: {A→D, D→E, B→D}.
   Decomposed relations: R1(A, B, C) , R2(B, C, D) , R3(A, C, E).

Using the Chase Algorithm,

	A	**B**	**C**	**D**	**E**
R1	a	b	c	d_1	e_1
R2	a_2	b	c	d	e_2
R3	a	b_3	c	d_3	e

We take the FD:A→D. So, we get,

	A	**B**	**C**	**D**	**E**
R1	a	b	c	d	e_1
R2	a_2	b	c	d	e_2
R3	a	b_3	c	d	e

Then, we take the FD: D→E. So, we get,

	A	**B**	**C**	**D**	**E**
R1	a	b	c	d	e
R2	a_2	b	c	d	e
R3	a	b_3	c	d	e

Then, we take the FD: B→D

	A	**B**	**C**	**D**	**E**
R1	a	b	c	d	e
R2	a_2	b	c	d	e
R3	a	b_3	c	d	e

Therefore, R does not have the lossless join property.

d) We have relation R(A, B, C, D, E) with FD's: {A→D, CD→E, E→D}.
Decomposed relations: R1(A, B, C) , R2(B, C, D) , R3(A, C, E).
Using the Chase Algorithm,

	A	**B**	**C**	**D**	**E**
R1	a	b	c	d_1	e_1
R2	a_2	b	c	d	e_2
R3	a	b_3	c	d_3	e

We take the FD: A→D. So, we get,

	A	**B**	**C**	**D**	**E**
R1	a	b	c	d	e_1
R2	a_2	b	c	d	e_2
R3	a	b_3	c	d	e

Then, we take the FD: CD→E. So, we get,

	A	**B**	**C**	**D**	**E**
R1	a	b	c	d	e
R2	a_2	b	c	d	e
R3	a	b_3	c	d	e

Then, we take the FD: E→D. So, we get,

	A	**B**	**C**	**D**	**E**
R1	a	b	c	d	e
R2	a_2	b	c	d	e
R3	a	b_3	c	d	e

Therefore, R does not have the lossless join property.

3. **Exercise 3.6.2:**

We have a relation R(n, s, b, cn, cs, cb, as, am) such that:
1. n is the name of the person with Social Security number s.
2. b is n 's birthdate.
3. cn is the name of one of n 's children.
4. cs is cn's Social Security number.
5. cb is cn's birthdate.
6. as is the serial number of one of n 's automobiles.
7. am is the make of the automobile with serial number as.

a) The Functional Dependencies that we would expect to hold in this relation are as follows:

s→ n, b
cs→cn, cb
as→am

The Multivalued Dependencies that we would expect to hold in this relation are as follows:

s↠as, am
s↠cs, cn, cb

b) Decomposition into 4NF:

From the violation s→ n, b, we get, R1(s, n, b) and R2(s cn, cs, cb, as, am). R1 is in 4NF and we recursively decompose R2.

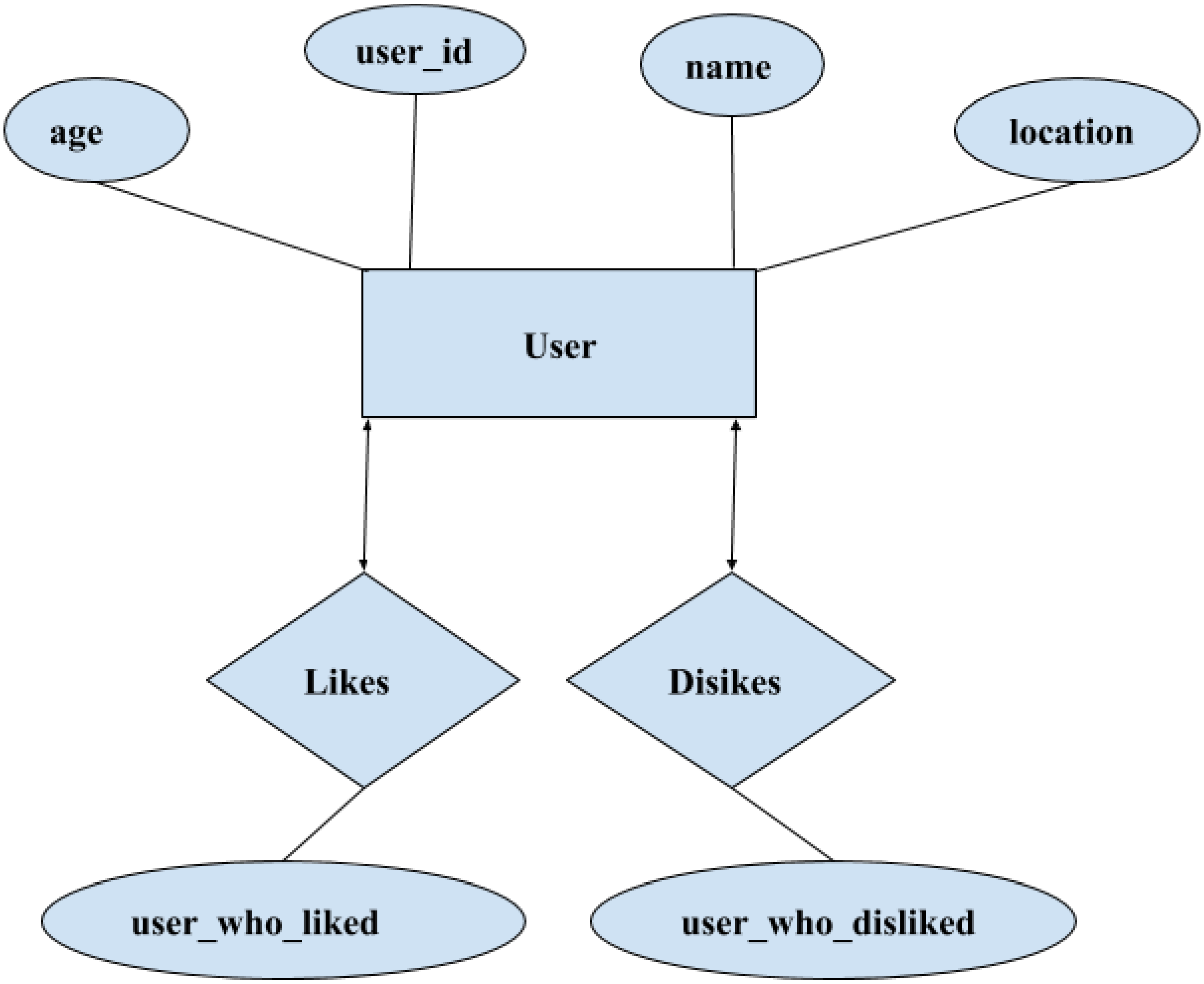
From the violation cs→cn, cb, we get, R3(cs, cn, cb) and R4(cs, s, as, am). R3 is in 4NF and we recursively decompose R4.

From the violation as→am, we get, R5(as, am) and R6(cs, s, as). R5 is in 4NF and R6 is also in 4NF as social security multivalue determines cs and as.

Therefore, our 4NF decomposition algorithm gives:

R1(s, n, b) , R3(cs, cn, cb) , R5(as, am) , and R6(cs, s, as).

4. The E/R diagram of the dating website might look like the following:



The two attributes can be added to the User Entity directly, but then it will lead to redundancy of the data. We thus use the divide table approach with 3 tables->

1. User table
2. User Like Table with User_id and the User_who_Liked_id
3. User dislike Table with User_id and the User_who_disliked_id