|  |
| --- |
|  |
| **Assignment 2**  **SYSC 5703 – Integrated Database Systems** |
| Fall 2013  Submitted To  Dr. Samuel A. Ajila  By  **Nikhil Nayyar (100 941 327)**  **Ferhan Jamal (100 953 487)**  Carleton University |

**Q1 Consider the following functional dependencies over the attribute set ABCDEFGH:**

**A E**

**AD BE**

**AC E**

**E B**

**BG F**

**BE D**

**BDH E**

**F A**

**D H**

**CD A**

**a. Find a minimal cover, and**

**b. Decompose into lossless 3NF.**

**c. After that, check if all the resulting relations are in BCNF. If you find a schema that is not, decompose it into a lossless BCNF. Explain all steps. (Exercise 6.20, page 248)**

**Solution 1a:**

First, we will change every functional dependency (FD) into the form X **** Y (with only one attribute on the right-hand side) which is also called Right-reducing step :

A**** E

AD**** B

AD**** E

AC**** E

E**** B

BG**** F

BE**** D

BDH**** E

F**** A

D**** H

CD**** A

Then we will check if any of the attributes on the left-hand side of the above Functional Dependencies are redundant or not. Left-reducing leads to the following set:

A **** E

BD **** E

A **** B

F **** A

E **** B

D **** H

BG **** F

CD **** A

E **** D

Now we check the new set of Functional Dependencies to see if any of them is redundant (i.e. they can be inferred from the others). By eliminating redundant ones, we obtain the following minimal cover of the given set of FDs which is the **minimal cover -**

**A  E**

**BD  E**

**E  B**

**F  A**

**BG  F**

**D  H**

**E  D**

**CD  A**

**Solution 1b:**

Firstly find the minimal cover which is –

**A  E, BD  E, E  B, F  A, BG  F, D  H, E  D, CD  A**

Secondly, each functional dependency gets its own schema and the attribute-set for each functional dependency x → y is given by x ∪ y which means -

* The functional dependency A → E obtains the new schema which is ({A, E}, {A → E}).
* The functional dependency BD → E obtains the new schema ({B, D, E}, {BD → E}).
* The functional dependency BG → F obtains the new schema ({B, G, F}, {B G → F}).
* The functional dependency F → A obtains the new schema ({F, A}, {F → A}).
* The functional dependency D → H obtains the new schema ({D, H}, {D → H}).
* The functional dependency CD → A obtains the new schema ({C, D, A}, {C D → A}).

Now, we will check if any one candidate key out of these {{A, C, G}, {B, C, G}, {C, D, G}, {C, E, G}, {C, F, G}} is part of a schema or not? And no schema contains any candidate key.

So below is the lossless 3NF –

(AE; {A→E}), (BDE; {BD →E, E →D, E →B}), (BGF; {BG →F}), (FA; {F →A}), (DH; {D →H}), (CDA; {CD →A)

**Solution 1c:**

After analyzing all the above relations, we found out that BGF and CDA are not in BCNF.

* BGF is not in BCNF because we can derive F → B from the original FD’s ( A → E, E → B which means A → B and we have F → A, now combining these two will give us F → B) and because of this reason they are not in BCNF.
* Similarly CDA is also not in BCNF because we can derived A → D from the original FD’s.

Now BGF can be decomposed into {BF; B → F} and {GF; } and CDA can be decomposed into {AD; A → D} and {AC; }

**Q2 Find a projection of the following set of dependencies on the attribute AFE:**

**A BC, E HG, C FG, G A**

**Solution 2:**

**Below are the projections of the above FD’s on attribute AFE which we will get after simplifying above FD’s as much as possible and after that we will find out those FD’s which are getting projected on AFE –**

Split the RHS to single attribute:

i. A B

ii A C

iii. EH

iv. E G

v. CF

vi. CG

vii GA

From (ii) and (v) we have AF(transitive dependency).

And From (iv) and (vii) we have EA(transitive dependency) which leads us to –

A F, E AF

OR

Similarly, we can get the below dependency using the above FD’s that we have split up.

A F, E A, E F

OR

And similarly this as well -

A F, E A

So the answers are -

A F, E AF

**OR**

A F, E A, E F

**OR**

A F, E A

**Q3** **For the attribute set ABCDEFG, let the MVDs be**

**ABCD DEFG**

**ABCE ABDFG**

**ABD CDEFG**

**Find a lossless decomposition into 4NF. Is it unique? Why or why not?**

**Solution 3:**

In the question below is the attribute set we have been given –

ABCDEFG

Now, we are going to apply first Multi Valued Dependency and after that we get the following decomposition –

ABCD, DEFG

The decomposition of this attributes from (a) gives us below relations –

R1 = ABCD R2 = DEFG

Now if we decompose further from (b), then it will gives us -

R3 = ABC R4 = ABD & R5 = DEFG

which means the second Multi Valued Dependency applies to ABCD as well and thus results into

ABC, ABD

Now, we will use the third Multi Valued Dependency to decompose ABC but it looks like, we cannot decompose ABC further. So the resulting output we will get is –

ABC, ABD, and DEFG

The above decomposition that we have generated is not unique in our opinion.

Now if we apply the third Multi Valued Dependency on the attribute set we have got, we would obtain the following result set:

ABD, CD, and DEFG.

**Q4 Using the schema given below, define trigger that fire when a student grade average drops below certain threshold. (For simplicity, assume that there is a function, grade\_avg (), which takes a student Id and returns the student average grade.)**

**Student (*Id, Name, Country*)**

**Course (*CrsCode, CrsName, Type, Instructor*)**

**Results (*Id, CrsCode, Grade)***

**Solution 4:**

We have assumed that the student passing grade (Threshold) is 40%.

CREATE TRIGGER **Check\_GPA**

AFTER UPDATE, DELETE, INSERT ON Results

FOR EACH STATEMENT

WHEN (EXISTS (SELECT Id FROM Results R

WHERE grade\_avg (Id) < 40%));

**Q5. It is possible to convert Datalog rules into equivalent relational algebra expressions. For each of the following Datalog rules, write an expression of relational algebra that defines the same relation as the head of the rule:**

a. P(x, y) <- Q(x, z) AND (R(z, y)

b. P(x, y) <-Q(x, z) AND Q(z, y)

c. P(x, y) <- Q(x, z) AND R(z, y) AND x < y

**Solution5:**

1. P(x, y) <- Q(x, z) AND (R(z, y)

**Solution (a):** Π x, y ( Q (X, Z)joinz = z’R (Z’, y))

1. P(x, y) <-Q(x, z) AND Q(z, y)

**Solution (b):** Π x, y ( Q (X, Z) joinz = z’Q (Z’, y))

1. P(x, y) <- Q(x, z) AND R(z, y) AND x < y

**Solution (c):** Π x, y ( Q (X, Z)joinz = z’AND X<yR (Z’, y))

**Q6.**

1. **In data mining, is a typical fact table in 4NF? Explain?**
2. **Consider the table below for a binary classification problem. This is the same example we saw in class.**

**Compute Gini index for the overall collection of training examples.**

**Solution 6:**

1. A fact table stores quantitative information for analysis and is often denormalized which means some fact tables reflect aggregated level data so it might be possible that some fact tables are in 4NF and some fact tables are not which means *Fact tables are not always in 4 Normal Form (NF).*  
     
   Consider employees, skills, and languages, where an employee may have several skills and several languages. We have here two many-to-many relationships, one between employees and skills, and one between employees and languages. Under fourth normal form, these two relationships should not be represented in a single record which violates 4NF such as  
     
    (Employee, Skill, Language)  
     
   Instead, they should be represented in the two records -  
     
    (Employee, Skill) and (Employee, Language)
2. By using the Gini index formula as mentioned below –

**i. Compute Gini index for the overall collection of training examples.**

Gini (t) =

= 1- [(6/20) 2 + (14 / 20) 2

**Gini (t)= .48**

**ii. Compute Gini index for the Married attribute**

**Solution:**

After analyzing the above table we can figure out that in the Married attribute –

There are total **14 YES and 6 NO** in the given attribute.

1. Out of this 14 YES, we have 11 No and 3 yes in the default attribute.
2. Out of this 6 NO, we have3 No and 3 yes in the default attribute.

YES = 1- [(11 / 14) 2 + (3 / 14) 2] = .3368

NO = 1 – [(3 / 6) 2 + (3 / 6)2] = .5

**WEIGHTED AVERAGE: = (14 / 20 \* .3368) + (6 / 20 \* .5)**

**= .390**

**iii. Computer Gini index for the Previous Default attribute**.

**Solution:** After analyzing the above table for the Previous Default Attribute –

There are total 6 YES and 14 NO in the attribute set.

1. Out of this 6 YES, we have 3 NO and 3 YES in the default attribute.
2. Out of this 14 NO, we have 3 YES and 11 NO in the default attribute.

YES = 1- [(3 / 6) 2 + (3 / 6) 2] = .5

NO = 1 – [(11 / 14) 2 + (3 / 14) 2] = .340

**WEIGHTED AVERAGE: = (6 / 20 \* .5) + (14 / 20 \* .3368)**

**= .390**

**iv. Compute Gini index for the Income attribute setting the threshold for default at less than 50.**

**Solution:** After analyzing the above table, we have found out that -

The income attribute will have the condition:-

THRESHOLD < 50 have 2 YES and 8 NO

THRESHOLD >= 50 have 4 YES and 6 NO

THRESHOLD < 50 = 1- [(2 / 10) 2 + (8 / 10) 2] = .70

THRESHOLD > 50 = 1 – [(4 / 10) 2 + (6 / 10) 2] = .46

**WEIGHTED AVERAGE: = (10 / 20 \* .72) + (10 / 20 \* .48)**

**= .65**

**v. Which attribute (Married or Previous Default or Income) is better? Is there any difference between these results and the Entropy results?**

**Solution**: Below are the entropy results of the given attributes –

Married: 0.39

Previous Default: 0.39

Income: 0.65

After looking at the above results, it looks like that Married attribute and Previous Default attribute is looking much better after taking Gini Index into consideration which is the distribution of the sample with zero reflecting the most distributed sample set. Therefore, Married and Previous Default attribute is better whose value is **0.39**.