

NAME: Nicholas Jacob  
STUDENT ID: # 113578513  
GRADED HOMEWORK NUMBER: 4  
COURSE: CS/DSA 4513 DATABASE MANAGEMENT  
SECTION: ONLINE  
SEMESTER: FALL 2023  
INSTRUCTOR: DR. LE GRUENWALD  
SCORE:

1. (a) Find the name of each employee who works for “Big Bank”.

$$\Pi_{person\_name}(\sigma_{company\_name=“BigBank”}(works))$$

- (b) Find the name and city of each employee who works for “Big Bank”.

$$\Pi_{person\_name,city}(\sigma_{company\_name=“BigBank”}(employee \bowtie_{person\_name} works))$$

- (c) Find the name, street address, and city of each employee who works for “Big Bank” and earns more than \$10 000.

$$\Pi_{person\_name,street,city}(\sigma_{(salary>10000)\wedge(company\_name=“BigBank”)}(employee \bowtie_{person\_name} works))$$

- (d) Find the name of each employee who lives in the same city as the company for which they work.

$$\Pi_{person\_name}(\sigma_{employee.city=company.city}((employee \bowtie_{person\_name} works) \bowtie_{company\_name} company))$$

2. Given the relational schema Employee(id, name, classid,gender, manager,salary) and the set of Functional Dependencies

FD =

{(classid,id,gender) → (salary, manager),  
 name → (age,id),  
 id → name,  
 manager → (gender, age, classid,id)}

- (a) To find all candidate keys, we will first examine the closure  $FD^+$ .  
 Using decomposition, we see that

manager → gender,  
 manager → age,  
 manager → classid,  
 manager → id,

Then we grab the penultimate and the last here and use transitivity so,  
 manager → name.

To show manager is a candidate key, I still need salary. We see from decomposition that

$(\text{classid}, \text{id}, \text{gender}) \rightarrow \text{salary}$ , and

$\text{manager} \rightarrow (\text{gender}, \text{classid}, \text{id})$  so transitivity gives us

$\text{manager} \rightarrow \text{salary}$ . Thus manager is a superkey. You cannot simplify manager so it is also a candidate key.

The other superkey is  $(\text{classid}, \text{id}, \text{gender})$ . We see right away that its closure contains salary and manager (decomposition rule). With this addition, we can also add age (decomposition again). We also get id trivially (reflexivity) so name will be included. Thus we have shown that all elements belong to the superkey  $(\text{classid}, \text{id}, \text{gender})$ . We note that this set is minimal. Closure on any of the individual or combination of elements will not give you manager. Thus  $(\text{classid}, \text{id}, \text{gender})$  is a candidate key.

We do not see any other keys. id and name had the best shot but both do not imply manager so you are stuck.

(b) Normal Forms

- i. This relational schema is in the first normal form since all entries are atomic.
- ii. To check for second normal form, we would need for any non-prime attribute to be fully dependent on the candidate key. We see that name is partially dependent on the candidate key  $(\text{classid}, \text{id}, \text{gender})$  since  $\text{id} \rightarrow \text{name}$ .
- iii. We are not third normal form either.  $\text{name} \rightarrow \text{age}$  is in the closure (by decomposition). But age is not in any of the candidate keys so not third normal form as age is not a prime attribute.
- iv. Since we were not third normal form, we will not be BCNF. We can see this by definition too.  $\text{name} \rightarrow \text{age}$  is again a issue. name is not a superkey.

(c) Since I was 1NF and 2NF but not 3NF, I will decompose with the desired result being 3NF. We first take each of the relations and build a schema from each.

$R_1(\text{classid}, \text{id}, \text{gender}, \text{salary}, \text{manager})$

$R_2(\text{name}, \text{age}, \text{id})$

$R_3(\text{id}, \text{name})$

$R_4(\text{manager}, \text{age}, \text{classid}, \text{id}, \text{gender})$

We see that  $R_1, R_4$  contain a candidate key. So we proceed to the next step of eliminating redundant relations.

$R_1(\text{classid}, \text{id}, \text{gender}, \text{salary}, \text{manager})$

$R_2(\text{name}, \text{age}, \text{id})$

I think I have removed all the redundant sets, so now it is time to identify the functional dependencies on each. The candidate key for  $R_1$  is (classid, id, gender). For  $R_2$  it is name. The functional dependencies would be (classid, id, gender)  $\rightarrow$  (salary, manager) and manager  $\rightarrow$  (gender, age, classid, id) on  $R_1$ . name  $\rightarrow$  (age, id) and id  $\rightarrow$  name on  $R_2$ . (name, id) is a superkey on  $R_2$  with name being a candidate key and (classid, id, gender, manager) is a superkey with (classid, id, gender) or manager being a candidate key.

- (d) This is a lossless join. As long as I did it correct, the 3NF decomposition is supposed to be lossless but we'll examine the details. We need to show that the intersection of all the sets create a functional dependency that was in the closure of the original. Start with  $R_1$ . We intersect that with  $R_2$  and see that the element id survives. We note that id  $\rightarrow$  (name, age, id) so this abides the lossless requirements  $R_1 \cap R_2 \rightarrow R_1$ .
- (e) This is a dependency preforming join. Again the 3NF decomposition is supposed to be dependency preserving. I haven't done it correctly because I lost manager  $\rightarrow$  age.

Try again...

- (f) I think the decomposition should be

$R_1(\text{CIGS})$

$R_2(\text{CIGM})$

$R_3(\text{NA})$

$R_4(\text{NI})$

$R_5(\text{MA})$

then superkeys are cig, cigm, n, ni, m. Dependencies preserved are cig  $\rightarrow$  s, cig  $\leftrightarrow$  m, n  $\rightarrow$  a, n  $\leftrightarrow$  i, m  $\rightarrow$  a.

- (g) I am having trouble showing the lossless join. I think I join 1 and 2. Join that with 5. Then join up 3 and 4 (all of these are lossless) then join together. This is called the chase (<https://stackoverflow.com/questions/21560069/lossy-join-decomposition-more-than-two-relations>)