# HOMEWORK -4-

*Doren Calliku*
*Group 8*

## Homework exercise:

Study the Northshore Hospital case study file available on Canvas, using the following Figure1:

a. Identify the functional dependencies represented by the attributes shown in the form in Figure 1. State any assumptions that you make about the data and the attributes shown in this form.

b. Describe and illustrate the process of normalizing the attributes shown in Figure 1 to produce a set of well-designed 3NF relations.

c. Identify the primary and foreign keys in your 3NF relations.

d. Examine the 3NF relations created to represent the attributes shown in the Hospital form shown in Figure 1.

**Patient Medication Form**

Patient Number: P10034
Full Name: Robert MacDonald
Ward Number: Ward 11
Bed Number: 84
Ward Name: Orthopaedic

| Drug Number | Name | Description | Dosage | Method of Admin | Units per Day | Start Date | Finish Date |
|---|---|---|---|---|---|---|---|
| 10223 | Morphine | Pain Killer | 10mg/ml | Oral | 50 | 24/03/13 | 24/04/14 |
| 10334 | Tetracyclene | Antibiotic | 0.5mg/ml | IV | 10 | 24/03/13 | 17/04/13 |
| 10223 | Morphine | Pain Killer | 10mg/ml | Oral | 10 | 25/04/14 | 02/05/15 |

Figure 1. Hospital Patient Medication Form.

A) Identify the functional dependencies represented by the attributes shown in the form in Figure 1. State any assumptions that you make about the data and the attributes shown in this form.

Functional dependencies are the constraints that describe the relationship between attributes in a relation. It happens between two sets of attributes. So, if we have a set of attributes called A in a relation X, and each value of it is related(associated) with exactly one value of another set of attributes B, then we have a functional dependency from A to B. Another way to interpret the functional dependencies is that F(A)→B, so B is dependent on value of A, or two tuples(data) sharing the same value of A will have the same value of B.
If a row is added where the student had a different value of semester then the functional dependency, F would no longer exist. Functional dependencies are dependent on data.
In order for us to express functional dependencies we have to have tables (or relations) in which we define which attributes are connected to which attributes by a functional dependency. So, we got only the table for the drugs, as to include a table in which only one tuple is given, as the patient medication form, it is practically meaningless. We also assume to take some of the features of the normalization process, in order to not make meaningless associations between data and attributes.
Functional Dependencies are the following:

Drug Number → {Name, Description, Dosage}

{Drug Number, Start Date} → {Method of Admin, Units/day/ Finish Date}
OR in place of Start date can be any of the other attributes. The start date was chosen for logical implication that Start Date would define the other ones, as the medication might have been given on that day. The other possibilities are still mathematically possible.

Assuming that our assumption that we have to deal only with the table might be wrong (because we got a little number of results) we can discuss about “Expected Functional Dependencies” which are the ones that one would expect to happen, but because of lack of data we cannot say that they are dependencies.
Patient Number → {Full Name, Ward Number, Bed Number}
Ward Number → Ward Number
Note: The dates in the “Finish Date” are a little unbelievable.

b) Describe and illustrate the process of normalizing the attributes shown in Figure 1 to produce a set of well-designed 3NF relations.

Normalization process is used to avoid or eliminate the problems that can arrive with data in a database system, those being: insertion, deletion and updating.
First normal form of the relation is when all of its attributes are simple, so none of them is a relation. We can do this by creating super-keys from tables, by uniting two primary keys. Then fully functionally dependent keys are registered.
Second normal form happens when all the non-primary attributes of the relation are fully functional on the primary key. So, we check the attributes which have functional dependencies and try to divide them so that we do not get any non-primary attribute.
Third Normal Form happens when a second normal form relation if all non-primary attributes (that is attributes that are not parts of the primary key or of any candidate key) have non-transitivity dependency on the primary key.

The process of normalizing the relation will assume that we take the whole medication form and also it will assume that some logical implication follow based on previous information in previous homework, and the North Case Study. So the decisions will not be based only on the information in the form.
Point 1: The relation between the big relation and the relation included in it can be expressed by two tables, first one having as a primary key the patient number, and the second one the drug number and the start date (with candidate keys: drug number+ units per day, drug number + finish date).
Point 2: The information about the patient can be divided into two tables, first one including patient number as primary key and also the connection between ward number and ward name is brought alone. To mention is that we do not include the bed number as defined from the primary keys of these two primary keys because it would allow needless redundancy. Still in the future it might be allowed.
Point 3: We continue to divide the table of the medication as we started in point one.

Table 1.

| Ward Number | Ward Name |
|---|---|
| 11 | Orthopaedic |

Table 2.

| Patient Number | Name | Bed | **Ward Number** |
|---|---|---|---|
| P10034 | Robert McDonald | 84 | **11** |

Table 3.

| **Drug Number** | Units per day | Start Date | Finish Date |
|---|---|---|---|
| **10223** | 50 | 24/03/13 | 24/04/14 |
| **10334** | 10 | 24/03/13 | 17/04/13 |
| **10223** | 10 | 25/04/14 | 02/05/15 |

Table 4.

| Drug Number | Name | Description | Dosage | Method of Admin |
|---|---|---|---|---|
| 10223 | Morphine | Pain Killer | 10mg/ml | Oral |
| 10334 | Tetracyclene | Antibiotic | 0.5mg/ml | IV |

Table 5.

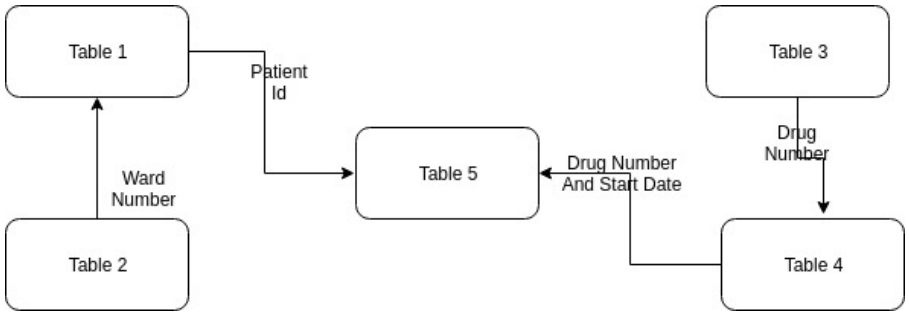
| **Patient Number** | **DrugNumber** | **Start Date** |
|---|---|---|
| **P10034** | **10223** | **24/03/13** |
| **P10034** | **10334** | **24/03/13** |
| **P10034** | **10223** | **25/04/14** |

C) Identify the primary and foreign keys in your 3NF relations.

D) Examine the 3NF relations created to represent the attributes shown in the Hospital form shown in Figure 1.

Identified primary keys on yellow, foreign keys on **Bold**.

Examination and analysis is done on question b. Still, to mention is the last table which is to connect the other tables together. The visual connection is shown in the picture.



## Extra queries:

The relation shown in Figure 2 describes hospitals (hospitalName) that require certain items (itemDescription), Which are supplied by suppliers (supplierNo) to the hospitals (hospitalName).

Furthermore, whenever a hospital (h) requires a certain item (i), and a supplier (s) supplies that item (i), and the supplier (s) already supplies at least one item to that hospital (h), then the supplier (s) will also supply the required item (i) to the hospital (h). In this example, assume that a description of an item (itemDescription) uniquely identifies each type of item.

a. Describe why the relation shown in Figure 2 is not in 4NF.

b. Describe and illustrate the process of normalizing the relation shown in Figure 2 to 4NF.

| Hospital name | Item Description | Supplier No. |
|---|---|---|
| Northshore Hospital | Antiseptic Wipes | S1 |
| Northshore Hospital | Paper towels | S2 |
| Northshore Hospital | Antiseptic Wipes | S2 |
| Northshore Hospital | Antiseptic Wipes | S2 |

**A)**

In order for a table to be turned into the fourth normal form it should be in the third normal form, BCNF and then we should add multivalued dependencies condition. A table is in fourth normal form if and only if for every multivalued non-trivial dependencies X→ → Y, where X is a super-key.

A non-trivial multivalued dependency happens if we choose any x actually occurring in the table, and combine a list of where it is found, and then we will see that y is connected to x no matter how z (another attribute) behaves. Practically, what multivalued dependency tries to constrain is the possibility of a data corruption where a value of x is not spread to y independently to z. So we are trying to express combinations of groups(attributes) so that certain values appear in all combinations.

For the existing table in this exercise let us check whether it is in third normal form:
Third Normal Form happens when a second normal form relation if all non-primary attributes (that is attributes that are not parts of the primary key or of any candidate key) have non-transitivity dependency on the primary key.

First, there is redundancy in the table. Rows 1 and 4 are repeated, which cannot be part of the fourth normal form. So, the necessary condition is not fulfilled. The fourth normal form is a way of following multivalued dependencies in an existing third normal form. The table cannot be in BCNF, because it cannot be in third normal form with that type of redundancy.

**B)**

Normalization removes redundant data from relations to improve database efficiency and veracity by: storage efficiency, data integrity, and scalability. Normalization is the process of splitting tables into other tables which should be rejoined every time we call a query. For better normalization we use the fourth normal form.

Point 1:

We reduce the redundancy of the table by deleting one entry which is represented earlier. This entry might be the first or the last tuple. We choose the last one.

| Hospital name | Item Description | Supplier No. |
|---|---|---|
| Northshore Hospital | Antiseptic Wipes | S1 |
| Northshore Hospital | Paper towels | S2 |
| Northshore Hospital | Antiseptic Wipes | S2 |

Point 2:

Multivalued dependency A->>B holds in a relation R if in every instance of R, for every pair of tuples t and u in R that agree on all the A's, we can find a tuple v in R that agrees

1. With both t and u on A's
2. With t on the b's, and
3. With u on all those attributes of R that are not A's or B's

The relation that we have the table is represented by Medication Supply (Hospital Name, Item Description, Supplier No.). In this relation each Item Description can be supplied from a Supplier No. , and each Item Description can be needed in different Hospital Name's. Each Hospital Name connects to every Supplier No. when needing an Item Description. The relation is in BCNF since there are no non-trivial functional dependencies. Can we swap i→h or i→s and have the resulting tuple is still in the relation?

We add the last row as we know from the conditions that s1 supplies paper towels too.

Now we have:

| Hospital Name | Item Description | Supplier No. |
|---|---|---|
| Northshore Hospital {t} | Antiseptic Wipes | S1 |
| Northshore Hospital {u} | Paper Towels | S2 |
| Northshore Hospital {v1} | Antiseptic Wipes | S2 |
| Northshore Hospital {v2} | Paper Towels | S1 |

Now we can see that Hospital Name ->> Item Description is a multiple dependency. For each pair of tuples t and u that agree on Hospital Name, we can find a tuple which agrees

1. With both t and u on Hospital Name
2. With t on Item Description {v1} and with u on Supplier No {v2}.

Same is true for Hospital Name ->> Supplier No. .

Decomposition is dependent on Multiple Dependencies, so now we have:

| **Hospital Name** | Item Description |
|---|---|
| Northshore Hospital | Antiseptic Wipes |
| Northshore Hospital | Paper Towels |

| **Hospital Name** | Supplier No. |
|---|---|
| Northshore Hospital | S1 |
| Northshore Hospital | S2 |

Note: Extra queries would not take this much text, but I tried to be explanatory.

References that might be a good source for explaining fourth normal form and multiple dependencies in general:

http://courses.cs.vt.edu/~cs4604/Fall08/lectures/lecture16.pdf

Please if you have time check it so other students can check it too. It explains them simply.