

# Advanced Data Management (CMM524)

## Laboratory #4: Normalisation

### 1. Aims

- To design a database that conforms to the third normal form (3NF).

### 2. Outcomes

In completing this exercise, you should be able to:

- Identify functional dependency among attributes in a table.
- Identify full and partial functional dependency.
- Identify transitive functional dependency.
- Apply normalisation techniques to a given relational database design up to 3NF.

### 3. The Problem Domain

In this domain, we are storing inspection information of properties of an estate agent.

- A property has a unique property number and an address.
- From time to time, staff have to inspect a property and record their findings as comment. A staff will take a company car to do the visit.
- Each staff has a unique staff number and a name. The name may not be unique.
- Each company car has a unique registration number.
- We need to know the date and time an inspection is done, the staff who did it, the company car used, and comments on the inspection.
- There may be multiple visits to the same property on the same date but the company makes sure that only one inspection is done to the same property at the same time.

#### 3.1. The Original Table

The original table design is as followed:

propertyNo	pAddress	iDate	iTime	comments	staffNo	sName	carReg
PG4	6 Lawrence Street, Glasgow	18/10/2016	10:00	Need to replace crockery.	SG36	Ann Beech	M231JGR
		22/04/2017	09:00	In good order.	SG14	David Ford	M533HDR
		01/10/2017	12:00	Damp rot in bathroom.	SG14	David Ford	N721HFR
PG16	5 Novar Drive, Glasgow	22/04/2017	13:00	Replace living room carpet.	SG14	David Ford	M533HDR
		24/10/2017	14:00	Good condition.	SG36	Ann Beech	N721HFR

### Notes:

- **DO NOT blindly copy this example to your coursework.**
- This table is an extreme case of poor design. We use it to illustrate different design flaws and how normalisation can fix the issues.
- In most cases, if you have done some sensible ER modelling, you should not end up with 1 single big table but multiple smaller tables.
- Your normalisation process should start from these multiple smaller tables rather 1 single big table.
- This table is not in 1NF.
  - How can you tell?

### 3.2. First Normal Form & Primary Key

The table is then revised:

propertyNo	pAddress	iDate	iTime	comments	staffNo	sName	carReg
PG4	6 Lawrence Street, Glasgow	18/10/2016	10:00	Need to replace crockery.	SG36	Ann Beech	M231JGR
PG4	6 Lawrence Street, Glasgow	22/04/2017	09:00	In good order.	SG14	David Ford	M533HDR
PG4	6 Lawrence Street, Glasgow	01/10/2017	12:00	Damp rot in bathroom.	SG14	David Ford	N721HFR
PG16	5 Novar Drive, Glasgow	22/04/2017	13:00	Replace living room carpet.	SG14	David Ford	M533HDR
PG16	5 Novar Drive, Glasgow	24/10/2017	14:00	Good condition.	SG36	Ann Beech	N721HFR

- Is this table in 1NF? How can you tell?
- Choose a primary key for this table.
  - Note: The primary key can be a composite key comprises multiple attributes. It must be able to uniquely identify a row in the table.

### 3.3. Second Normal Form

- Identify all functional dependencies in the table. For example:

propertyNo → pAddress

- **The sample data DO NOT show all possible cases.** You should use your understanding of the domain to identify the functional dependencies.
- List all function dependencies found using the above notation.
- List all full functional dependency on the primary key.
- List all see any partial functional dependency on the primary key.
- Is the table in 2NF? How can you tell?
- Using the functional dependency identified, transform this table into 2NF.

- Update the primary keys in all tables if necessary.

### 3.4. Third normal Form

- Identify and list all any transitive functional dependencies in your tables.
- Is your table now in 3NF? How can you tell?
- Using the functional dependency identified, further transform this table into 3NF.

## 4. The Patient Record Domain

- In a hospital, a patient is admitted to a ward on the admission date. There will be a discharge date when the patient leaves.
- The patient will be give a bed on admission.
- Throughout his/her lifetime, a patient may be admitted several times (and potentially to different wards).
- Each patient has a patient number which uniquely identifies his/her records in the hospital.
- Each ward has a unique ward number, and a name.
- While the patient is in hospital, he/she may be prescribed with drugs on a certain date and time. Note that multiple different drugs can be prescribed at the same time.
- Each drug has a unique drug ID, and a name.
- Each prescription has a dosage.

The followings attributes/columns are identified:

patientNo, pName, wardNo, wName, bedNo, admissionDate, dischargeDate, drugID, dName, dosage, pDate, pTime

### Notes:

- You can do ER modelling which will give you multiple tables to normalise, or you can skip ER modelling and put all attributes into 1 single table.
- In either case, the normalisation process will recognise design flaws and fix them.
- Again, in your coursework you should do normalisation on the multiple tables from ER modelling, not from a single big table.
- Identify a primary key (which may comprises multiple attributes) for the table.
- List all functional dependencies among the attributes.
- Transform the table to 2NF and 3NF.