

# Chapter 5

# Database normalisation

## Chapter summary

When moving data from the real world to relational databases, we often face difficulties trying to ensure that data is only held once within the system and that the same data does not reappear in another table. Apart from being wasteful of space, a far more compelling reason is to ensure that any change made to a specific item (say a telephone number) is automatically used by every part of the database system. If two copies of the same telephone number existed then discrepancies could easily arise should only one of these numbers be changed. This chapter looks in detail at the process of normalisation, in particular the first, second and third normal forms.

## Learning outcomes

After studying this chapter, you should check your knowledge against the outcomes below and test your achievement by answering the questions at the end of the chapter. You should be able to:

### **Outcome 1: Understand the reason for normalisation.**

You should be able to describe the phases of normalisation and what must be done at each step.

### **Outcome 2: Carry out normalisation to third normal form.**

Starting with a set of un-normalised data available in relational table format, you should be able to go through the steps of normalisation (1NF, 2NF and 3NF).

## How will you be assessed on this?

Normalisation is always popular for exams. You will likely be asked to define the first three stages of normalisation (1NF, 2NF and 3NF), setting out clearly what you must do to achieve the various stages. After that you will likely be given a sample problem with a description or a relationship with its known dependencies. Either way you will be asked to normalise the data to 3NF clearly showing your intermediate steps.

## Section 1

# Introduction to normalisation

Normalisation is a fairly methodical process exercising a set of definable rules in order to arrive at a robust table arrangement. As we will see, the process relies heavily on knowledge of the data. There are several levels of normalisation which we will group as follows:

- 1NF – First normal form
- 2NF – Second normal form
- 3NF – Third normal form
- BCNF – Boyce-Codd normal form
- 4NF – Fourth normal form
- 5NF – Fifth normal form
- DK/NF – Domain-key normal form

In terms of importance, 1NF, 2NF and 3NF are by far the most important and should be considered as mandatory for all practical relational databases. BCNF is a more rigorous form of 3NF and should be applied to production databases. 4NF, 5NF and DK/NF are not generally considered as these deal with situations considered to be very rare.

Initially only three forms of normalisation (1NF, 2NF and 3NF) were put forward by E. F. Codd in 1972. The Boyce-Codd normal form was later introduced by R. Boyce and E. F. Codd in 1974. The later forms were mainly the work of R. Fagin in the period from 1977 through to 1981.

Going beyond third normal form is outside the scope of this book. Before we go into detail on the process of normalisation, we will first summarise the steps to be taken to achieve third normal form. These are shown in Figure 5.1.

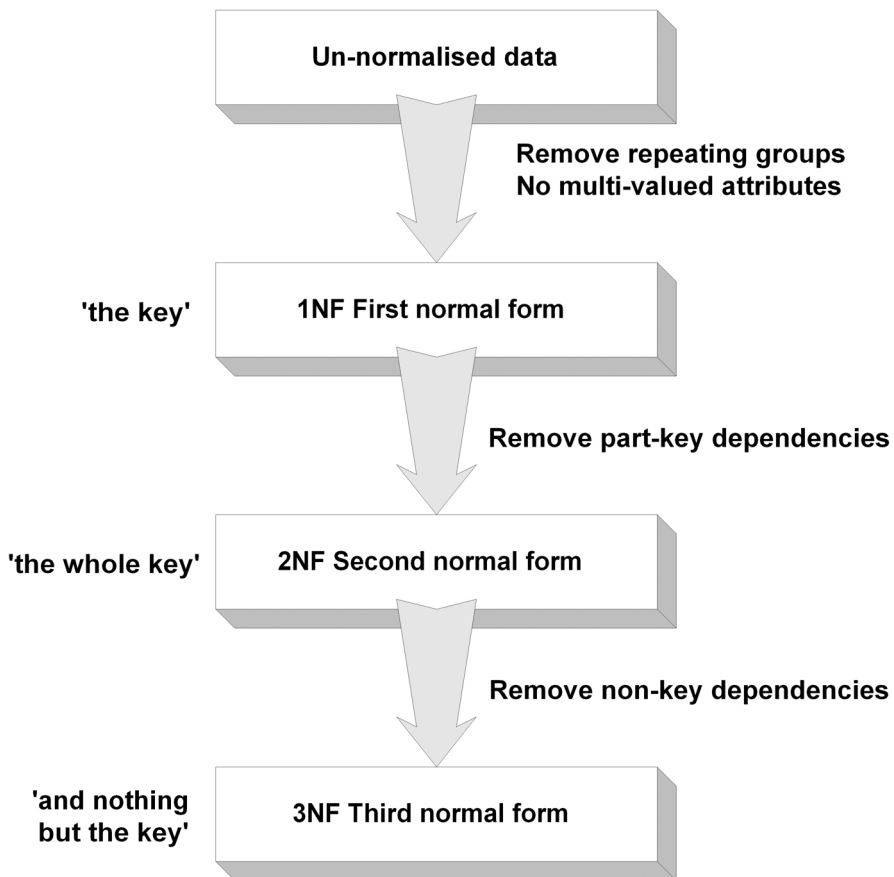


Figure 5.1 Steps to achieve third normal form

## Case study

Let's consider an example where we may want to commit to a database the details of packing notes raised by a supplier. The fields involved on the packing note might be as shown in Figure 5.2.

### Packing Note No: <sup>300</sup>

---

Packed By: <i>JW</i>	Customer Name: <i>Bloggs</i>	Customer Address: <i>Perth</i>
-------------------------	---------------------------------	-----------------------------------

No	Qty	Part No	Description
1	<i>200</i>	<i>1234</i>	<i>Nuts</i>
2	<i>200</i>	<i>2234</i>	<i>Bolts</i>
3	<i>200</i>	<i>3334</i>	<i>Washers</i>
4			
5			
6			
7			
8			

Figure 5.2 Packing note

## Un-normalised data

The first important fact to realise is that there are fields which appear only once on the packing note (those in the header group) and there are fields that repeat for every separate item listed on the packing note (those in the invoice body group). If we were to try to make one record for each packing note, this would result as shown in Table 5.1:

Table 5.1 Un-normalised data

NoteNo	Packer	Name	Address	ItemNo	Qty	PartNo	Desc
300	JW	Bloggs	Perth	1	200	1234	Nuts
				2	200	2234	Bolts
				3	200	3334	Washers

Here we can clearly identify repeating groups. But fields must be 'atomic' in the sense that there can only be one value in any field (no multi-valued attributes).

In theory we could extend the number of columns and introduce the following fields: