

**Employee**

Name	HireDate	Email	DeptNo
Jones	Feb 1, 2010	Jones@ourcompany.com	100
Smith	Dec 3, 2012	Smith@ourcompany.com	200
Chau	March 7, 2012	Chau@ourcompany.com	100
Greene	July 17, 2011	Greene@ourcompany.com	100

**Department**

DeptNo	DeptName
100	Accounting
200	Marketing
300	Information Systems

**Figure 5-23**

Two Normalized Tables

with its own theme. In this case, we create an *Employee* table and a *Department* table, as shown in Figure 5-23.

As mentioned, there are dozens of ways that tables can be poorly formed. Database practitioners classify tables into various **normal forms** according to the kinds of problems they have. Transforming a table into a normal form to remove duplicated data and other problems is called *normalizing* the table.<sup>2</sup> Thus, when you hear a database designer say, “Those tables are not normalized,” she does not mean that the tables have irregular, not-normal data. Instead, she means that the tables have a format that could cause data integrity problems.

### Summary of Normalization

As a future user of databases, you do not need to know the details of normalization. Instead, understand the general principle that every normalized (well-formed) table has one and only one theme. Further, tables that are not normalized are subject to data integrity problems.

Be aware, too, that normalization is just one criterion for evaluating database designs. Because normalized designs can be slower to process, database designers sometimes choose to accept non-normalized tables. The best design depends on the users’ processing requirements.

### Representing Relationships

Figure 5-24 shows the steps involved in transforming a data model into a relational database design. First, the database designer creates a table for each entity. The identifier of the entity becomes the key of the table. Each attribute of the entity becomes a column of the table. Next, the resulting tables are normalized so that each table has a single theme. Once that has been done, the next step is to represent relationship among those tables.

- Represent each entity with a table
  - Entity identifier becomes table key
  - Entity attributes become table columns
- Normalize tables as necessary
- Represent relationships
  - Use foreign keys
  - Add additional tables for N:M relationships

**Figure 5-24**

Transforming a Data Model into a Database Design

<sup>2</sup>See David Kroenke and David Auer, *Database Processing*, 12th ed. (Upper Saddle River, NJ: Prentice Hall, 2012) for more information.