

Because department names are duplicated in this table, every row that has a value of "Accounting" must be changed to "Accounting and Finance."

Data Integrity Problems

Suppose the Accounting name change is correctly made in two rows, but not in the third. The result is shown in Figure 5-22 (b). This table has what is called a **data integrity problem**: Some rows indicate that the name of Department 100 is "Accounting and Finance," and another row indicates that the name of Department 100 is "Accounting." This problem is easy to spot in this small table. But consider a table like the *Customer* table in the Amazon.com database or the eBay database. Those databases may have millions of rows. Once a table that large develops serious data integrity problems, months of labor will be required to remove them.

Data integrity problems are serious. A table that has data integrity problems will produce incorrect and inconsistent information. Users will lose confidence in the data, and the system will develop a poor reputation. Information systems with poor reputations become serious burdens to the organizations that use them.

Normalizing for Data Integrity

The data integrity problem can occur only if data are duplicated. Because of this, one easy way to eliminate the problem is to eliminate the duplicated data. We can do this by transforming the table in Figure 5-22 into two tables, as shown in Figure 5-23. Here, the name of the department is stored just once; therefore no data inconsistencies can occur. Of course, to produce an employee report that includes the department name, the two tables in Figure 5-23 will need to be joined back together. Because such joining of tables is common, DBMS products have been programmed to perform it efficiently, but it still requires work. From this example, you can see a trade-off in database design: Normalized tables eliminate data duplication, but they can be slower to process. Dealing with such trade-offs is an important consideration in database design.

The general goal of normalization is to construct tables such that every table has a *single* topic or theme. In good writing, every paragraph should have a single theme.

This is true of databases as well: every table should have a single theme. The problem with the table in Figure 5-22 is that it has two independent themes: employees and departments. The way to correct the problem is to split the table into two tables, each

Figure 5-22
A Poorly Designed
Employee Table

Employee

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

(a) Table Before Update

Employee

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

(b) Table with Incomplete Update