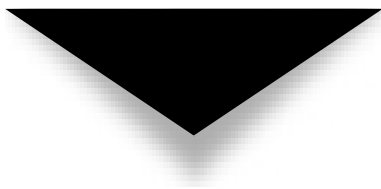

PART I



Installation and Administration



CHAPTER 1



What Is PostgreSQL?

There have always been a handful of different commercial database systems available for Microsoft Windows users and developers to choose from. The choices vary widely, from simple user database systems such as Microsoft's Access or FoxPro to more advanced systems such as Microsoft's SQL Server, IBM's DB2, or the Oracle suite of database software packages. However, now there's yet another player in the Microsoft database world.

If you are new to Open Source software, you may not have ever heard of the PostgreSQL database system. It has been around in the Unix and Linux worlds for quite some time, gathering quite a following of users and developers. Unfortunately, in earlier versions of PostgreSQL you had to be pretty knowledgeable and computer-savvy to get PostgreSQL to work on a Windows platform. This left PostgreSQL as an unknown for most Windows database users. However, as of PostgreSQL version 8, installing and running PostgreSQL in Windows is a snap. Now any Windows developer and common user can create professional databases using the high-quality, free PostgreSQL package.

This chapter introduces PostgreSQL, and explains the myriad of features available that make it a great choice for both Windows application developers and normal Windows users when creating database applications. You will see that just because a software package is free doesn't mean that it cannot compete with high-quality, expensive commercial products.

THE OPEN SOURCE MOVEMENT

Usually Windows developers and users reach for commercial products as the first solution to provide software for projects. The term "free software" conjures up memories from the old days of sloppily written freeware, packages with pop-up advertisements in them, or limited shareware applications. The Open Source movement cannot be farther from that concept. Open Source projects are written by teams of both amateur and professional programmers working to produce commercial-quality applications, mostly for the love of programming.

One of the first misconceptions of Windows users when starting out with Open Source software is the definition of the term *free*. The free part of Open Source is more related to sharing than price. Under Open Source software rules, a company or organization is allowed to charge a price for distributing Open Source software (although many do not). The free part comes from the program source code being freely sharable to anyone who wants to view and modify it.

Since sharing is the cornerstone of Open Source, any modifications made to Open Source code must also be shared. This process encourages improvements and feature enhancements from both developers and users. Many programmers feel this is the main reason Open Source software has enjoyed the popularity it has. This method of sharing new ideas quickly propels simple software ideas into mainstream applications.

There are many different types of licenses that Open Source software is released under. The most popular is the GNU General Public License (GPL). The GNU organization (www.gnu.org) supports Open Source software projects, and has published the GPL as a guide for how Open Source projects should be licensed to the public. If you have had any dealings with the popular Linux operating system, no doubt you have heard of the GPL. The GPL stipulates that any changes made to an Open Source project's code must be publicly published and available at no cost. While this is great for hobbyists and academics, it can cause problems for commercial organizations wanting to use Open Source code.

The developers of PostgreSQL have decided to release PostgreSQL under a slightly different Open Source license. PostgreSQL uses the BSD license, developed at the University of California (UC), Berkeley for public projects. This license is less restrictive than the GPL. It allows organizations to modify the code for internal use without being bound to publicly release the changes. This allows corporations (and private users as well) to use PostgreSQL however they want. This has provided a catalyst for many companies to use the PostgreSQL database as an internal database engine for many different commercial applications, as well as using PostgreSQL as the back-end database for some web sites.

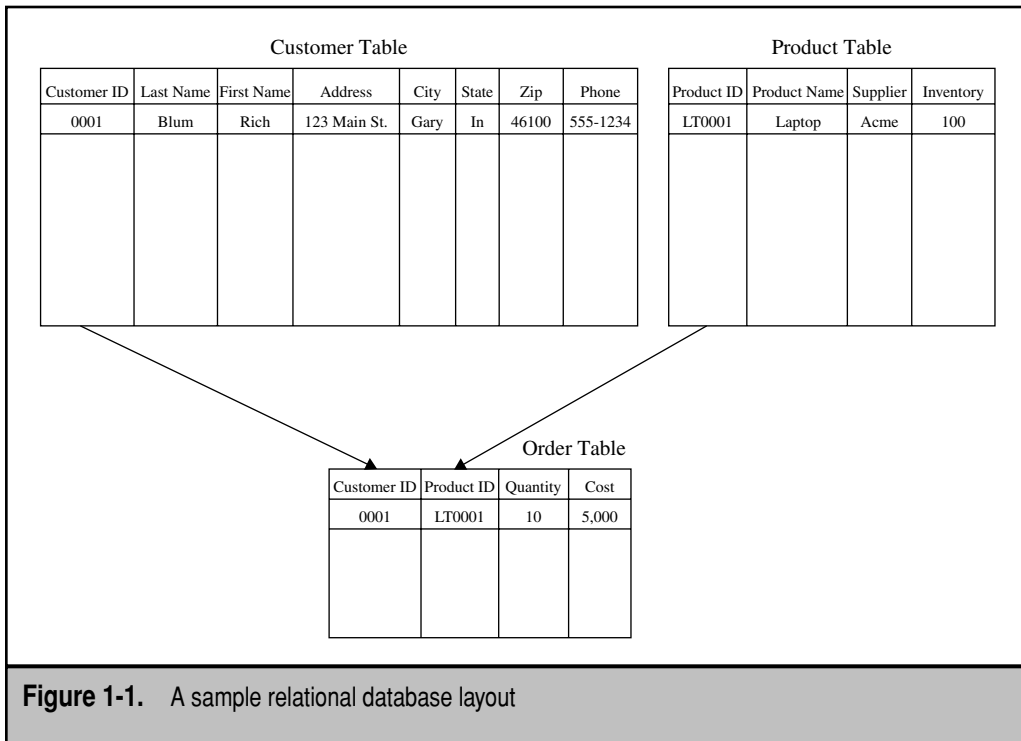
Under the BSD license, the developers of PostgreSQL are able to provide PostgreSQL free of charge at the same time that a few companies provide their versions of PostgreSQL as a for-profit commercial product. If you want to use PostgreSQL as-is on your own, you are free to download it and use it for whatever purposes you want. If you want to use PostgreSQL for a high-visibility production application that requires 24-hour support, you are able to purchase it from a company that provides such services. This is the best of both worlds.

THE HISTORY OF POSTGRESQL

To fully appreciate PostgreSQL, it helps to know where it came from. PostgreSQL started life as an academic database project at UC Berkeley. Professor Michael Stonebraker is credited as the father of PostgreSQL. In 1986 he started a project (then called Postgres) as a follow-up to another popular database packaged called Ingres. Ingres started out as an academic project to prove theoretical database concepts about relational database structures. In relational database theory, data is arranged in tables. Tables of data can be connected together by related data. This was a radical idea, compared to the existing types of database models at the time.

A classic example of a relational database is a typical store computer system. This database must contain information on the store's customers, the products it carries, and the current inventory. It must also keep track of orders made by customers. In the past, all of this data would be jumbled together in multiple data files, often duplicating information between the files.

In relational database theory, data is divided into separate groups, called tables. Customer information is stored in the Customer table. The Customer table contains data pertinent to a customer, such as the customer name, address, and billing information. Each customer is assigned a unique ID in the Customer table, with each customer record



being a separate row in the Customer table. Similarly, product data is stored in a separate Product table. The Product table contains detailed information about each product, including a unique product ID, with each product being a separate row of data in the Product table. This is demonstrated in Figure 1-1.

As shown in Figure 1-1, to track orders, database programmers create a separate Order table using the unique IDs from the Customer and Product tables. The Order table relates a customer to the products that are bought. This relationship shows that a single customer can be related to multiple product orders, but each product order belongs to a single customer.

Ingres was one of the first database products available to handle these types of data relationships. With its success, Ingres quickly became a commercial product, and Dr. Stonebraker started working on another database system. Postgres was started in a similar manner as Ingres, attempting to prove the academic theory of object-relational databases.

Object-relational databases take relational databases one step further. In object-oriented programming, data can inherit properties from other data, called a *parent*. The object-oriented principle of inheritance is applied in object-relational databases. Tables can inherit fields from base tables (also called parent tables). For example, a database table of cars can inherit properties (fields) from a parent table of vehicles. This is demonstrated in Figure 1-2.

