24.1 Introduction

A **database** is an organized collection of data. There are many different strategies for organizing data to facilitate easy access and manipulation. A **database management system** (**DBMS**) provides mechanisms for storing, organizing, retrieving and modifying data for many users. Database management systems allow for the access and storage of data without concern for the internal representation of data.

Structured Query Language

Today's most popular database systems are *relational databases* (Section 24.2). A language called **SQL**—pronounced "sequel," or as its individual letters—is the international standard language used almost universally with relational databases to perform **queries** (i.e., to request information that satisfies given criteria) and to manipulate data. [*Note:* As you learn about SQL, you'll see some authors writing "a SQL statement" (which assumes the pronunciation "sequel") and others writing "an SQL statement" (which assumes that the individual letters are pronounced). In this book we pronounce SQL as "sequel."]

Popular Relational Database Management Systems

Some popular proprietary **relational database management systems** (**RDBMSs**) are Microsoft SQL Server, Oracle, Sybase and IBM DB2, PostgreSQL, MariaDB and MySQL are popular *open-source* DBMSs that can be downloaded and used *freely* by anyone. JDK 8 comes with a pure-Java RDBMS called Java DB—the Oracle-branded version of Apache Derby™.

JDBC

Java programs interact with databases using the **Java Database Connectivity (JDBC™) API**. A **JDBC driver** enables Java applications to connect to a database in a particular DBMS and allows you to manipulate that database using the JDBC API.

Software Engineering Observation 24.1

The JDBC API is portable—the same code can manipulate databases in various RDBMSs.

Most popular database management systems provide JDBC drivers. In this chapter, we introduce JDBC and use it to manipulate Java DB databases. The techniques we demonstrate here can be used to manipulate other databases that have JDBC drivers. If not, third-party vendors provide JDBC drivers for many DBMSs.

Java Persistence API (JPA)

In online Chapter 29, we introduce Java Persistence API (JPA). In that chapter, you'll learn how to autogenerate Java classes that represent the tables in a database and the relationships between them—known as object-relational mapping—then use objects of those classes to interact with a database. As you'll see, storing data in and retrieving data from a database will be handled for you—many of the JDBC techniques you learn in this chapter typically are hidden from you by JPA.

JDK 9 Note

9

As of JDK 9, Oracle no longer bundles Java DB with the JDK. If you're using JDK 9 with this chapter, follow the download and installation instructions for Apache Derby at

```
http://db.apache.org/derby/papers/DerbyTut/install_so
```

before proceeding with this chapter’s examples.