2.10. **MySQL**

Windows programs written in other languages can call this API. It's so low-level that even Microsoft encourages developers to use higher-level libraries (e.g., the Database Classes of Microsoft Foundation Classes) that hide the complexities of ODBC. The power of ODBC is due to the facts that

- Despite the complexity of the API, data is accessed simply by passing SQL strings to a specific ODBC function.

- Practically all database systems available for the Windows platform have ODBC drivers. This means that data access works the same whether your data reside in an Access database, an Excel spreadsheet, an Oracle database, or whatever.

- The data source can be decided at run-time. Even the database system to which the ODBC calls are to be directed need not be chosen until runtime. Recall from Figure 2.42 that the executable of embedded SQL program can execute only for a single database system. In fact, the database system needs to be determined at pre-process-time for embedded SQL programs. With programs that access ODBC data sources, the data source can be chosen dynamically at run time. Vendors can build — and sell — applications that access any ODBC source.

We discuss ODBC in the context of Visual Studio programs in Section 2.11.

JDBC is a newer standard API for accessing databases. It's used only in Java programs, but is not limited to Windows environments. It's a higher-level API than ODBC and supports a more flexible architecture for accessing databases on the Web. It shares the advantages described above for the ODBC API.

### 2.10 MySQL

#### 2.10.1 Overview

For those of you who despise Microsoft and its march towards world domination, there are open-source alternatives to Access and Microsoft's other database systems, Foxpro and SQL Server. Perhaps the most popular alternative is MySQL. MySQL is available at no cost for many platforms, including Windows and various flavors of Unix.

MySQL is much more than Access; it's a high-powered multi-user, transaction-based database system. As its name suggests, it has strong support for executing SQL requests, described in Section 2.9. What it doesn't have is a software development environment like that of Access. If you want to develop software that access MySQL databases, you do so using other environments such as Microsoft Visual Studio or IBM's Eclipse. Your application accesses the database by making calls to ODBC or JDBC functions, as described in Section 2.9.6. Or your application can use MySQL's extensive native API.
2.10.2 Installing MySQL

To install MySQL, first download the appropriate zip file from

http://dev.mysql.com/downloads/

Extract the contents of the zip file and run the installation program (SETUP.EXE on Windows). The download includes a manual with extensive documentation about installing and running MySQL.

The MySQL server runs on Windows as a process. To run MySQL, you must start this process. Windows XP and recent variants of Windows NT support processes running as Windows services. On such a machine run the Control Panel Applet named “Services” (on some versions of Windows, found in Control Panel’s Administrative Tools sub-menu). Look for the service named “MySQL” (or perhaps a different name you gave when you installed MySQL), as shown in Figure 2.43. In the figure, MySQL5 is shown as started. If your MySQL service is not started, right-click and select “Start”. The figure also shows the MySQL service (MySQL5) as starting automatically, i.e., the service starts when Windows starts up. You can change your start-up behavior by right-clicking MySQL, selecting and editing “Properties”.

Also available at the MySQL download site is an administrative tool, MySQL Administrator. Once you download and install it, it is available from the Start menu, using menu items similar to

Start → All Programs → MySQL → MySQL Administrator

The tool, shown in Figure 2.44, allows you to inspect and modify many properties of the MySQL server. For example, the “Database” tab of this tool shows you all the databases created in the server so far, and all the relations in each
database. Note that the database named “MySQL” already exists; installing MySQL creates the “MySQL” database for you.

2.10.3 The MySQL Monitor

Now client processes can talk to the server process using TCP/IP network protocol (see 6.1). MySQL provides one useful client: the MySQL monitor. To start this client, select Windows menu items similar to

Start → All Programs → MySQL → MySQL Command Line Client

The monitor, shown in Figure 2.45, supports the execution of SQL and other commands against the MySQL server’s databases.

To see what databases exist in the server, type in the command

show databases;

The monitor displays the database names as a relation:

+-------------------+
| Database           |
+-------------------+
| information_schema |
| mysql             |
| test              |
+-------------------+
3 rows in set (0.00 sec)
To create a new database named “studentdb”, type in the command

```sql
create database studentdb;
```

To work in a specific database, say `studentdb`, type in

```sql
use studentdb;
```

Now you can create relations, insert and delete tuples, and query relations in the `studentdb` database. Just use SQL statements like those shown in Section 2.9. You can also run a script of SQL statements by typing

```sql
source C:\SQLExams\createdb.sql;
```

where the path to the script is given. To create a copy of the student database described earlier in this section, you may download a script from

http://www.cs.uml.edu/~john/books/SoftwareDevelopment/createdbMySQL.sql

## 2.10.4 MySQL’s ODBC Driver

One way to access a MySQL server from your application is via ODBC. To do this, download the Windows ODBC driver for MySQL from the MySQL Web site (http://dev.mysql.com/downloads/). Then install the ODBC driver.

Now create a MySQL database named “student”. Follow these steps to create a DSN (data source name) for the database:

1. Open the Data Sources (ODBC) applet in the Control Panel. On a Windows XP machine, this applet may be located in the “Administrative Tools” folder of the Control Panel.
2. In the User DSN tab of the applet, shown in Figure 2.46, press the “Add” button.

3. Create a User DSN named “studentMySQL” for this database. Select the MySQL ODBC driver, and press finish.

4. Configure the driver as shown in Figure 2.47.

Now that you’ve registered an ODBC driver for the MySQL database, you can access this database from many applications. For example, you can link to one the MySQL’s relations from an Access database, making the MySQL relation available as if it were an ordinary relation in the Access database. Here’s how:

1. Open an Access database.

2. Select “Tables” in the main database dialog box.

3. Right-click on the dialog box and select “Link Tables...” from the context menu.

4. In the “Files of Type” text box of the Link dialog box, select ODBC Databases. Select the MySQL data source and press the OK button.

5. In the resulting MySQL ODBC driver configuration dialog box, fill in any missing connection information.

6. Close the dialog box. A new dialog box should appear, asking which tables you’d like to make links to. Choose some.
7. Close the dialog box. Now new, linked tables should appear in Access’s Tables tab. You should be able to use the tables in Access as if they were ordinary Access tables.

2.11 Accessing Databases from Visual Studio .NET Programs

We’ve already mentioned that you can access databases by ODBC calls. Visual Studio .NET provides higher-level abstractions for accessing databases from your code:

- **DataSet** an in-memory copy of some database (or other) data
- **DataAdaptor** manager of the translation of data from a database (or other source) and a DataSet, or vice versa
- **DataConnection** a connection to a particular database

These abstractions are the heart of the ADO.NET data-access framework, available in Visual Studio. Each of these abstractions corresponds to a Visual Studio toolbox component that you can drag into your application. To demonstrate, let’s create an application that displays results from a database.

First you need to set up a data source. To do that in MySQL, download from www.mysql.com the MySQL Connector/ODBC, as described in Section 2.10.4. Create a database called “studentdb”. Download the SQL script at

http://www.cs.uml.edu/~john/books/SoftwareDevelopment/createdbMySQL.sql