

Chapter 13

An introduction to database programming

Murach's Visual Basic 2008, C13

© 2008, Mike Murach & Associates, Inc.

Slide 1

Objectives

Knowledge

- Describe the hardware components of a typical multi-user system.
- Describe the software components of a typical multi-user database application.
- Explain how a table in a relational database is organized.
- Explain how the tables in a relational database are related.
- Describe the use of these SQL statements: Select, Insert, Update, Delete.
- Describe the use of these ADO.NET components: data adapter, data command, connection, data reader, dataset, data table.
- Compare the structure of a dataset with the structure of a relational database.

Murach's Visual Basic 2008, C13

© 2008, Mike Murach & Associates, Inc.

Slide 2

Objectives (continued)

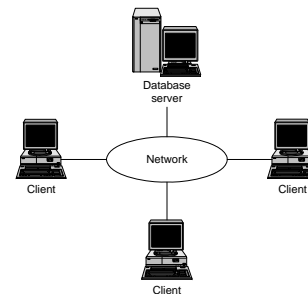
- Describe concurrency, optimistic concurrency control, and "last in wins."
- Describe two ways that you can create ADO.NET components.

Murach's Visual Basic 2008, C13

© 2008, Mike Murach & Associates, Inc.

Slide 3

A simple client/server system



Murach's Visual Basic 2008, C13

© 2008, Mike Murach & Associates, Inc.

Slide 4

The three hardware components of a client/server system

- The *clients* are the PCs, Macintoshes, or workstations of the system.
- The *server* is a computer that stores the files and databases of the system and provides services to the clients. When it stores databases, it's often referred to as a *database server*.
- The *network* consists of the cabling, communication lines, and other components that connect the clients and the servers of the system.

Murach's Visual Basic 2008, C13

© 2008, Mike Murach & Associates, Inc.

Slide 5

Client/server system implementations

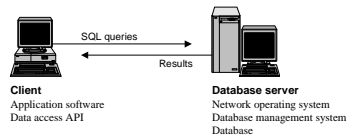
- In a simple *client/server system*, the server is typically a high-powered PC that communicates with the clients over a *local area network (LAN)*.
- The server can also be a midrange system, like an IBM iSeries or a Unix system, or it can be a mainframe system. Then, special hardware and software components are required to make it possible for the clients to communicate with the midrange and mainframe systems.
- A client/server system can also consist of one or more PC-based systems, one or more midrange systems, and a mainframe system in dispersed geographical locations. This type of system is commonly referred to as an *enterprise system*.
- Individual systems and LANs can be connected and share data over larger private networks, such as a *wide area network (WAN)* or a public network like the Internet.

Murach's Visual Basic 2008, C13

© 2008, Mike Murach & Associates, Inc.

Slide 6

Client software, server software, and the SQL interface



Murach's Visual Basic 2008, C13

© 2008, Mike Murach & Associates, Inc.

Slide 7

Server software

- To manage the network, the server runs a *network operating system* such as Windows Server 2003.
- To store and manage the databases of the client/server system, each server requires a *database management system (DBMS)* such as Microsoft SQL Server.
- The processing that's done by the DBMS is typically referred to as *back-end processing*, and the database server is referred to as the *back end*.

Murach's Visual Basic 2008, C13

© 2008, Mike Murach & Associates, Inc.

Slide 8

Client software

- The *application software* does the work that the user wants to do. This type of software can be purchased or developed.
- The *data access API (application programming interface)* provides the interface between the application and the DBMS. The newest data access API is ADO.NET 3.5, which is a part of Microsoft's .NET Framework.
- The processing that's done by the client software is typically referred to as *front-end processing*, and the client is typically referred to as the *front end*.

Murach's Visual Basic 2008, C13

© 2008, Mike Murach & Associates, Inc.

Slide 9

The SQL interface

- The application software communicates with the DBMS by sending *SQL queries* through the data access API. When the DBMS receives a query, it provides a service like returning the requested data (the *query results*) to the client.
- *SQL*, which stands for *Structured Query Language*, is the standard language for working with a relational database.

Murach's Visual Basic 2008, C13

© 2008, Mike Murach & Associates, Inc.

Slide 10

Client/server versus file-handling systems

- In a client/server system, the processing done by an application is typically divided between the client and the server.
- In a file-handling system, all of the processing is done on the clients. Although the clients may access data that's stored in files on the server, none of the processing is done by the server. As a result, a file-handling system isn't a client/server system.

Murach's Visual Basic 2008, C13

© 2008, Mike Murach & Associates, Inc.

Slide 11

The Products table in the MMABooks database

ProductCode	Description	UnitPrice	OrderQuantity
A2C#	Murach's ASP.NET 2.0 Web Programming with C# 2005	52.5000	4637
A2VB	Murach's ASP.NET 2.0 Web Programming with VB 2005	52.5000	3974
ADV2	Murach's ADO.NET 2.0 Database Programming VB 2005	52.5000	4530
CRFC	Murach's CICS Desk Reference	50.0000	1865
DB1R	DB2 for the COBOL Programmer, Part 1 (2nd Edition)	42.0000	4825
DB2R	DB2 for the COBOL Programmer, Part 2 (2nd Edition)	45.0000	621
JSE6	Murach's JAVA SE 6	52.5000	3485
MC#5	Murach's C# 2005	52.5000	5136
MC2L	Murach's Structured COBOL	62.5000	2366
MCCP	Murach's CICS for the COBOL Programmer	54.0000	2368
HTSP	Murach's JAVA Servlets and JSP	49.5000	4999
MVB5	Murach's Visual Basic 2005	52.5000	0
SQLS	Murach's SQL Server 2005	52.5000	2465
ZLR	Murach's OS/390 and z/os JCL	62.5000	677

Murach's Visual Basic 2008, C13

© 2008, Mike Murach & Associates, Inc.

Slide 12

How a table is organized

- A relational database uses tables to store and manipulate data:
 - Each table consists of one or more records, or rows, that contain the data for a single entry.
 - Each row contains one or more fields, or columns, with each column representing a single item of data.
- Most tables contain a primary key that uniquely identifies each row in the table.
- Some database management systems let you define one or more non-primary keys. In SQL Server, these keys are called unique keys, and they're implemented using unique key constraints.
- A table can also be defined with one or more indexes. An index provides an efficient way to access data from a table based on the values in specific columns. An index is automatically created for a table's primary and non-primary keys.

Two related tables: Invoices and InvoiceLineItems

InvoiceID	CustomerID	InvoiceDate	ProductTotal	SalesTax	Shipping	InvoiceTotal
41	333	4/13/2007 12:00:00	62.5000	4.6900	3.7500	70.9400
42	666	4/13/2007 12:00:00	62.5000	4.6900	3.7500	70.9400
43	332	4/13/2007 12:00:00	52.5000	3.9375	3.7500	60.1875
44	995	4/13/2007 12:00:00	62.5000	4.6900	3.7500	70.9400
45	213	4/13/2007 12:00:00	62.5000	4.6900	3.7500	70.9400
46	20	4/13/2007 12:00:00	212.5000	15.9375	7.5000	235.9375
47	10	4/13/2007 12:00:00	242.5000	18.1900	8.7500	269.4400

InvoiceID	ProductCode	UnitPrice	Quantity	ItemTotal
41	A208	62.5000	1	62.5000
42	A208	62.5000	1	62.5000
43	208	52.5000	1	52.5000
44	A208	62.5000	1	62.5000
45	A208	62.5000	1	62.5000
46	A208	62.5000	1	62.5000
46	0638	45.0000	1	45.0000
46	M085	52.5000	1	52.5000
46	5055	52.5000	1	52.5000
47	A208	62.5000	1	62.5000
47	0638	45.0000	4	180.0000

How the tables in a database are related

- The tables in a relational database are related to each other through their key columns. A column that identifies a related row in another table is called a foreign key.
- Usually, a foreign key corresponds to the primary key in the related table. In SQL Server, however, a foreign key can also correspond to a unique key in the related table.
- When two tables are related via a foreign key, the table with the foreign key is referred to as the foreign key table and the table with the primary key is referred to as the primary key table.
- The relationships between the tables in a database correspond to the relationships between the entities they represent. The most common type of relationship is a one-to-many relationship. A table can also have a one-to-one relationship or a many-to-many relationship with another table.

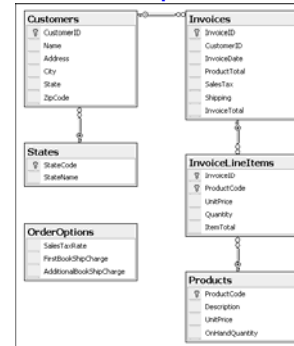
The Server Explorer design view window for the Products table



Common SQL Server data types

- bit
- char, varchar, text
- datetime, smalldatetime
- decimal, numeric
- float, real
- bigint, int, smallint, tinyint
- money, smallmoney

The tables that make up the MMABooks database



Simplified syntax of the Select statement

```

Select column-1 [, column-2]...
From table-1
[Where selection-criteria]
[Order By column-1 [Asc|Desc] [, column-2 [Asc|Desc]]...]
    
```

A Select statement that retrieves and sorts selected columns and rows from the Customers table

```

Select Name, City
From Customers
Where State = 'WA'
Order By Name
    
```

The result set defined by the Select statement

Name	City
Allen, Craig	Pullman
Antalocy, S.	Seattle
Cassara, Glenn	Van
Graka, Eric	Seattle
Hester, Maurice	Kennewickm
Howell, Kim	Renkon
Millen, G	Kirkland
Millard, Dwayne	Seattle
Ornel, Ri	Olympia
Seaver, Glenda	Mountlake Terrace
Smith, Lloyd	Pullman
Sunderam, Kelly	Oregon

How to query a single table

- The result of a Select statement is a *result table*, or *result set*.
- The Select clause lists the columns to be included in the result set. This list can include *calculated columns* that are calculated from other columns.
- The From clause names the table the data will be retrieved from.
- The Where clause specifies which rows should be retrieved. To retrieve all rows from a table, omit the Where clause.
- The Order By clause lists the columns that the results are sorted by and indicates whether each column is sorted in ascending or descending sequence.
- To select all of the columns in a table, you can code an asterisk (*) in place of the column names:
Select * From Customers

The syntax of the Select statement for joining two tables

```

Select column-list
From table-1
    [Inner] Join table-2
    On table-1.column-1 {=|<|>|<=|>=|<>} table-2.column-2
[Where selection-criteria]
[Order By column-list]
    
```

A Select statement that joins data from the InvoiceLineItems and Products tables

```

Select InvoiceLineItems.InvoiceID,
InvoiceLineItems.ProductCode,
Products.Description, InvoiceLineItems.UnitPrice,
InvoiceLineItems.Quantity
From InvoiceLineItems
Inner Join Products
    On Products.ProductCode =
    InvoiceLineItems.ProductCode
Where InvoiceID = 46
    
```

The result set defined by the Select statement

InvoiceID	ProductCode	Description	UnitPrice	Quantity
46	AZVB	Murach's ASP.N...	62.5000	1
46	DECR	DE2 For the COB...	45.0000	1
46	MHBS	Murach's Visual ...	52.5000	1
46	SQLS	Murach's SQL Se...	52.5000	1

How to join data from two or more tables

- A *join* lets you combine data from two or more tables into a single result set.
- The most common type of join is an *inner join*. This type of join returns rows from both tables only if their related columns match.

How to add a single row to a table

The syntax of the Insert statement for adding a single row

```
Insert [Into] table-name [(column-list)]
    Values (value-list)
```

A statement that adds a single row to the Products table

```
Insert Into Products (ProductCode, Description, UnitPrice,
    OnHandQuantity)
    Values ('MVB8', 'Murach's Visual Basic 2008', 54.50,
    3000)
```

How to update rows in a table

The syntax of the Update statement

```
Update table-name
    Set expression-1 [, expression-2]...
    [Where selection-criteria]
```

A statement that updates the UnitPrice column for a specified product

```
Update Products
    Set UnitPrice = 54.00
    Where ProductCode = 'MCCP'
```

How to delete rows from a table

The syntax of the Delete statement

```
Delete [From] table-name
    [Where selection-criteria]
```

A statement that deletes a specified customer

```
Delete From Customers
    Where CustomerID = 558
```

.NET data provider core objects

Object	Description
Connection	Establishes a connection to a database.
Command	Represents an individual SQL statement that can be executed against the database.
Data reader	Provides read-only, forward-only access to the data in a database.
Data adapter	Provides the link between the command and connection objects and a dataset object.

Data providers included with the .NET Framework

Provider	Namespace	Lets you access...
SQL Server	System.Data.SqlClient	SQL Server databases
OLE DB	System.Data.OleDb	Any database that supports OLE DB
ODBC	System.Data.Odbc	Any database that supports ODBC
Oracle	System.Data.OracleClient	Oracle databases

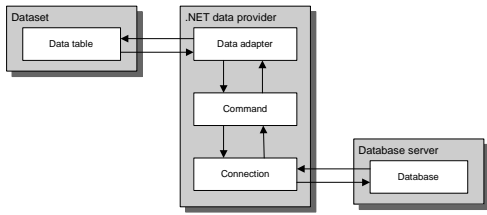
Class names for the data provider objects

SQL Server	OLE DB	ODBC	Oracle
SqlConnection	OleDbConnection	OdbcConnection	OracleConnection
SqlCommand	OleDbCommand	OdbcCommand	OracleCommand
SqlDataReader	OleDbDataReader	OdbcDataReader	OracleDataReader
SqlDataAdapter	OleDbDataAdapter	OdbcDataAdapter	OracleDataAdapter

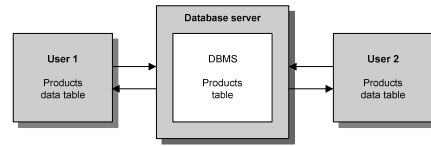
An Imports statement for the SQL Server data provider namespace

```
Imports System.Data.SqlClient
```

Basic ADO.NET components



Two users who are working with copies of the same data



Concurrency concepts

- When two or more users retrieve the data in the same row of a database table at the same time, it is called *concurrency*. Because ADO.NET uses a disconnected data architecture, the database management system can't prevent this from happening.
- If two users try to update the same row in a database table at the same time, the second user's changes could overwrite the changes made by the first user. Whether or not that happens depends on the *concurrency control* that the programs use.
- By default, ADO.NET uses *optimistic concurrency*. This means that the program checks to see whether the database row that's going to be updated or deleted has been changed since it was retrieved. If it has, a *concurrency exception* occurs and the update or deletion is refused. Then, the program should handle the exception.

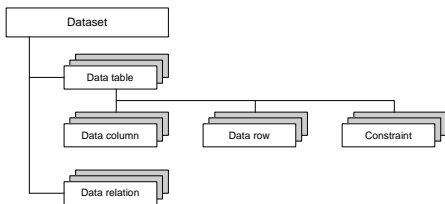
Concurrency concepts (cont.)

- If optimistic concurrency isn't in effect, the operation proceeds without throwing an exception. This is referred to as "*last in wins*" because the last update overwrites any previous update. And this can lead to errors in the database.

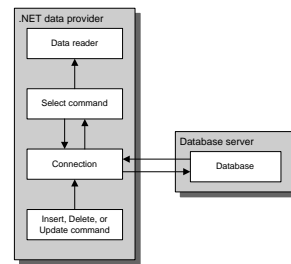
How to avoid concurrency errors

- For many applications, concurrency errors rarely occur. As a result, optimistic concurrency is adequate.
- If concurrency is likely to be a problem, a program can be designed so it updates the database and refreshes the dataset frequently. That way, concurrency errors are less likely to occur.
- Another alternative is to design a program so it retrieves and updates just one row at a time. That way, there's less chance that two users will retrieve and update the same row at the same time.

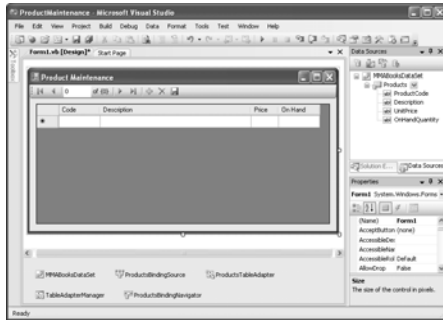
The basic dataset object hierarchy



ADO.NET components for accessing a database directly



ADO.NET objects created using the Data Sources window



Murach's Visual Basic 2008, C13

© 2008, Mike Murach & Associates, Inc.

Slide 37

ADO.NET objects created using code

```

Dim connectionString As String _
    = "Data Source=localhost\SqLExpress;" _
      & "Initial Catalog =MMABooks;" _
      & "Integrated Security=True"
Dim connection As New SqlConnection(connectionString)

Dim selectStatement As String _
    = "SELECT * FROM Products ORDER BY ProductCode"
Dim selectCommand As New SqlCommand( _
    selectStatement, connection)

Dim productsDataAdapter As New SqlDataAdapter(selectCommand)

Dim productsDataSet As DataSet = New DataSet
  
```

Murach's Visual Basic 2008, C13

© 2008, Mike Murach & Associates, Inc.

Slide 38