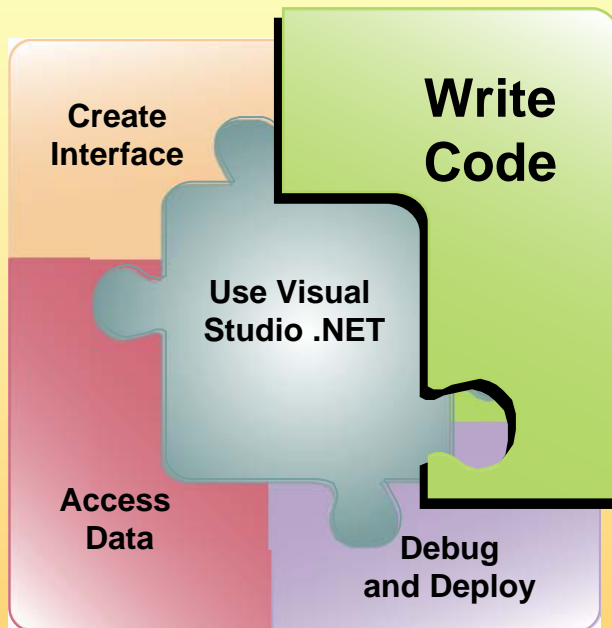


Module 11: Using ADO.NET

Overview



- Database Concepts
- Overview of ADO.NET
- Working with Data

Lesson: Database Concepts

- Database Terminology
- How Database Programming Works
- What Is SQL?
- What Is a Connected Environment?
- What Is a Disconnected Environment?

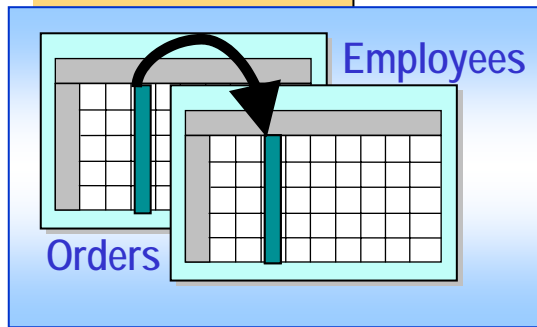
Database Terminology

Employees Table

Employee ID	Last name	First name
3	Small	Tony
5	Smith	James
▪	▪	▪
▪	▪	▪
▪	▪	▪

Rows
(records)

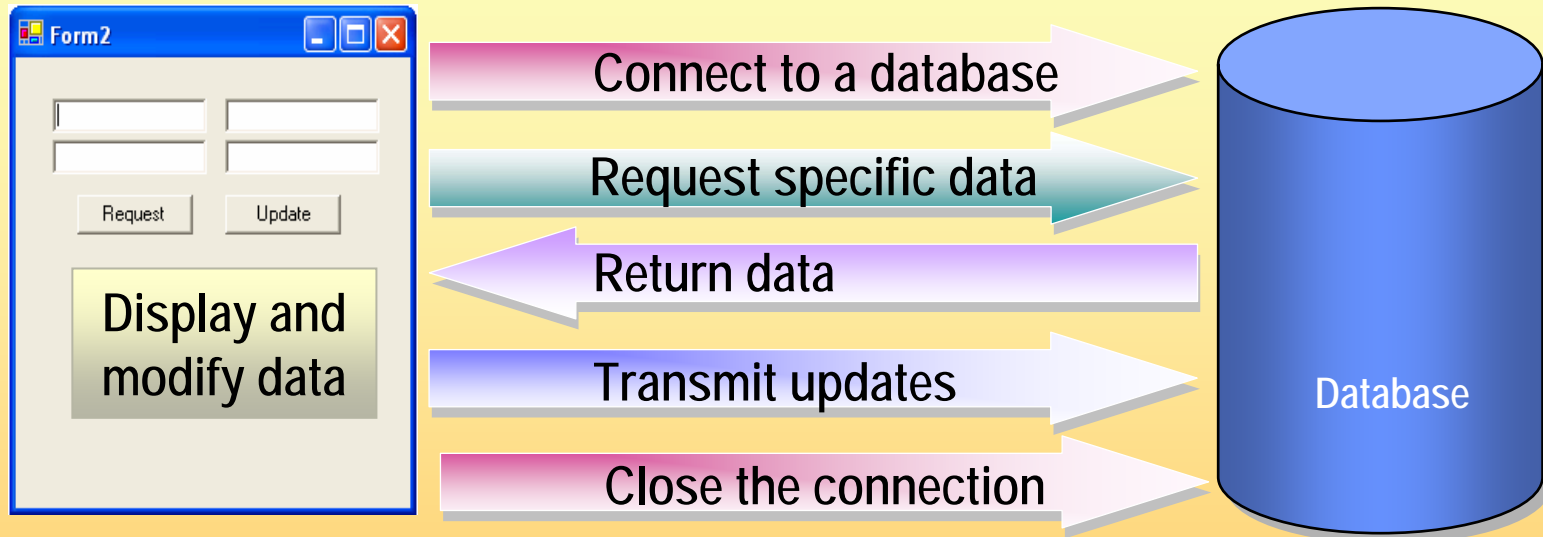
Relationships



Columns (fields)

How Database Programming Works

Common Tasks in Database Programming



In many applications, the connection is closed after the user accesses the data and reopened when the user transmits updates or makes further requests

What Is SQL?

Definition: SQL is an industry-standard language that has evolved into the most widely accepted means to query and modify data in a database

■ Syntax for common SQL statements

- To specify exactly which records you want to retrieve, use
`SELECT Field FROM Table`
- To limit the selection of records, use
`SELECT * FROM Table WHERE Field = "String"`
- To return records in ascending order, use
`SELECT * FROM Table ORDER BY Field ASC`

■ Example

```
SELECT FirstName FROM Employees
```

What Is a Connected Environment?

- A connected environment is one in which users are continuously connected to a data source
- Advantages:
 - Environment is easier to maintain
 - Concurrency is more easily controlled
 - Data is more likely to be current than in other scenarios
- Disadvantages:
 - Must have a constant network connection
 - Limited scalability

What Is a Disconnected Environment?

- A disconnected environment is one in which data can be modified independently and changes are written to the database later
- Advantages:
 - Connections are used for the shortest possible time, allowing fewer connections to serve more users
 - A disconnected environment improves the scalability and performance of applications
- Disadvantages:
 - Data is not always up-to-date
 - Change conflicts can occur and must be resolved

Practice: Identifying Connected or Disconnected Data Scenarios

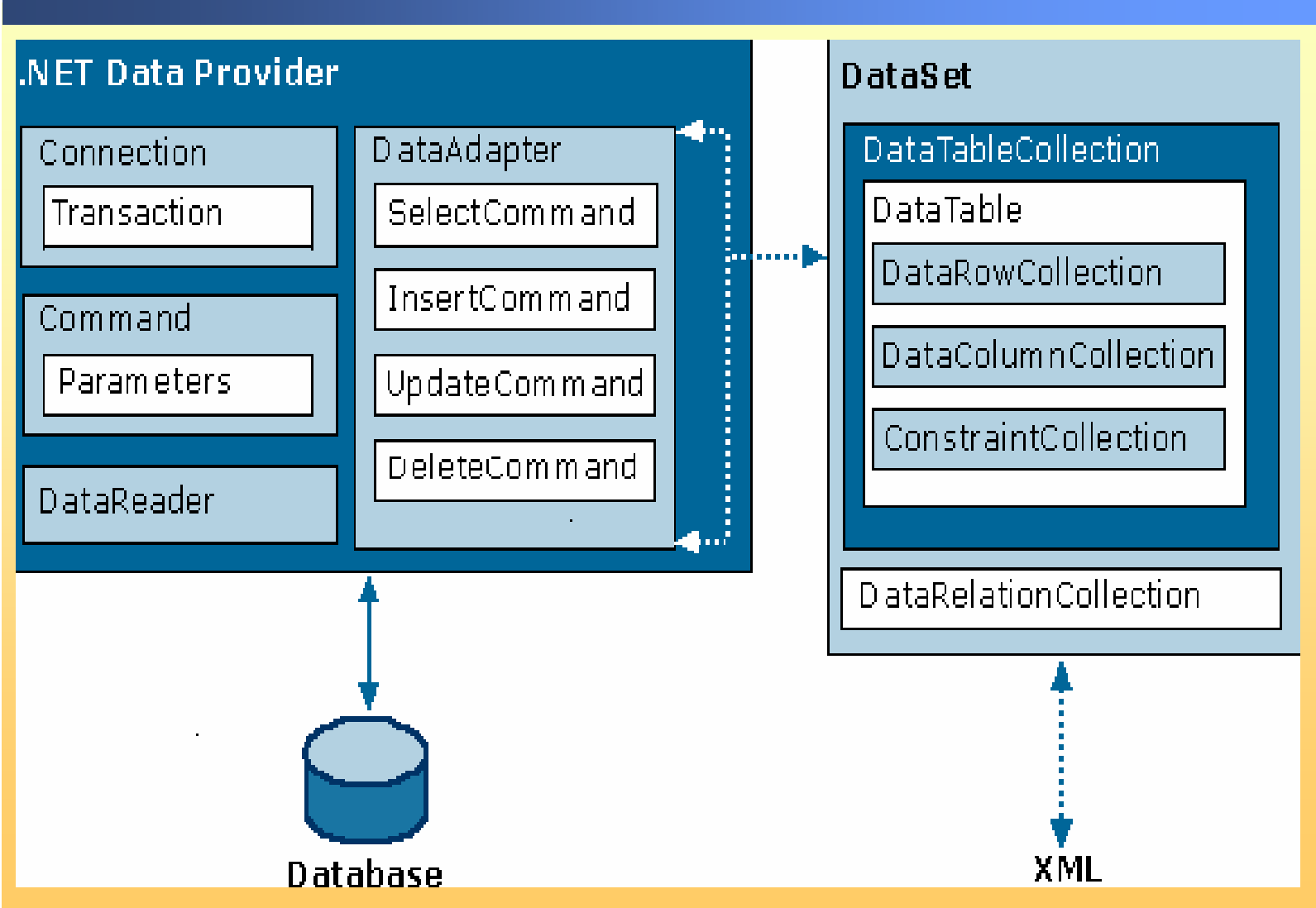


- In this practice, you will:
 - Work in pairs to analyze five business scenarios in which data access is required
 - For each scenario, choose a connected or disconnected environment, depending on the application requirements
 - Determine whether read-only or read/write access is required

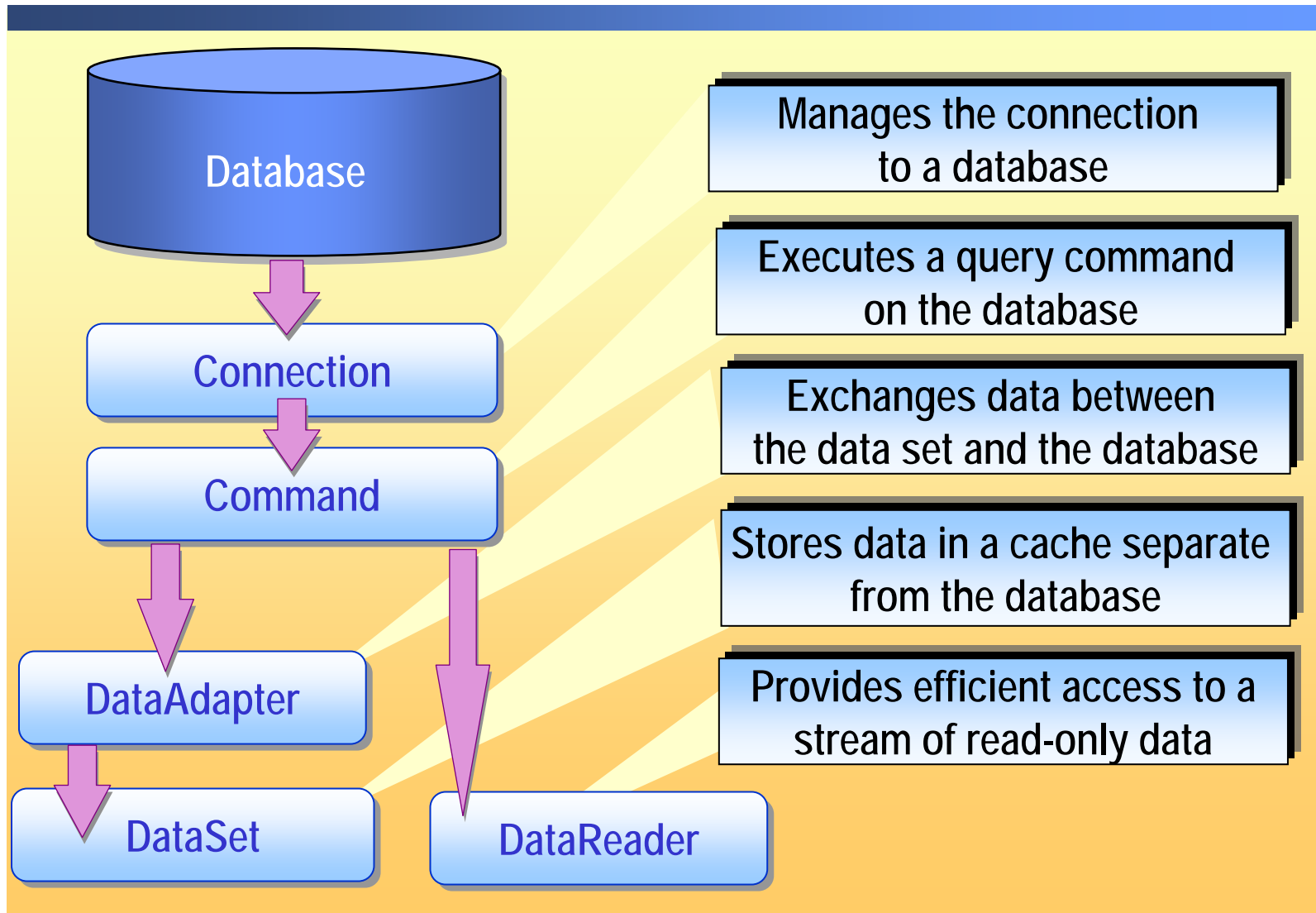
Lesson: Overview of ADO.NET

- What Is ADO .NET?
- Common ADO.NET Objects
- How to Work with Databases in Server Explorer
- ADO.NET and XML

What Is ADO.NET?



Common ADO.NET Objects

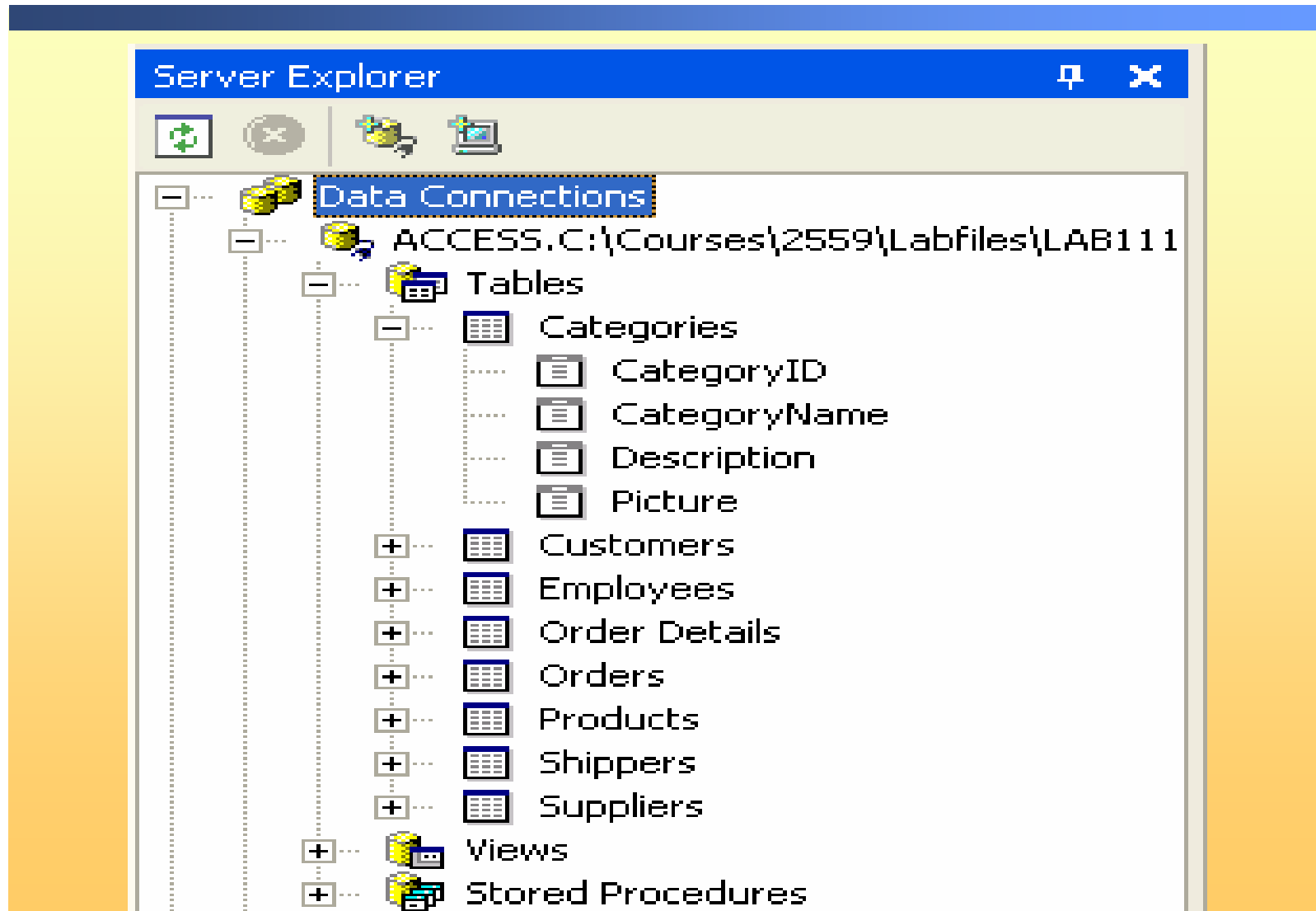


Demonstration: Using Server Explorer



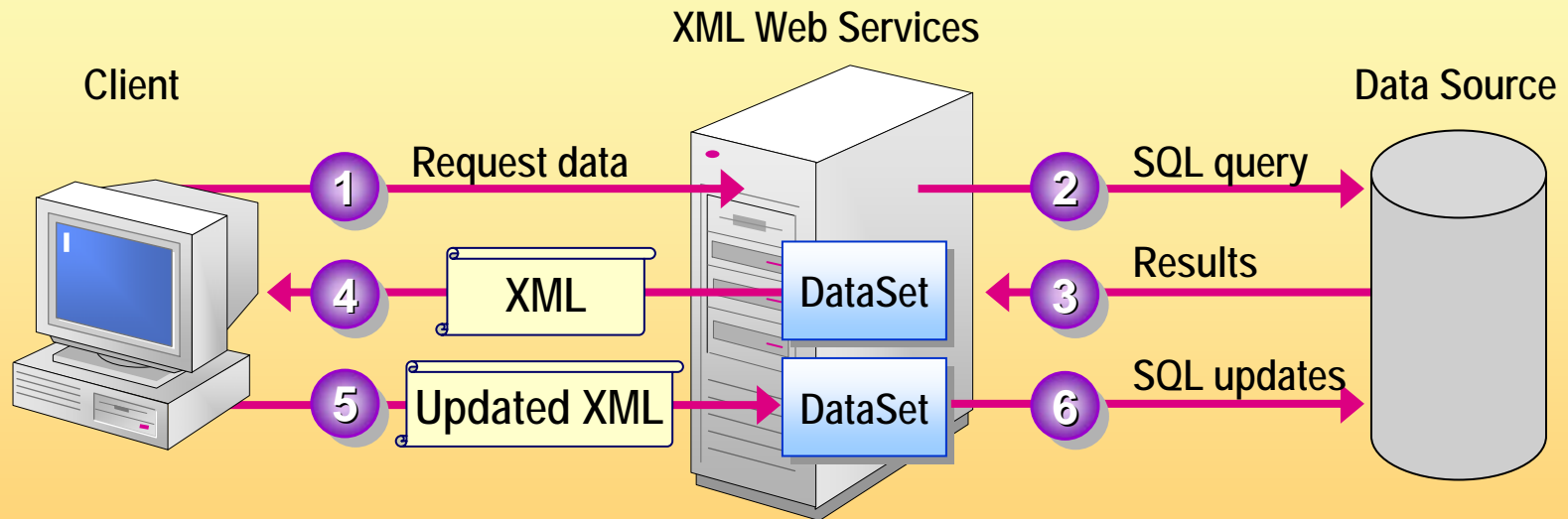
- In this demonstration, you will learn how to use Server Explorer to add database connections and view database elements such as tables

How to Work with Databases in Server Explorer



ADO.NET and XML

- ADO.NET is closely integrated with XML



Example of Using XML in a Disconnected ADO.NET Application

Practice: Accessing Read-Only Data



- In this practice, you will:
 1. Start a new Windows Forms application
 2. Choose the connection type and database
 3. Add the **Connection** and the **DataAdapter** to the form
 4. Generate the **DataSet**
 5. Add a **DataGrid** control to the form and set its properties
 6. Use the **Fill** method to populate the **DataSet**
 7. Run the application to view read-only data

Lesson: Working with Data

- How to Use a Connection Object
- How to Use a DataAdapter Object
- How to Use a DataSet Object
- How to Use a DataGrid Control
- How to Use the Data Form Wizard

How to Use a Connection Object

- You use Connection to:
 - Choose the connection type
 - Specify the data source
 - Open the connection to the data source
 - Close the connection to the data source
- Example of connection to a SQL Server database

```
Dim PubsSQLConn As SqlClient.SqlConnection
PubsSQLConn = New SqlClient.SqlConnection( )
PubsSQLConn.ConnectionString = "Integrated Security=True;" & _
    "Data Source=localhost;Initial Catalog=Pubs;"
PubsSQLConn.Open( )
```

How to Use a DataAdapter Object

- To create a DataAdapter

- Declare with the **Dim** keyword
- Pass a query string and a **Connection** object as parameters

```
Dim PubsAdapter As SqlDataAdapter = New SqlDataAdapter _  
    ("Select * from Titles", PubsSQLConn)
```

- Key methods of DataAdapter:

- **Fill** method populates a data set
- **Update** method transmits changes to the data store

How to Use a DataSet Object

■ How DataSets work

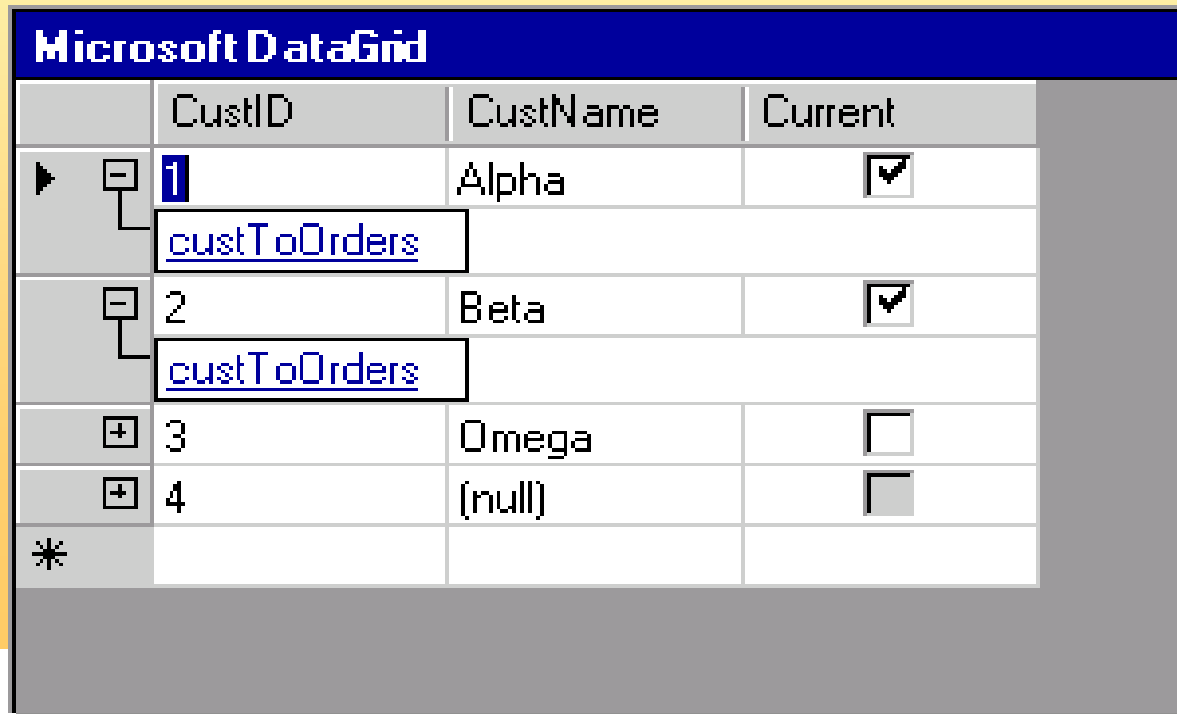
- Store data in a disconnected cache
- Use a hierarchical object model of tables, rows, and columns

■ You can populate a DataSet by:

- Using the **Fill** method
- Manually populating the tables
- Reading an XML document or stream
- Merging or copying the contents of another **DataSet**

How to Use a DataGrid Control

- Use a DataGrid control to display data from a single data source or multiple data sources
 - From the Toolbox, add a **DataGrid** control to a form
 - Set the properties of the **DataGrid** control



The screenshot shows a Microsoft DataGrid control with a blue header bar. The table has three columns: CustID, CustName, and Current. The first two rows are expanded, showing a sub-table of 'custToOrders' for each customer. The third and fourth rows are collapsed. The first row has CustID 1, CustName Alpha, and Current checked. The second row has CustID 2, CustName Beta, and Current checked. The third row has CustID 3, CustName Omega, and Current unchecked. The fourth row has CustID 4, CustName (null), and Current unchecked. A '*' symbol is in the first column of the fifth row.

	CustID	CustName	Current
▶ -	1	Alpha	<input checked="" type="checkbox"/>
	custToOrders		
-	2	Beta	<input checked="" type="checkbox"/>
	custToOrders		
+	3	Omega	<input type="checkbox"/>
+	4	(null)	<input type="checkbox"/>
*			

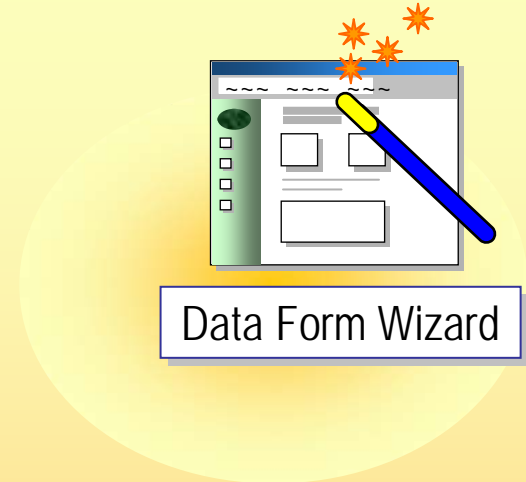
How to Use the Data Form Wizard

■ How to open the Data Form Wizard

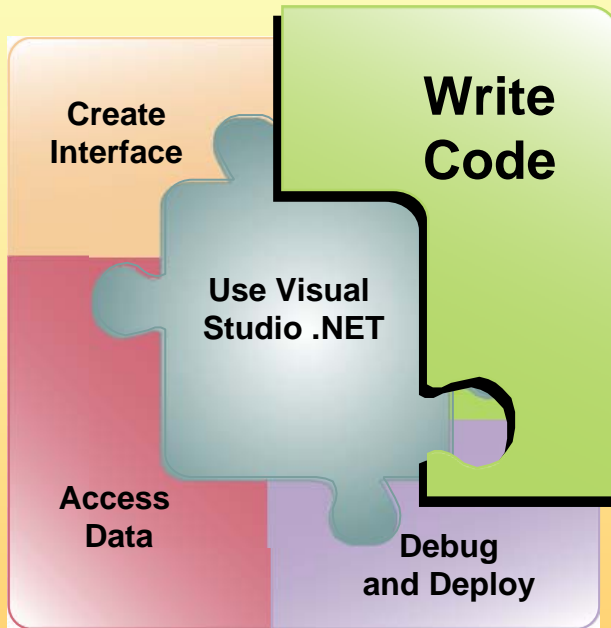
- On the **File** menu, click **Add New Item**
- In the **Templates** pane, select **Data Form Wizard**
- Follow the steps in the wizard

■ The Data Form Wizard provides:

- A Windows form with data-bound controls
- All necessary data components for the form
- Methods that you can call to read and write data

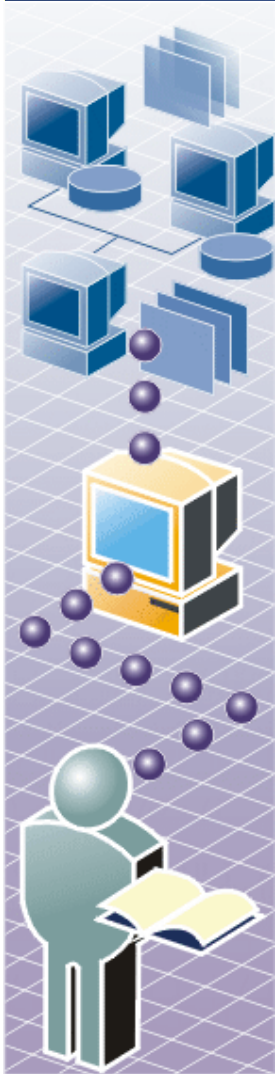


Review



- Database Concepts
- Overview of ADO.NET
- Working with Data

Lab 11.1: Accessing Data with ADO.NET



- Exercise 1: Using the Data Form Wizard