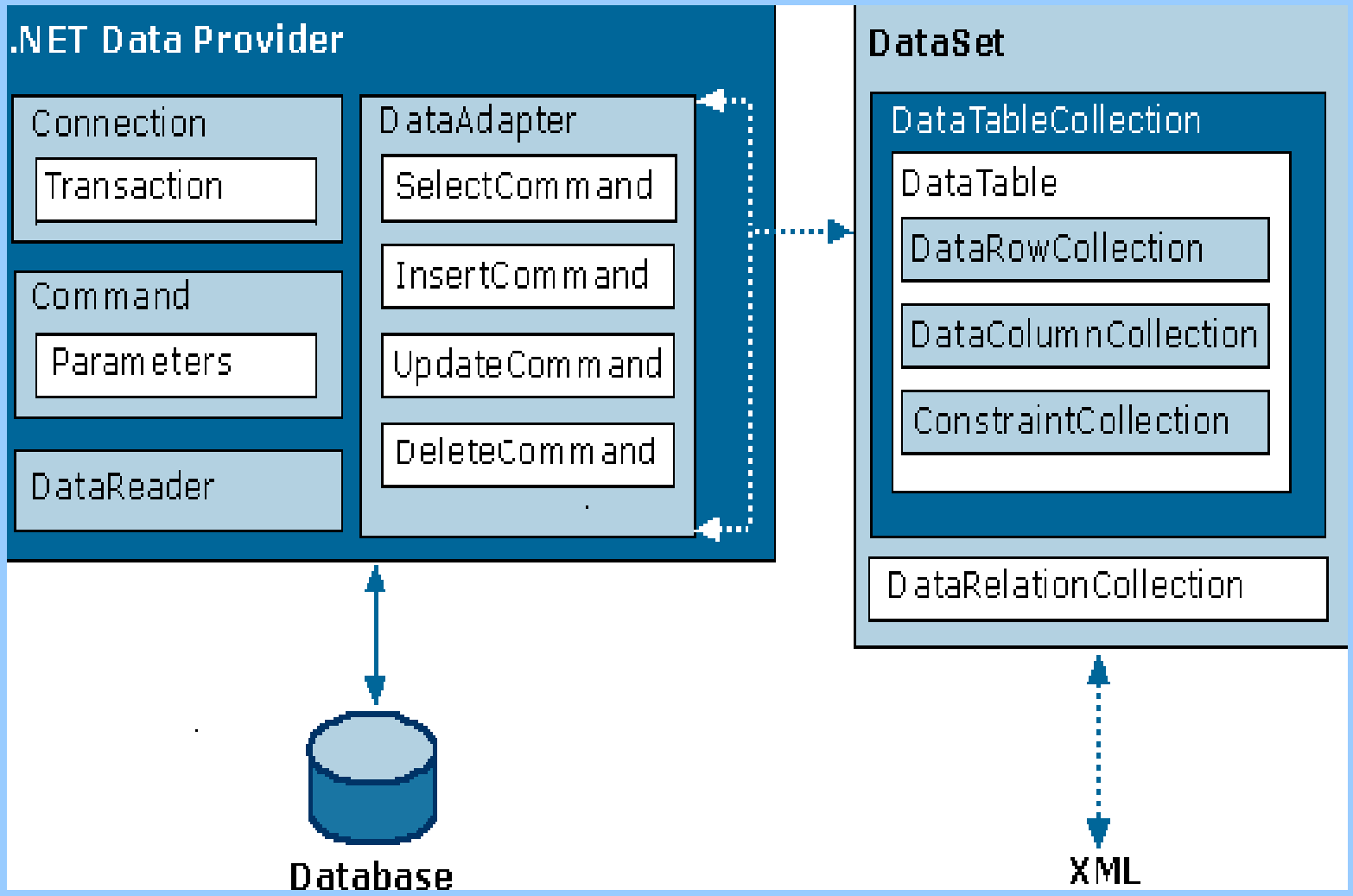
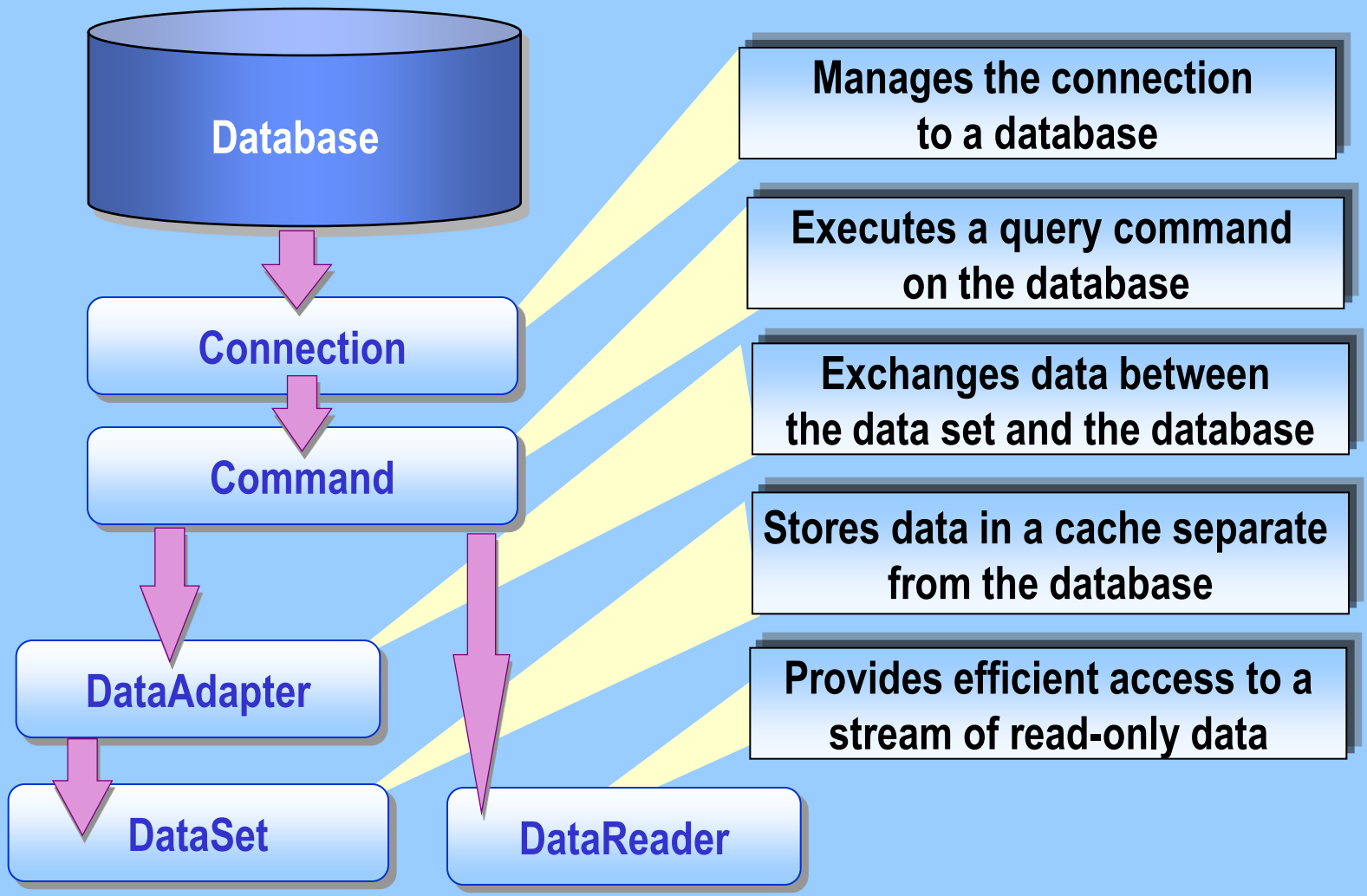
Accessing Relational Data Using Microsoft Visual Studio .NET

What Is ADO.NET?

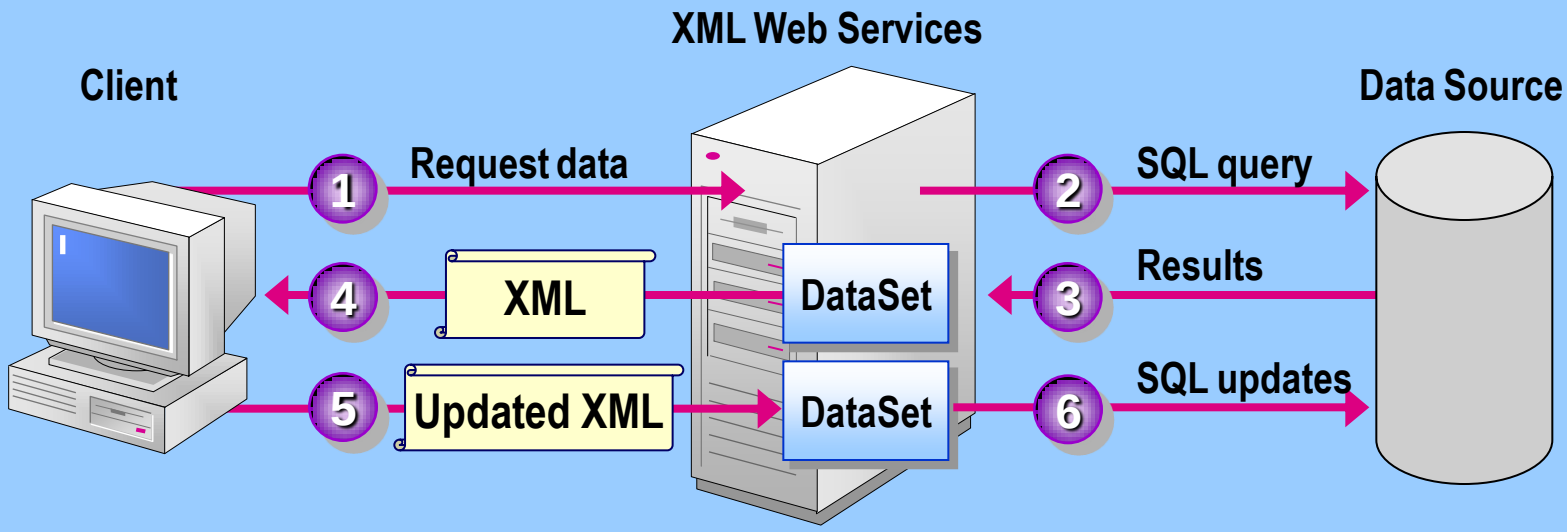
- **ADO.NET provides a set of classes for working with data**
- **ADO.NET is:**
  - An evolutionary, more flexible successor to ADO
  - A system designed for disconnected environments
- **ADO.NET provides:**
  - A programming model with advanced XML support
  - A set of classes, interfaces, structures, and enumerations that manage data access from within the .NET Framework

What Is ADO.NET?




Database
Connection
Command
DataAdapter
DataSet
DataReader
Manages the connection to a database
Executes a query command on the database
Exchanges data between the data set and the database
Stores data in a cache separate from the database
Provides efficient access to a stream of read-only data

- ADO.NET is tightly integrated with XML



XML in a disconnected ADO.NET application

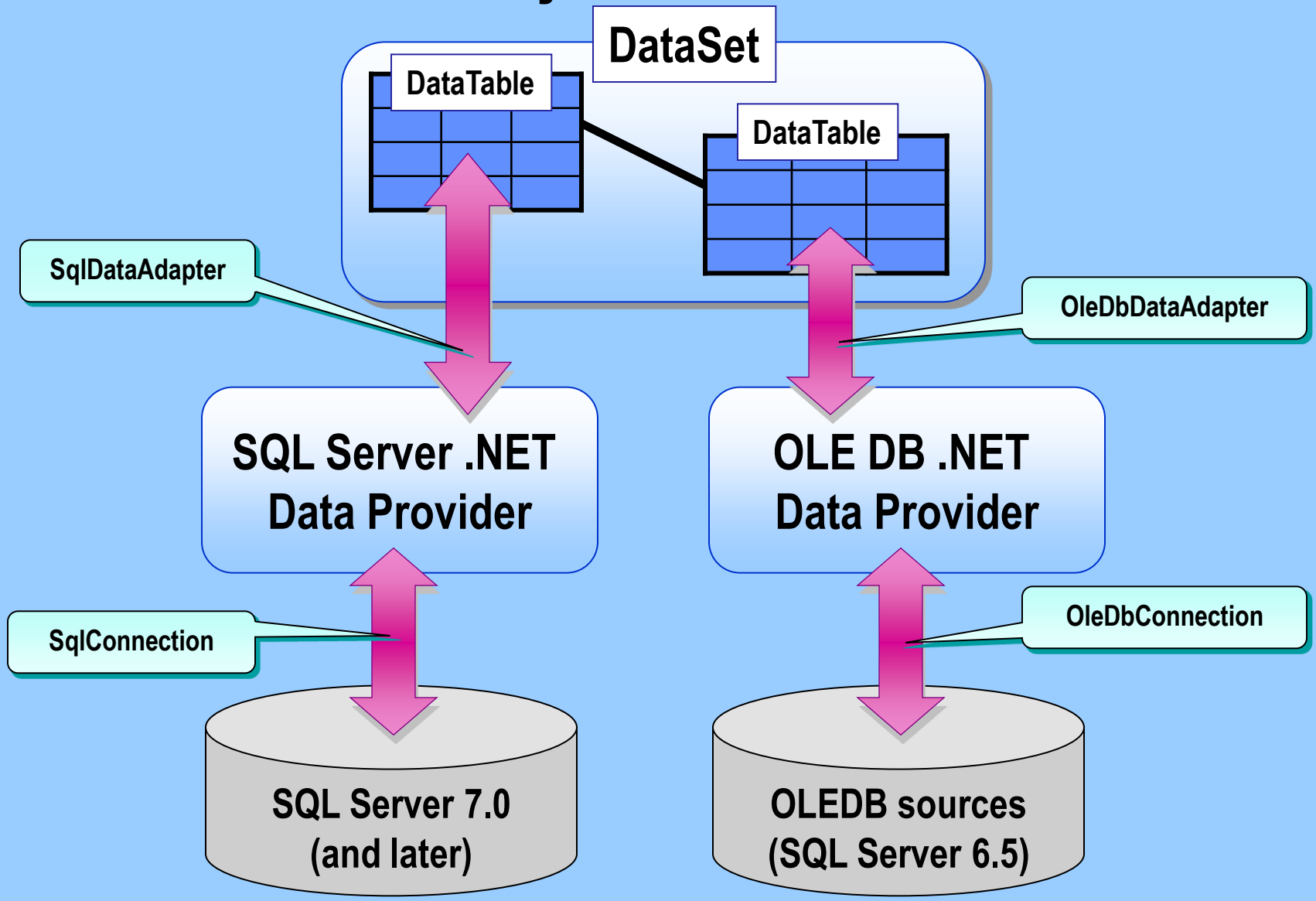
Using Namespaces

- Use the Imports statement to import namespaces

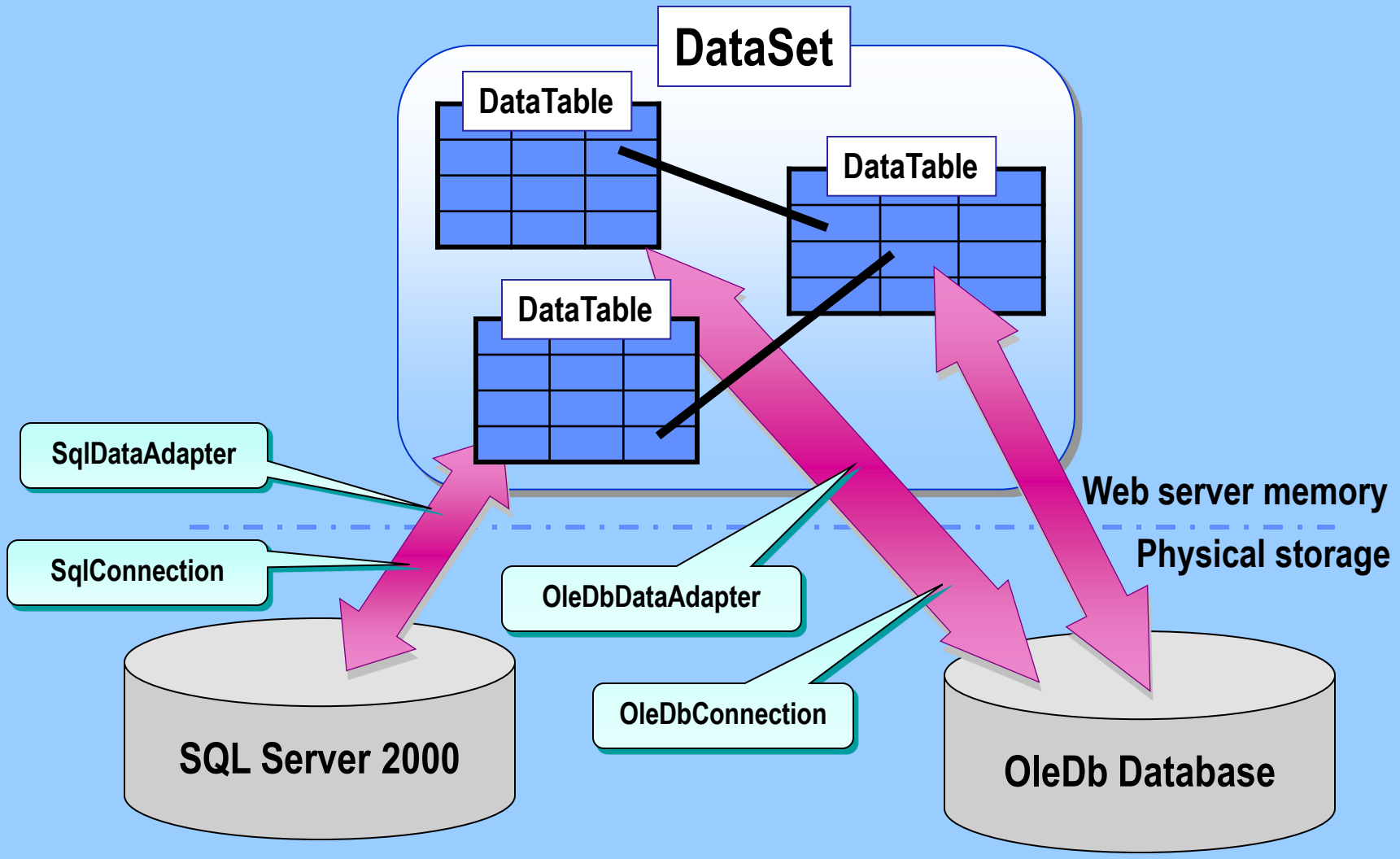
```
Imports System.Data
Imports System.Data.SqlClient
```

- Namespaces used with ADO.NET include:
  - System.Data
  - System.Data.SqlClient
  - System.Data.OleDb

The ADO.NET Object Model

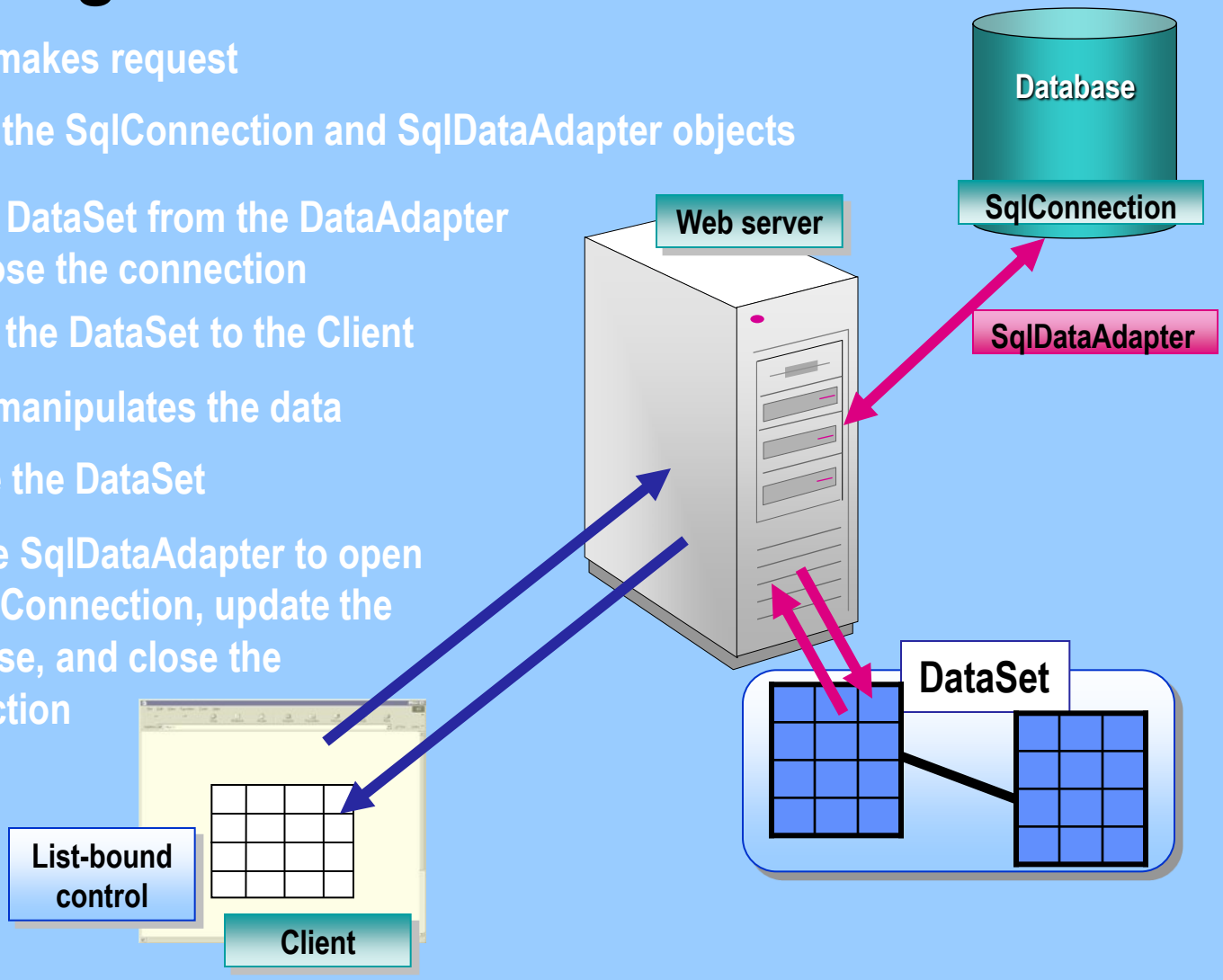


What Is a Dataset?



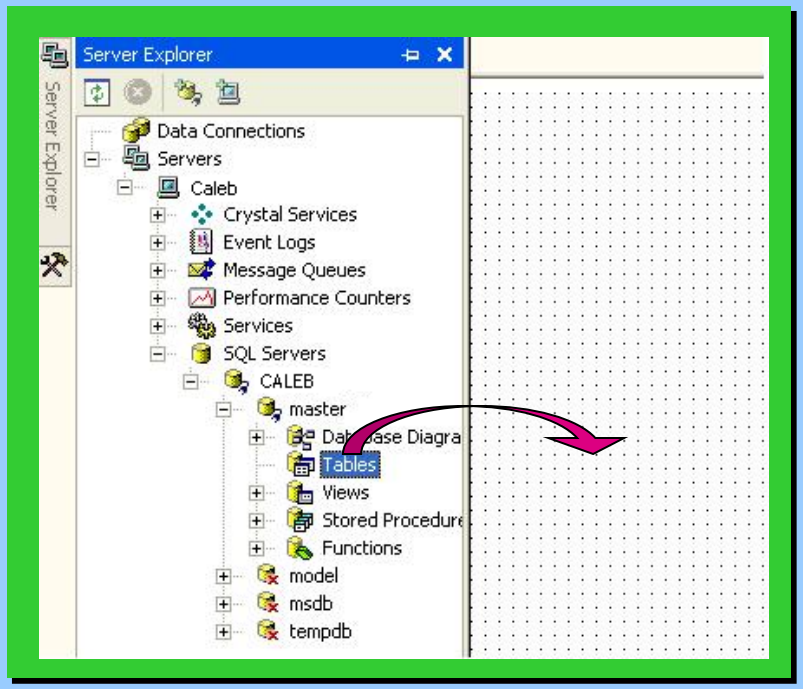
Accessing Data with ADO.NET

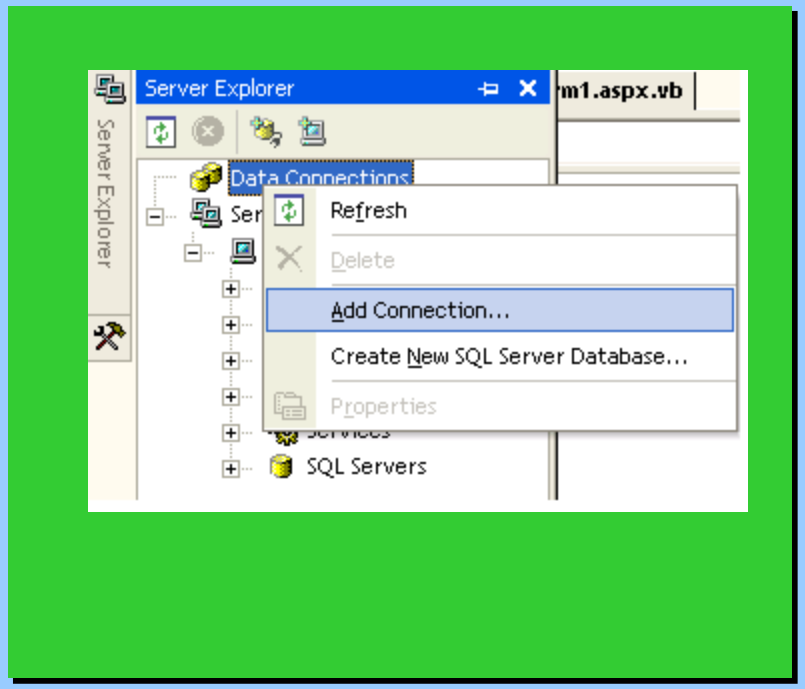
1. Client makes request
2. Create the SqlConnection and SqlDataAdapter objects
3. Fill the DataSet from the DataAdapter and close the connection
4. Return the DataSet to the Client
5. Client manipulates the data
6. Update the DataSet
7. Use the SqlDataAdapter to open the SqlConnection, update the database, and close the connection



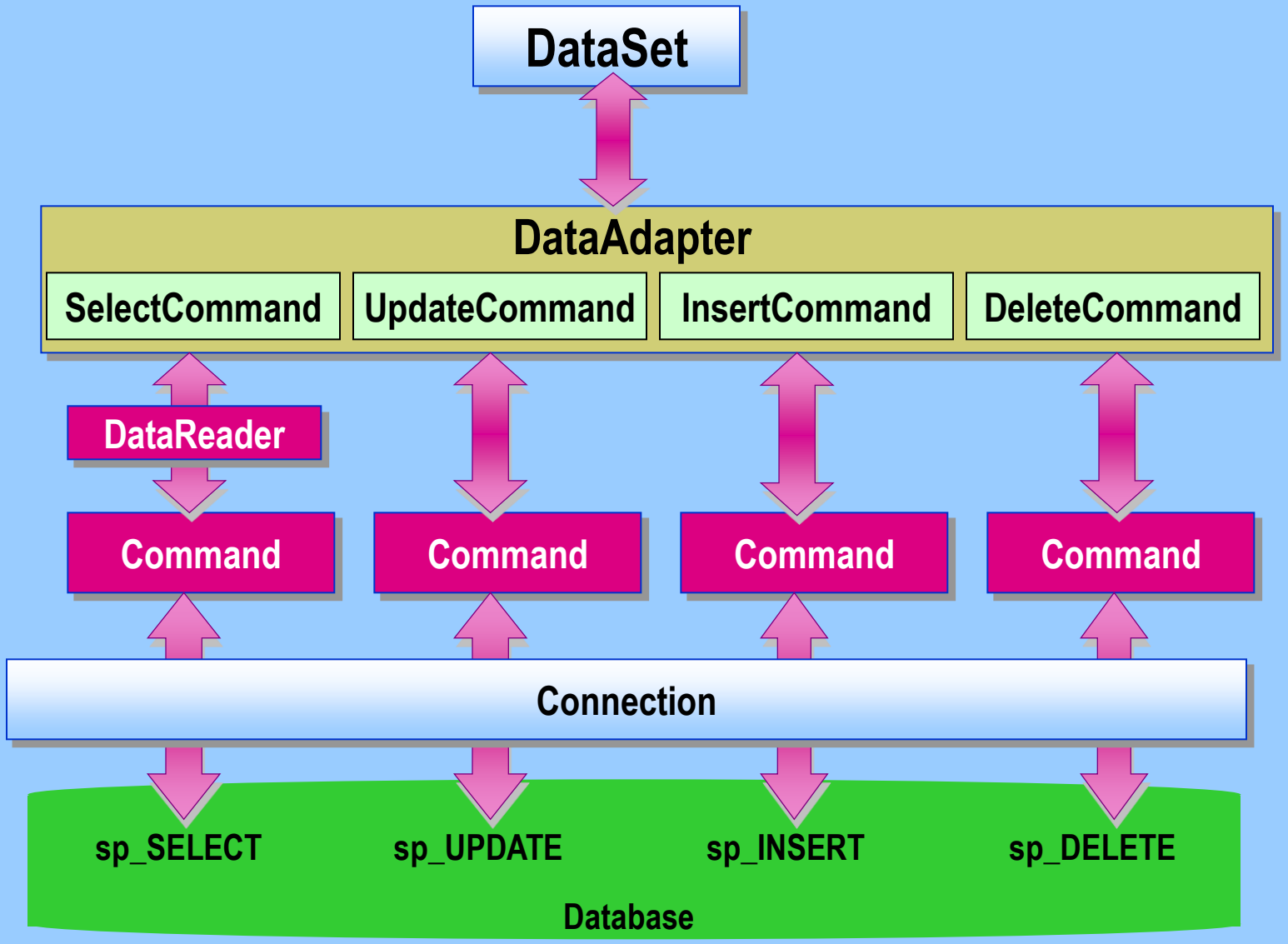
Using Server Explorer to Generate a Connection

- Create a new data connection by dragging from Server Explorer
- Create a new data connection using the Data Links dialog box





The DataAdapter Object Model



Demonstration: Connecting to a Database

1. Expand Server Explorer to a table in a SQL Server database
2. Drag the table to an ASP.NET page
3. View the data using xxxDataAdapter
4. View the source of the page that was created

Generating a Dataset

- **You can generate a dataset…**
  - …through the UI…
    - Creates a dataset that allows you to access data as an object
  - …or through code…

```
Dim ds As New DataSet()
```

- **…and then fill the DataSet from the DataAdapter(s)**

```
DataAdapter1.Fill(ds)
DataAdapter2.Fill(ds)
```

What Are List-Bound Controls?

- Controls that connect to a data source and display
  the data
- List-bound controls include the following:
  - DropDownList
  - ListBox
  - CheckBoxList
  - RadioButtonList
  - DataGrid
  - DataList
  - Repeater

Displaying DataSet Data in List-Bound Controls

- Set the properties

Property	Description
DataSource	▪ The **DataSet** containing the data
DataMember	▪ The **DataTable** in the DataSet
DataTextField	▪ The field in the **DataTable** that is displayed
DataValueField	▪ The field in the **DataTable** that becomes the value of the selected item in the list

- Fill the DataSet, then call the DataBind method

```
DataAdapter1.Fill(ds)
lstEmployees.DataBind()
```

Demonstration: Binding List-Bound Controls to a Database

1. Add a DataGrid to a Web Form
2. Set the DataSource and DataMember properties
3. Fill the DataSet
4. Call DataBind()

Practice: Using a DataGrid

- **Students will:**
  1. Create a **SqlConnection**
  2. Create a **SqlDataAdapter**
  3. Generate a **DataSet**
  4. Place a **DataGrid** on a Web Form
  5. Bind the **DataGrid** to the **DataSet**
- **Time: 5 minutes**