

Module 4: Building DataSets

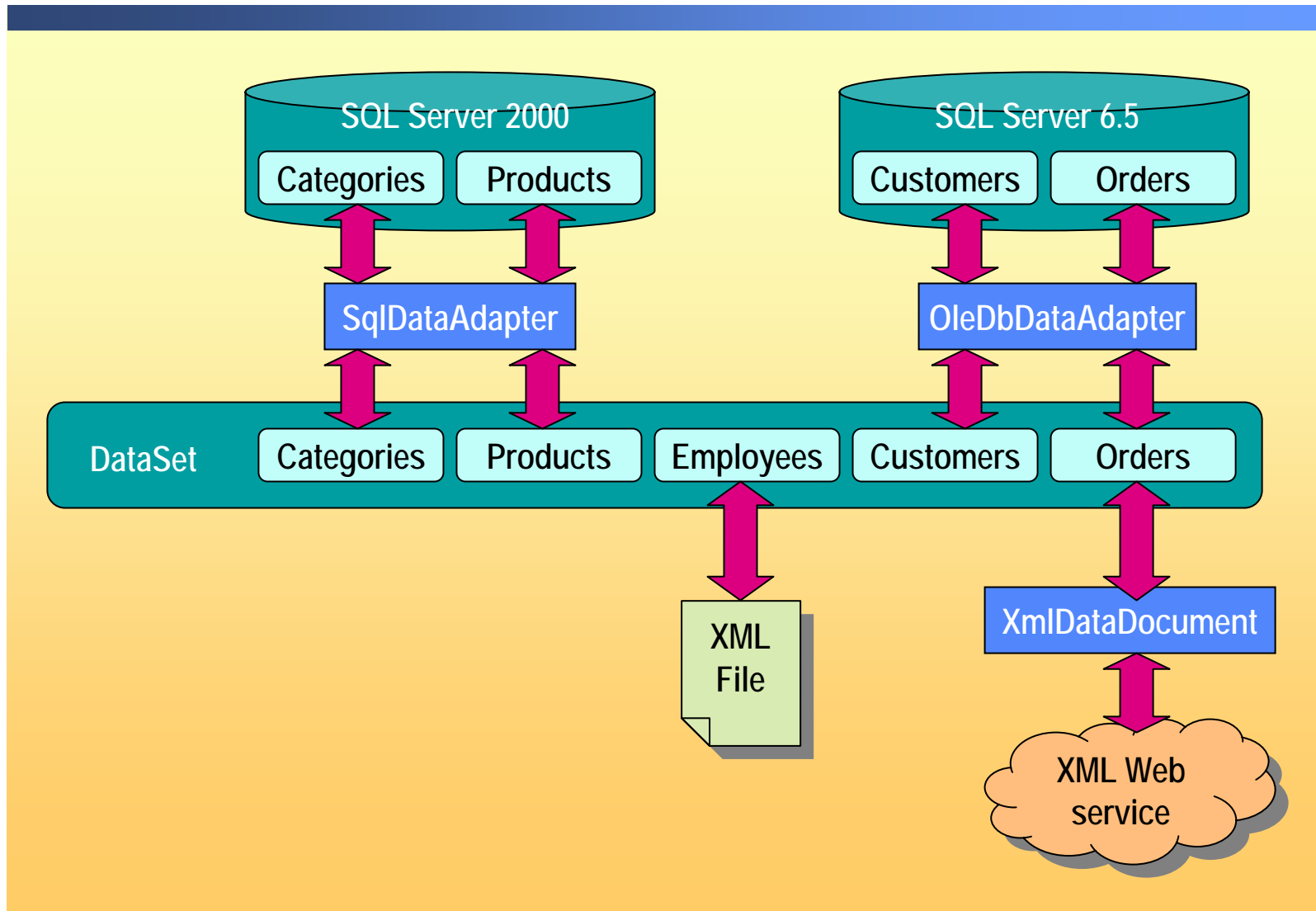
Overview

- Working in a Disconnected Environment
- Building DataSets and DataTables
- Binding and Saving a DataSet
- Defining Data Relationships
- Modifying Data in a DataTable
- Sorting and Filtering

Lesson: Working in a Disconnected Environment

- Typically disconnected scenarios
- .NET Framework classes used in scenarios

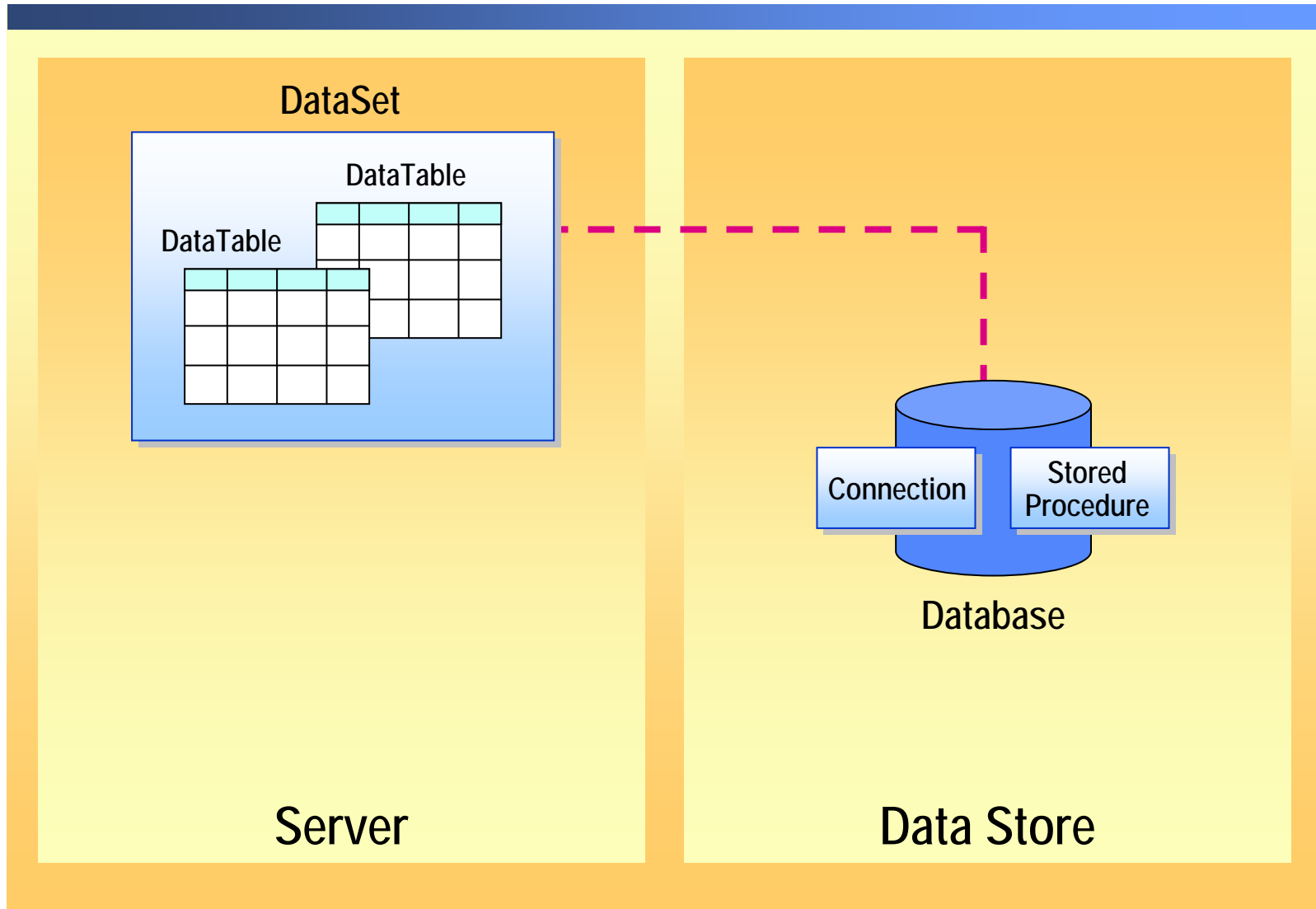
Disconnected Architecture



Lesson: Building DataSets and DataTables

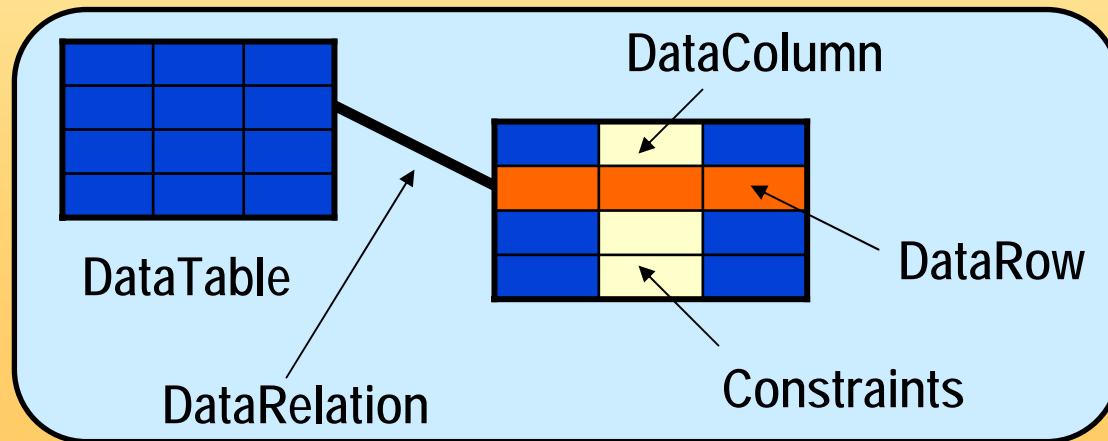
- What Are DataSets, DataTables, and DataColumnns?
- The DataSet Object Model
- How to Create a DataSet, a DataTable, and a DataColumn
- How to Create a Primary Key Constraint
- Using Unique Constraints
- Creating Custom Expressions

What Are DataSets, DataTables, and DataColumnns?



The DataSet Object Model

- Common collections
 - Tables (collection of DataTable objects)
 - Relations (collection of DataRelation objects)
- Data binding to Web and Windows controls supported
- Schema can be defined programmatically or using XSD



How to Create a DataSet, a DataTable, and a DataColumn

■ Creating a DataSet

- Drag and drop a DataSet control from the Toolbox

■ Creating a DataTable

- Edit the Tables collection of a DataSet by using the Properties window

■ Creating a DataColumn and adding it to a DataTable

- Edit the Columns collection of a DataTable by using the Properties window

How to Create a Primary Key Constraint

- **Set the PrimaryKey property of the DataTable**
 - Select the columns in order
 - Does not allow the naming of the constraint
- **Edit the Constraints collection of the DataTable**
 - Add a UniqueConstraint
 - Name the constraint
 - Select the columns
 - Check the primary key box

Using Unique Constraints

- Two types of constraints
 - UniqueConstraint
 - ForeignKeyConstraint
- Creating a constraint
 - Edit the Constraints collection of a DataTable
- UniqueConstraint
 - Array of DataColumnns
 - Can be the primary key for the DataTable

Creating Custom Expressions

- **Definition**

- Custom expressions are columns derived from calculations, rather than stored values

- **Using the DataColumn Expression property**

- `Sum([Unit Price] * [Quantity])`

- **Aggregate functions can use parent/child relationships**

- Avg, Count, Sum, Max, Min

Lesson: Binding and Saving a DataSet

- How to Bind Data to a Windows Control
- How to Bind a DataSet to a DataGrid
- How to Save and Open a DataSet

How to Bind Data to a Windows Control

■ Simple data binding

- Binding to a simple control that can only display a single value – for example, a text box

■ Complex data binding

- Binding to a more complex control that can display multiple values – for example, a data grid

■ All controls have a (DataBindings) collection

- Use the Properties window to edit data bindings

■ Best practice: use an intermediate DataView

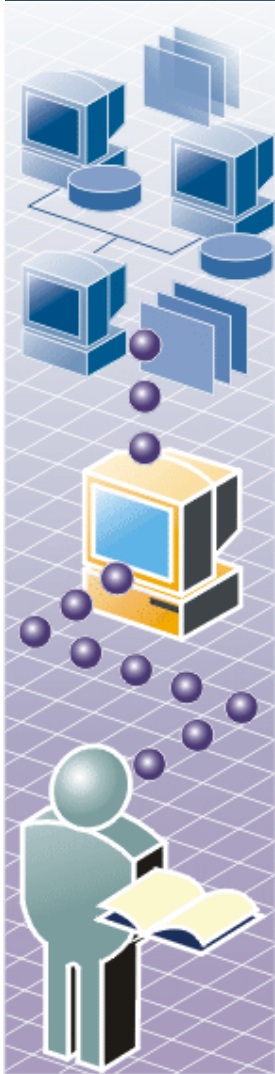
How to Bind a DataSet to a DataGrid

- **DataSource property**
 - Can be a DataSet, DataTable, or DataView
- **DataMember**
 - Can be a DataTable if the DataSource is a DataSet
- **Or, use the (DataBindings) collection**
- **Best practice: Use an intermediate DataView**

How to Save and Open a DataSet

- Two DataSet methods
 - ReadXml
 - WriteXml
 - Pass the path and filename as a string parameter
- ReadXml raises an exception if the file does not exist
- WriteXml overwrites existing files

Lab 4.1: Building, Binding, Opening, and Saving DataSets



- Exercise 1: Building the Custom DataSet
- Exercise 2: Opening and Saving the DataSet
- Exercise 3: Coding the Menus in the Windows Application

Lesson: Defining Data Relationships

- Using Foreign Key Constraints to Restrict Actions
- How to Create a Foreign Key Constraint
- What Is a DataRelation Object?
- How to Create a DataRelation Object
- How to Navigate Related DataTables

Using Foreign Key Constraints to Restrict Actions

- A ForeignKeyConstraint enforces referential integrity
 - If the EnforceConstraints property of a DataSet is True
- Restricting actions performed in related tables
 - DeleteRule and UpdateRule properties

Action	Description
Cascade	Deletes or updates related rows. This is the default.
SetNull	Sets values in related rows to DBNull.
SetDefault	Sets values in related rows to the DefaultValue.
None	No action is taken, but an exception is raised.

How to Create a Foreign Key Constraint

- **Use Properties window**
 - DataSet must have at least two tables
 - Parent table must have a primary key
 - Add a ForeignKeyConstraint to the child table
 - Choose the matching columns
 - Choose Update and Delete rules
- **Write code**

What Is a DataRelation Object?

- **Definition**
 - A DataRelation object defines a navigational relationship, NOT a constraint relationship
- **Used by presentation objects (for example, a DataGrid) to allow easier navigation (for example, “drill down” capability from parent rows to child rows)**
- **Used by expression columns to calculate aggregates**
- **A DataSet has a Relations collection**

How to Create a DataRelation Object

- Use the Properties window
- Or, write code

```
dsNorthwind.Relations.Add( _  
    "FK_CustomersOrders", _  
    dtCustomers.Columns("CustomerID"), _  
    dtOrders.Columns("CustomerID"), _  
    True) ' create a ForeignKeyConstraint too
```

How to Navigate Related DataTables

- The GetChildRows method of the DataRow
 - Pass a DataRelation name as the parameter
- Example:

```
Dim drCustomer As DataRow
Dim drOrder As DataRow
For Each drCustomer In _
    dsNorthwind.Tables("Customer").Rows
    For Each drOrder In drCustomer.GetChildRows( _
        "FK_CustomersOrders")
        ' process row
    Next
Next
Next
```

Lesson: Modifying Data in a DataTable

- How to Insert a New Row
- How to Position on a Row
- Modifying Data in a Table
- How to Delete a Row
- What Are the RowState and RowVersion Properties?
- How to Handle the DataTable Events

How to Insert a New Row

- Creating a new row

```
Dim drNewEmployee As DataRow = _  
    dtEmployees.NewRow()
```

- Filling the new row

```
drNewEmployee("EmployeeID") = 11  
drNewEmployee("LastName") = "Smith"
```

- Appending the row to a DataTable

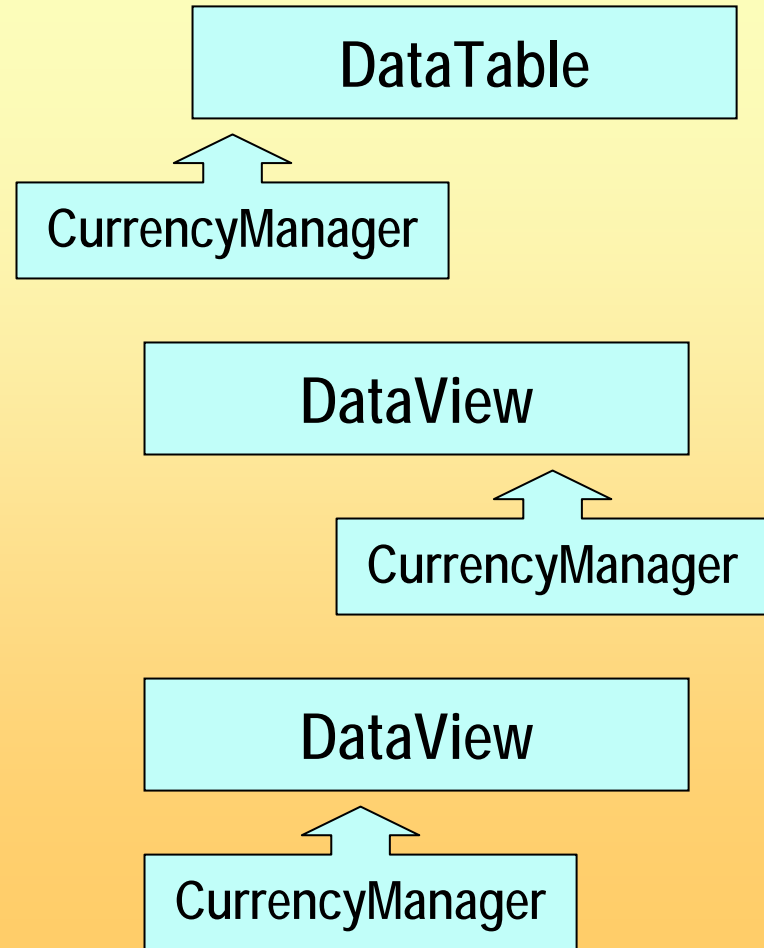
```
dtEmployees.Rows.Add(drNewEmployee)
```

- Creating, filling, and appending a row simultaneously

```
dtEmployees.Rows.Add(_  
    New Object() {11, "Smith"})
```

How to Position on a Row

- CurrencyManager object
 - Position property points to current row
 - Position = 0 is first row
- Each DataTable and DataView can have one CurrencyManager



Modifying Data in a Table

- The BeginEdit method of DataRow class
 - Disables the raising of events and exceptions
- EndEdit and CancelEdit methods of DataRow class
 - Enable the raising of events and exceptions
- How to modify data in a table

```
Dim drEmployee As DataRow = dtEmployees.Rows(3)
drEmployee.BeginEdit()
drEmployee("FirstName") = "John"
drEmployee("LastName") = "Smith"
drEmployee.EndEdit()
```

How to Delete a Record

- The Remove method of the DataRowCollection class
 - Completely removes the row from the collection

- Example:

```
dtEmployees.Rows.Remove(drEmployee)
```

- The Delete method of the DataRow class
 - Marks the row as deleted
 - Hidden, but still accessible if necessary

- Example:

```
drEmployee.Delete
```

What Are the RowState and RowVersion Properties?

- RowState property of a DataRow
 - Added, Deleted, Detached, Modified, Unchanged
- DataRowState enumeration
 - Used with a DataView to filter rows of a certain state
 - CurrentRows, OriginalRows, and so on
- DataRowVersion enumeration is used when retrieving values using the Item property
 - Current, Default, Original, Proposed
- HasVersion method of a DataRow
- AcceptChanges and RejectChanges methods

How to Handle the DataTable Events

- Example:

```
Private WithEvents dtProducts As DataTable

Private Sub dtProducts_RowDeleted( _
    ByVal sender As Object, _
    ByVal e As System.Data.DataRowChangeEventArgs) _
    Handles dtProducts.RowDeleted

    ' write code here

End Sub
```

- DataTable events list

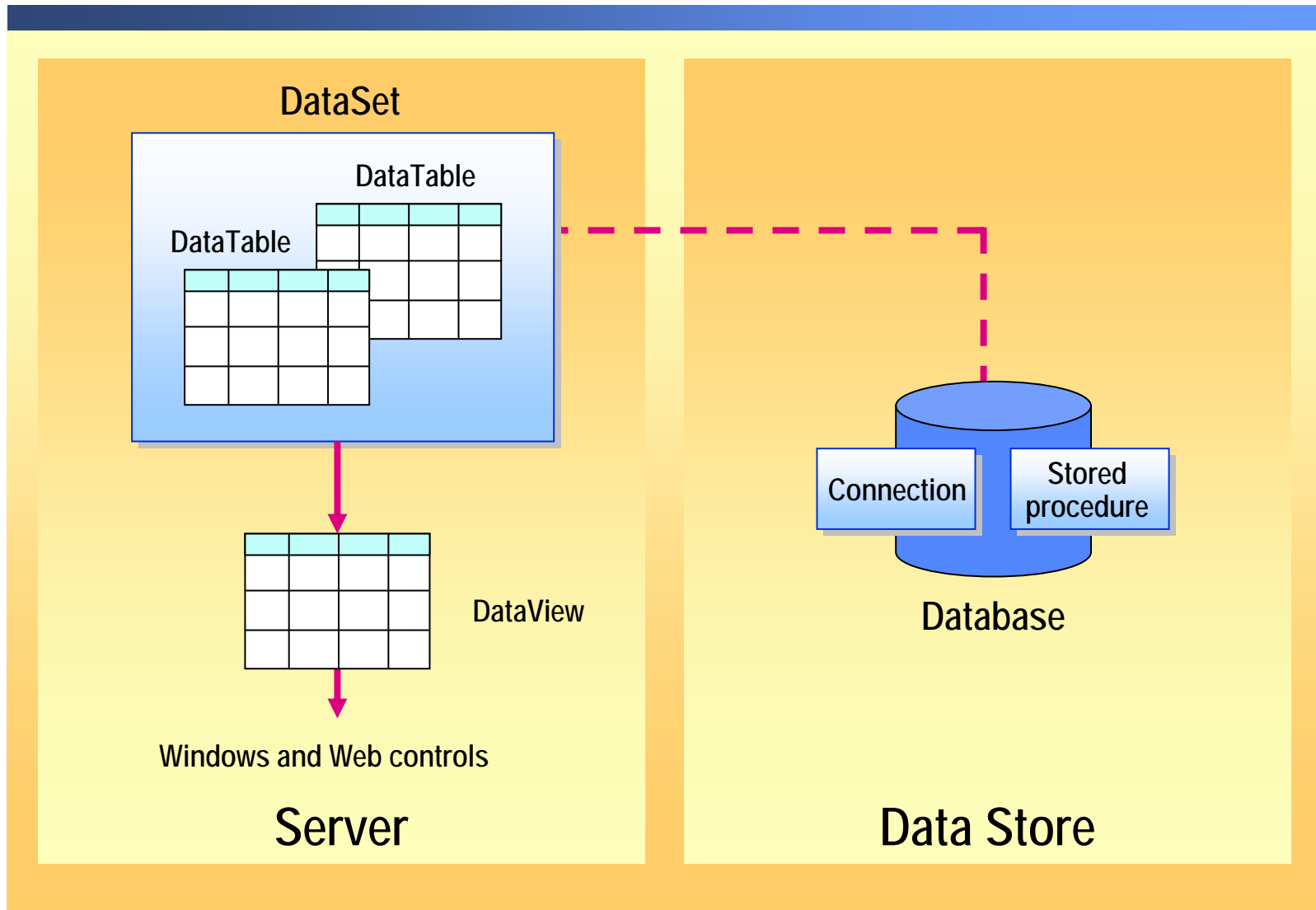
Lesson: Sorting and Filtering

- How to use the Select method
- What Is a DataView Object?
- How to Define a DataView
- How to Use a DataView at Design Time

How to Use the Select Method

- **DataTables have a Select method**
 - Gets an array of DataRow objects that match the filter in the order of the sort, and that match the specified state
- **Three optional parameters**
 - Filter expression, for example, "City='London'"
 - Sort, for example, "CompanyName ASC"
 - DataRowState, for example, Deleted

What Is a DataView Object?



How to Define a DataView

- Creating a DataView by using form controls
- Creating a DataView programmatically

```
Dim dvProducts As New _  
    DataView(dsNorthwind.Tables("Products"))  
dvProducts.Sort = "UnitPrice"  
dvProducts.RowFilter = "CategoryID > 4"  
grdProducts.DataSource = dvProducts
```

- Applying a DataView to a DataTable

```
dvProducts.Table = _  
    dsNorthwind.Tables("Products")
```

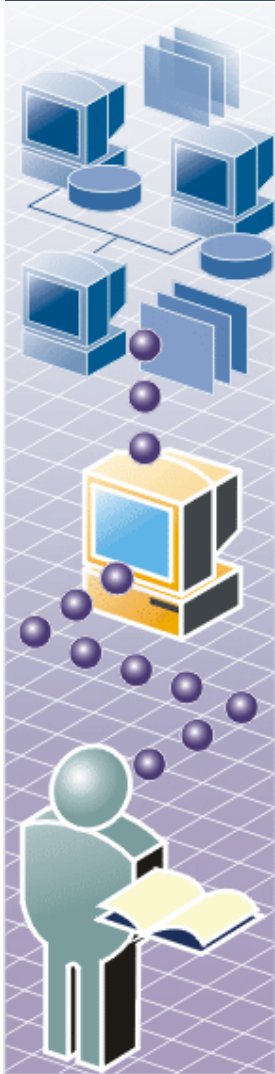
How to Use a DataView at Design Time

- DataViews allow sorting and filtering at design time
- A DataView cannot span multiple DataTables, unlike the View object in SQL Server
 - Use presentation-level objects instead (for example, the DataGrid control, report designers, and so on)
- Every DataTable has a DefaultView property

Review

- Working in a Disconnected Environment
- Building DataSets and DataTables
- Binding and Saving a DataSet
- Defining Data Relationships
- Modifying Data in a DataTable
- Sorting and Filtering

Lab 4.2: Manipulating DataSets and Modifying Data



- Exercise 1: Creating Relationships
- Exercise 2: Navigating Relationships
- Exercise 3: Editing Rows in a DataTable
- Exercise 4: Sorting and Filtering with DataViews