

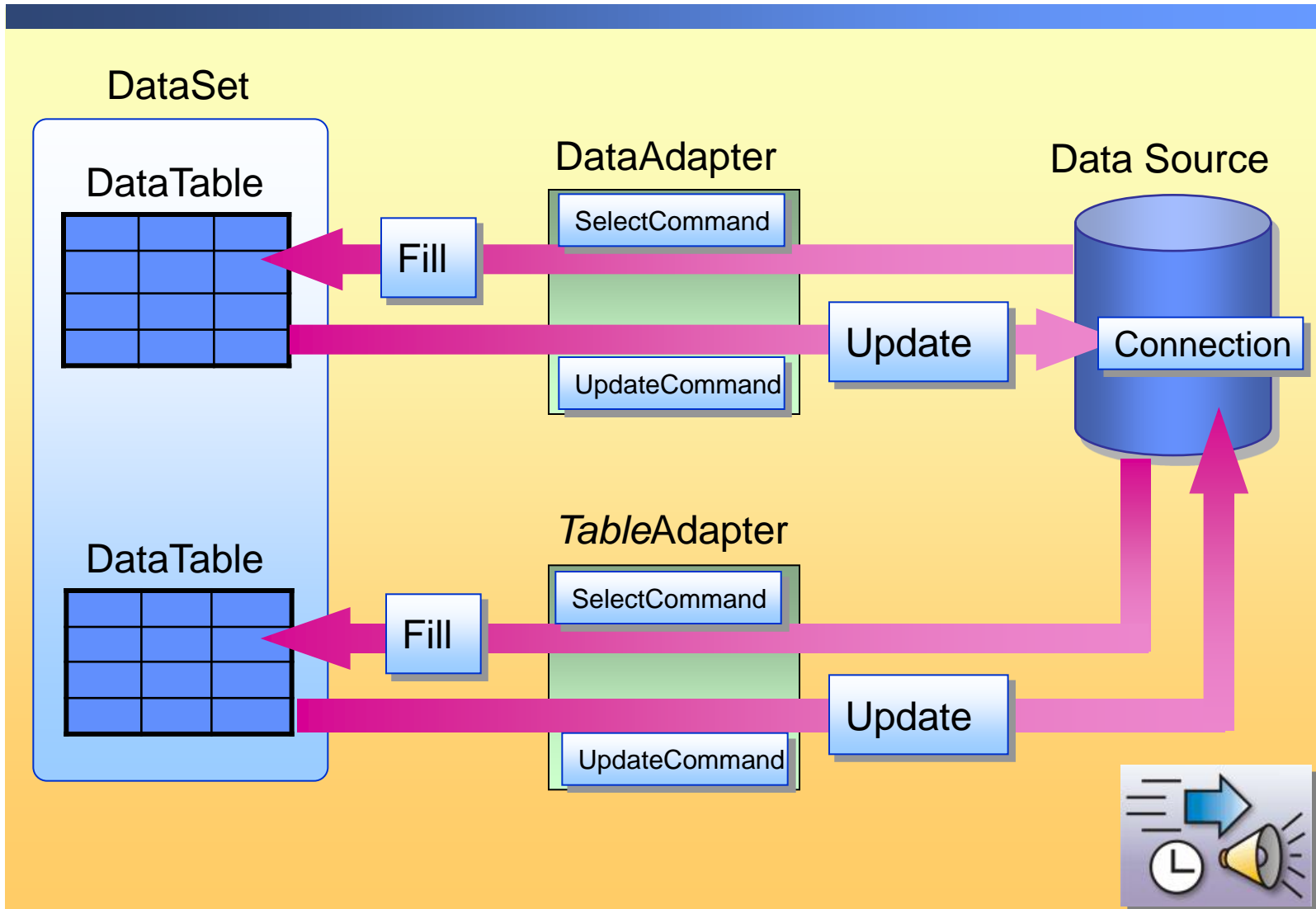
# ADO .NET

Programiranje korisničkih interfejsa

Bojan Furlan

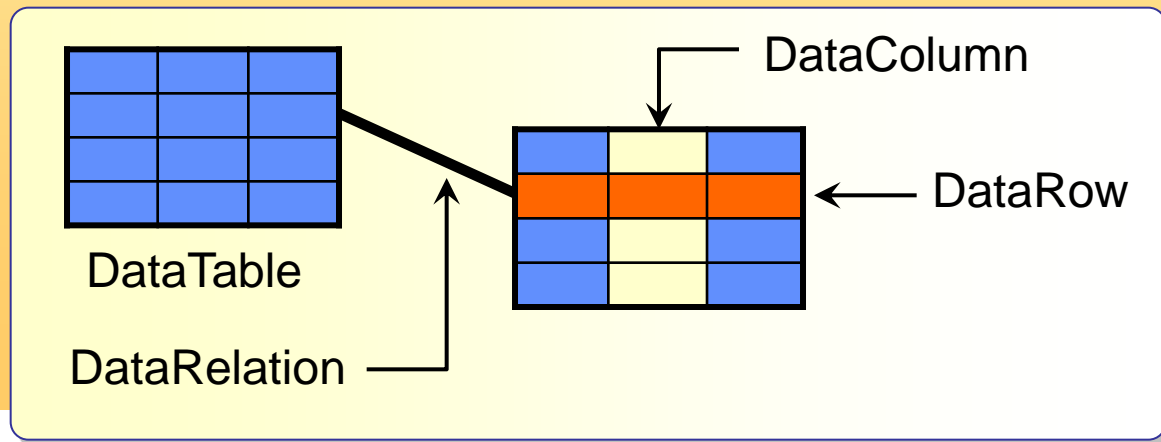


# ADO.NET Objects



# What Is a DataSet?

- Datasets can include multiple DataTables
- Relationships between tables are represented using DataRelations
- Constraints enforce primary and foreign keys
- Use the DataRow and DataColumn to access values in Tables



# What Is a Typed DataSet?

## ■ Typed datasets

- Derive from the base **DataSet** class
- Provide type checking at compile time
- Provide faster access to tables and columns in the dataset
- Generated from XML Schema (.xsd) files by using the XSD.exe tool

## ■ To access tables and columns

- Untyped dataset

```
PubsDataSet.Tables("Titles");
```

- Typed dataset

```
PubsDataSet.Titles;
```

# How to Add ADO.NET Objects to and Configure ADO.NET Objects in a Windows Forms Application

- 1** Drag an OleDbDataAdapter or SqlDataAdapter object from the Toolbox to a form
- 2** Specify connection and SQL command information
- 3** Select the adapter or adapters that will be used to transfer data between the data source and the dataset
- 4** On the Data menu, choose Generate Dataset
- 5** Select New and then specify a name for the new dataset

# How to Populate a Dataset

- Use the DataAdapter object to fill the dataset

```
SqlDataAdapter storesSQLDataAdapter;  
SqlCommand storesSelectSqlCommand;  
storesSelectSqlCommand.CommandText = "SELECT * FROM  
stores";  
storesSelectSqlCommand.Connection = SqlConnection1;  
storesSQLDataAdapter.SelectCommand =  
storesSelectSqlCommand;  
storesSQLDataAdapter.Fill(storesDataSet.Tables["Stores"]);
```

- How to pass Parameter

```
SELECT stor_id, ord_num, qty, ord_date, payterms, title_id  
FROM sales WHERE (stor_id = @stor_id)  
...  
salesSqlDataAdapter.SelectCommand.Parameters["@stor_id"].  
Value = stor_id;
```

# How to “Manually” Update Data in a DataSet

## ■ Adding rows

```
DataRow newDataRow =  
    pubsDataSet.Tables["Titles"].NewRow();  
newDataRow["title"] = "New Book";  
newDataRow["type"] = "business";  
pubsDataSet.Tables["Titles"].Rows.Add(newDataRow);
```

## ■ Editing rows

```
changeDataRow.BeginEdit( );  
changeDataRow["Title"] = changeDataRow["Title"].ToString()  
    + " 1";  
changeDataRow.EndEdit( );
```

## ■ Deleting data

```
DataRow deleteDataRow =  
    pubsDataSet.Tables["Titles"].Rows[0];  
pubsDataSet.Tables["Titles"].Rows.Remove(deleteDataRow);
```

# How to Explicitly Update Data to a Data Source

- Explicitly specifying the updates

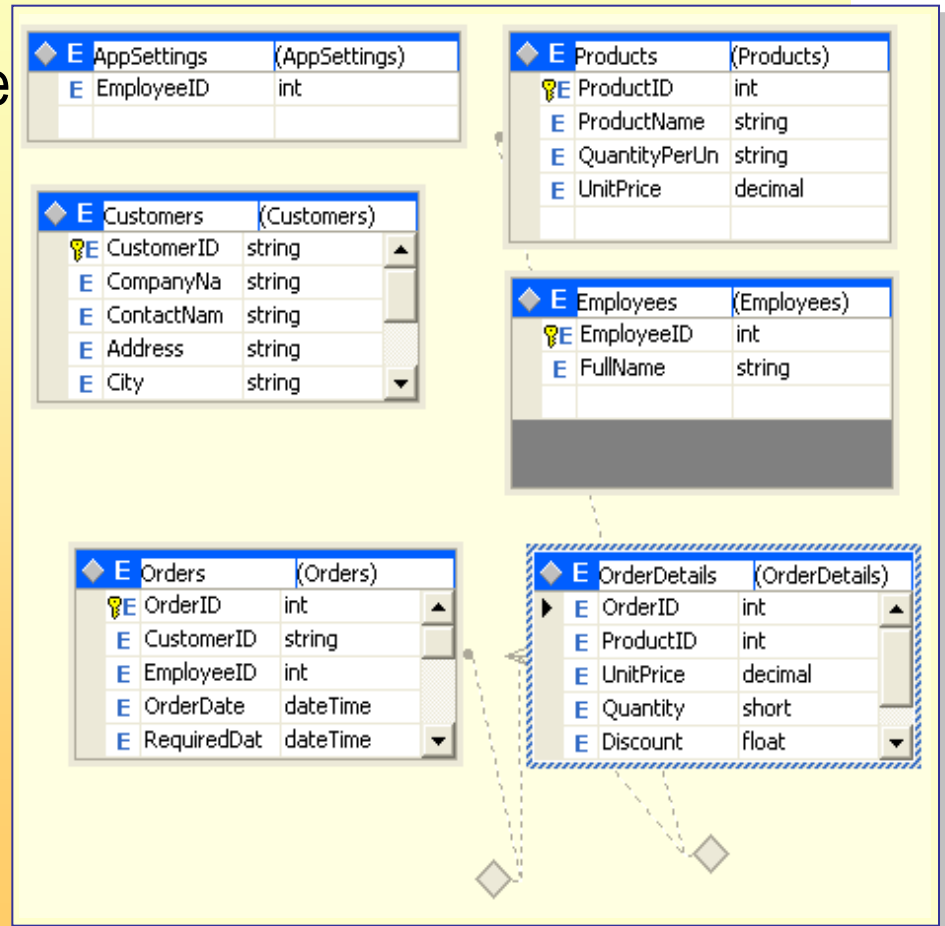
```
SqlCommand insertTitlesCommand = new SqlCommand
("Insert titles (title_id, title, type)
values (@title_id,@title,@type)");

insertTitlesCommand.Parameters.Add
    ("@title_id", SqlDbType.VarChar, 6);
insertTitlesCommand.Parameters.Add
    ("@title", SqlDbType.VarChar, 80);
insertTitlesCommand.Parameters.Add
    ("@type", SqlDbType.Char, 12);
//Similar for update
titlesSQLDataAdapter.InsertCommand = insertTitlesCommand;
titlesSQLDataAdapter.Update(pubsDataSet, "titles");
```



# How to Create Database Schema on the Client

- XML Schema (.xsd) files enforce data integrity on the client
- Use the XML Designer to create and modify XML Schema files
  1. Determine the schema design
  2. On the Project menu, click Add New Item
  3. Add the schema
  4. Create the schema



# How to Perform Simple Binding by Using the DataBindings Property

To use the DataBindings Collection to bind a control to a data source, set the DataBinding property of the control to the data source

- Bind any property of Control

Property of the control to which data is bound

```
txtCustomerAddress.DataBindings.Add("Text",  
    dsNorthwindData1.Customers, "Address");  
txtCustomerCity.DataBindings.Add("Text",  
    dsNorthwindData1.Customers, "City");
```

Table from the data source

Column in the table

# How to Perform Complex Data Binding by Using the DataBound Windows Forms Controls

## ■ Complex data binding

- Bind a control property to a data table
- Use with combo boxes, list boxes, data grids
- Can bind controls to DataSets, Arrays, and ArrayLists

## ■ Complex databinding at design time

- Set the **DataSource** and **DataMember** properties

## ■ Complex databinding at run time

```
DataGrid1.DataSource = productsDataSet;  
DataGrid1.DataMember = "Products";
```

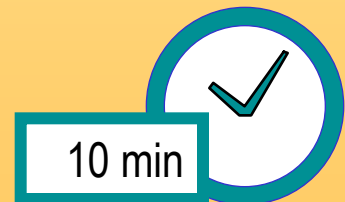
# Practice: Binding Controls to Data



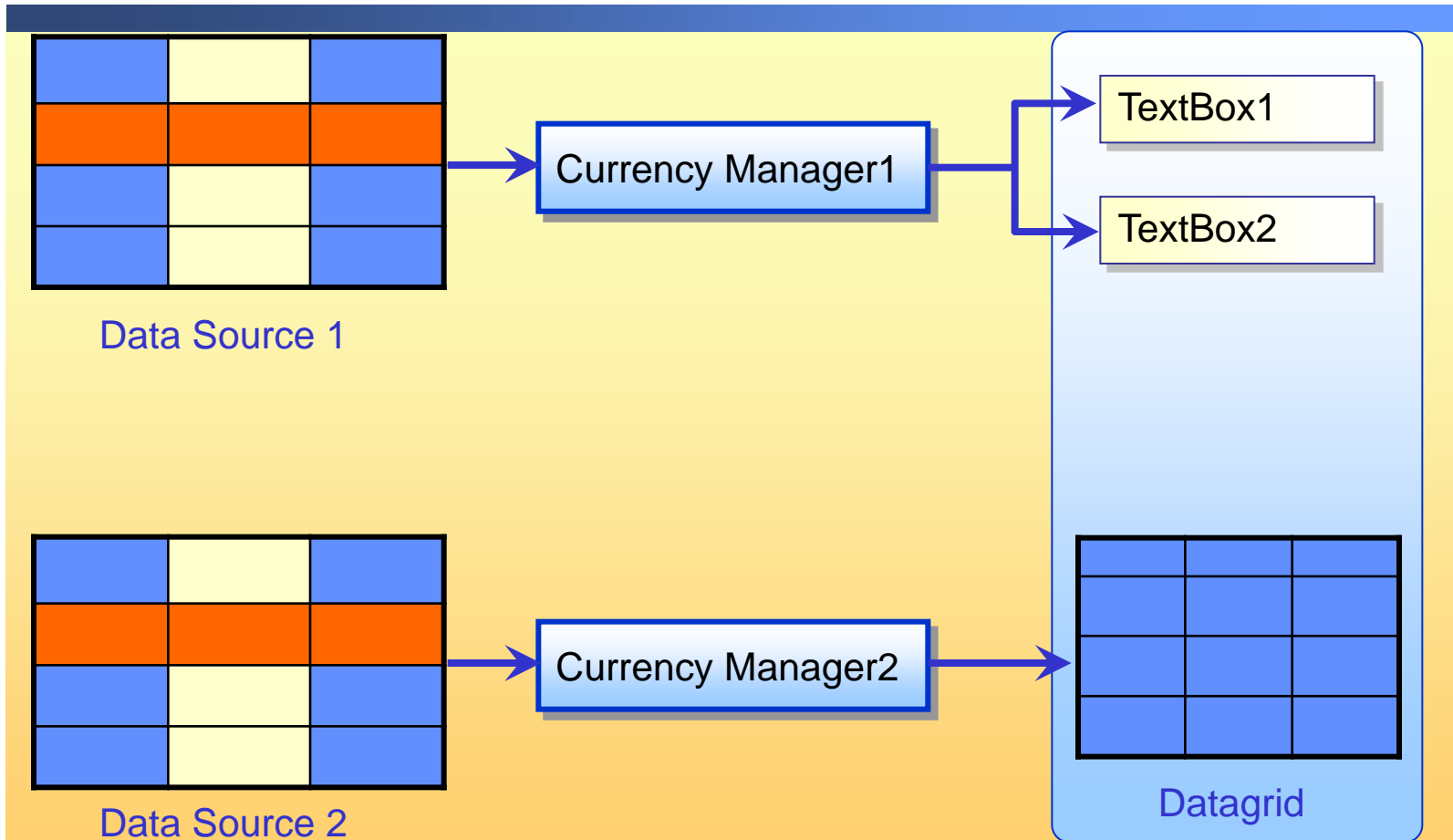
## In this practice, you will

- Configure the **SQLConnection** control to connect to the database
- Bind controls to columns in the dataset at design time
- Bind controls to columns in the dataset at run time

Begin reviewing the objectives for this practice activity

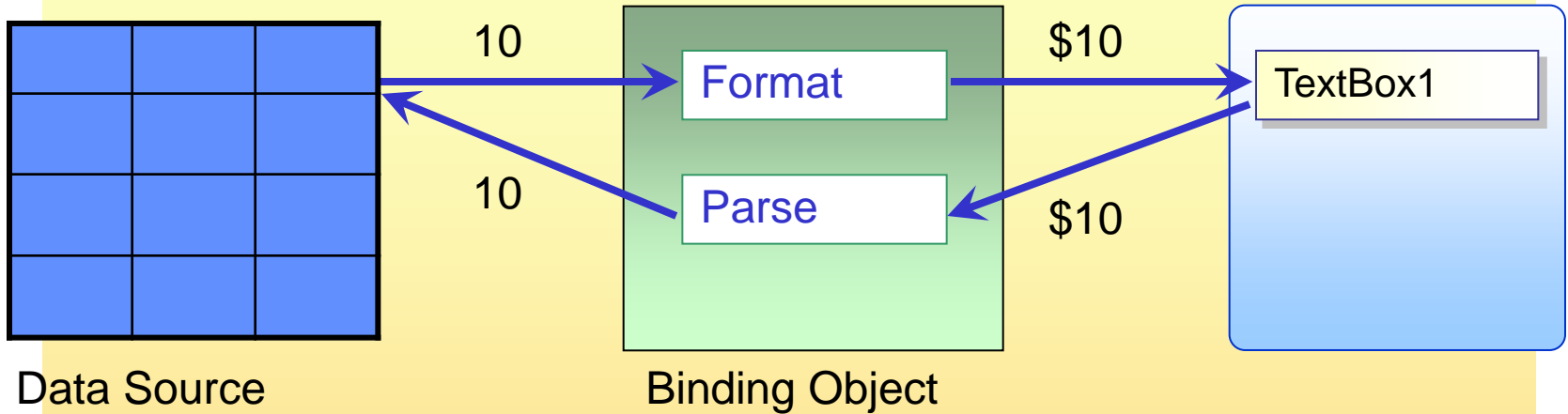


# How to Maintain the Currency of a Control by Using CurrencyManager



```
CurrencyManager cm;  
cm = (CurrencyManager)this.BindingContext[pubsDataSet,  
"Authors"];  
cm.Position += 1;
```

# How to Format and Parse Data Bound Values



```
Binding priceBinding = new
    Binding("Text", titlesDataSet1, "titles.price");
priceBinding.Format += new
    ConvertEventHandler(this.FormatDecimalToString);
priceBinding.Parse += new
    ConvertEventHandler(this.ParseStringToDecimal);
PriceTextBox.DataBindings.Add(priceBinding);
```