

CGT 456 Lecture 9

Parameters



SqlParameterCollection class

- The command object works in conjunction with two classes, the first of which is:
 - SqlParameterCollection

- The SqlParameterCollection is accessed through the **Parameters** property
 - Esposito Ch. 10



SqlParameter

- The second is:
 - SqlParameter

- SqlParameter represents a single command parameter stored in the collection.
 - Esposito Ch. 10



SELECT (overview)

- ❑ `Sql = "SELECT * FROM Products_Lab6 WHERE ProductID = @prodID";`
- ❑ `myCommand = new SqlCommand(Sql, oConn);`
- ❑ `myCommand.Parameters.Add(
new SqlParameter(
"@prodID", System.Data.SqlDbType.Int, 4));`
- ❑ `myCommand.Parameters["@prodID"].Value =
dt.Rows[0][0].ToString();`
- ❑ `results.DataSource = myCommand.ExecuteReader();`
- ❑ `results.DataBind();`



SELECT

- `// Create the SQL statement`
 - `// The @prodID represents a parameter that`
 - `// serves as a placeholder that's value can`
 - `// be passed into the SQL statement later.`
-
- `Sql = "SELECT * FROM Products_Lab6 WHERE ProductID = @prodID";`



SELECT (cont.)

- `// Create a new SQL command using`
- `// the SQL command and connection object`

- `myCommand = new SqlCommand(Sql, oConn);`


SELECT (cont.)

- `// Parameters gets the`
- `// SqlParameterCollection.`
- `// Add a new parameter matching the`
- `// parameter(s) used in the above`
- `// SQL statement.`

- `myCommand.Parameters.Add(
new SqlParameter(
"@prodID", System.Data.SqlDbType.Int, 4));`

■ **The above should all be on one line*

Design Table 'Products_Lab6' in 'cgt356inst' o...

Column Name	Data Type	Length	Allow Nulls
 ProductID	int	4	
title	varchar	100	✓
authors	varchar	100	✓
copyrightDate	varchar	6	✓
edition	varchar	6	✓
isbn	varchar	25	✓
coverart	varchar	100	✓
description	varchar	100	✓
price	varchar	100	✓

Columns

Description	
Default Value	
Precision	10
Scale	0
Identity	Yes
Identity Seed	1
Identity Increment	1
Is RowGuid	No
Formula	
Collation	



SELECT (cont.)

- `// Assign a value to the @prodID`
- `// placeholder in the above SQL statement`

- `myCommand.Parameters["@prodID"].Value = dt.Rows[0][0].ToString();`

- **The above should all be on one line*



SELECT (cont.)

- `// set the data source`
- `results.DataSource = myCommand.ExecuteReader();`

- `// bind the data source to the DataGrid`
- `results.DataBind();`



SELECT (overview)

- ❑ `Sql = "SELECT * FROM Products_Lab6 WHERE ProductID = @prodID";`
- ❑ `myCommand = new SqlCommand(Sql, oConn);`
- ❑ `myCommand.Parameters.Add(
new SqlParameter(
"@prodID", System.Data.SqlDbType.Int, 4));`
- ❑ `myCommand.Parameters["@prodID"].Value =
dt.Rows[0][0].ToString();`
- ❑ `results.DataSource = myCommand.ExecuteReader();`
- ❑ `results.DataBind();`


INSERT (overview)

- `Sql = "INSERT INTO Products_Lab6 (title, authors, copyrightDate, edition, isbn, coverart, description, price) VALUES(@title, @authors, @copyrightDate, @edition, @isbn, @coverart, @description, @price)";`

- `oConn = new SqlConnection(ConnStr);`
- `myCommand = new SqlCommand(Sql, oConn);`

- `myCommand.Parameters.Add(new SqlParameter("@title", System.Data.SqlDbType.VarChar, 100));`
- `myCommand.Parameters.Add(new SqlParameter("@authors", System.Data.SqlDbType.VarChar, 100));`
- `myCommand.Parameters.Add(new SqlParameter("@copyrightDate", System.Data.SqlDbType.VarChar, 6));`
- `myCommand.Parameters.Add(new SqlParameter("@edition", System.Data.SqlDbType.VarChar, 6));`
- `myCommand.Parameters.Add(new SqlParameter("@isbn", System.Data.SqlDbType.VarChar, 25));`
- `myCommand.Parameters.Add(new SqlParameter("@coverart", System.Data.SqlDbType.VarChar, 100));`
- `myCommand.Parameters.Add(new SqlParameter("@description", System.Data.SqlDbType.VarChar, 100));`
- `myCommand.Parameters.Add(new SqlParameter("@price", System.Data.SqlDbType.VarChar, 100));`

Design Table 'Products_Lab6' in 'cgt356inst' o...

Column Name	Data Type	Length	Allow Nulls
 ProductID	int	4	
title	varchar	100	✓
authors	varchar	100	✓
copyrightDate	varchar	6	✓
edition	varchar	6	✓
isbn	varchar	25	✓
coverart	varchar	100	✓
description	varchar	100	✓
price	varchar	100	✓

Columns

Description	
Default Value	
Precision	10
Scale	0
Identity	Yes
Identity Seed	1
Identity Increment	1
Is RowGuid	No
Formula	
Collation	



INSERT (overview cont.)

- ❑ `myCommand.Parameters["@title"].Value = "Microsoft Visual";`
- ❑ `myCommand.Parameters["@authors"].Value = "Jon Jagger";`
- ❑ `myCommand.Parameters["@copyrightDate"].Value = "2003";`
- ❑ `myCommand.Parameters["@edition"].Value = "2003";`
- ❑ `myCommand.Parameters["@isbn"].Value = "0735619093";`
- ❑ `myCommand.Parameters["@coverart"].Value = "Microsoft";`
- ❑ `myCommand.Parameters["@description"].Value = "Teach yourself";`
- ❑ `myCommand.Parameters["@price"].Value = "26.39";`

- ❑ `oConn.Open();`
- ❑ `myCommand.ExecuteNonQuery();`
- ❑ `oConn.Close();`

- ❑ `Select_Click(Source, E);`

INSERT

- // The @xxxx represents parameters that
- // serve as a placeholders that's value
- // can be passed into the SQL statement
- // later.

- ```
Sql = "INSERT INTO Products_Lab6 (title,
authors, copyrightDate, edition, isbn,
coverart, description, price)
VALUES(@title, @authors, @copyrightDate,
@edition, @isbn, @coverart, @description,
@price)";
```

■ *\*The above should all be on one line*



## INSERT (cont.)

---

- `// Create a new connection object using`
- `// the connection string above.`
- `// oConn represents an open connection to`
- `// a SQL Server database.`
  
- `oConn = new SqlConnection(ConnStr);`





## INSERT (cont.)

---

- `// Create a new SQL command using the SQL`
- `// command and connection object.`
- `// myCommand represents a SQL statement`
- `// or stored procedure to execute`
- `// against a SQL Server database.`
  
- `myCommand = new SqlCommand(Sql, oConn);`



# INSERT (cont.)

---

- `// Parameters gets the`
- `// SqlParameterCollection.`
- `// Add a new parameter matching the`
- `// parameter(s) used in the above`
- `// SQL statement`
  
- `myCommand.Parameters.Add(  
new SqlParameter("@title", System.Data.SqlDbType.VarChar, 100));`



# INSERT (cont.)

---

- `myCommand.Parameters.Add(  
new SqlParameter("@authors", System.Data.SqlDbType.VarChar, 100));`
- `myCommand.Parameters.Add(  
new SqlParameter("@copyrightDate", System.Data.SqlDbType.VarChar, 6));`
- `myCommand.Parameters.Add(  
new SqlParameter("@edition", System.Data.SqlDbType.VarChar, 6));`
- `myCommand.Parameters.Add(  
new SqlParameter("@isbn", System.Data.SqlDbType.VarChar, 25));`
- `myCommand.Parameters.Add(  
new SqlParameter("@coverart", System.Data.SqlDbType.VarChar, 100));`
- `myCommand.Parameters.Add(  
new SqlParameter("@description", System.Data.SqlDbType.VarChar, 100));`
- `myCommand.Parameters.Add(  
new SqlParameter("@price", System.Data.SqlDbType.VarChar, 100));`



# INSERT (cont.)

---

- ❑ `// Assign a value to the placeholders in the`
- ❑ `// above SQL statement`
- ❑ `myCommand.Parameters["@title"].Value = "Microsoft Visual";`
- ❑ `myCommand.Parameters["@authors"].Value = "Jon Jagger";`
- ❑ `myCommand.Parameters["@copyrightDate"].Value = "2003";`
- ❑ `myCommand.Parameters["@edition"].Value = "2003";`
- ❑ `myCommand.Parameters["@isbn"].Value = "0735619093";`
- ❑ `myCommand.Parameters["@coverart"].Value = "Microsoft";`
- ❑ `myCommand.Parameters["@description"].Value = "Teach yourself";`
- ❑ `myCommand.Parameters["@price"].Value = "26.39";`



# INSERT (cont.)

---

- `// Open the connection to the database`
  
- `oConn.Open( );`
  
- `// Execute the SQL statement against the`
- `// connection. ExecuteNonQuery will`
- `// return the number of rows affected,`
- `// but does not return a recordset.`
  
- `myCommand.ExecuteNonQuery( );`



## INSERT (cont.)

---

- `// Close the connection to the database`
- `oConn.Close();`
- `// Call the Select_Click() function to`
- `// redisplay the data in the browser.`
- `Select_Click(Source, E);`

# INSERT (overview)

---

- `Sql = "INSERT INTO Products_Lab6 (title, authors, copyrightDate, edition, isbn, coverart, description, price) VALUES(@title, @authors, @copyrightDate, @edition, @isbn, @coverart, @description, @price)";`
  
- `oConn = new SqlConnection(ConnStr);`
- `myCommand = new SqlCommand(Sql, oConn);`
  
- `myCommand.Parameters.Add(new SqlParameter("@title", System.Data.SqlDbType.VarChar, 100));`
- `myCommand.Parameters.Add(new SqlParameter("@authors", System.Data.SqlDbType.VarChar, 100));`
- `myCommand.Parameters.Add(new SqlParameter("@copyrightDate", System.Data.SqlDbType.VarChar, 6));`
- `myCommand.Parameters.Add(new SqlParameter("@edition", System.Data.SqlDbType.VarChar, 6));`
- `myCommand.Parameters.Add(new SqlParameter("@isbn", System.Data.SqlDbType.VarChar, 25));`
- `myCommand.Parameters.Add(new SqlParameter("@coverart", System.Data.SqlDbType.VarChar, 100));`
- `myCommand.Parameters.Add(new SqlParameter("@description", System.Data.SqlDbType.VarChar, 100));`
- `myCommand.Parameters.Add(new SqlParameter("@price", System.Data.SqlDbType.VarChar, 100));`

# INSERT (overview cont.)

---

- ❑ `myCommand.Parameters["@title"].Value = "Microsoft Visual";`
- ❑ `myCommand.Parameters["@authors"].Value = "Jon Jagger";`
- ❑ `myCommand.Parameters["@copyrightDate"].Value = "2003";`
- ❑ `myCommand.Parameters["@edition"].Value = "2003";`
- ❑ `myCommand.Parameters["@isbn"].Value = "0735619093";`
- ❑ `myCommand.Parameters["@coverart"].Value = "Microsoft";`
- ❑ `myCommand.Parameters["@description"].Value = "Teach yourself";`
- ❑ `myCommand.Parameters["@price"].Value = "26.39";`
  
- ❑ `oConn.Open();`
- ❑ `myCommand.ExecuteNonQuery();`
- ❑ `oConn.Close();`
  
- ❑ `Select_Click(Source, E);`