Software Engineering Lab

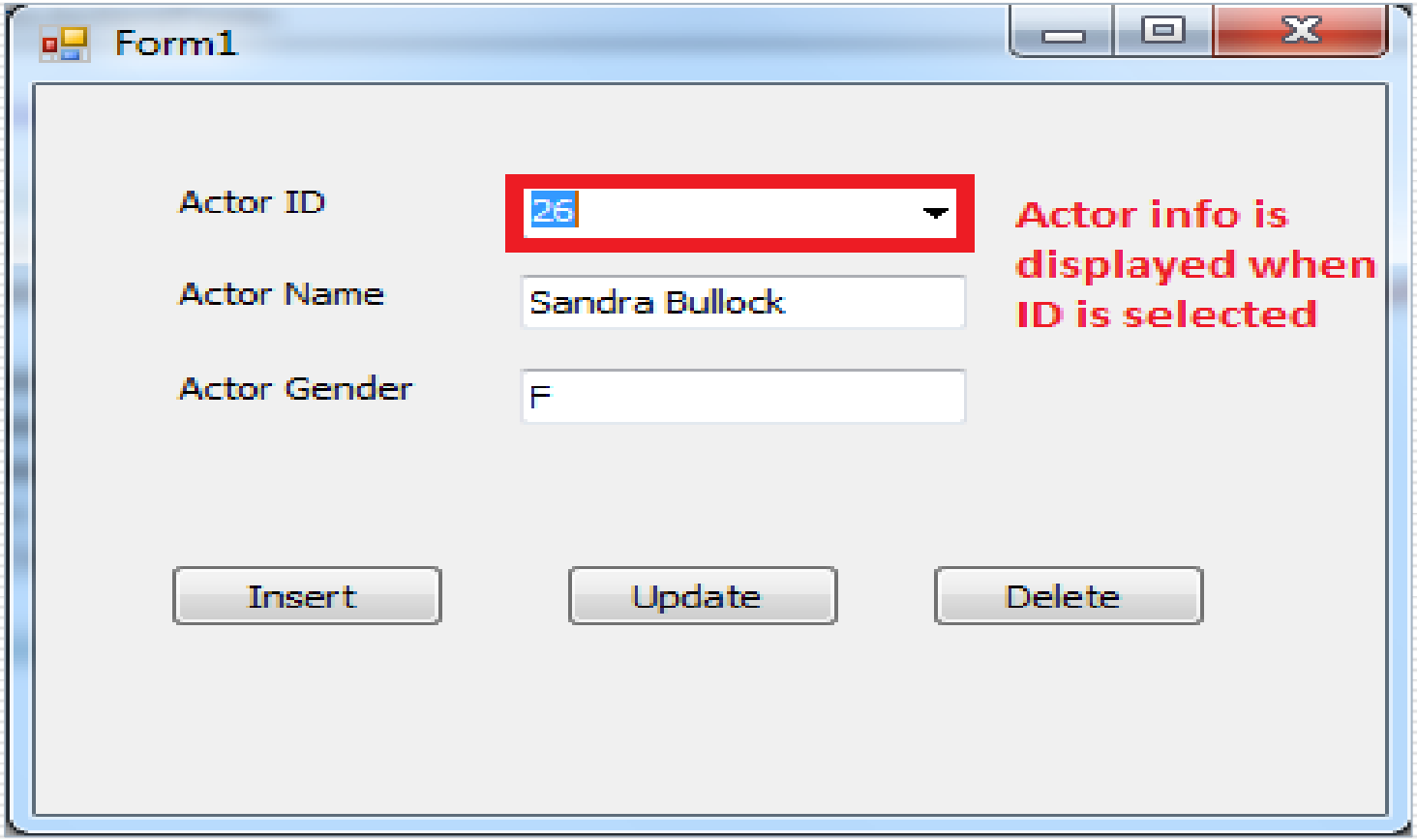
Building .NET Applications on the Oracle Database with Microsoft Visual Studio

Lab 2 – ODP.Net
DML statements in connected layer

Agenda

- Retrieving Actor Information for selected ID using Bind Variables
- DML Operations using Bind Variables
  - Inserting New Actor Information
  - Updating Actor Information for selected ID
  - Deleting Actor Information for selected ID

1. Retrieving Actor Info for the selected ID



Retrieving Actor Info for the selected ID

- ComboBox **SelectedIndexChanged** Event:

  Occurs when the SelectedIndex property of the comboBox has changed. (When the user selects value from the list)

Parameterized Query

- In order to pass values from the c# application to the SQL statement, we should use:

  1. Bind Variables in the SQL Statement of the OracleCommand
  2. Put the value in a parameter and add it to the OracleCommand

Parameterized Query: 1- Bind Variables

- Bind variables allow you to create *parameterized queries* where user input is used to build the SQL statement.

- Bind variables can be used with *SELECT*, *INSERT*, *UPDATE* , and *DELETE* statements

Parameterized Query: 1- Bind Variables (cont.)

- Bind variables are placeholders in a SQL statement that begin with a single colon (":")

```
OracleCommand c = new OracleCommand();
c.Connection = conn;
c.CommandText = "select actorname,gender
 from Actors where ActorID=:id";
c.CommandType = CommandType.Text;
```

Parameterized Query: 2- Oracle Parameters

- In order to use bind variables from your ODP.NET programs, use the OracleParameter objects associated with the OracleCommand.

```
c.Parameters.Add("id", cmb_ID.SelectedItem.ToString());
```

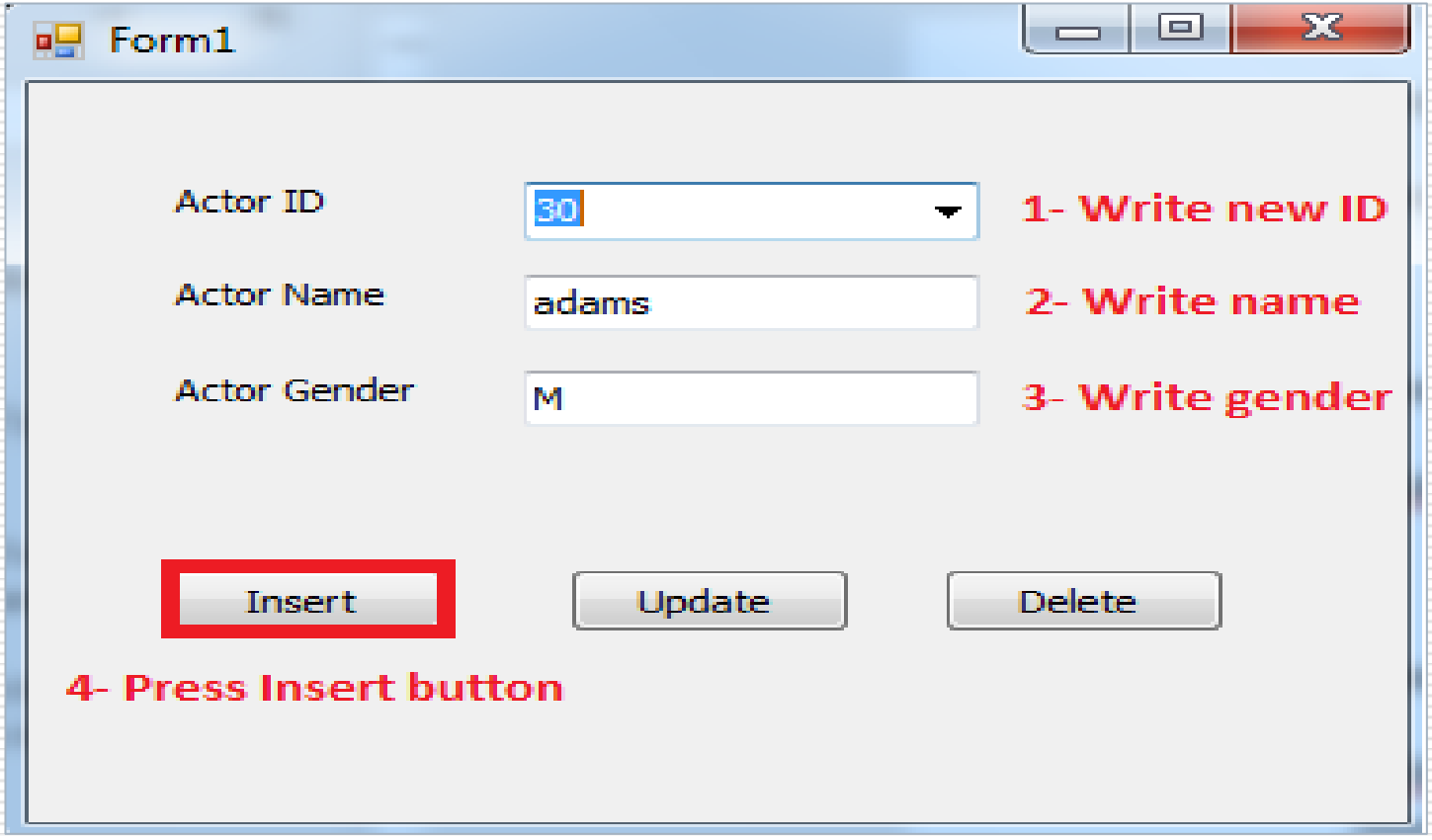
Retrieving Actor Info for the selected ID

```
private void cmb_ID_SelectedIndexChanged(object sender, EventArgs e)
{
    OracleCommand c = new OracleCommand();
    c.Connection = conn;
    c.CommandText = "select actorname,gender from Actors where ActorID=:id";
    c.CommandType = CommandType.Text;
    c.Parameters.Add("id", cmb_ID.SelectedItem.ToString());
    OracleDataReader dr = c.ExecuteReader();
    if (dr.Read())
    {
        txt_Name.Text = dr[0].ToString();
        txt_Gender.Text = dr[1].ToString();
    }
    dr.Close();
```

2- Adding New Actor

(Insert Statement)

Adding New Actor (Insert)



Before inserting new ID, make sure that it doesn't exist in the combobox

Parameterized Query: 1- Bind Variables

- Bind variables are placeholders in a SQL statement that begin with a single colon (":")

```
OracleCommand cmd = new OracleCommand();
cmd.Connection = conn;
cmd.CommandText = "insert into Actors
                   values (:id,:name,:gender)";
```

Parameterized Query: 2- Oracle Parameters

- In order to use bind variables from your ODP.NET programs, use the OracleParameter objects associated with the OracleCommand.

```
cmd.Parameters.Add("id", cmb_ID.Text);
cmd.Parameters.Add("name", txt_Name.Text);
cmd.Parameters.Add("gender", txt_Gender.Text);
```

- you must add the parameters in the ***same order*** as they appear in the SQL statement.

ExecuteNonQuery()

- This method executes a SQL statement and returns the number of rows affected.

- Used with *Insert*, *Update*, *Delete*

```
int r = cmd.ExecuteNonQuery();
```

- It returns -1 if no rows are affected

Adding New Actor (Insert)

```
private void btn_Insert_Click(object sender, EventArgs e)
{
    OracleCommand cmd = new OracleCommand();
    cmd.Connection = conn;
    cmd.CommandText = "insert into Actors values (:id,:name,:gender)";
    cmd.Parameters.Add("id", cmb_ID.Text);
    cmd.Parameters.Add("name", txt_Name.Text);
    cmd.Parameters.Add("gender", txt_Gender.Text);
    int r = cmd.ExecuteNonQuery();
    if (r != -1)
    {
        cmb_ID.Items.Add(cmb_ID.Text);
        MessageBox.Show("New Actor is added");
    }
}
```

After the new actor has been inserted successfully, add its ID to the combobox

3- Updating Actor Information for a selected ActorID

(Update Statement)

Updating Actor Information for a selected ActorID

- Remember Update syntax:

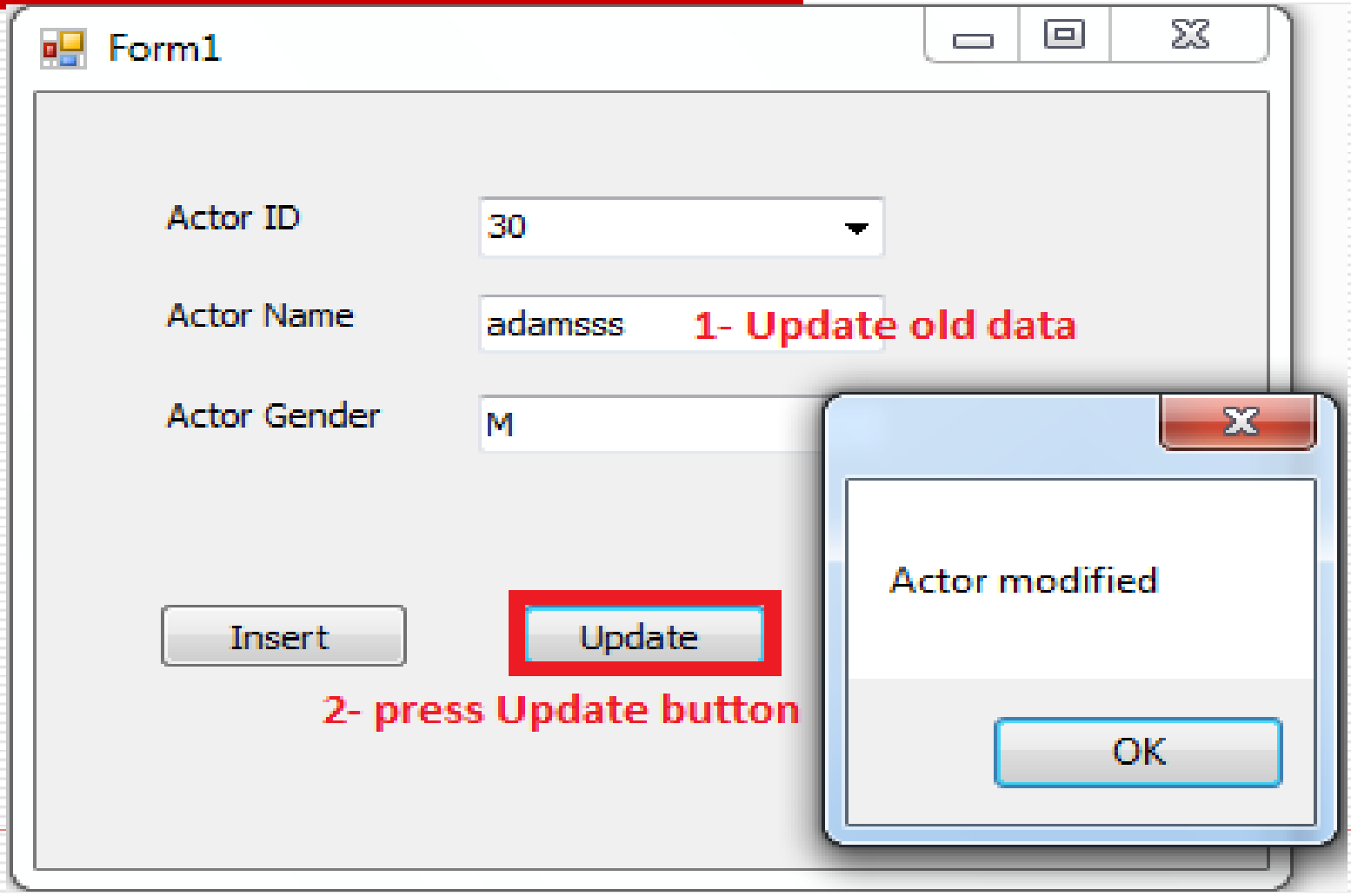
```
UPDATE      table
SET         column = value [, column = value, ...]
[WHERE      condition];
```

Updating Actor Information

```
private void btn_Update_Click(object sender, EventArgs e)
{
    OracleCommand c = new OracleCommand();
    c.Connection = conn;
    c.CommandText = "update Actors set ActorName=:name, gender=:gender where ActorID =:id";

    c.Parameters.Add("name", txt_Name.Text);
    c.Parameters.Add("gender", txt_Gender.Text);
    c.Parameters.Add("id", cmb_ID.SelectedItem.ToString());
    int r = c.ExecuteNonQuery();
    if (r != -1)
    {
        MessageBox.Show("Actor modified");
    }
}
```

Updating Actor Information



4- Deleting Actor Information for a selected ActorID

(Delete Statements)

Deleting Actor Information for a selected ActorID

- Remember Delete syntax:

```
DELETE [FROM]   table
[WHERE          condition];
```

Deleting Actor Information

```
private void btn_Delete_Click(object sender, EventArgs e)
{
    // after the row is successfully deleted, you should clear the values of the controls on the form
    OracleCommand c = new OracleCommand();
    c.Connection = conn;
    c.CommandText = "Delete from Actors where ActorID=:id";
    c.Parameters.Add("id", cmb_ID.Text);
    int r = c.ExecuteNonQuery();
    if (r != -1)
    {
        MessageBox.Show("Actor deleted");
        cmb_ID.Items.RemoveAt(cmb_ID.SelectedIndex);
        txt_Name.Text = "";
        txt_Gender.Text = "";
    }
}
```

Thank You