TUNKU ABDUL RAHMAN UNIVERSITY OF MANAGEMENT AND TECHNOLOGY

Database Programming

Chapter 4 (Part 2)

What Are You Going To Learn?

- At the end of this lesson, you will be able to:
  - Create, retrieve, update and delete (CRUD) records on database.
  - Advanced Database Handling
  - use data bound controls
  - apply data binding to data bound controls, such as dropdownlist

Common Database Task

- In this section, you will learn to perform common database tasks using the ADO.NET's data objects:
  - Create & open a database connection.
  - **C:** Create/Add new database records.
  - **R:** Retrieve & display database records.
  - **U:** Update existing database records.
  - **D:** Delete database records.

Create (Add) a Database Record

- 4 steps to create a database record:
  1. Create & open a database connection.
  2. Create SQL **Insert** **statement** and **SqlCommand** object.
  3. Execute the Command to add the record into database
     - Display the status (success?).
  4. Close the database connection

Add a New Database Record

Step 2: Create command object

- SQL Insert Statement (Syntax):

```
Insert Into Tablename (field1, field2,….)
Values (value1, value2,…)
```

- Example:

```
Insert Into Student (Id, name, programme)
Values (‘S001’, ‘Zhen Mei Li’, ‘RSD2’)
```

Add Record with Parameters

Step 2: Create command object

- If the values are obtained from Textboxes

```
string strAdd = “Insert Into Student (Id, name, programme)
Values (@ID, @name, @prog)”;

SqlCommand cmdAdd = new SqlCommand(strAdd,con);

cmdAdd.Parameters.AddWithValue(“@ID”, txtID.Text);
cmdAdd.Parameters.AddWithValue(“@name”, txtName.Text);
cmdAdd.Parameters.AddWithValue(“@prog”, txtProg.Text);
```

Add a New Database Record

Step 3: Execute the Command

- Use *ExecuteNonQuery* method

```
string strAdd = “Insert Into Student (Id, name, programme) Values
(@ID, @name, @prog)”;
SqlCommand cmdAdd = new SqlCommand(strAdd,con);
cmdAdd.Parameters.AddWithValue(@ID, txtID.Text);
cmdAdd.Parameters.AddWithValue(@name, txtName.Text);
cmdAdd.Parameters.AddWithValue(@prog, txtProg.Text);
int n = cmdAdd.ExecuteNonQuery();
if(n>0)
        Response.Write(“Record is added”);
else
        Response.Write(“Ops! Unable to insert record.”);
```

this returns the number of rows affected

Usability heuristic No 1 - Visibility of system status

Update (Edit) a Database Record

- 4 steps to create a database record:
  1. Create & open a database connection.
  2. Create SQL **Update** **statement** and **SqlCommand** object.
  3. Execute the Command to add the record into database
     - Display the status (success?).
  4. Close the database connection

Update a Database Record

Step 2: Create command object

- SQL Update Statement (Syntax):

```
Update table
Set  column1 = value1, column2 = value2…..
Where search condition
```

- Example:
  - given a table Student (**Id**, name, programme)

```
Update Student
Set name= ‘Hun Mei Li’, programme=‘RSD1’
Where Id = ‘S001’
```

Update Several Database Records

- Unlike SQL Insert command, Update command might affect more than one record at a time.

```
Update Event
Set Status = ‘open’
Where Eventtype =‘contest’
```

Id	Title	Eventtype	Status
1	Badminton Tom’s Cup	contest	open
2	IT Fair	exhibition	close
3	Imagine Cup	contest	open

Event table

Add Record with Parameters

Step 2: Create command object

- If the values are obtained from Textboxes

```
string strEdit = “Update Student
Set name= @name, programme=@prog Where Id = @ID”;

SqlCommand cmdAdd = new SqlCommand(strEdit,con);

cmdAdd.Parameters.AddWithValue(@ID, txtID.Text);
cmdAdd.Parameters.AddWithValue(@name, txtName.Text);
cmdAdd.Parameters.AddWithValue(@prog, txtProg.Text);
```

Update Database Records

- Step 1, Step 3 and Step 4 are the same as the steps to Insert a Database Record
- Use *ExecuteNonQuery* method to execute the command

Delete a Database Record

Step 2: Create command object

- SQL Delete Statement (Syntax):

```
Delete From table Where search condition
```

- Example:

```
Delete From Student Where Id= ‘S001’
```

Delete Several Database Records

- Similar to SQL Update command, Delete command might affect more than one record at a time.

```
Delete From Event
Where Eventtype =‘contest’
```

Id	Title	Eventtype	Status
~~1~~	~~Badminton Tom’s Cup~~	~~contest~~	~~open~~
2	IT Fair	exhibition	close
~~3~~	~~Imagine Cup~~	~~contest~~	~~open~~

Event table

Delete Database Records

- Step 1, Step 3 and Step 4 are the same as the steps to Insert a Database Record
- Use *ExecuteNonQuery* method to execute the command

Question

Assume that you are going to build a Promotion module. Suggest which SQL statement (insert, select, update, delete) and method (ExecuteReader, ExecuteScalar and ExecuteNonQuery) would you use to:

1. display the number of products cost below RM1
2. display the details of the products that are going to expire soon, such as name and price
3. add a new promotion details
4. edit a product’s selling price
5. clear all the products that are already expired from the database

Next Week

- Using Data bound controls with Data Source controls
- Data binding on data bound controls