Module 6

Accessing a Database

Microsoft

Module Overview

- Creating and Using Entity Data Models
- Querying Data by Using LINQ

Lesson 1: Creating and Using Entity Data Models

- Introduction to the ADO.NET Entity Framework
- Using the ADO.NET Entity Data Model Tools
- Demonstration: Creating an Entity Data Model
- Customizing Generated Classes
- Reading and Modifying Data by Using the Entity Framework

Introduction to the ADO.NET Entity Framework

- The ADO.NET Entity Framework provides:
  - EDMs
  - Entity SQL
  - Object Services
- The ADO.NET Entity Framework supports:
  - Writing code against a conceptual model
  - Easy updating of applications to a different data source
  - Writing code that is independent from the storage system
  - Writing data access code that supports compile-time type-checking and syntax-checking

Using the ADO.NET Entity Data Model Tools

- Tools support:
  - Database-first design by using the Entity Data Model Wizard
  - Code-first design by using the Generate Database Wizard
- They also provide:
  - Designer pane for viewing, updating, and deleting entities and their relationships
  - Update Model Wizard for updating a model with changes that are made to the data source
  - Mapping Details pane for viewing, updating, and deleting mappings

Text Continuation

Text Continuation

Customizing Generated Classes

- Do not modify the automatically generated classes in a model
- Use partial classes and partial methods to add business functionality to the generated classes

```
public partial class Employee
{
    partial void OnDateOfBirthChanging(DateTime? value)
    {
        if (GetAge() < 16)
        {
            throw new Exception("Employees must be 16 or over");
        }
    }
}
```

Reading and Modifying Data by Using the Entity Framework

- Reading data

```
FourthCoffeeEntities DBContext = new FourthCoffeeEntities();

// Print a list of employees.
foreach (FourthCoffee.Employees.Employee emp in
DBContext.Employees)
{
    Console.WriteLine("{0} {1}", emp.FirstName, emp.LastName);
}
```

- Modifying data

```
var emp = DBContext.Employees.First(e => e.LastName ==
"Prescott");
if (emp != null)
{
    emp.LastName = "Forsyth";
    DBContext.SaveChanges();
}
```

Text Continuation

Lesson 2: Querying Data by Using LINQ

- Querying Data
- Querying Data by Using Anonymous Types
- Demonstration: Querying Data by Using Anonymous Types
- Forcing Query Execution

Querying Data

- Use LINQ to query a range of data sources, including:
  - .NET Framework collections
  - SQL Server databases
  - ADO.NET data sets
  - XML documents
- Use LINQ to:
  - Select data
  - Filter data by row
  - Filter data by column

Text Continuation

Querying Data by Using Anonymous Types

- Use LINQ and anonymous types to:
  - Filter data by column
  - Group data
  - Aggregate data
  - Navigate data

Text Continuation

Forcing Query Execution

- Deferred query execution—default behavior for most queries
- Immediate query execution—default behavior for queries that return a singleton value
- Forced query execution—overrides deferred query execution:
  - **ToArray**
  - **ToDictionary**
  - **ToList**

```
IList<Employee> emp = (from e in FCEntities.Employees
                       orderby e.LastName
                       select e).ToList();
```

Text Continuation

Text Continuation

Lab: Retrieving and Modifying Grade Data

- Exercise 1: Creating an Entity Data Model from The School of Fine Arts Database
- Exercise 2: Updating Student and Grade Data by Using the Entity Framework
- Exercise 3: Extending the Entity Data Model to Validate Data

Logon Information

Estimated Time: 75 minutes

Lab Scenario

You have been asked to upgrade the prototype application to use an existing SQL Server database. You begin by working with a database that is stored on your local machine and decide to use the Entity Data Model Wizard to generate an EDM to access the data. You will need to update the data access code for the Grades section of the application, to display grades that are assigned to a student and to enable users to assign new grades. You also decide to incorporate validation logic into the EDM to ensure that students cannot be assigned to a full class and that the data that users enter when they assign new grades conforms to the required values.

Module Review and Takeaways

- Review Question(s)