شركة تحالف الإمارات للحلول التقنية ذ.م.م.
TAHALUF AL EMARAT TECHNICAL SOLUTIONS L.L.C.

C# Object Oriented Programming

Tahaluf Training Center 2021

Day 9

1. **Accessing a Database**
2. Customizing Generated Classes
3. Reading and Modifying Data
4. Querying Data

The ADO.NET Entity Framework is an object-relational mapping (ORM) framework for the .NET Framework.

Developers can use it to create data access applications by programming against a conceptual application model instead of directly against a relational storage schema.

The ADO.NET Entity Framework provides:

1. EDMs: Entity Data Model , is a client-side data model and it is the core of the Entity Framework.

2. Entity SQL: Entity SQL is another way to create a query. It is processed by the Entity Framework's Object Services directly.

It returns ObjectQuery instead of IQueryable.

The ADO.NET Entity Framework supports:

1. Writing code against a conceptual model
2. Easy updating of applications to a different data source
3. Writing code that is independent from the storage system
4. 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

Day 9

1. Accessing a Database
2. **Customizing Generated Classes**
3. Reading and Modifying Data
4. Querying Data

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");
        }
    }
}
```

شركة تحالف الإمارات للحلول التقنية ذ.م.م.
TAHALUF AL EMARAT TECHNICAL SOLUTIONS L.L.C.

Day 9

1. Accessing a Database
2. Customizing Generated Classes
3. **Reading and Modifying Data**
4. Querying Data

- 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();
}
```

Day 9

1. Accessing a Database
2. Customizing Generated Classes
3. Reading and Modifying Data
4. **Querying Data**

Querying Data

شركة تحالف الإمارات للحلول التقنية ذ.م.م.
TAHALUF AL EMARAT TECHNICAL SOLUTIONS L.L.C.

- A query is a request for data or information from a database table or combination of tables. This data may be generated as results returned by Structured Query Language (SQL) or as pictorials, graphs or complex results, e.g., trend analyses from data-mining tools.

- One of several different query languages may be used to perform a range of simple to complex database queries.

Querying a Collection

- Use LINQ expressions to query collections

```
var drinks =
    from string drink in prices.Keys
    orderby prices[drink] ascending
    select drink;
```

Forcing Query Execution

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

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

On the E-Learning Portal