Session 4

Working with ADO.NET and Entity Framework
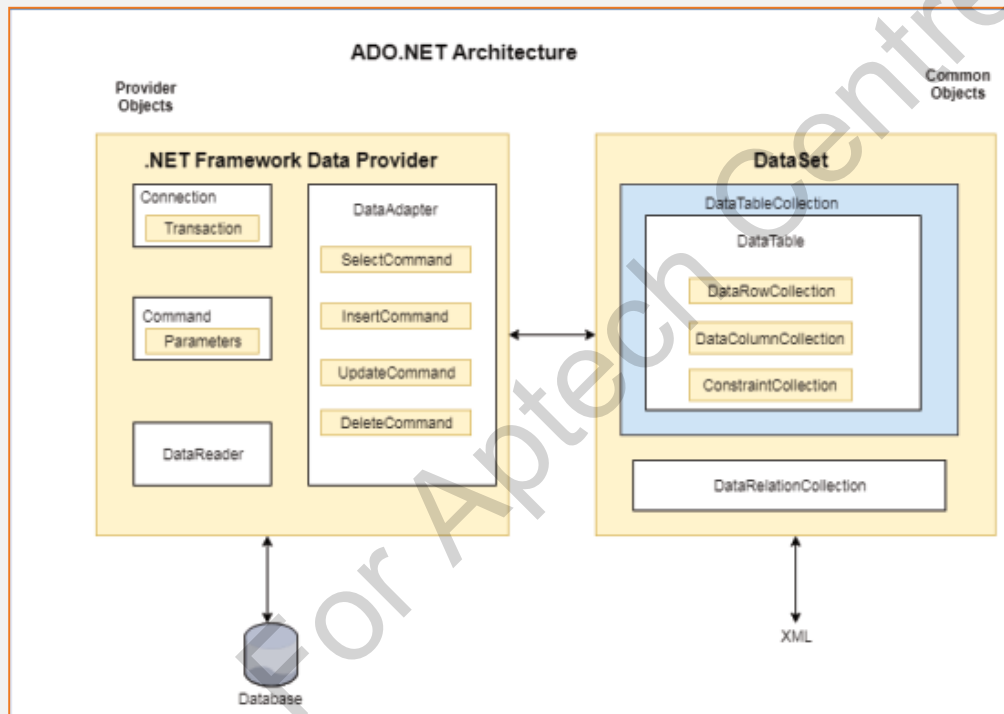
Session Overview

- Explain ADO.NET

- Describe how to connect to a database

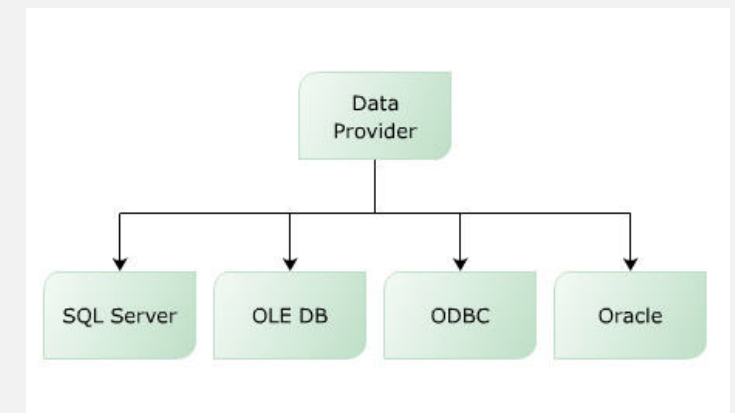- Explain how to access data with ADO.NET

- Define the Entity Framework

- Core component of .NET Framework
- Is used for establishing a connection between an application and data sources

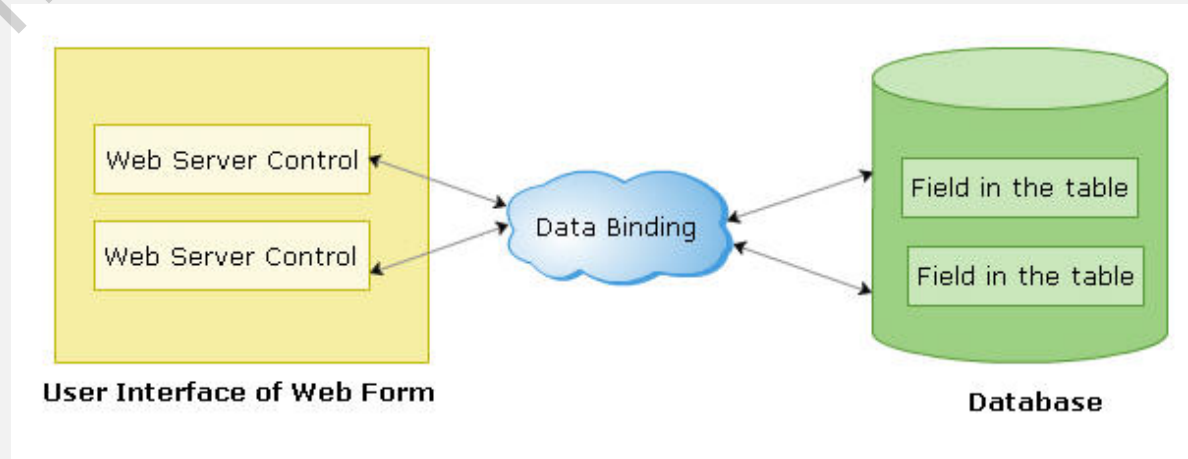*ADO.NET Architecture*

Data Providers

Advantages of ADO.NET are:

Single Object-oriented API

Managed Code

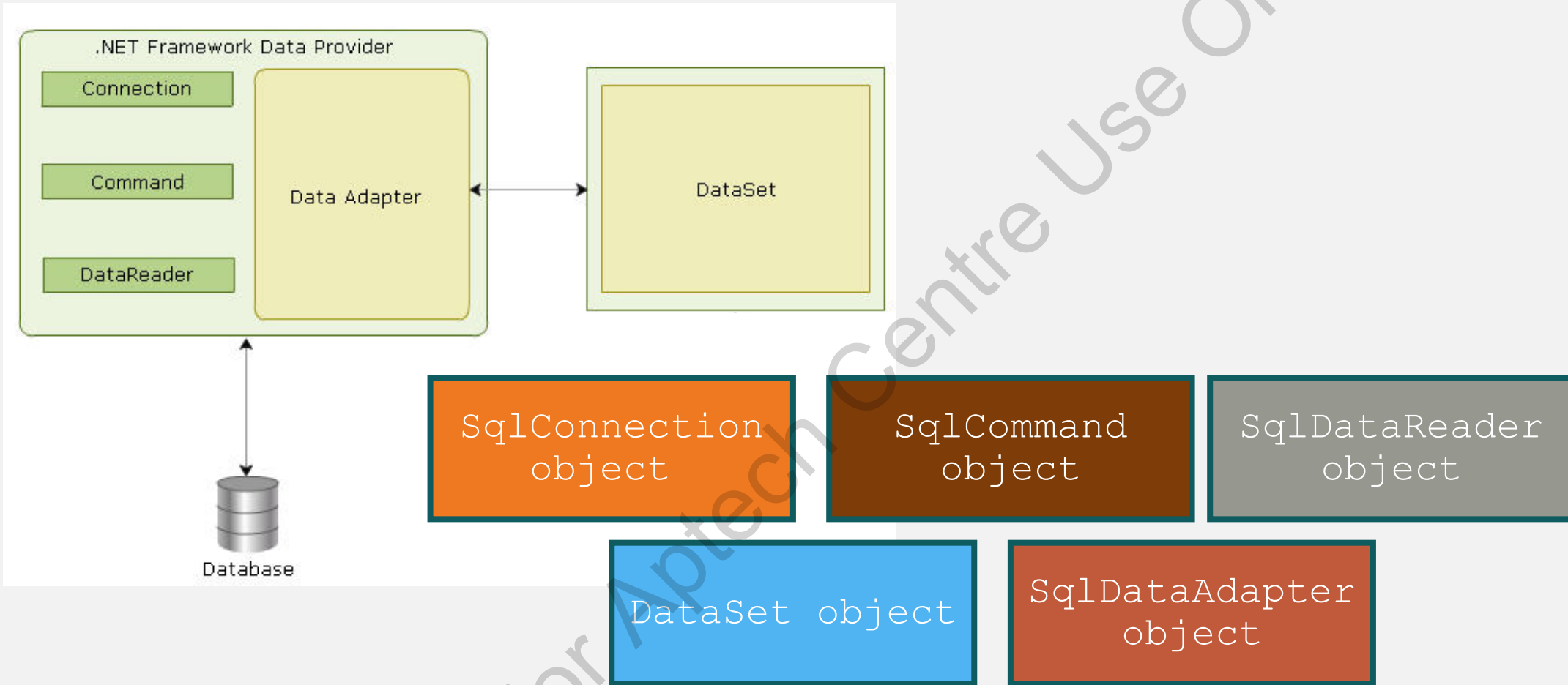Extensible Markup Language (XML) Support

Visual Data Components

Performance and Scalability

Web Server Control

Web Server Control

Data Binding

Field in the table

Field in the table

User Interface of Web Form

Database

SqlConnection object

SqlCommand object

SqlDataReader object

DataSet object

SqlDataAdapter object

Client

Need Connection Object for Connection

Database

Parameter Name	Description
Data Source	Local machine, machine domain name, or IP address that identifies the server
Initial Catalog	Name of the database
Integrated Security	Often set to SSPI to make a connection with the user's Window login. This parameter is optional
User ID	User's ID configured in SQL Server
Password	Password matching SQL Server UserID
Connection String	

Connecting to a Database (3-6)

Establish the SqlConnection

Open the connection

Transfer the connection to other ADO.NET objects

Execute database operations with other ADO.NET objects

Close the connection
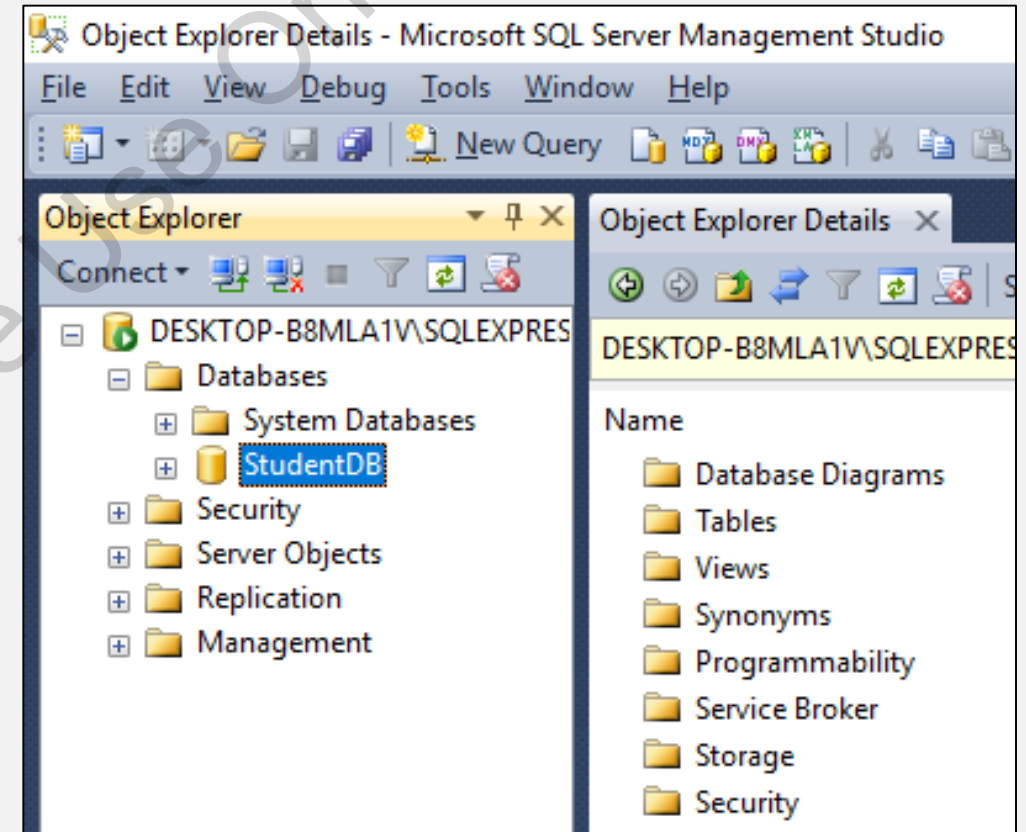
Executing SQL Commands using the Command Object

Client
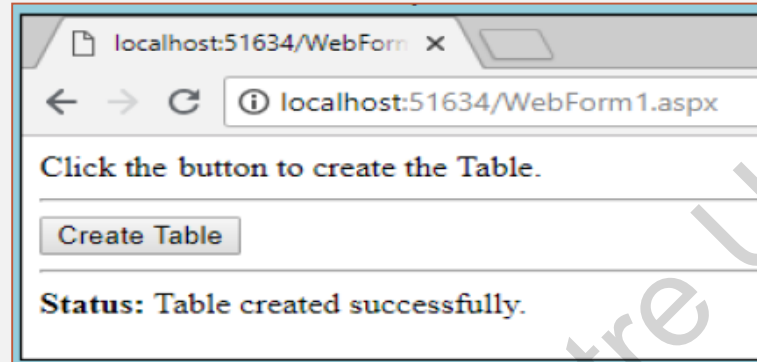
Database

Connecting to a Database (4-6)
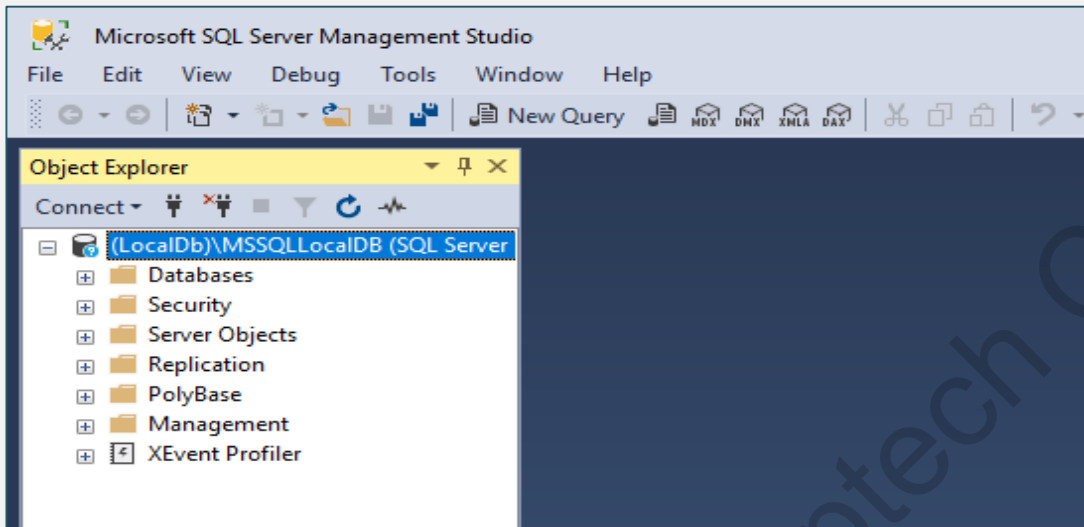
SQL Server Management Tool

StudentDB Database Created
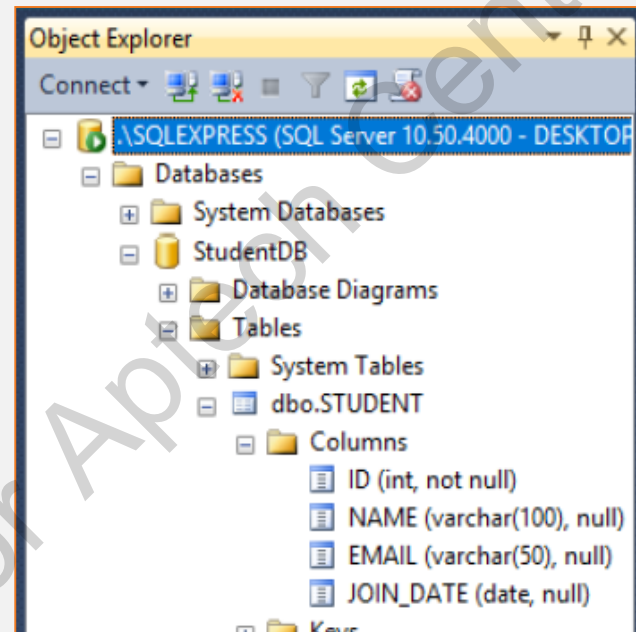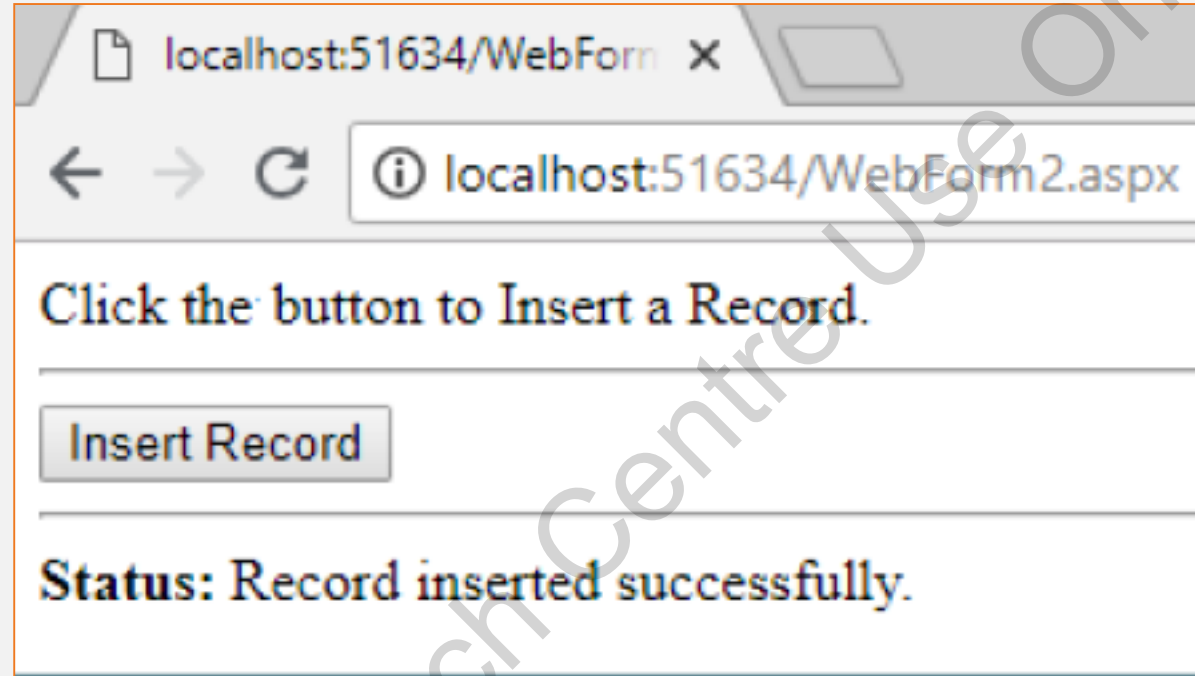
Connecting to a Database (5-6)

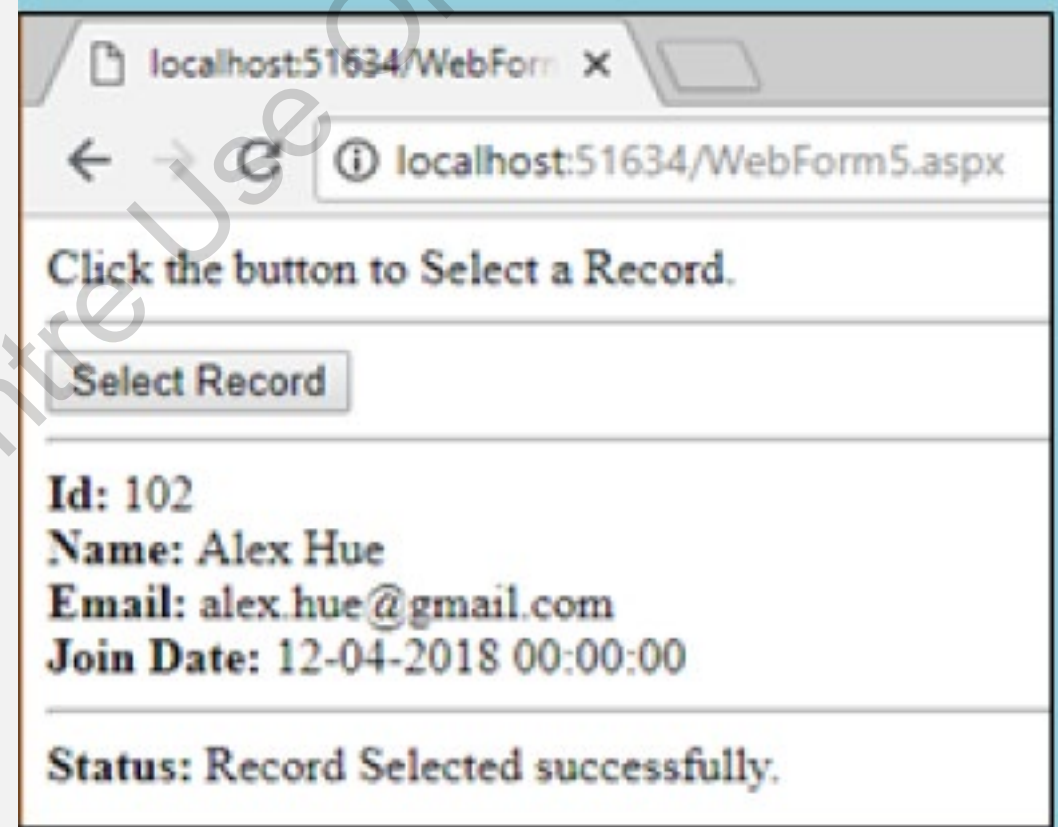*Create Table Web Page*

Table Structure in SSMS

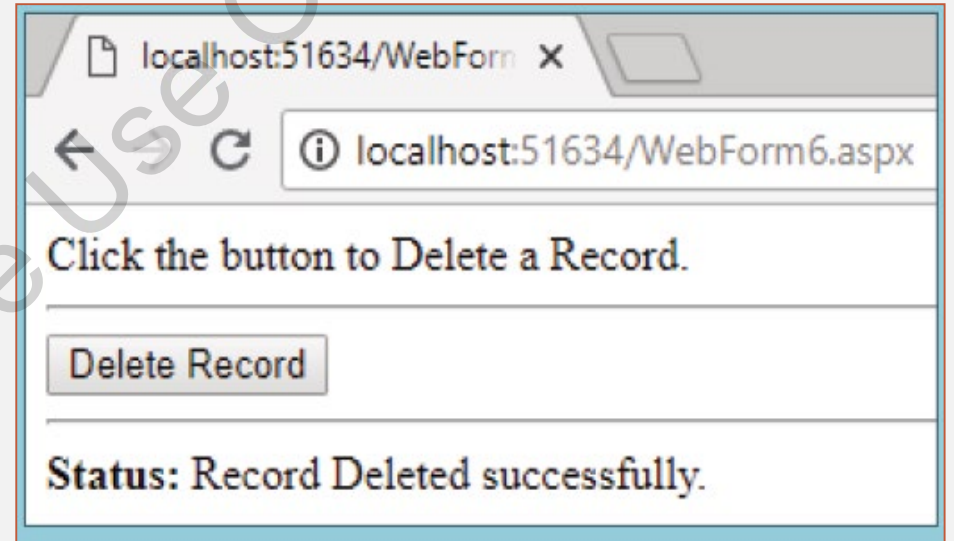# Connecting to a Database (6-6)

Insert Record Web Page

Accessing Data with ADO.NET (1-2)

- Retrieving data inserted using C# code

Select Record Web Page

- Deleting a row using C# code

Delete Record Web Page

Entity Framework (1-2)

Entity Framework (EF) is an Object Relational Mapping (ORM) tool that is used to connect to the database.

In the EF, the ADO.NET methods and classes used for performing data operations are not written by the developer.

Using EF, business objects and entities are generated according to the database tables.

It provides the mechanism for:

- Working with basic CRUD operations
- Supporting one to one, one to many, and many to many relationships
- Creating inheritance relationships between entities

Entity Framework (2-2)

It provides following benefits:

- Uses higher level languages to write data access logic
- Uses relationships among entities to represent conceptual model in a better way
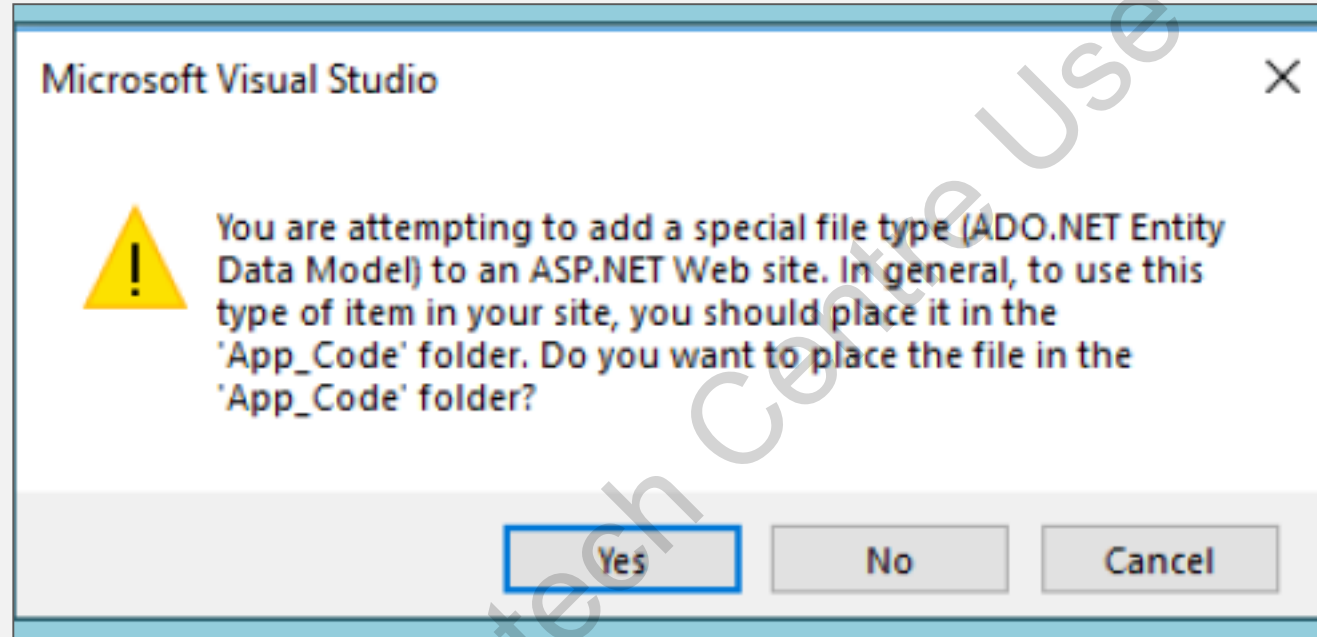- Replaces all underlying data store without much overhead as data access logic is at a higher level

Connection to the database is through the following three approaches:

Database First
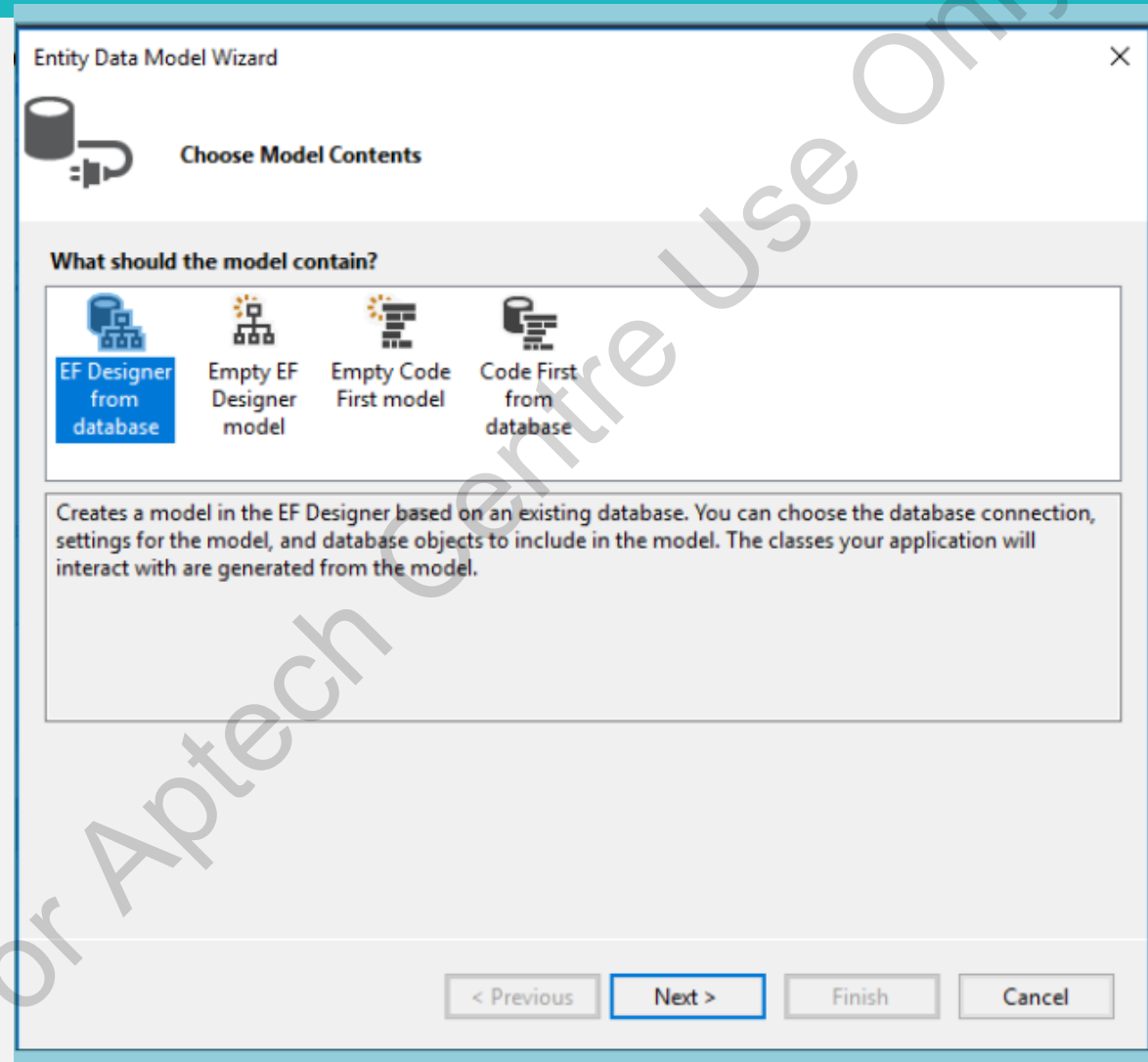
Model First

Code First

Usage in ASP.NET (1-6)

Microsoft Visual Studio Dialog Box

Entity Data Model Wizard

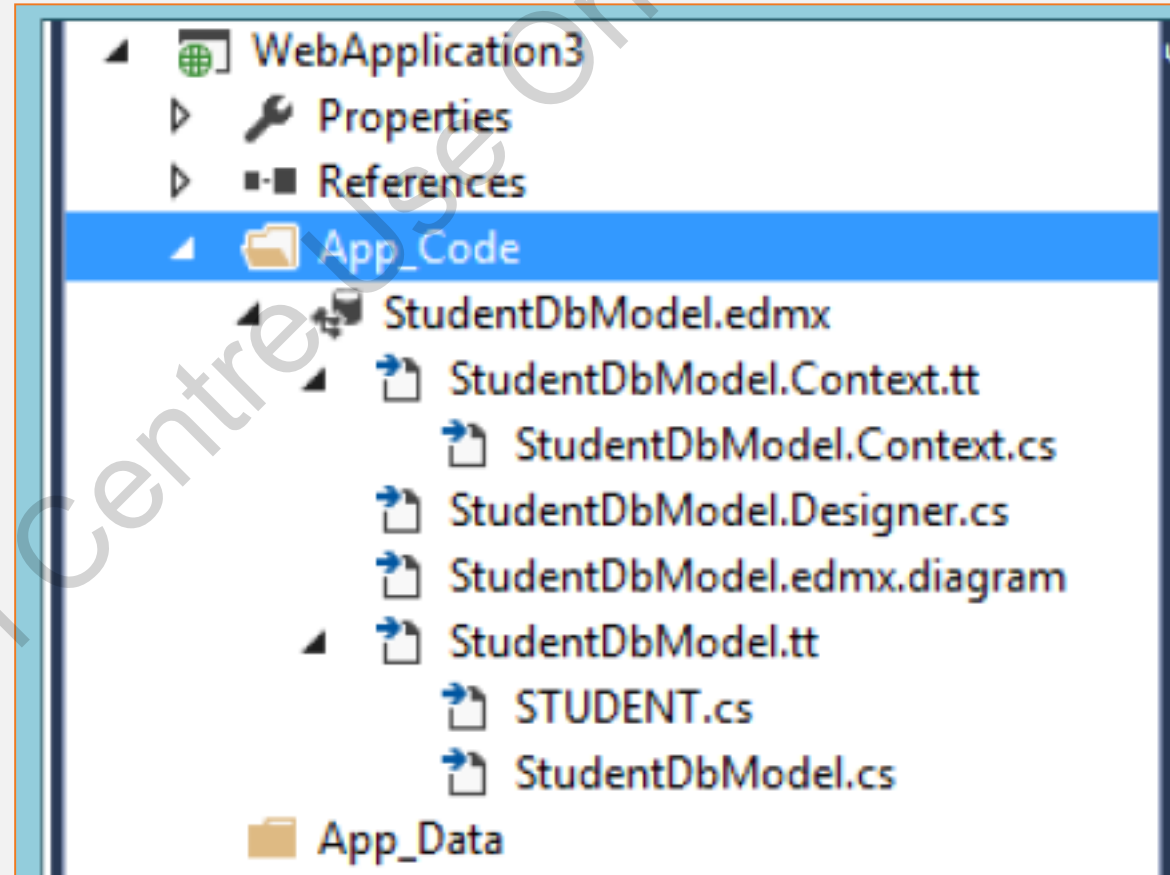*Connection Properties*

Entity Data Model Wizard
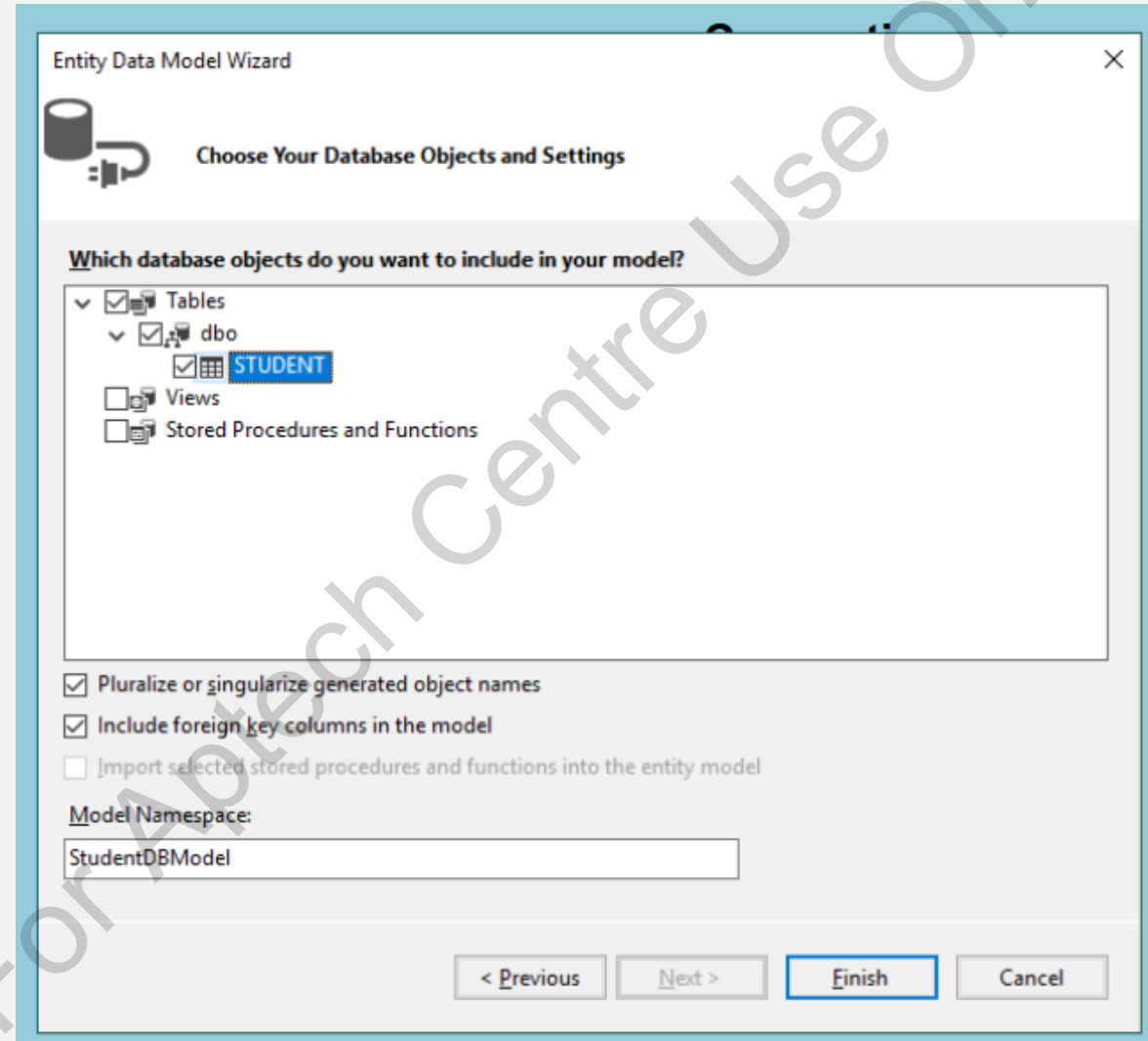
Usage in ASP.NET (5-6)
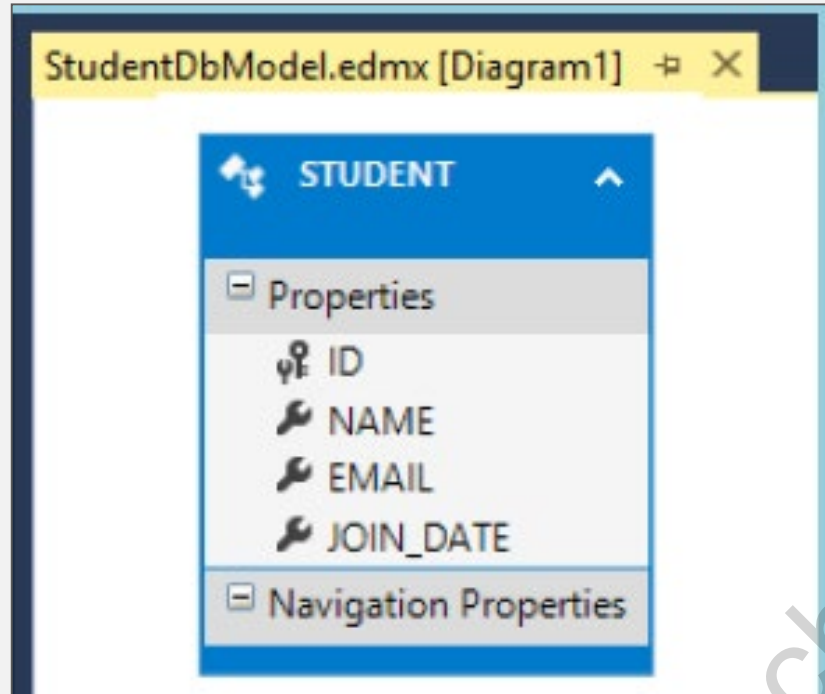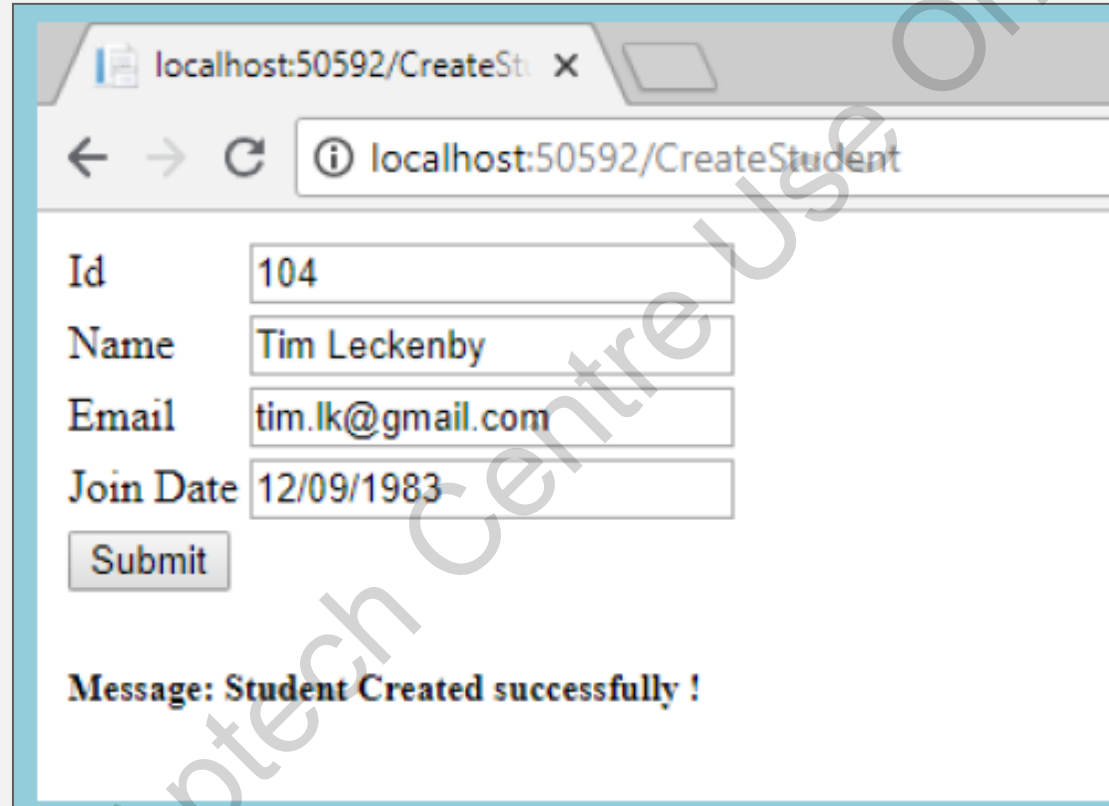
Database Diagram

Models in Solutions Explorer

Usage in ASP.NET (6-6)

Output for CreateStudent Form

Entity Framework Core

EF Core is the new version released by Entity Framework post EF 6.x.

Some of its characteristics include:

- Open-source, Lightweight , Extensible, and Cross-platform

EF Core can be used with:

- .NET Core applications
- Framework-based applications (.NET Framework 4.5+)

Code-First and Database-First are supported by EF Core.

Summary

- ADO.NET is a core component of .NET Framework and is used for establishing a connection between an application and its data sources.

- Various ADO.NET objects for an SQL data source are namely, SqlConnection object, SqlCommand object, SqlDataReader object, DataSet object, and SqlDataAdapter object.

- Command objects use the connection object to know which database to execute the command on.

- SQL statements are sent to the database through a command object.

- Entity Framework is an Object Relational Mapper (ORM). Business objects and entities are generated through this Framework according to the database tables.