MCSE 541: Web Computing and Mining

Entity Framework Core Architecture

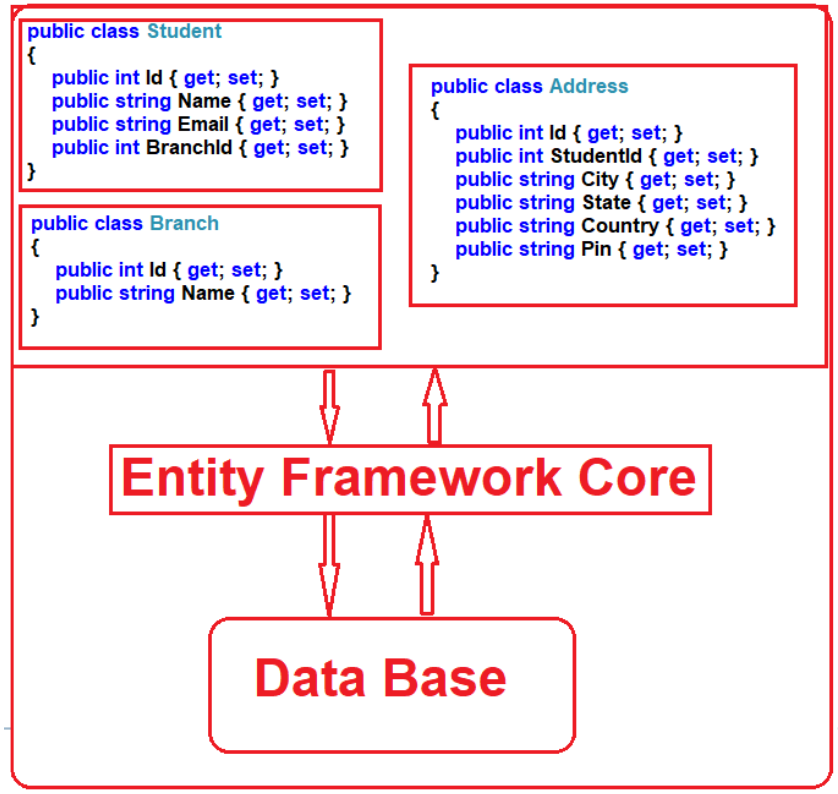
Prof. Dr. Shamim Akhter

EF Core

It is an extensible, lightweight, Open Source, and cross-platform version of Entity Framework data access technology. It works on multiple operating systems like Windows, Mac, and Linus.

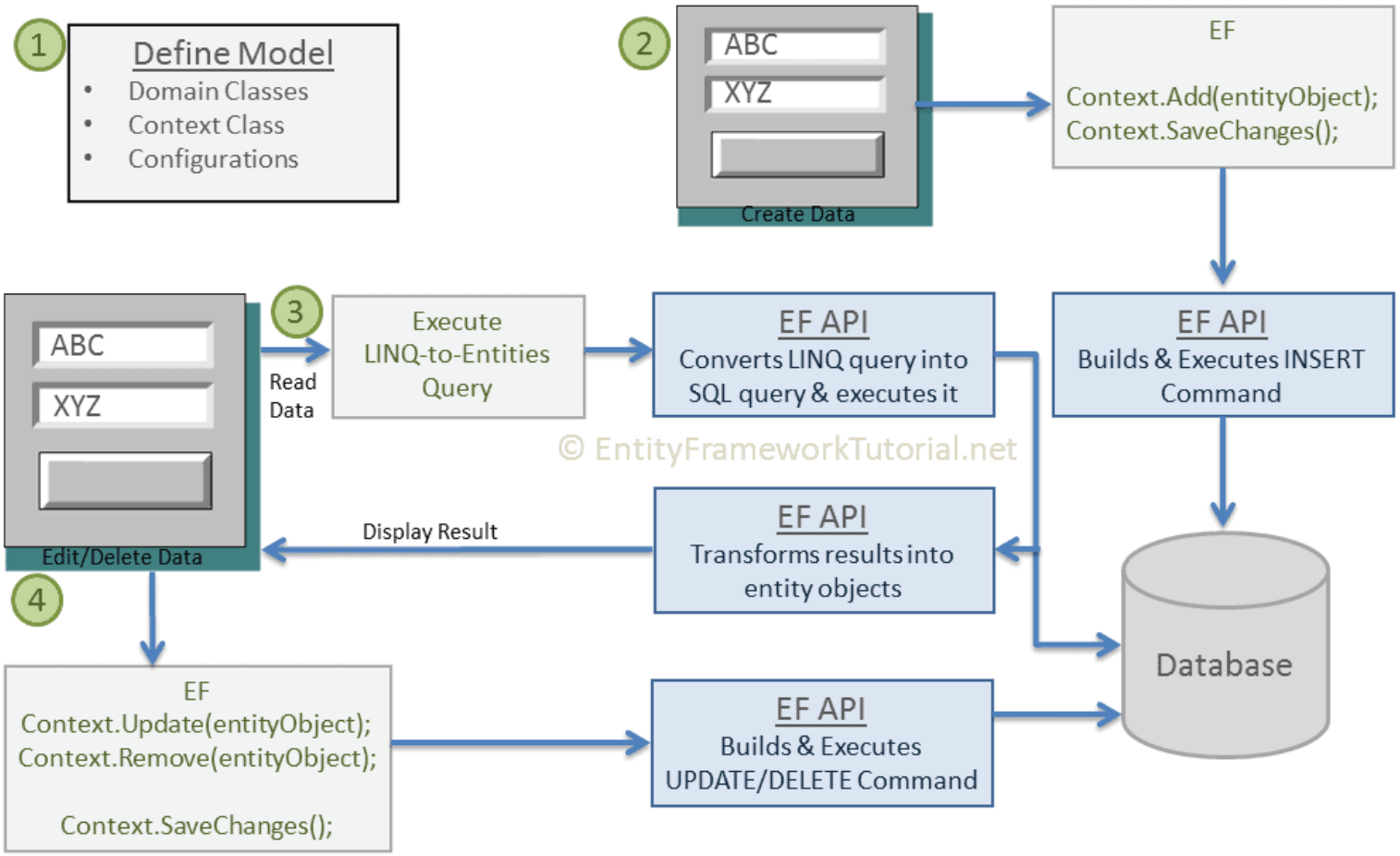
EF is an object-relational mapper (O/RM)

Object-Relational Mapper and it automatically creates classes based on database tables and the vice versa is also true. It can also generate the necessary SQL to create the database based on the classes.



Custom code to map the database data to our model classes like Student, Department, Address, etc. is slow.

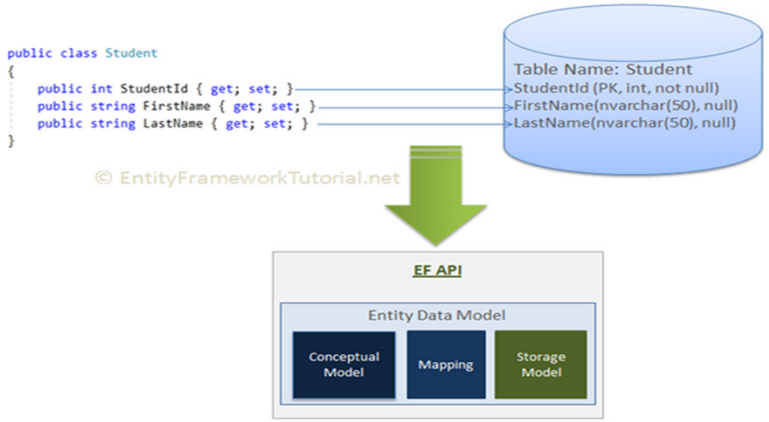
Basic Workflow in Entity Framework

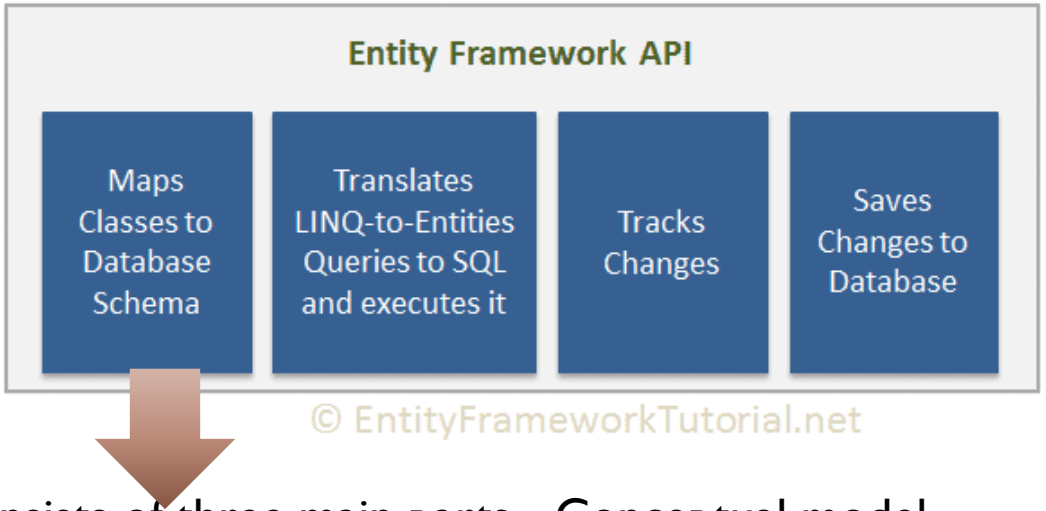


1. First of all, you need to define your model. Defining the model includes defining your domain classes, context class derived from DbContext, and configurations (if any). EF will perform CRUD operations based on your model.
2. To insert data, add a domain object to a context and call the SaveChanges() method. EF API will build an appropriate INSERT command and execute it to the database.
3. To read data, execute the LINQ(Language Integrated Query)-to-Entities query in your preferred language (C#/VB.NET). EF API will convert this query into SQL query for the underlying relational database and execute it. The result will be transformed into domain (entity) objects and displayed on the UI.
4. To edit or delete data, update or remove entity objects from a context and call the SaveChanges() method. EF API will build the appropriate UPDATE or DELETE command and execute it to the database.

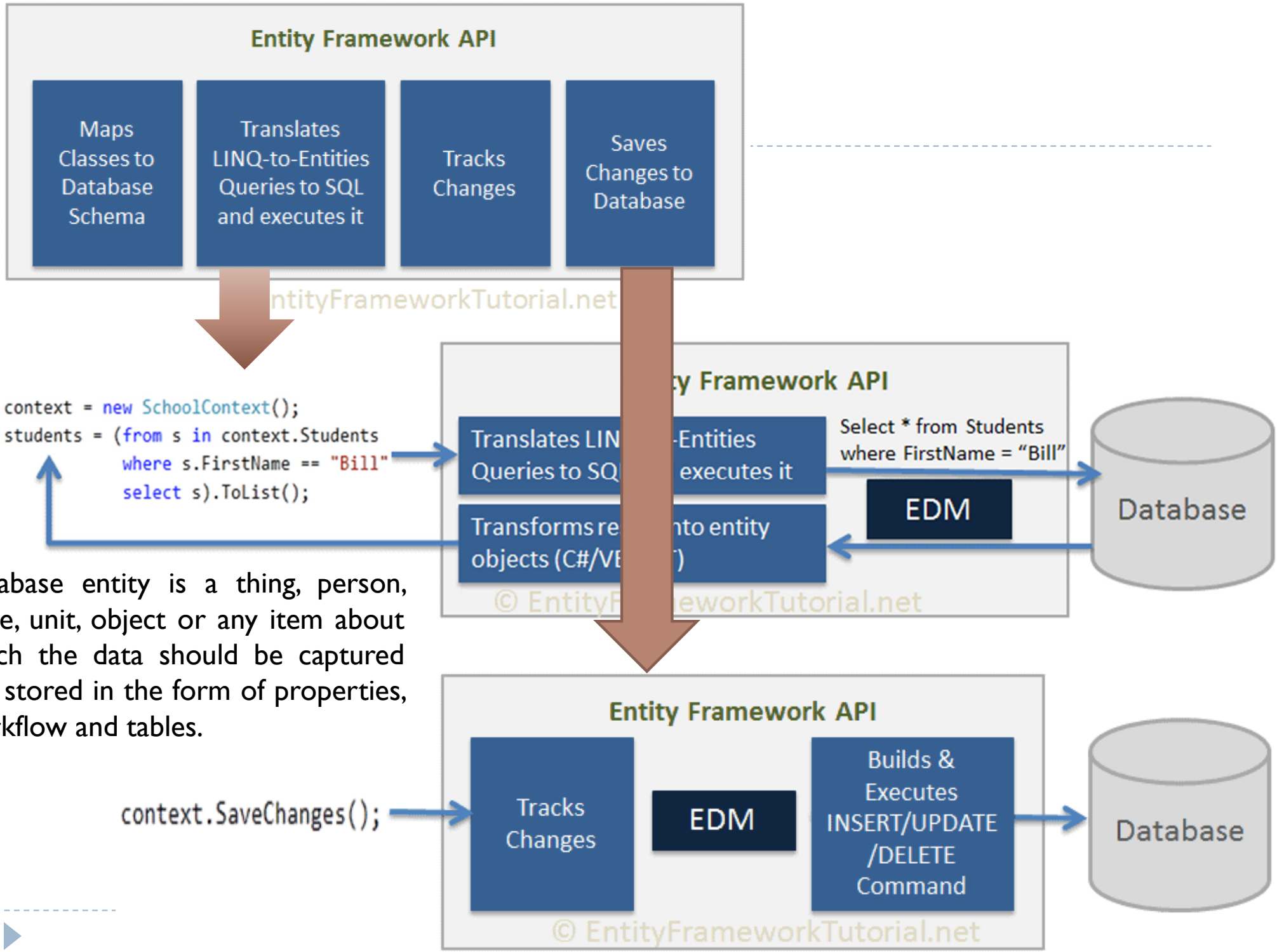
How Does Entity Framework Work?

- EF works on EF API





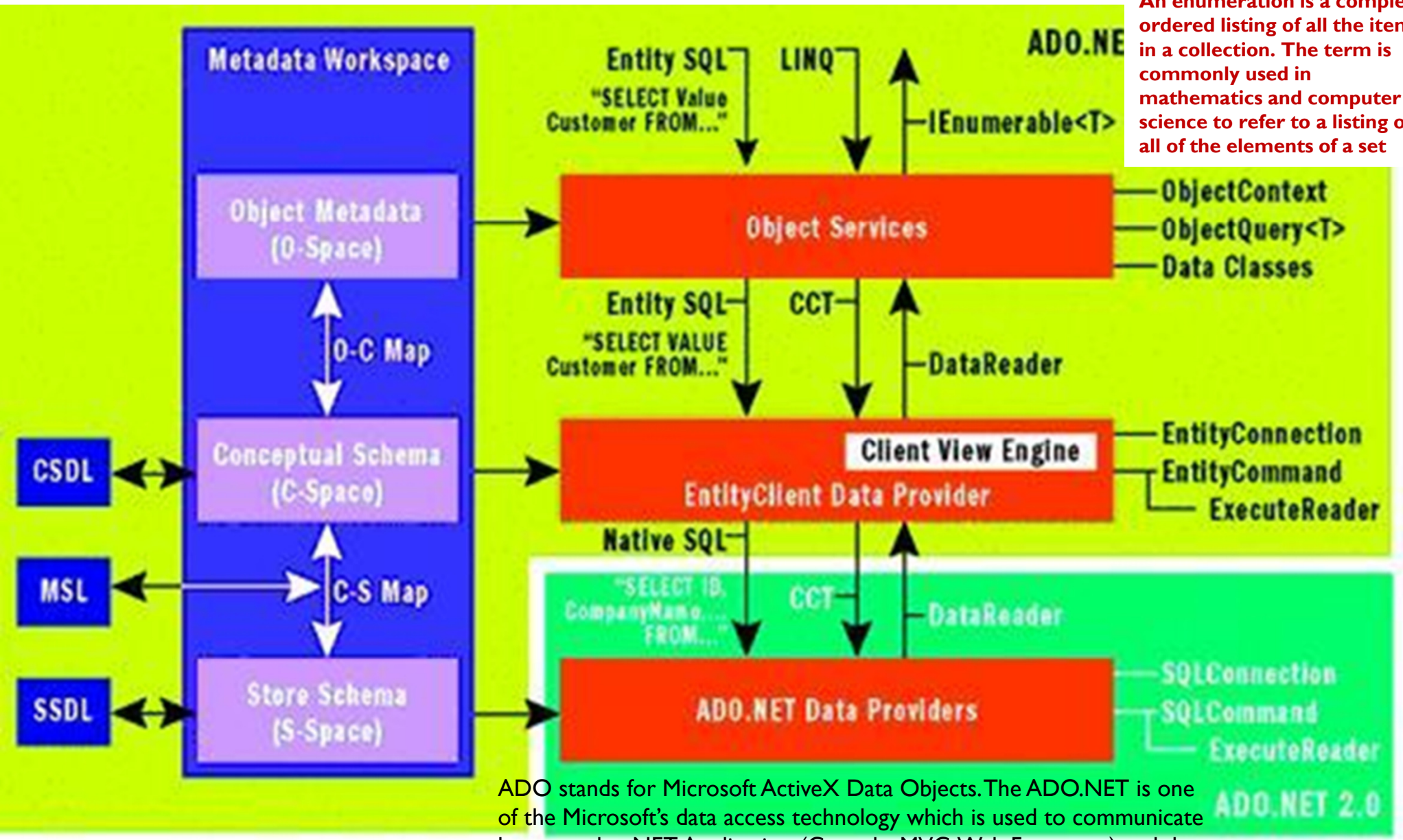
- **EDM (Entity Data Model):** EDM consists of three main parts - Conceptual model, Mapping and Storage model.
  - **Conceptual Model (C-Space)** contains the model classes and their relationships*(In XML). Conceptual Schema Definition Language (.CSDL) is used to map the entity types used in the conceptual model.* This will be independent from your database table design.
  - **Storage Model(S-Space)/Logical Model:** is the database design model which includes tables, views, stored procedures, and their relationships and keys. *Store Schema Definition Language (.SSDL) is used to map the schema information of the logical layer.*
  - In the code-first approach, this will be inferred from the conceptual model. In the database-first approach, this will be inferred from the targeted database.
  - **Mapping(C-S Space)** consists of information about how the conceptual model is mapped to the storage model. *Mapping Schema Language (.MSL) is used to map the logical model to the conceptual model.*


Entity Framework API
Maps Classes to Database Schema
Translates LINQ-to-Entities Queries to SQL and executes it
Tracks Changes
Saves Changes to Database
ntityFrameworkTutorial.net
var context = new SchoolContext();
var students = (from s in context.Students
where s.FirstName == "Bill"
select s).ToList();
y Framework API
Translates LIN -Entities
Queries to SQ executes it
Transforms re nto entity
objects (C#/VB)
Select * from Students
where FirstName = "Bill"
EDM
Database
© EntityF eworkTutorial.net
Database entity is a thing, person, place, unit, object or any item about which the data should be captured and stored in the form of properties, workflow and tables.
Entity Framework API
context.SaveChanges();
Tracks Changes
EDM
Builds & Executes INSERT/UPDATE /DELETE Command
Database
© EntityFrameworkTutorial.net

ADO.Net (ActiveX Data Object.NET (ADO.NET))

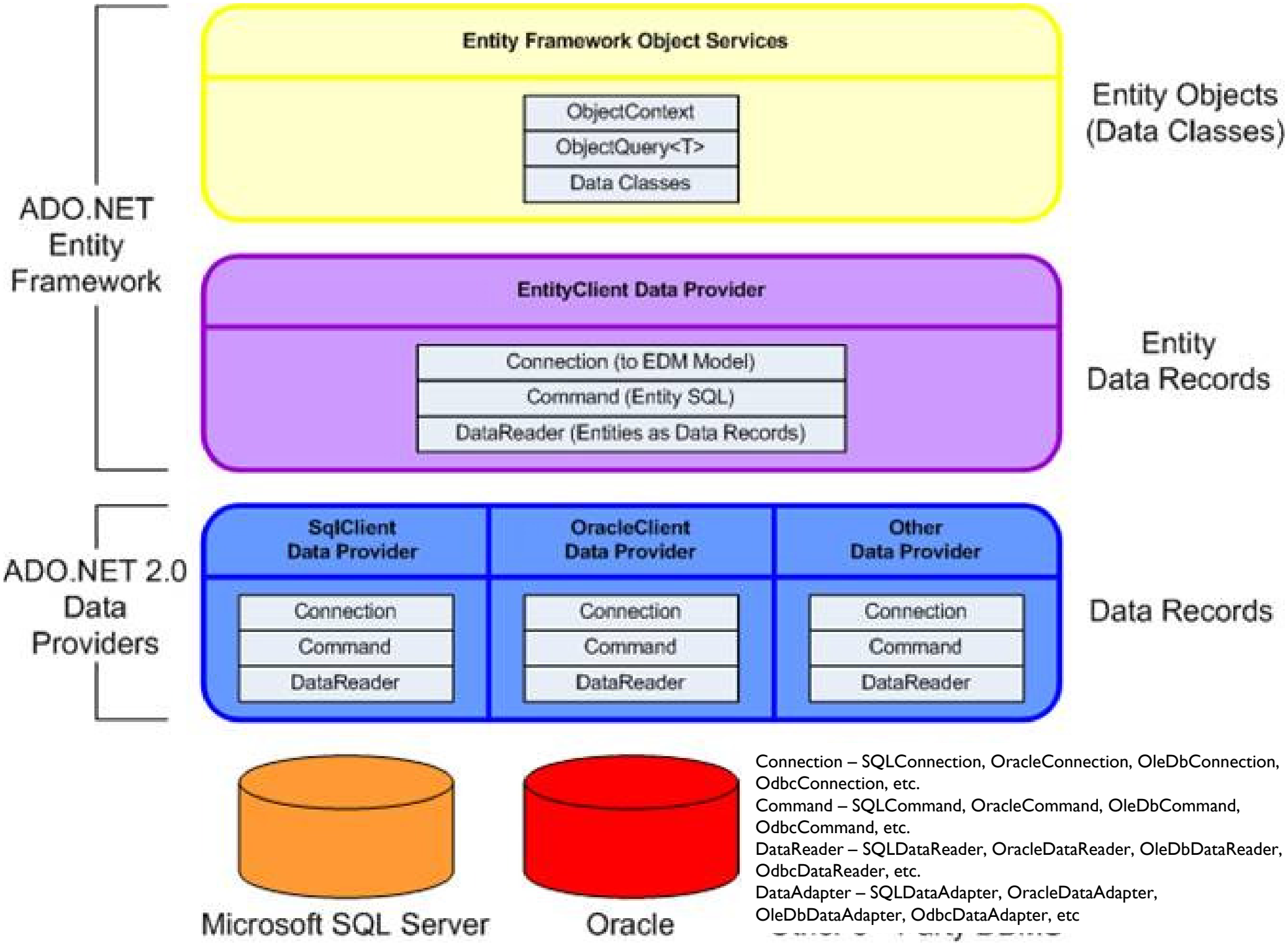
Data Provider

An enumeration is a complete, ordered listing of all the items in a collection. The term is commonly used in mathematics and computer science to refer to a listing of all of the elements of a set



ADO stands for Microsoft ActiveX Data Objects. The ADO.NET is one of the Microsoft's data access technology which is used to communicate between the .NET Application (Console, MVC, Web Form, etc.) and data sources such as SQL Server, Oracle, MySQL, XML document, etc.

- **LINQ to Entities:** LINQ-to-Entities (L2E) is a query language used to write queries against the object model. It returns entities, which are defined in the conceptual model.
- **Entity SQL:** Entity SQL is another query language (For EF 6 only) just like LINQ to Entities. However, it is a little more difficult than L2E and the developer will have to learn it separately.
- **Object Service:** Object service is a main entry point for accessing data from the database and returning it back. Object service is responsible for materialization, which is the process of converting data returned from an entity client data provider (next layer) to an entity object structure.
- **Entity Client Data Provider:** The main responsibility of this layer is to convert LINQ-to-Entities or Entity SQL queries into a SQL query which is understood by the underlying database. It communicates with the ADO.Net data provider which in turn sends or retrieves data from the database.
- **ADO.Net (**ActiveX Data Object.NET (ADO.NET)**) Data Provider:** This layer communicates with the database using standard ADO.Net.


ADO.NET Entity Framework
Entity Framework Object Services
ObjectContext
ObjectQuery<T>
Data Classes
Entity Objects (Data Classes)
EntityClient Data Provider
Connection (to EDM Model)
Command (Entity SQL)
DataReader (Entities as Data Records)
Entity Data Records
ADO.NET 2.0 Data Providers
SqlClient Data Provider
OracleClient Data Provider
Other Data Provider
Connection
Command
DataReader
Connection
Command
DataReader
Connection
Command
DataReader
Data Records
Microsoft SQL Server
Oracle
Connection – SQLConnection, OracleConnection, OleDbConnection, OdbcConnection, etc.
Command – SQLCommand, OracleCommand, OleDbCommand, OdbcCommand, etc.
DataReader – SQLDataReader, OracleDataReader, OleDbDataReader, OdbcDataReader, etc.
DataAdapter – SQLDataAdapter, OracleDataAdapter, OleDbDataAdapter, OdbcDataAdapter, etc

ADO.NET data provider

```
OracleConnection connection = new OracleConnection("data source=.;
database=TestDB; integrated security=SSPI");

OracleCommand command = new OracleCommand("Select * from Customers",
connection);

connection.Open();

OracleDataReader myReader = command.ExecuteReader();

while (myReader.Read())
{
Console.WriteLine("\t{0}\t{1}", myReader.GetInt32(0), myReader.GetString(1));
}
connection.Close();
```

Entity Client Provider

```
string connectionString = "specify your connection string here…";

EntityConnection entityConnection = new EntityConnection(connectionString);

entityConnection.Open();


String queryString = "Select value a from IDGEntities.Author as a";

EntityCommand entityCommand = new EntityCommand(queryString, entityConnection);



EntityDataReader entityDataReader =
entityCommand.ExecuteReader(CommandBehavior.SequentialAccess);

 while (entityDataReader.Read())
        {
            Console.WriteLine(entityDataReader.GetValue(1));
        }
```

Entity Framework Object Services

The **ObjectContext** appears to have a very similar role to the **DataContext** in the LINQ to SQL world

Constructing an ObjectContext

We can construct an **ObjectContext** either by giving it a connection string, an existing **EntityConnection**

```
1 using (ObjectContext ctx = new ObjectContext("Name=NorthwindEntities"))
```

Starting to Query with an ObjectContext

Once we've got an **ObjectContext**, we can ask it to go ahead and create an **ObjectQuery<T>** for us by feeding it a piece of eSQL. For instance;

```
2 ObjectQuery<DbDataRecord> query = ctx.CreateQuery<DbDataRecord>(
  "select c.CompanyName, c.ContactName from NorthwindContext.Customers as c");
```

```
using System;
using System.Data.Common;
using System.Data.Objects;
using Northwind;

namespace ConsoleApplication4
{
  class Program
  {
    static void Main(string[] args)
    {
      using (ObjectContext ctx = new ObjectContext("Name=NorthwindEntities"))
      {
        ObjectQuery<Customers> query = ctx.CreateQuery<Customers>(
          "select value c from NorthwindContext.Customers as c");

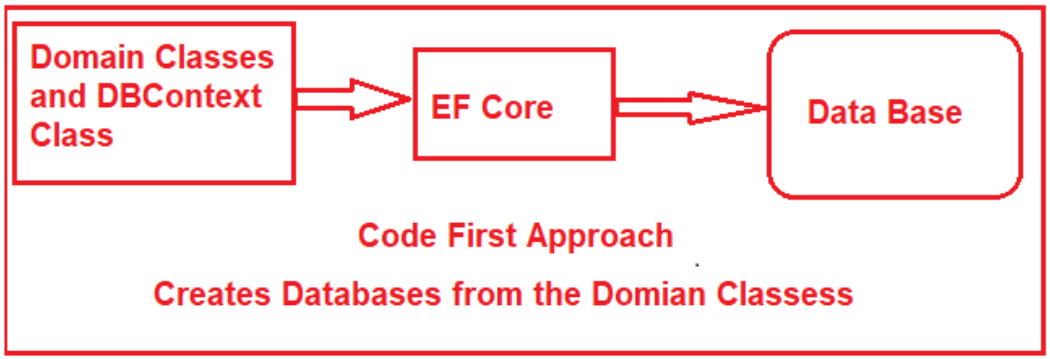
        ObjectResult<Customers> results = query.Execute(MergeOption.NoTracking);

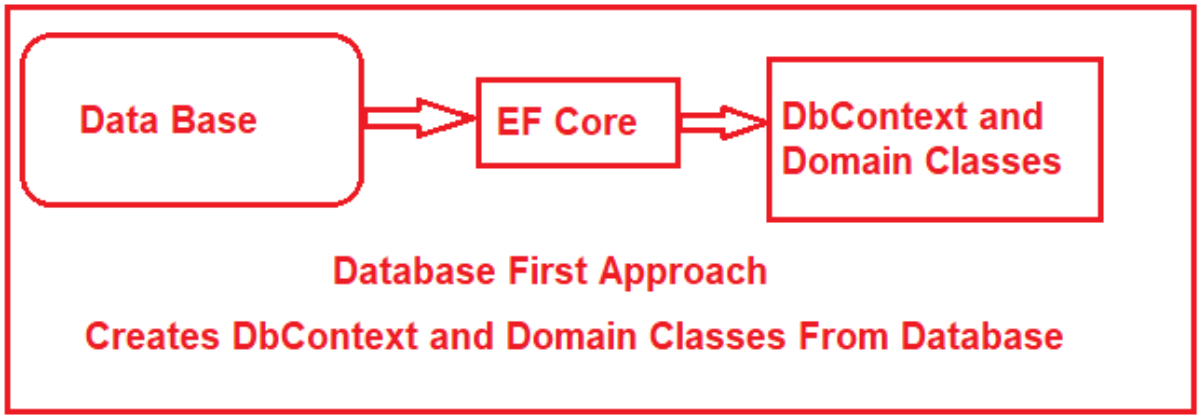
        foreach (Customers c in results)
        {
          Console.WriteLine("Company [{0}], Contact [{1}]",
            c.CompanyName, c.ContactName);
        }
      }
    }
```

EF Core Development Approaches

- Entity Framework Core supports two development approaches. They are as follows:

1. **Code-First Approach**
2. **Database-First Approach**





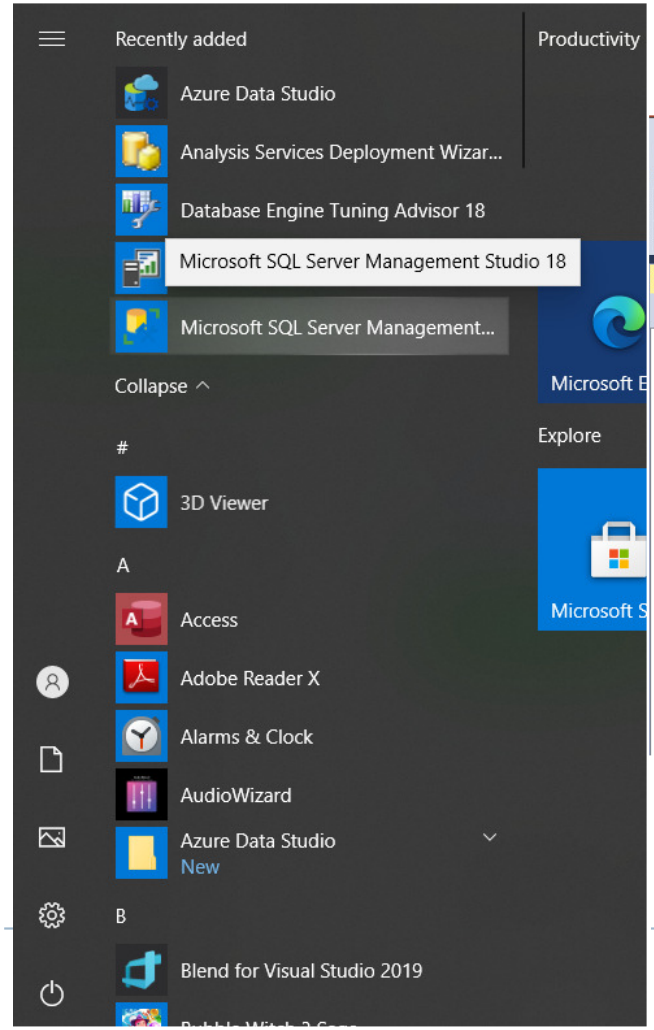
MCSE 541: Web Computing and Mining

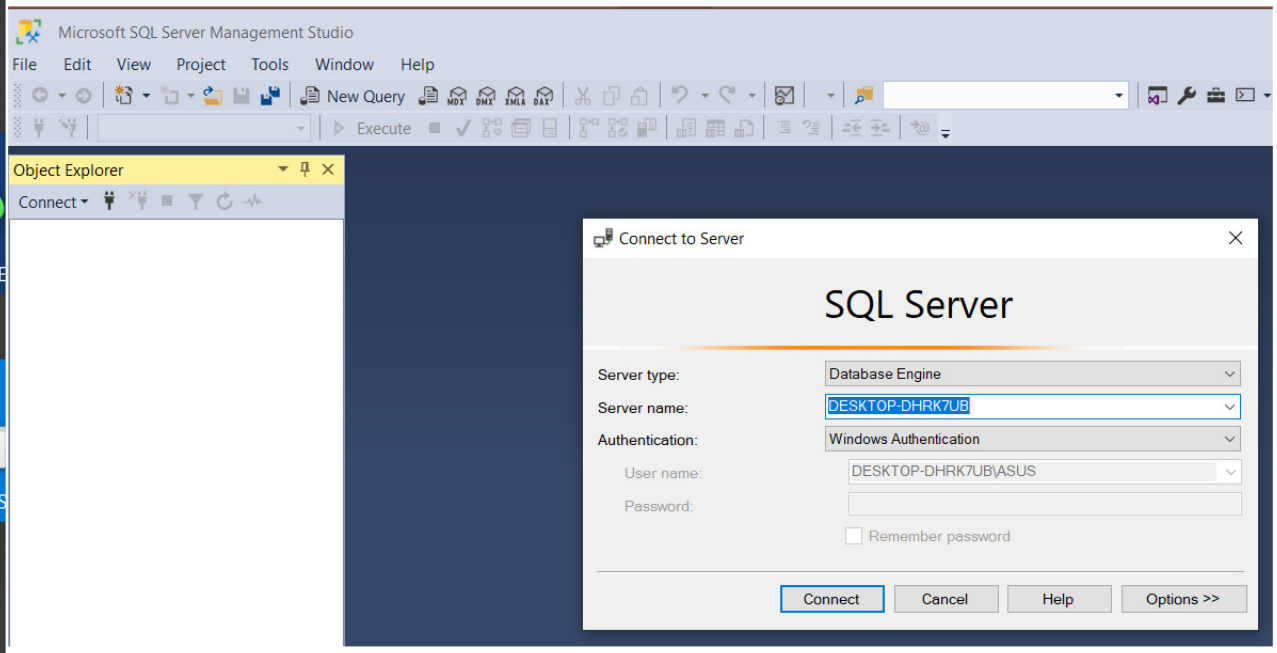
Entity Framework in ASP.NET
DB First EF Convention

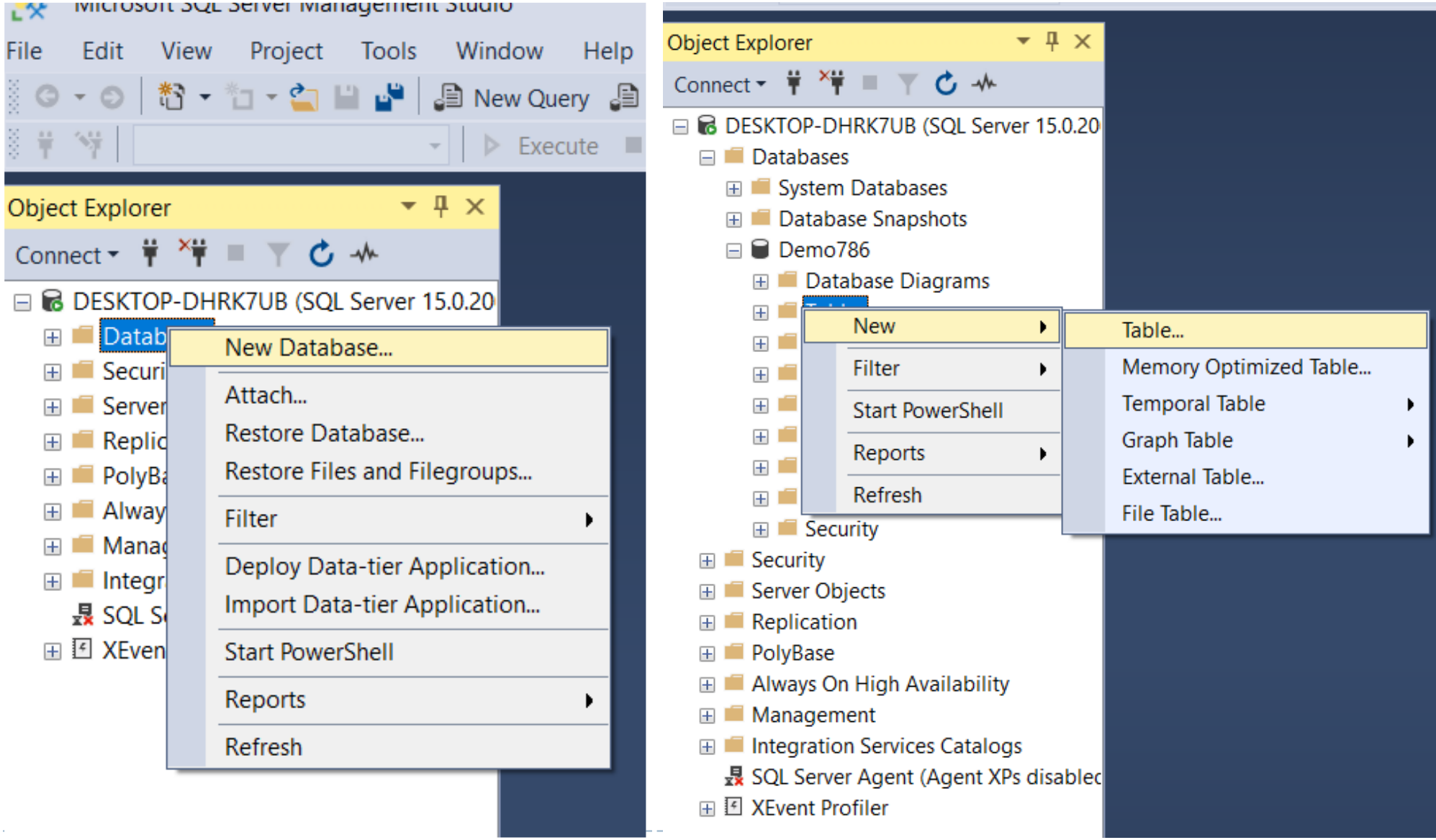
Prof. Dr. Shamim Akhter
shamimakhter@iubat.edu

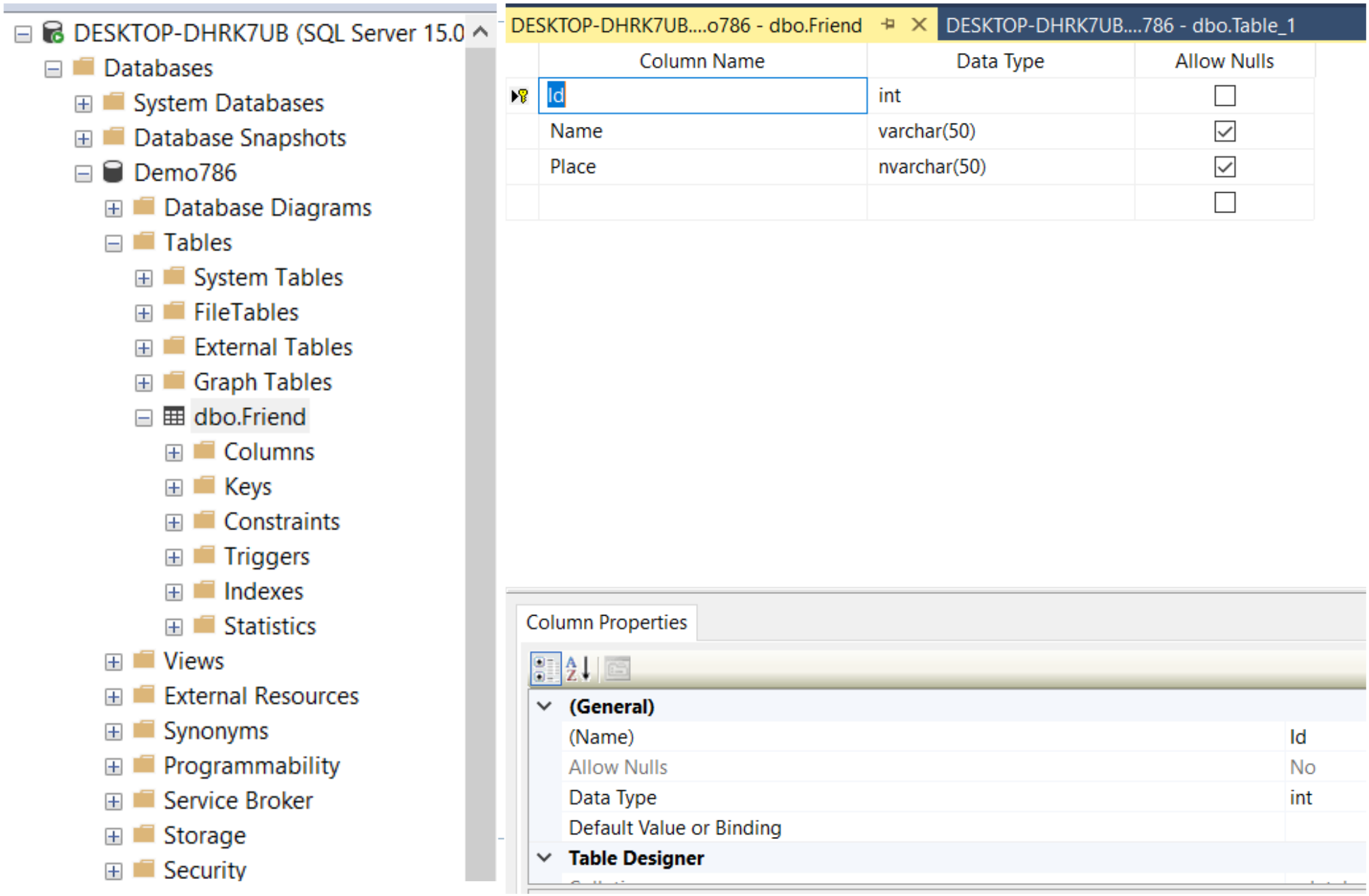
Install SQL Server

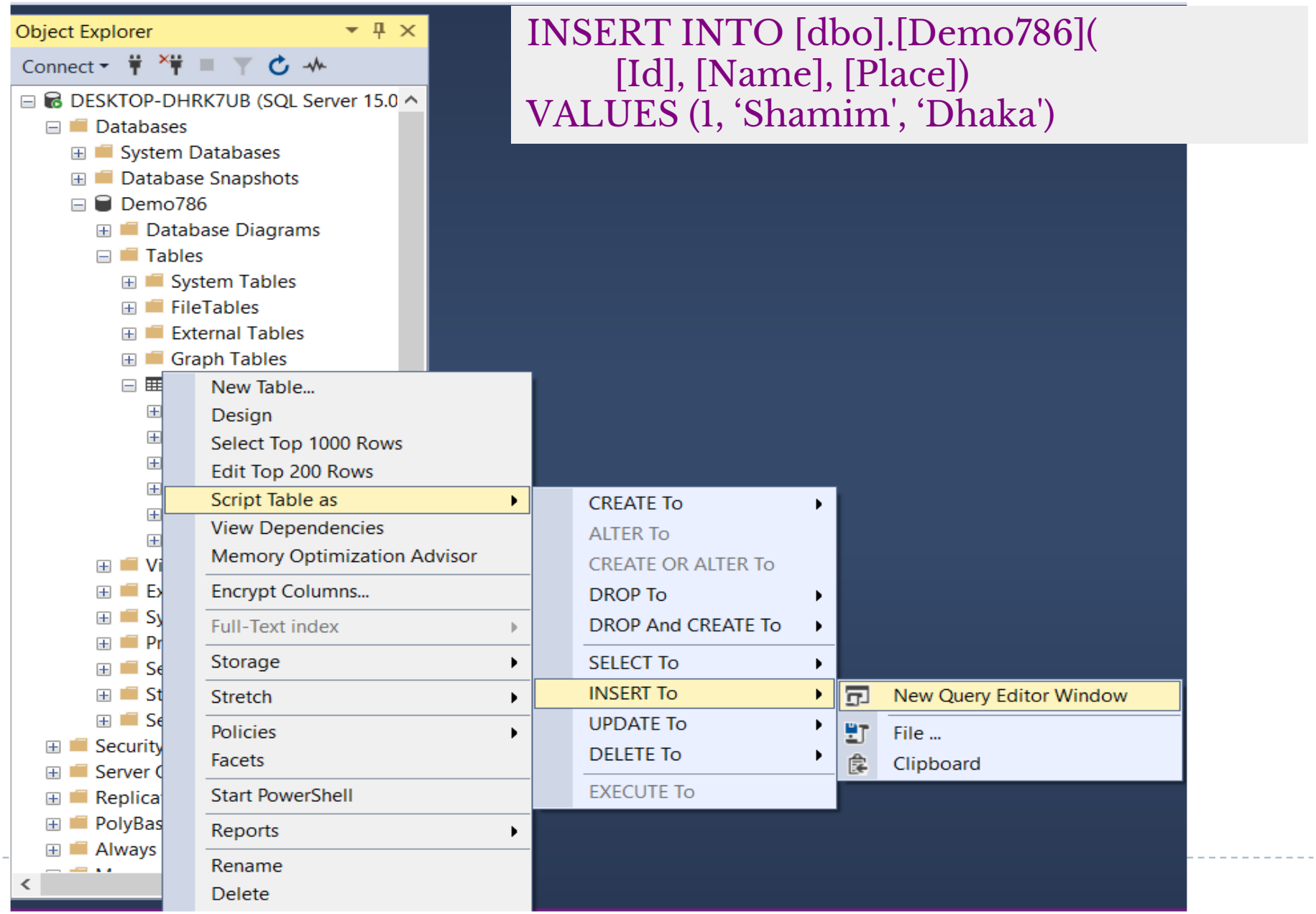
- How to download and install Microsoft SQL Server 2019
  - https://www.youtube.com/watch?v=QsXWszvjMBM

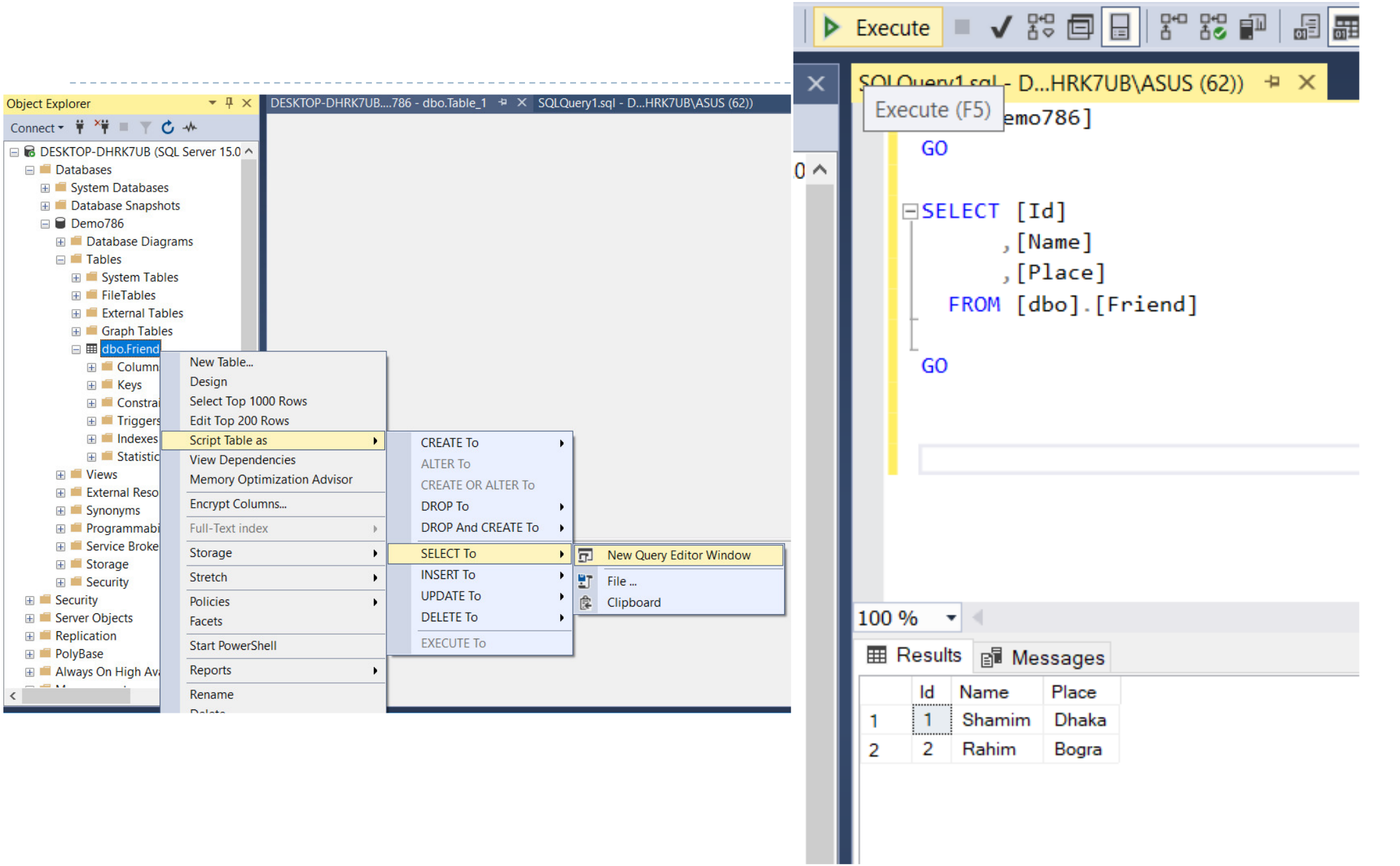




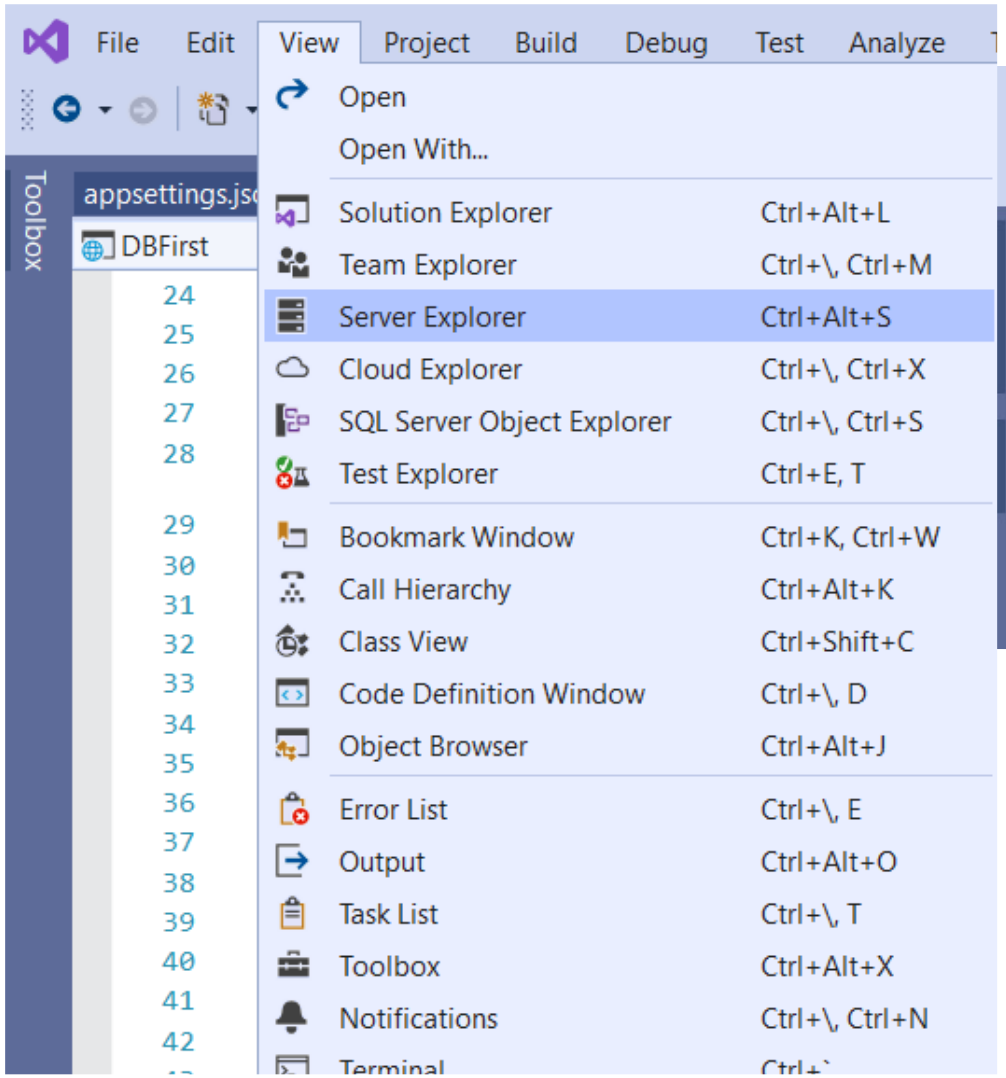

File Edit View Project Tools Window Help
New Query
Execute
Object Explorer
Connect
DESKTOP-DHRK7UB (SQL Server 15.0.20
New Database...
Attach...
Restore Database...
Restore Files and Filegroups...
Filter
Deploy Data-tier Application...
Import Data-tier Application...
Start PowerShell
Reports
Refresh
Databases
System Databases
Database Snapshots
Demo786
Database Diagrams
New
Filter
Start PowerShell
Reports
Refresh
Table...
Memory Optimized Table...
Temporal Table
Graph Table
External Table...
File Table...
Security
Server Objects
Replication
PolyBase
Always On High Availability
Management
Integration Services Catalogs
SQL Server Agent (Agent XPs disabled
XEvent Profiler

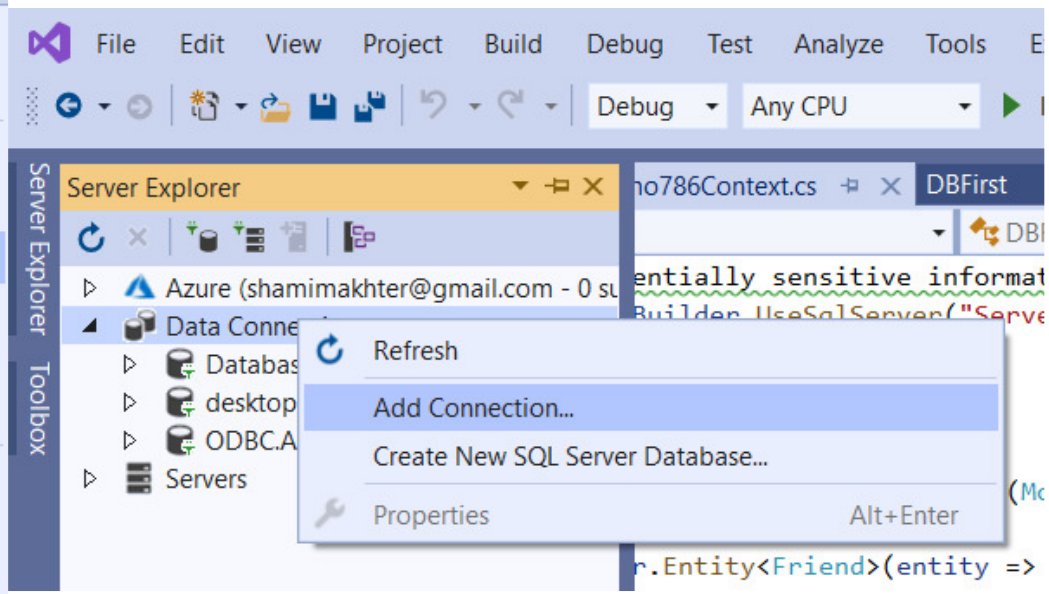

DESKTOP-DHRK7UB (SQL Server 15.0
Databases
System Databases
Database Snapshots
Demo786
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.Friend
Columns
Keys
Constraints
Triggers
Indexes
Statistics
Views
External Resources
Synonyms
Programmability
Service Broker
Storage
Security
DESKTOP-DHRK7UB....o786 - dbo.Friend
DESKTOP-DHRK7UB....786 - dbo.Table_1
Column Name
Data Type
Allow Nulls
Id
int
Name
varchar(50)
Place
nvarchar(50)
Column Properties
(General)
(Name)
Id
Allow Nulls
No
Data Type
int
Default Value or Binding
Table Designer

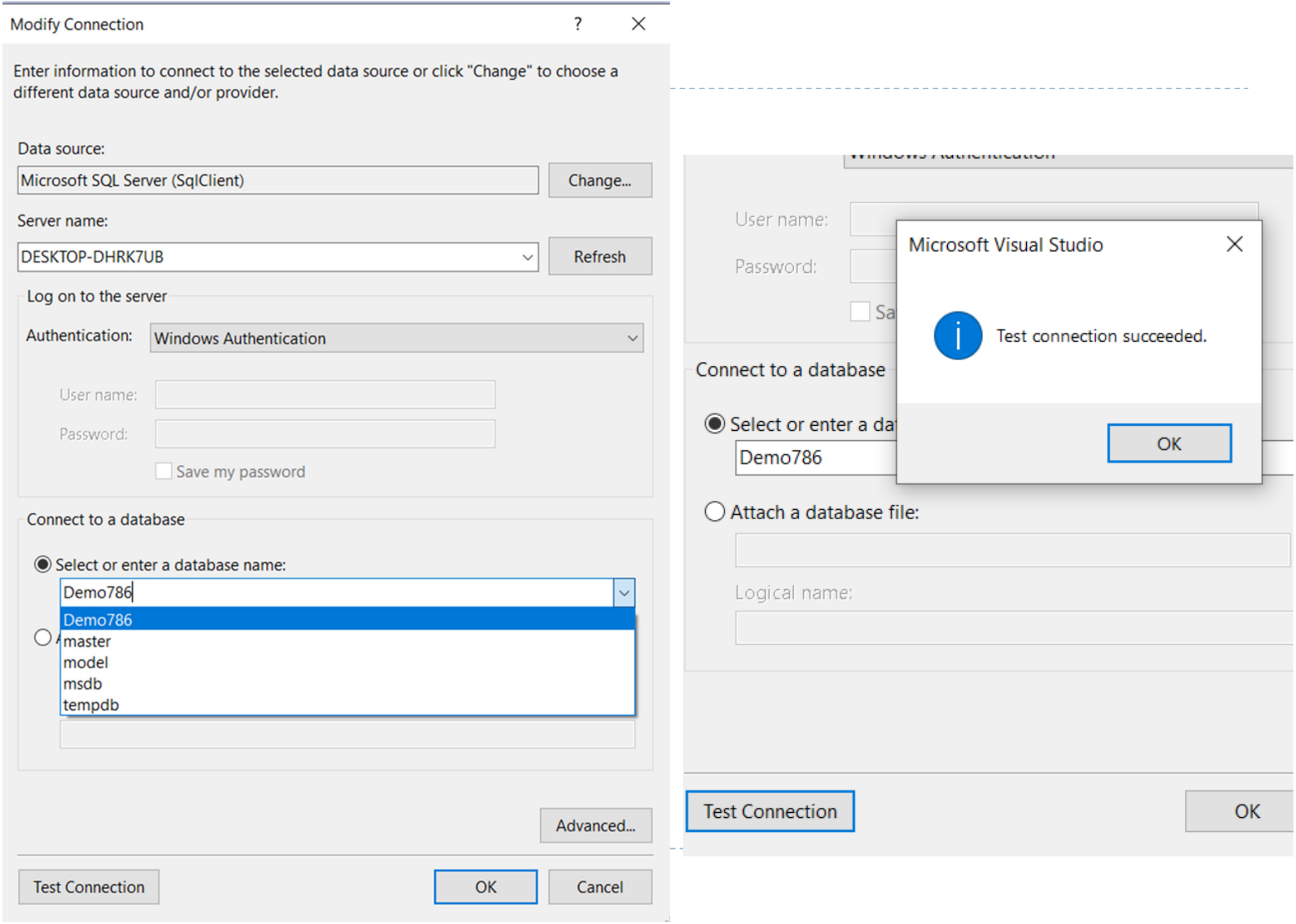

Object Explorer
Connect
DESKTOP-DHRK7UB (SQL Server 15.0
Databases
System Databases
Database Snapshots
Demo786
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
New Table...
Design
Select Top 1000 Rows
Edit Top 200 Rows
Script Table as
View Dependencies
Memory Optimization Advisor
Encrypt Columns...
Full-Text index
Storage
Stretch
Policies
Facets
Start PowerShell
Reports
Rename
Delete
CREATE To
ALTER To
CREATE OR ALTER To
DROP To
DROP And CREATE To
SELECT To
INSERT To
UPDATE To
DELETE To
EXECUTE To
New Query Editor Window
File ...
Clipboard
INSERT INTO [dbo].[Demo786](
[Id], [Name], [Place])
VALUES (1, 'Shamim', 'Dhaka')


Execute
Execute (F5)
SQLQuery1.sql - D...HRK7UB\ASUS (62))
emo786]
GO
SELECT [Id]
,[Name]
,[Place]
FROM [dbo].[Friend]
GO
Object Explorer
Connect
DESKTOP-DHRK7UB (SQL Server 15.0
Databases
System Databases
Database Snapshots
Demo786
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.Friend
Column
Keys
Constrai
Triggers
Indexes
Statistic
Views
External Reso
Synonyms
Programmabi
Service Broke
Storage
Security
Security
Server Objects
Replication
PolyBase
Always On High Av
DESKTOP-DHRK7UB....786 - dbo.Table_1
SQLQuery1.sql - D...HRK7UB\ASUS (62))
New Table...
Design
Select Top 1000 Rows
Edit Top 200 Rows
Script Table as
View Dependencies
Memory Optimization Advisor
Encrypt Columns...
Full-Text index
Storage
Stretch
Policies
Facets
Start PowerShell
Reports
Rename
CREATE To
ALTER To
CREATE OR ALTER To
DROP To
DROP And CREATE To
SELECT To
INSERT To
UPDATE To
DELETE To
EXECUTE To
New Query Editor Window
File ...
Clipboard
100 %
Results
Messages
Id
Name
Place
1
1
Shamim
Dhaka
2
2
Rahim
Bogra

Visual Studio






Modify Connection
Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.
Data source:
Microsoft SQL Server (SqlClient)
Change...
Server name:
DESKTOP-DHRK7UB
Refresh
Log on to the server
Authentication:
Windows Authentication
User name:
Password:
Save my password
Connect to a database
Select or enter a database name:
Demo786
Demo786
master
model
msdb
tempdb
Advanced...
Test Connection
OK
Cancel
Microsoft Visual Studio
Test connection succeeded.
OK
Connect to a database
Select or enter a da
Demo786
Attach a database file:
Logical name:
Test Connection
OK

Scaffold-DbContext Command

- EF Core does not support visual designer for DB model and wizard to create the entity and context classes similar to EF 6.
- So, we need to do reverse engineering using the Scaffold-DbContext command.
  - This reverse engineering command creates model and context classes (by deriving DbContext) based on the schema of the existing database.

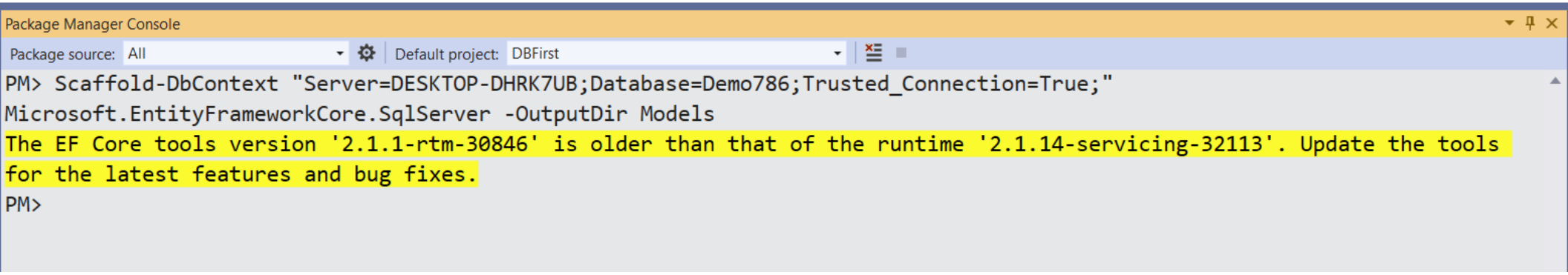
Use Scaffold-DbContext to create a model based on your existing database. The following parameters can be specified with Scaffold-DbContext in Package Manager Console:

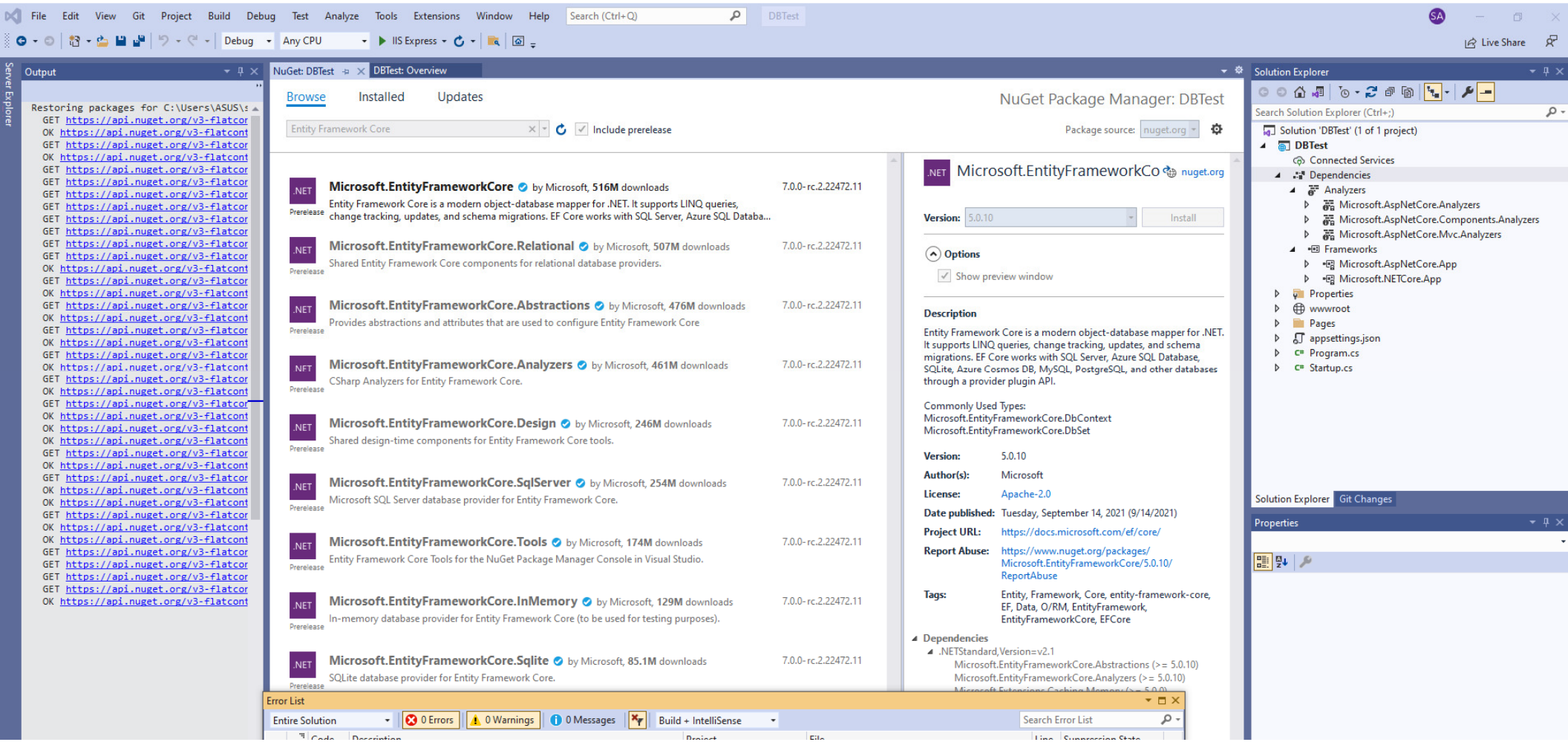
Scaffold-DbContext [-Connection] [-Provider] [-OutputDir] [-Context] [-Schemas>] [-Tables>] [-DataAnnotations] [-Force] [-Project] [-StartupProject] [<CommonParameters>]

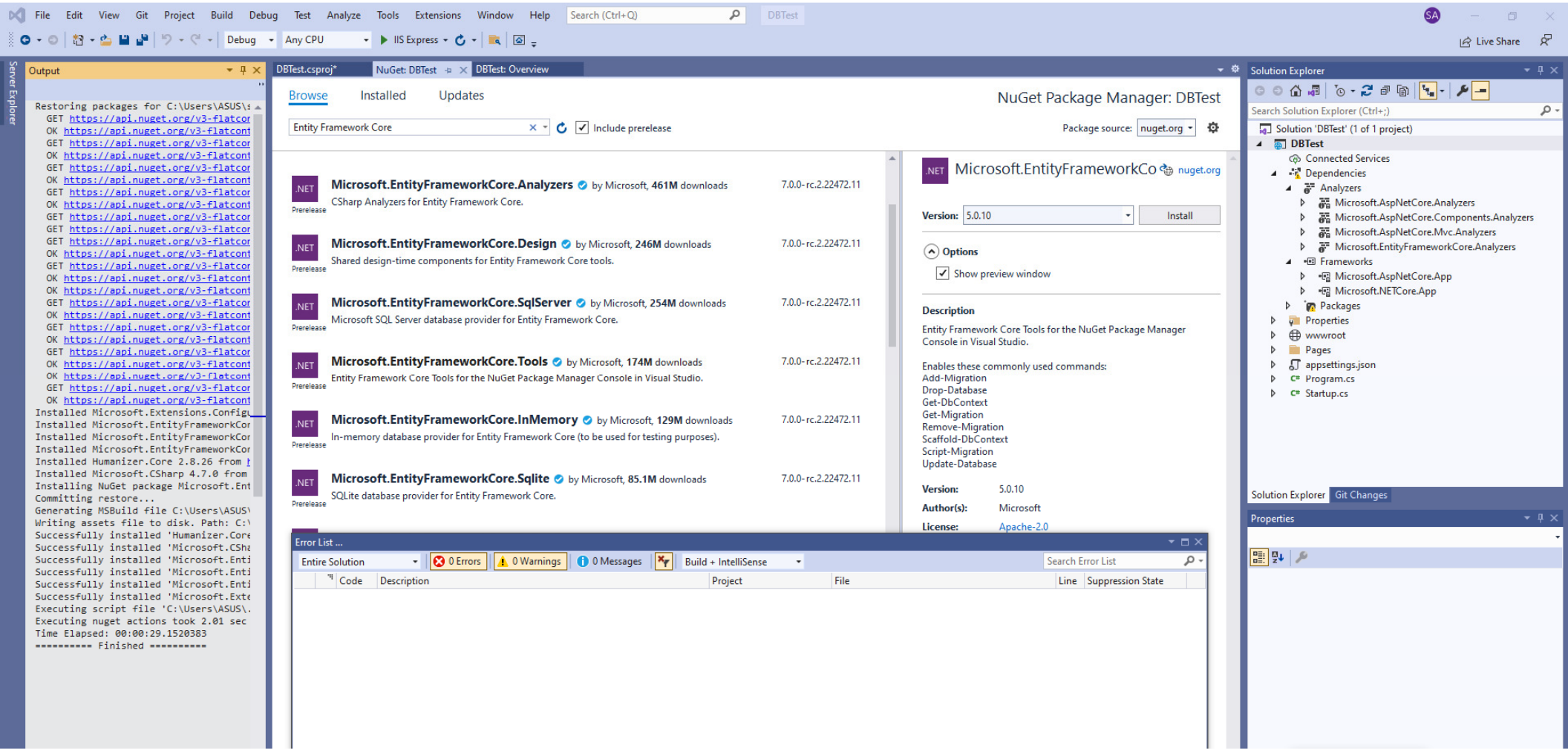
```
Scaffold-DbContext
"Server=.\SQLExpress;Database=SchoolDB;Trusted_Connection=True;"
Microsoft.EntityFrameworkCore.SqlServer -OutputDir Model
```

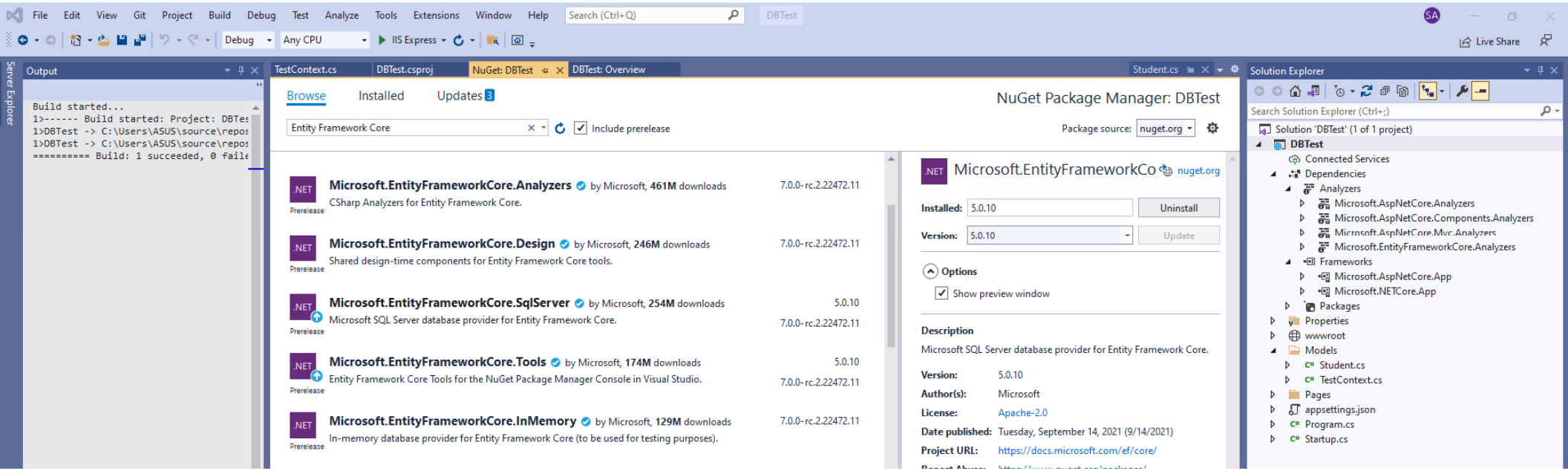
The first parameter is a connection string which includes three parts: DB Server, database name and security info. Here, Server=.\SQLExpress; refers to local SQLEXPRESS database server. Database=SchoolDB; specifies the database name "SchoolDB" for which we are going to create classes. Trusted_Connection=True; specifies the Windows authentication. It will use Windows credentials to connect to the SQL Server. The second parameter is the provider name. We use provider for the SQL Server, so it is Microsoft.EntityFrameworkCore.SqlServer. The -OutputDir parameter specifies the directory where we want to generate all the classes which is the Models folder in this case.

- ```
  Scaffold-DbContext "Server=DESKTOP-
  DHRK7UB;Database=StudentDB;Trusted_Connection=True;"
  Microsoft.EntityFrameworkCore.SqlServer -OutputDir Models
  ```

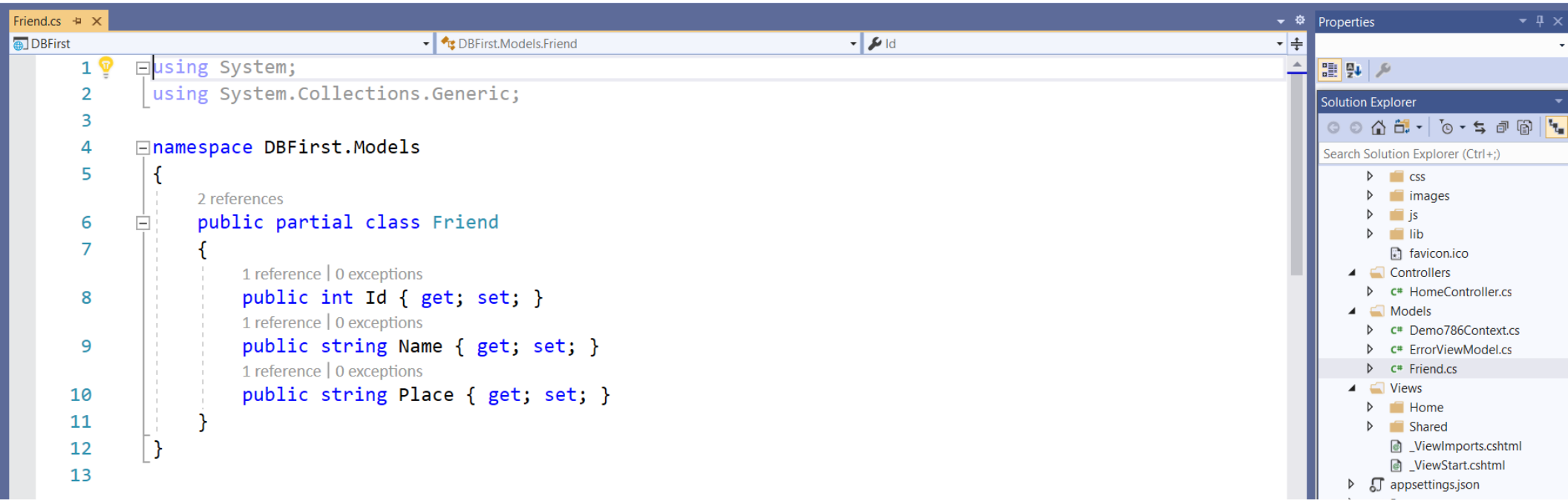

  ```


File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
DBTest
Debug
Any CPU
IIS Express
Live Share
Output
Restoring packages for C:\Users\ASUS\s
GET https://api.nuget.org/v3-flatcor
OK https://api.nuget.org/v3-flatcont
NuGet: DBTest
DBTest: Overview
Browse
Installed
Updates
NuGet Package Manager: DBTest
Entity Framework Core
Include prerelease
Package source: nuget.org
Microsoft.EntityFrameworkCore by Microsoft, 516M downloads 7.0.0-rc.2.22472.11
Entity Framework Core is a modern object-database mapper for .NET. It supports LINQ queries, change tracking, updates, and schema migrations. EF Core works with SQL Server, Azure SQL Databa...
Microsoft.EntityFrameworkCore.Relational by Microsoft, 507M downloads 7.0.0-rc.2.22472.11
Shared Entity Framework Core components for relational database providers.
Microsoft.EntityFrameworkCore.Abstractions by Microsoft, 476M downloads 7.0.0-rc.2.22472.11
Provides abstractions and attributes that are used to configure Entity Framework Core
Microsoft.EntityFrameworkCore.Analyzers by Microsoft, 461M downloads 7.0.0-rc.2.22472.11
CSharp Analyzers for Entity Framework Core.
Microsoft.EntityFrameworkCore.Design by Microsoft, 246M downloads 7.0.0-rc.2.22472.11
Shared design-time components for Entity Framework Core tools.
Microsoft.EntityFrameworkCore.SqlServer by Microsoft, 254M downloads 7.0.0-rc.2.22472.11
Microsoft SQL Server database provider for Entity Framework Core.
Microsoft.EntityFrameworkCore.Tools by Microsoft, 174M downloads 7.0.0-rc.2.22472.11
Entity Framework Core Tools for the NuGet Package Manager Console in Visual Studio.
Microsoft.EntityFrameworkCore.InMemory by Microsoft, 129M downloads 7.0.0-rc.2.22472.11
In-memory database provider for Entity Framework Core (to be used for testing purposes).
Microsoft.EntityFrameworkCore.Sqlite by Microsoft, 85.1M downloads 7.0.0-rc.2.22472.11
SQLite database provider for Entity Framework Core.
Prerelease
Microsoft.EntityFrameworkCo
nuget.org
Version: 5.0.10
Install
Options
Show preview window
Description
Entity Framework Core is a modern object-database mapper for .NET. It supports LINQ queries, change tracking, updates, and schema migrations. EF Core works with SQL Server, Azure SQL Database, SQLite, Azure Cosmos DB, MySQL, PostgreSQL, and other databases through a provider plugin API.
Commonly Used Types:
Microsoft.EntityFrameworkCore.DbContext
Microsoft.EntityFrameworkCore.DbSet
Version: 5.0.10
Author(s): Microsoft
License: Apache-2.0
Date published: Tuesday, September 14, 2021 (9/14/2021)
Project URL: https://docs.microsoft.com/ef/core/
Report Abuse: https://www.nuget.org/packages/Microsoft.EntityFrameworkCore/5.0.10/ReportAbuse
Tags: Entity, Framework, Core, entity-framework-core, EF, Data, O/RM, EntityFramework, EntityFrameworkCore, EFCore
Dependencies
.NETStandard,Version=v2.1
Microsoft.EntityFrameworkCore.Abstractions (>= 5.0.10)
Microsoft.EntityFrameworkCore.Analyzers (>= 5.0.10)
Error List
Entire Solution
0 Errors
0 Warnings
0 Messages
Build + IntelliSense
Search Error List
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'DBTest' (1 of 1 project)
DBTest
Connected Services
Dependencies
Analyzers
Microsoft.AspNetCore.Analyzers
Microsoft.AspNetCore.Components.Analyzers
Microsoft.AspNetCore.Mvc.Analyzers
Frameworks
Microsoft.AspNetCore.App
Microsoft.NETCore.App
Properties
wwwroot
Pages
appsettings.json
Program.cs
Startup.cs
Solution Explorer
Git Changes
Properties


File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
DBTest
Debug
Any CPU
IIS Express
Live Share
Output
Restoring packages for C:\Users\ASUS\s
GET https://api.nuget.org/v3-flatcor
OK https://api.nuget.org/v3-flatcont
Installed Microsoft.Extensions.Configu
Installed Microsoft.EntityFrameworkCor
Installed Microsoft.EntityFrameworkCor
Installed Microsoft.EntityFrameworkCor
Installed Humanizer.Core 2.8.26 from
Installed Microsoft.CSharp 4.7.0 from
Installing NuGet package Microsoft.Ent
Committing restore...
Generating MSBuild file C:\Users\ASUS\
Writing assets file to disk. Path: C:\
Successfully installed 'Humanizer.Core
Successfully installed 'Microsoft.CSha
Successfully installed 'Microsoft.Enti
Successfully installed 'Microsoft.Enti
Successfully installed 'Microsoft.Enti
Successfully installed 'Microsoft.Exte
Executing script file 'C:\Users\ASUS\.
Executing nuget actions took 2.01 sec
Time Elapsed: 00:00:29.1520383
========== Finished ==========
DBTest.csproj*
NuGet: DBTest
DBTest: Overview
Browse
Installed
Updates
NuGet Package Manager: DBTest
Entity Framework Core
Include prerelease
Package source: nuget.org
Microsoft.EntityFrameworkCore.Analyzers by Microsoft, 461M downloads
CSharp Analyzers for Entity Framework Core.
7.0.0-rc.2.22472.11
Microsoft.EntityFrameworkCore.Design by Microsoft, 246M downloads
Shared design-time components for Entity Framework Core tools.
Microsoft.EntityFrameworkCore.SqlServer by Microsoft, 254M downloads
Microsoft SQL Server database provider for Entity Framework Core.
Microsoft.EntityFrameworkCore.Tools by Microsoft, 174M downloads
Entity Framework Core Tools for the NuGet Package Manager Console in Visual Studio.
Microsoft.EntityFrameworkCore.InMemory by Microsoft, 129M downloads
In-memory database provider for Entity Framework Core (to be used for testing purposes).
Microsoft.EntityFrameworkCore.Sqlite by Microsoft, 85.1M downloads
SQLite database provider for Entity Framework Core.
Prerelease
Microsoft.EntityFrameworkCo
nuget.org
Version: 5.0.10
Install
Options
Show preview window
Description
Entity Framework Core Tools for the NuGet Package Manager Console in Visual Studio.
Enables these commonly used commands:
Add-Migration
Drop-Database
Get-DbContext
Get-Migration
Remove-Migration
Scaffold-DbContext
Script-Migration
Update-Database
Version: 5.0.10
Author(s): Microsoft
License: Apache-2.0
Error List ...
Entire Solution
0 Errors
0 Warnings
0 Messages
Build + IntelliSense
Search Error List
Code
Description
Project
File
Line
Suppression State
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'DBTest' (1 of 1 project)
DBTest
Connected Services
Dependencies
Analyzers
Microsoft.AspNetCore.Analyzers
Microsoft.AspNetCore.Components.Analyzers
Microsoft.AspNetCore.Mvc.Analyzers
Microsoft.EntityFrameworkCore.Analyzers
Frameworks
Microsoft.AspNetCore.App
Microsoft.NETCore.App
Packages
Properties
wwwroot
Pages
appsettings.json
Program.cs
Startup.cs
Solution Explorer
Git Changes
Properties


File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
DBTest
Debug
Any CPU
IIS Express
Live Share
Output
Build started...
1>------ Build started: Project: DBTes
1>DBTest -> C:\Users\ASUS\source\repos
1>DBTest -> C:\Users\ASUS\source\repos
========== Build: 1 succeeded, 0 faile
TestContext.cs
DBTest.csproj
NuGet: DBTest
DBTest: Overview
Student.cs
Browse
Installed
Updates 3
NuGet Package Manager: DBTest
Entity Framework Core
Include prerelease
Package source: nuget.org
Microsoft.EntityFrameworkCore.Analyzers by Microsoft, 461M downloads
CSharp Analyzers for Entity Framework Core.
7.0.0-rc.2.22472.11
Microsoft.EntityFrameworkCore.Design by Microsoft, 246M downloads
Shared design-time components for Entity Framework Core tools.
7.0.0-rc.2.22472.11
Microsoft.EntityFrameworkCore.SqlServer by Microsoft, 254M downloads
Microsoft SQL Server database provider for Entity Framework Core.
5.0.10
7.0.0-rc.2.22472.11
Microsoft.EntityFrameworkCore.Tools by Microsoft, 174M downloads
Entity Framework Core Tools for the NuGet Package Manager Console in Visual Studio.
5.0.10
7.0.0-rc.2.22472.11
Microsoft.EntityFrameworkCore.InMemory by Microsoft, 129M downloads
In-memory database provider for Entity Framework Core (to be used for testing purposes).
7.0.0-rc.2.22472.11
Prerelease
Microsoft.EntityFrameworkCo
nuget.org
Installed: 5.0.10
Uninstall
Version: 5.0.10
Update
Options
Show preview window
Description
Microsoft SQL Server database provider for Entity Framework Core.
Version: 5.0.10
Author(s): Microsoft
License: Apache-2.0
Date published: Tuesday, September 14, 2021 (9/14/2021)
Project URL: https://docs.microsoft.com/ef/core/
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'DBTest' (1 of 1 project)
DBTest
Connected Services
Dependencies
Analyzers
Microsoft.AspNetCore.Analyzers
Microsoft.AspNetCore.Components.Analyzers
Microsoft.AspNetCore.Mvc.Analyzers
Microsoft.EntityFrameworkCore.Analyzers
Frameworks
Microsoft.AspNetCore.App
Microsoft.NETCore.App
Packages
Properties
wwwroot
Models
Student.cs
TestContext.cs
Pages
appsettings.json
Program.cs
Startup.cs

Friend.cs


Friend.cs
DBFirst
DBFirst.Models.Friend
Id
using System;
using System.Collections.Generic;
namespace DBFirst.Models
{
2 references
public partial class Friend
{
1 reference | 0 exceptions
public int Id { get; set; }
1 reference | 0 exceptions
public string Name { get; set; }
1 reference | 0 exceptions
public string Place { get; set; }
}
}
Properties
Solution Explorer
Search Solution Explorer (Ctrl+;)
css
images
js
lib
favicon.ico
Controllers
HomeController.cs
Models
Demo786Context.cs
ErrorViewModel.cs
Friend.cs
Views
Home
Shared
_ViewImports.cshtml
_ViewStart.cshtml
appsettings.json

Demo786Context.cs

Demo786Context.cs* | Friend.cs

DBFirst | DBFirst.Models.Demo786Context | OnModelCreating(ModelBu

```
7   public partial class Demo786Context : DbContext
8   {
        0 references | 0 exceptions
9       public Demo786Context()...
        0 references | 0 exceptions
12      public Demo786Context(DbContextOptions<Demo786Context> options)...
        0 references | 0 exceptions
16      public virtual DbSet<Friend> Friend { get; set; }
        0 references | 0 exceptions
17      protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
18      {
19          if (!optionsBuilder.IsConfigured)
20          {
21  #warning To protect potentially sensitive information in your connection string, you should move it out of source code. See
22              optionsBuilder.UseSqlServer("Server=DESKTOP-DHRK7UB;Database=Demo786;Trusted_Connection=True;");
23          }
24      }
        0 references | 0 exceptions
25      protected override void OnModelCreating(ModelBuilder modelBuilder)
26      {
27          modelBuilder.Entity<Friend>(entity =>
28          {
29              entity.Property(e => e.Id).ValueGeneratedNever();
30
31              entity.Property(e => e.Name)
32                  .HasMaxLength(50)
33                  .IsUnicode(false);
34
35              entity.Property(e => e.Place).HasMaxLength(50);
36          });
37      }
38  }
```

Context class connects Models with DB

```
namespace MVC_SQL_Server.Models
{
    public class AppDBContext: DbContext //connecting to DB
    {
        public AppDBContext(DbContextOptions<AppDBContext> options): base(options) {
            //injecting service collection as constructor parameter
        }

        public DbSet<FriendModel> FriendModels { get; set; } //mirrors as table in DB

        protected override void OnModelCreating(ModelBuilder modelBuilder) {
           //overridden to build entity model and initialized data

            modelBuilder.Entity<FriendModel>().HasData(
                new FriendModel { Id =1, Name="Faysal", Place="Sydney"},
                new FriendModel { Id =2, Name = "Sumon", Place = "Rajshahi" }
                );
        }
    }
}
```

LinQ

- An innovative concept in C#
- Encompasses a set of features (query) that lets you retrieve information from a data source.

- LINQ gives C# the ability to generate queries
  - For any compatible data source
  - Unique query syntax for any data source (RDBMS, Array, XML file. ADO.NET datasets)

- Two fundamentals steps are involved with LINQ
  - Query formulation, defines WHAT to retrieve from a data source
  - Executing the query, and obtains the results

LinQ Simple Example with Array

```
using System;
using System.Linq;  // LINQ Library namespace

class SimpQuery{
    static void Main(){
        int [] num={1, 2, -3 , -5, 3, 2};   //C# arrays are convertible to IEnumerable<T>

        var PosNum= from n in num   //PosNum is a query variable
                            where n> 0  // from range-variable in data source
                                select n;  // where boolean-expression

         foreach (int i in PosNum) Console.Write(i+ " "); // execute query

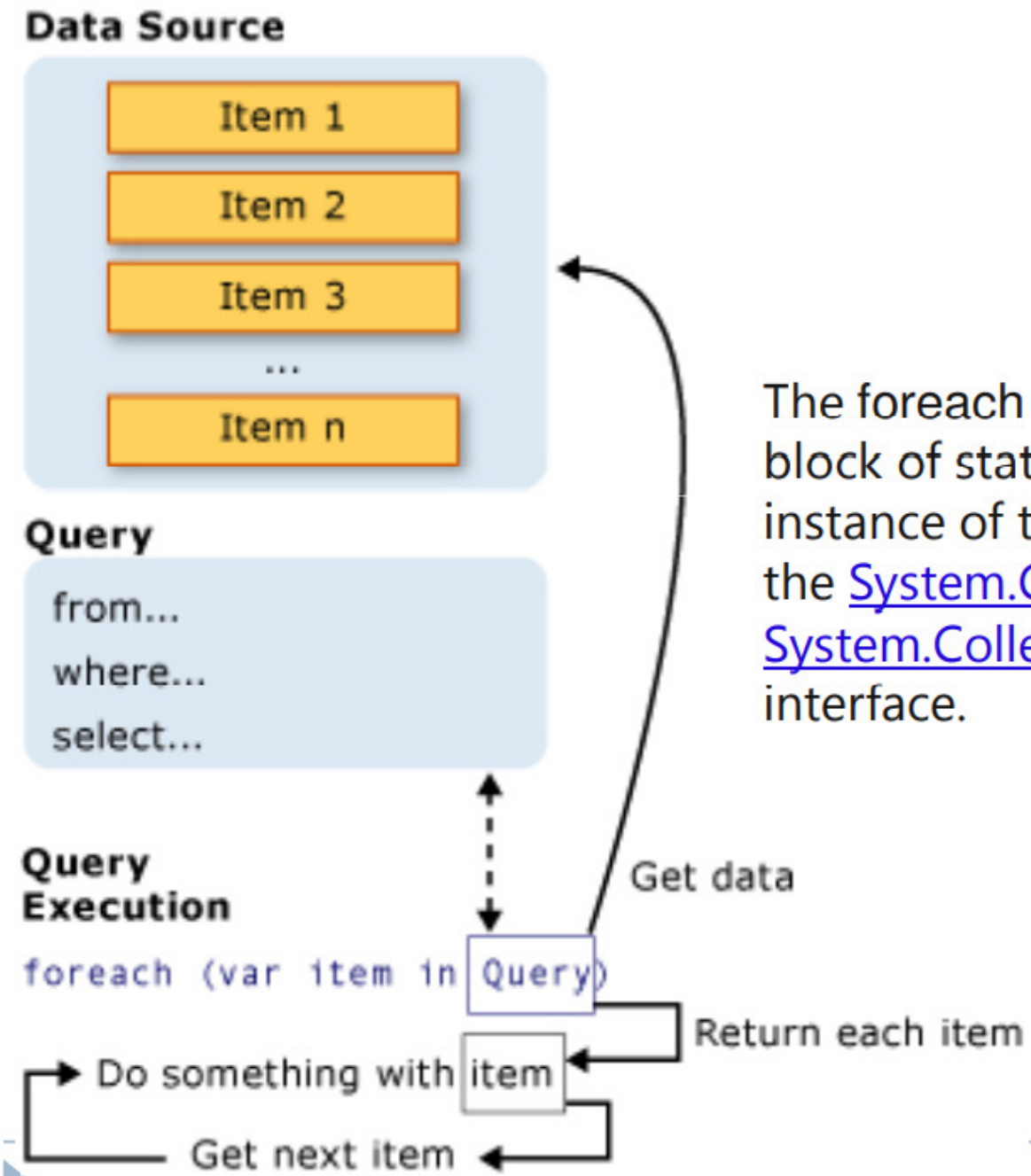
        Console.WriteLine();
    }
}                                                          int evenNumCount = PosNum.Count();
```

WHY var type PosNum?

NB: I) A Query can be executed more than once; num[2]=99;

NB: II) where n>0 && n<10 III) orderby n Iv) orderby n descending

https://www.tutorialspoint.com/compile_csharp_online.php



The foreach statement executes a statement or a block of statements for each element in an instance of the type that implements the System.Collections.IEnumerable or System.Collections.Generic.IEnumerable<T> interface.

LinQ in Database

```
private Demo786Context ctx = new Demo786Context();
        public IActionResult ShowAll()
        {
            IEnumerable<Friend> friends = (from s in ctx.Friend
                                           select s);

            return View(friends);
        }

        public IActionResult SearchFriend( )
        {
            var friend = (from s in ctx.Friend
                          where s.Name == "Shamim"
                          select s).FirstOrDefault<Friend>();

            return View(friend);
        }
```

Add and Remove APIs

```
public IActionResult Index()
{
    ctx.Add(new Participant { Id=4, Name="Maisa", Address="Norshingdi" });
    ctx.SaveChanges();


   /* IEnumerable<Participant> ps = (from s in ctx.Participants
                                     select s);*/


    ctx.Remove(ctx.Participants.Single(a => a.Id == 2));
    ctx.SaveChanges();

    IEnumerable<Participant> ps = (from s in ctx.Participants
                                     select s);


    return View(ps);
}
```