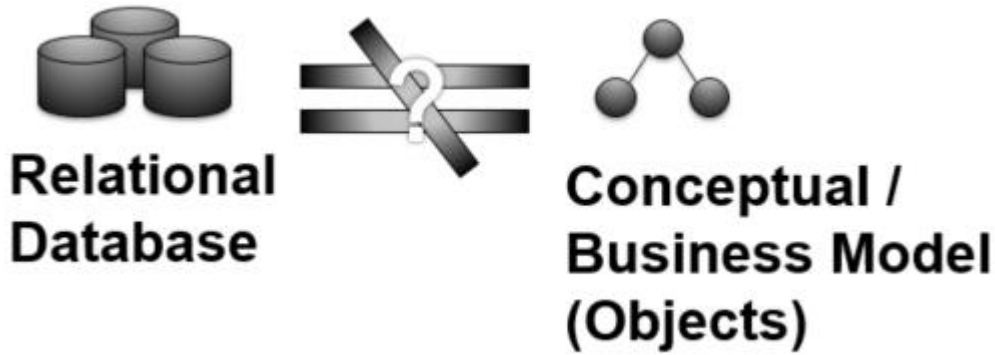


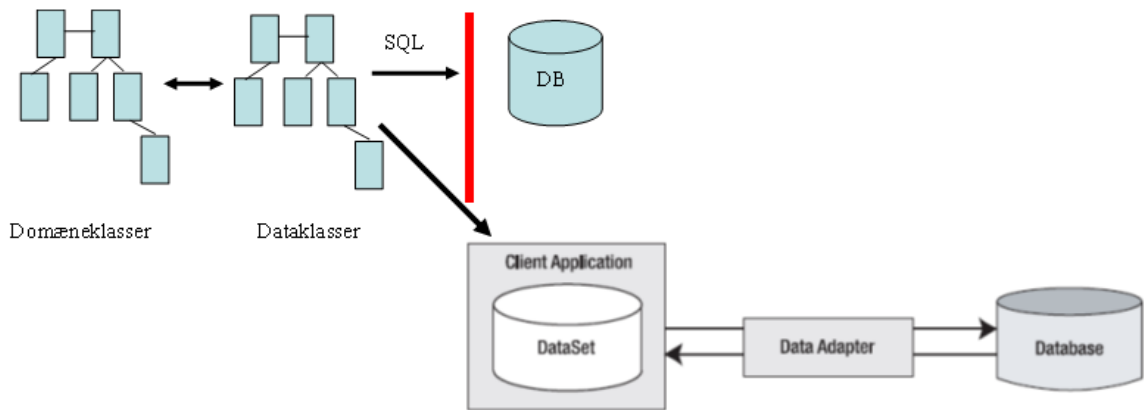
Entity Framework

Şimdiye kadar anlatılan konular üzerinden şu söylenebilir; “veri tabanı ile uygulama arasında bir bağ kurulabiliyor”. Bu ADO.NET ile mümkün kılınmıştı. Veri tabanı ile uygulama arasında veri alışverişi yapılabiliyordu. Ancak belirtmek gerekir ki bu bağ (ya da yöntem) bazı zorlukları ve sıkıntıları beraberinde getiriyor. Bugün database ile uygulama arasındaki iletişimi daha iyi yönetecek bir sistemden bahsedilecektir.



Geleneksel yöntem olarak; SQL'i OOP'a map'leyecek (haritalayacak ya da planlayacak yazmadım ☺) sabit bir katman oluşturulur. Uygulamanın bir çok yerden veri tabanına ulaşması için en iyi çözüm budur. Ama hala bir takım sıkıntılar doğmaktadır;

- Uygulama ve database arasında sıkı bir bağ vardır. Bunun dezavantajı şudur; uygulamadaki ilgili class'ları değiştirilmeden veritabanındaki tablolar değiştirilemez.
- Stabil kalması zordur. Mesela sql sorguları uygulama içinde pek çok yere dağılmış classlar vb yapılara bağlı olabilir. Bu classlarda yapılan en ufak bir değişiklik sorguların hatalı sonuçlanmasına yada sonuçlanmamasına sebep olabilir.
- Uygulama alanları ile veritabanı alanları aynı olmayabilir.



Tam bu noktalarda karşılaşılan sıkıntıları gidermek ve daha iyi çözüm yolları oluşturmak için yeni bir kavram ortaya çıkmaktadır:

ORM (Object Relational Mapping) nedir?

ORM, ilişkisel veri tabanı (RDBMS) ile nesne yönelimli programlama (OOP) arasındaki ilişkiyi kuran ve ilişkisel veri tabanındaki bilgileri yönetmek için nesne modellerini kullanan bir tekniktir. Yani nesneleri database deki tablolara bağlayarak uygulama ile database arasındaki alışverişi sağlar. ORM kavramı belli bir programlama diline bağlı değildir ve her OOP dilinde yazılabilir.

ORM sayesinde veritabanında herhangi bir tablo yaratılması halinde oop kodu tarafında bir nesne oluşturmaya gerek kalmadan otomatik olarak code generation tekniği kullanarak developer tarafından yazılması gereken kodu (class'ı) oluşturur.

.NET için ORM frameworkleri:

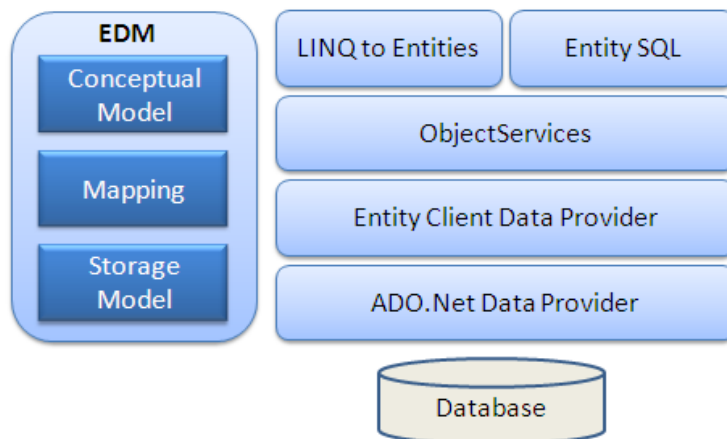
- Entity Framework
- Nhibernate
- .Net Persistence
- DataObjects.NET
- DotNorm
- FastObjects.NET
- Norm
- OJB.NET

Bu araçlar arasında .Net'de en çok kullanılan iki araç "Entity Framework" ve "Nhibernate" araçlarıdır. Entity Framework ise kullanım ve yaygınlık açısından açık ara farkla ilk sıradadır. Eğitim boyunca ORM aracı olarak Entity Framework kullanılacaktır.

Entity Framework nedir?

Entity Framework, demin belirtildiği gibi .NET platformu için geliştirilmiş olan ORM frameworklerinden biridir. Entity framework'ün aslında temel amacı uygulama geliştiricinin data işlemleri ile çok haşır neşir olmadan uygulama tarafına odaklanmasını sağlamaktır. Çok basit bir örnek olarak, klasik ADO.NET uygulamalarında bir bağlantının açılmasından ve kapatılmasından tamamen biz geliştiriciler sorumludur. Ancak entity framework kullandığınızda bu tür işlemlere siz karışmazsınız. Sorgunuzu hazırlar ver entity framework aracılığı ile bunu database'e iletirsiniz.

Entity Framework'un yapısı ise şu şekildedir:



EDM (Entity Data Model): EDM 3 bölümden oluşur. Conceptual Model, Mapping, Storage Model.

Conceptual Model: Bu alanda model sınıflarımız ve bu sınıfların ilişkileri yer alacaktır. Bu sınıflar veritabanı tasarımınızdan bağımsız olacaktır.

Storage Model: Bu alanda veritabanı tasarım modelimiz yer alır. Bu model içerisinde veritabanımıza ait tablolar, view'lar, stored procedure'ler ve bunlara ait ilişkiler ve key'ler yer alır.

Mapping: Bu alan ise model sınıflarımız ile tasarım modelimiz arasındaki haritalama işlemlerinin bilgilerinin tutulduğu alandır.

LINQ to Entities: Nesneleri sorgulamada kullanacağımız sorgulama dilidir. Bu sorgular bize model sınıflarını döndürecektir. Bu alanda LINQ'in yeteneklerini de kullanabiliriz.

Entity SQL: Yine LINQ to Entities gibi sorgulama yapabileceğimiz bir sorgulama dilidir. Ancak Linq To Entities'e göre daha zordur.

Object Services: Veritabanından sorgulama sonucunda alınan verilerin geri döndürülme aşamasında gereken convert (dönüştürme) işlemlerinin yapıldığı alandır.

Entity Client Data Provider: Bu alanın sorumluluğu hazırlanan LINQ to Entities yada Entity SQL sorgularını ADO.Net Data Provider'a gönderilmek üzere anlaşılır SQL sorgularına dönüştürmektir.

ADO.Net Data Provider: Bu katman ise standart ADO.NET kullanarak veritabanı ile iletişim kurar.

Entity Framework (dolayısıyla ORM) avantajları:

- Object Oriented Programming(OOP) için uygun ortam hazırlıyor.
- SQL sorgusu yazmanıza gerek kalmadan veritabanına bağlanıp uygulama yazabiliyorsunuz.
- Veritabanı türü ile bağımlılığı bulunmuyor.
- ORM araçları genellikle Open Source(ücretsiz).
- Test edilmesi kolay kodlar oluşturur.
- Yazılım şirketleri tarafından destekleniyor.
- Geleceğin programcılık mantığını şekillendiriyor.

Entity Framework dezavantajları:

- Performans sorunu yaşanabilir.

- Hakimiyet derecesi azalabilir, sebebi ise bilgi alışverişinin artık EF tarafından yapılmasıdır.

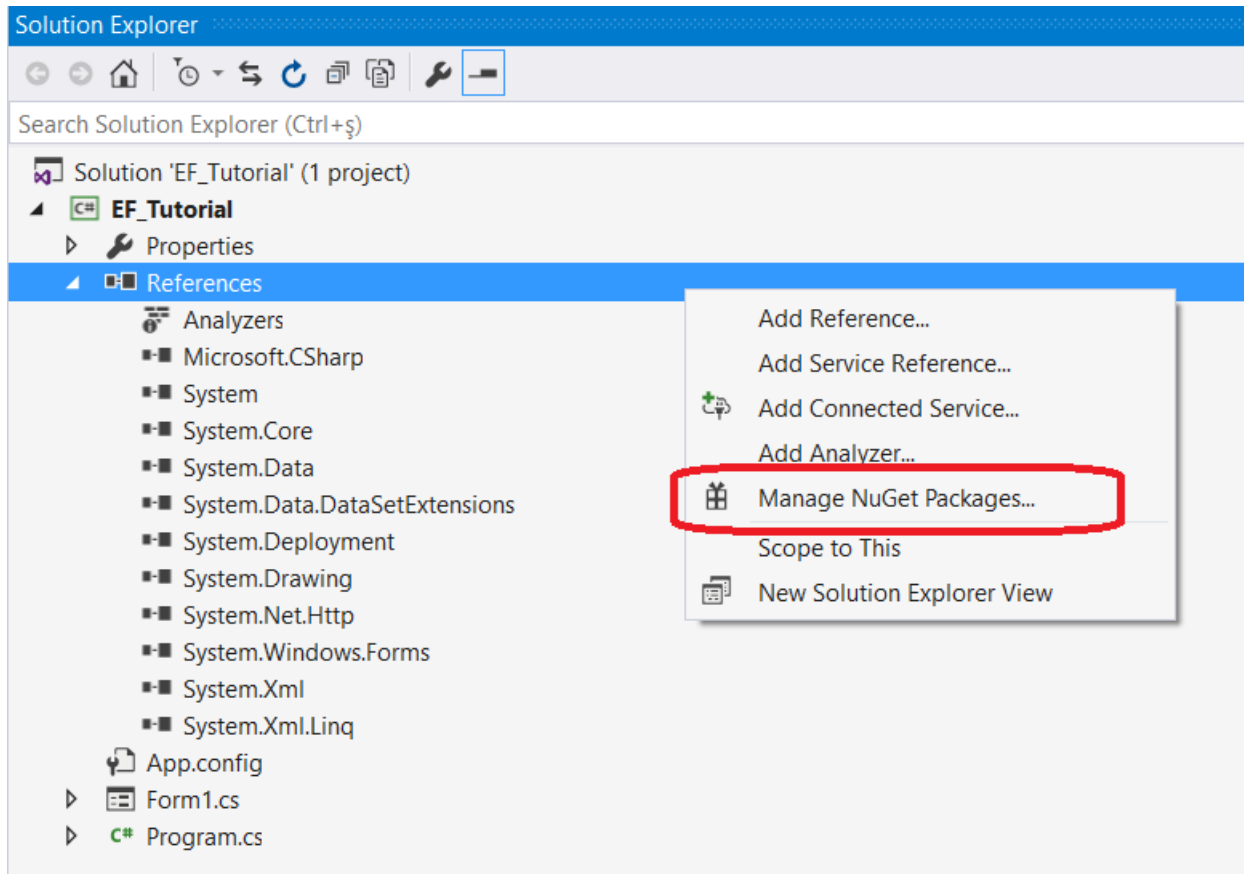
NuGet nedir?

Açık kaynak kütüphaneler ile uygulama geliştirirken karşılaştığımız en önemli sorunlardan biri de kütüphanelerin elde edilmesi, güncellenmesi ve bağlı olduğu diğer kütüphanelerin elde edilmesi sırasında ortaya çıkan sorunlardır. Örneğin bir kütüphaneyi indirdikten sonra tam projeye referans edip kullanmaya başlanacağı sırada indirilen kütüphanenin başka bir kütüphaneye bağlı olduğunu görebilmekte ve bu sefer gidip o kütüphaneyi indirmek gibi sıkıcı süreçlerle karşılaşılabilir.

Özellikle Microsoft cephesine bakıldığında bu şekilde bir süreci otomatize eden bir yapı bulunmamaktaydı. Ancak ASP.NET takımı tarafından geliştirilen NuGet doğrudan Visual Studio içerisine entegre olarak çalışmakta ve shell scriptleri ile ilgili kütüphanelerin ve diğer bağımlı kütüphanelerin indirilerek projeye referans edilmesini sağlamaktadır.

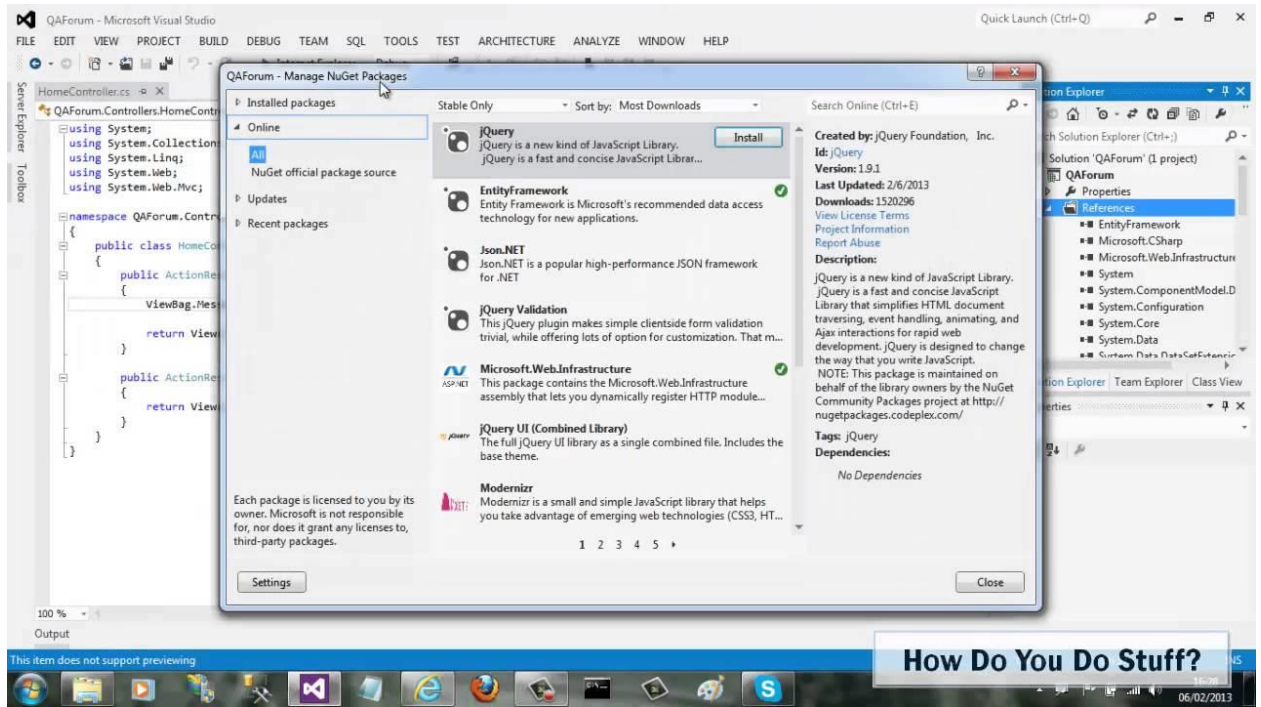
NuGet Kullanımı

Bir proje dosyasına Entity Framework aracı eklenmek istendiğinde aşağıdaki adımlar



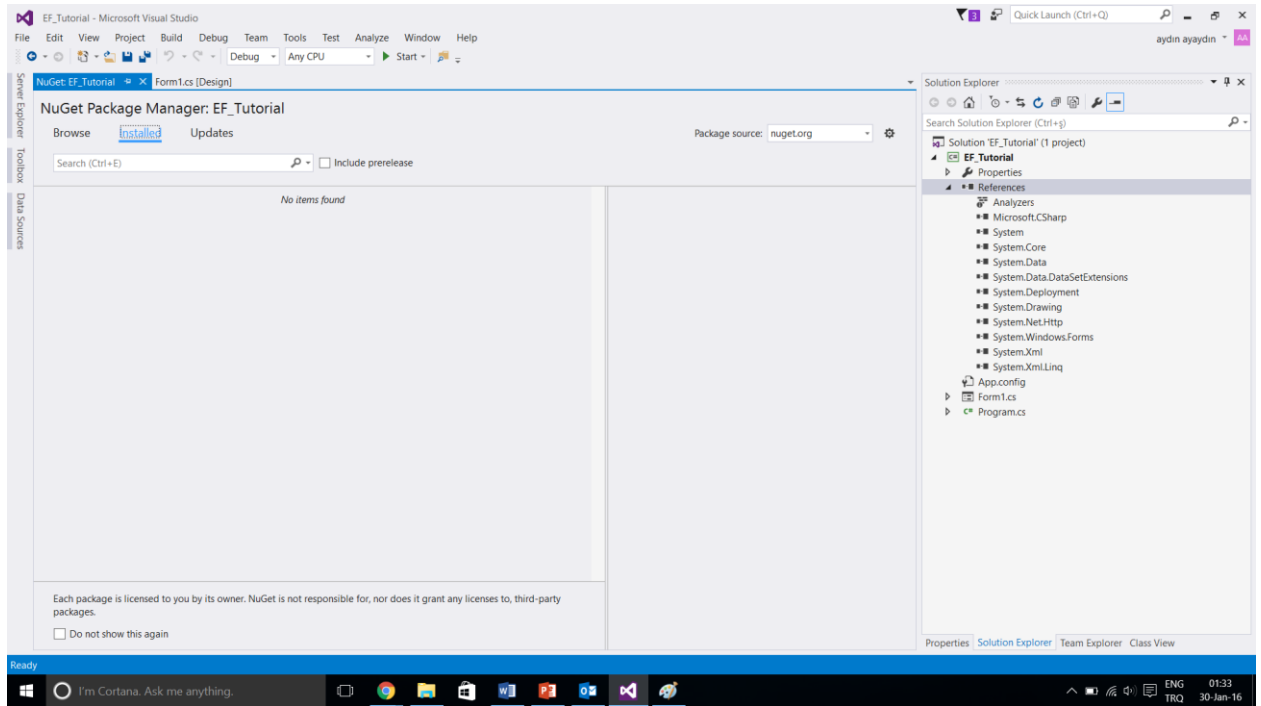
Buradaki "Manage NuGet Packages" komutu tıklandığında aşağıdaki ekranlar gelir:

VS 2013



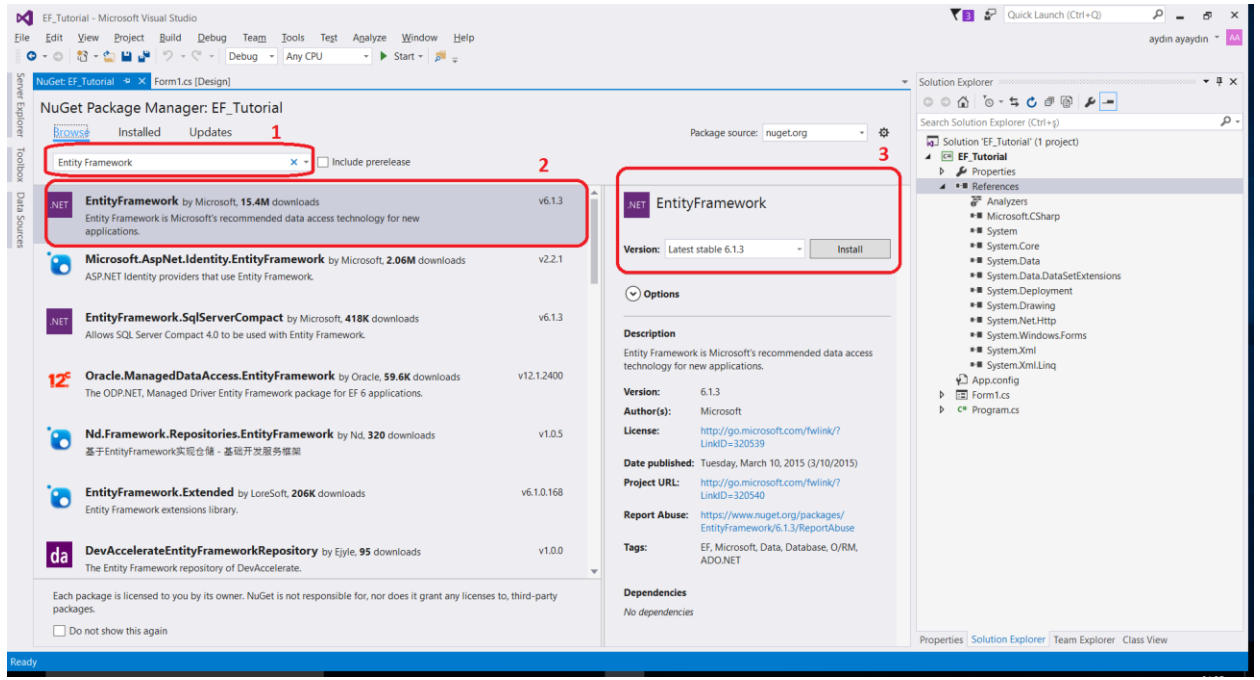
Bu versiyonda 2013'de direct new windows olarak çıkıyor.

Ancak VS 2015'de:

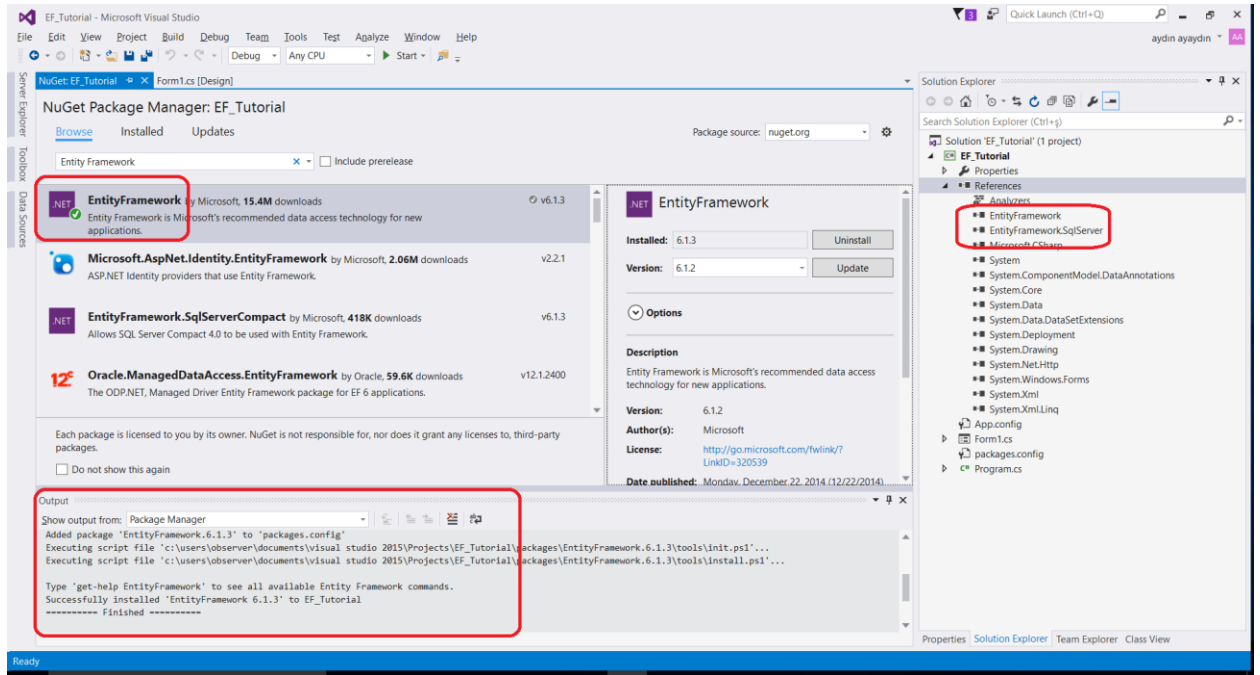


Ancak vs 2015 versiyonunda pencere olarak yazdırmadı.

Devam etmek gerekirse; entity framework NuGet de aranır ve o projeye yüklenir.

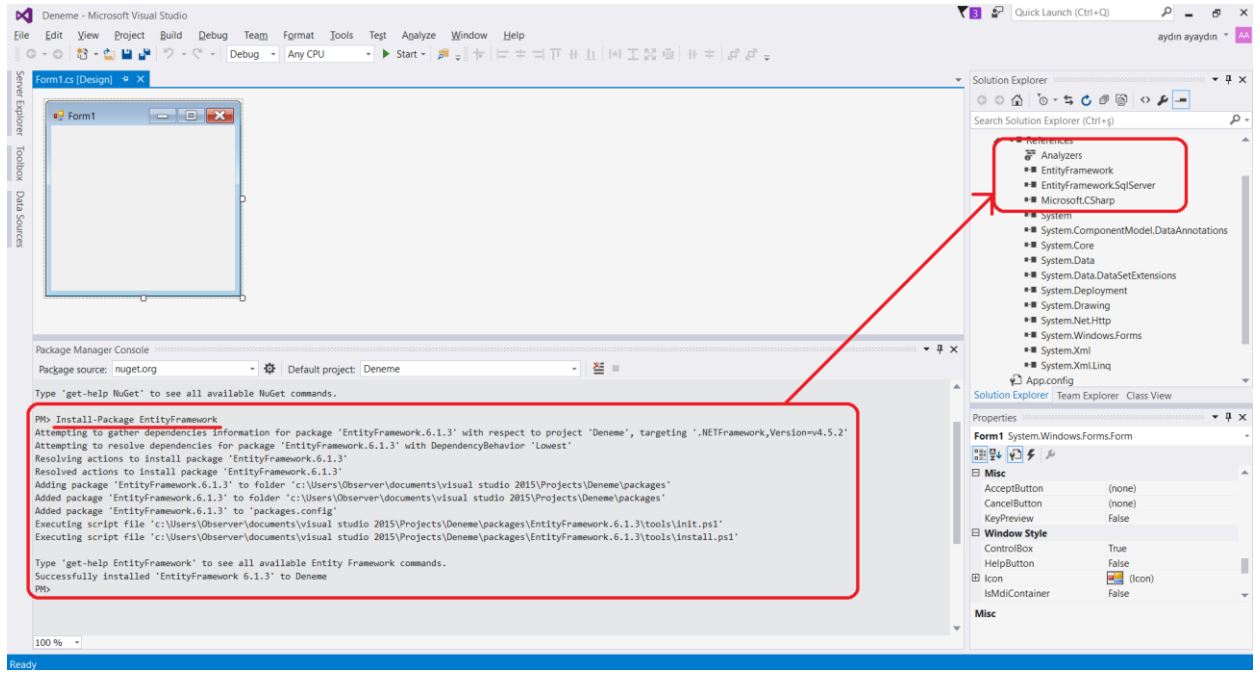


Yukarıdaki şekilde numaralandırılmış şekilde işlemleri sırasıyla uygulamak gerekir.



Burada görülüyor ki kurulum başarıyla yapıldı ve kullanıma hazır.

Bir diğer kurulum da Package Manager Console üzerinden yapmaktır. Bunun için package manager tool penceresi açık olmalıdır. Açmak için View > Other Window > Packed Manager Controller



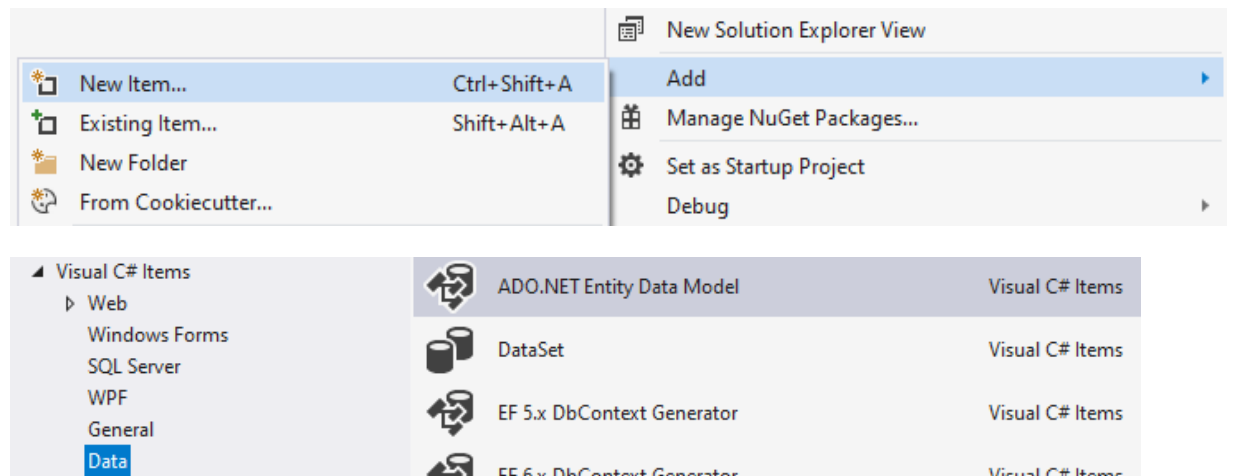
Derste yapılan EF Örnekleri:

dbOkul Örneği:

dbOkul adında veritabanı oluşturulup içerisine Öğrenci adında bir tablo oluşturuyoruz.

MD.dbOkul - dbo.ogrenci		MD.Northwind - dbo.Employees	
Column Name	Data Type	Allow Nulls	
OğrenciID	int	☐	
Numara	nvarchar(4)	☒	
Ad	nvarchar(50)	☒	
Soyad	nvarchar(50)	☒	
		☐	

Daha sonra visual studio üzerinden yeni bir form uygulaması açıyoruz. Açtığımız uygulamada form tasarımı yapmadan önce oluşturduğumuz veri tabanı modelini ekliyoruz.





Choose Model Contents

What should the model contain?

EF Designer
from
databaseEmpty EF
Designer
modelEmpty Code
First modelCode First
from
database

Creates a model in the EF Designer based on an existing database. You can choose the database connection, settings for the model, and database objects to include in the model. The classes your application will interact with are generated from the model.

< Previous

Next >

Finish

Cancel



Choose Your Data Connection

Which data connection should your application use to connect to the database?

md.dbOkul.dbo

New Connection...

This connection string appears to contain sensitive data (for example, a password) that is required to connect to the database. Storing sensitive data in the connection string can be a security risk. Do you want to include this sensitive data in the connection string?

- ☐ No, exclude sensitive data from the connection string. I will set it in my application code.
- ☐ Yes, include the sensitive data in the connection string.

Connection string:

```
metadata=res://*/Model2.csdl|res://*/Model2.ssdl|  
res://*/Model2.msl;provider=System.Data.SqlClient;provider connection string="data source=MD;initial  
catalog=dbOkul;integrated security=True;MultipleActiveResultSets=True;App=EntityFramework"
```

☒ Save connection settings in App.Config as:

dbOkulEntities1

< Previous

Next >

Finish

Cancel

Entity Data Model Wizard

Choose Your Database Objects and Settings

Which database objects do you want to include in your model?

☒ Tables

☐ Views

☐ Stored Procedures and Functions

☒ Pluralize or singularize generated object names

☒ Include foreign key columns in the model

☐ Import selected stored procedures and functions into the entity model

Model Namespace:

dbOkulModel1

< Previous Next > Finish Cancel

Model ekleme sihirbazı her uygulama için aynı şekilde yapılacaktır. O nedenle uygulamalar üzerinde tekrar tekrar gösterilmeyecektir. Daha sonra form tasarımına geçiyoruz.

Form1

ID

Numara

Ad

Soyad

Ekle Güncelle Sil

```

public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }
    dbOkulEntities db = new dbOkulEntities();

    void doldur()
    {
        dataGridView1.DataSource = db.ogrencis.ToList();
    }
    private void Form1_Load(object sender, EventArgs e)
    {
        doldur();
    }
    private void btnEkle_Click(object sender, EventArgs e)
    {
        ogrenci yeniOgrenci = new ogrenci();
        yeniOgrenci.Numara = txtNumara.Text;
        yeniOgrenci.Ad = txtAd.Text;
        yeniOgrenci.Soyad = txtSoyad.Text;
        db.ogrencis.Add(yeniOgrenci);
        db.SaveChanges();
        doldur();
    }
}

```

```

private void btnGuncelle_Click(object sender, EventArgs e)
{
    int guncelle = Convert.ToInt32(txtOgrenciID.Text);

    var guncellenecekOgrenci = db.ogrencis.Where(x => x.OgrenciID ==
        guncelle).FirstOrDefault();

    guncellenecekOgrenci.Numara = txtNumara.Text;
    guncellenecekOgrenci.Ad = txtAd.Text;
    guncellenecekOgrenci.Soyad = txtSoyad.Text;

    db.SaveChanges();
    doldur();
}

```

```

private void btnSil_Click(object sender, EventArgs e)
{
    int silinecek = Convert.ToInt32(txtOgrenciID.Text);

    var silinecekOgrenci = db.ogrencis.Where(x => x.OgrenciID ==
        silinecek).FirstOrDefault();

    db.ogrencis.Remove(silinecekOgrenci);
    db.SaveChanges();
    doldur();
}

```

Northwind dbFirst Categories Örneği:

```

public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }

    //1 Data Base tanımı
    int secilenid;
    private void Form1_Load(object sender, EventArgs e)
    {
        Doldur();
    }
}

```

```

private void Doldur()
{
    NorthwindEntities db = new NorthwindEntities();
    //1. Yol Lambda expresion
    //Db içinde Ketgori oku ve Data source eşitle
    // dataGridView1.DataSource = db.Categories.ToList();

    // 2. Yol Lambda Expression
    //ICollection<Categories> clist = db.Categories.ToList();
    //dataGridView1.DataSource = clist;

    //3. Yol Lambda Expression
    var clist = db.Categories.Select(x => new
    {
        x.CategoryID,
        KategoriAdı = x.CategoryName,
        Açıklama = x.Description,
        Resim = x.Picture
    }).Where(x => x.KategoriAdı.Contains(txtAra.Text)).OrderBy(x =>
        x.KategoriAdı).ToList();
    dataGridView1.DataSource = clist;
    dataGridView1.Columns[0].Visible = false;

    //4. Yol Linq
    //dataGridView1.DataSource = (from c in db.Categories
    //                             select new
    //                             {
    //                                 c.CategoryID,
    //                                 c.CategoryName,
    //                                 Açıklama = c.Description,
    //                                 c.Picture
    //                             }).ToList();
}

private void txtAra_TextChanged(object sender, EventArgs e)
{
    Doldur();
}

```

```

private void dataGridView1_CellMouseClick(object sender,
DataGridViewCellMouseEventArgs e)
{
    NorthwindEntities db = new NorthwindEntities();
    secilenid = Convert.ToInt32(dataGridView1.CurrentRow.Cells
        ["CategoryID"].Value);
    // Komumlama
    Categories c = db.Categories.Where(x => x.CategoryID ==
        secilenid).FirstOrDefault();
    txtAd.Text = c.CategoryName;
    txtAciklama.Text = c.Description;
    // byte[] imagesource = c.Picture;
    pictureBoxResim.Image = null;

    if (c.Picture != null)
    {
        ImageConverter converter = new ImageConverter();
        pictureBoxResim.Image = (Image)converter.ConvertFrom(c.Picture);
    }
}

```

```

private void btnEkle_Click(object sender, EventArgs e)
{
    NorthwindEntities db = new NorthwindEntities();
    // eklemede Tablo (Entity ) new ile intantance değişkeni alır
    // db.entity.Add() çalıştırılır
    // birde db nin savechanged methodu çalıştırılır
    Categories c = new Categories();
    c.CategoryName = txtAd.Text;
    c.Description = txtAciklama.Text;
    db.Categories.Add(c);
    db.SaveChanges();
    Doldur();
}

```

```

private void btnGuncelle_Click(object sender, EventArgs e)
{
    NorthwindEntities db = new NorthwindEntities();
    // Güncellemede konumlama yapılır . Gereki Alanlar eşitlenir
    // db.SaveChanges metot çalıştırılır

    // Konumlama 1. Yol
    //Categories c = db.Categories.Where(x => x.CategoryID ==
        secilenid).FirstOrDefault();

    // Konumlama 2. Yol
    Categories c = db.Categories.Find(secilenid);
    c.CategoryName = txtAd.Text;
    c.Description = txtAciklama.Text;
    db.SaveChanges();
    Doldur();
}

private void btnSil_Click(object sender, EventArgs e)
{
    NorthwindEntities db = new NorthwindEntities();
    try
    {
        Categories c = db.Categories.Find(secilenid);
        db.Categories.Remove(c);
        db.SaveChanges();
        Doldur();
    }
    catch (Exception ex)
    {
        MessageBox.Show("Yanlış Kayıt !" + ex.InnerException);
    }
}

```

Northwind dbFirst Products Örneği:

```
int secilenid;
private void Form1_Load(object sender, EventArgs e)
{
    Doldur();
    KategoriDoldur();
    TedarikciDoldur();
}
```

```
private void TedarikciDoldur()
{
    NorthwindEntities db = new NorthwindEntities();
    var list = db.Suppliers.Select(x => new
    {
        x.SupplierID,
        x.CompanyName
    }).ToList();
    cmbTedarikci.DisplayMember = "CompanyName";
    cmbTedarikci.ValueMember = "SupplierID";
    cmbTedarikci.DataSource = list;
}
```

```
private void KategoriDoldur()
{
    NorthwindEntities db = new NorthwindEntities();
    var list = db.Categories.Select(x => new
    {
        x.CategoryID,
        x.CategoryName
    }).ToList();
    cmbKategori.DisplayMember = "CategoryName";
    cmbKategori.ValueMember = "CategoryID";
    cmbKategori.DataSource = list;
}
```

```
private void Doldur()
{
    NorthwindEntities db = new NorthwindEntities();

    var plist = db.Products.Select(x => new
    {
        x.ProductID,
        x.Categories.CategoryName,
        x.ProductName,
        x.Suppliers.CompanyName,
    });
}
```

```

        x.Suppliers.ContactName,
        x.QuantityPerUnit,
        x.UnitPrice,
        x.UnitsInStock
    })).Where(x => x.ProductName.Contains(txtAra.Text)).ToList();

    dataGridView1.DataSource = plist;
    // var top = db.Products.Sum(x => x.UnitPrice);
    // var ave = db.Products.Average(x => x.UnitPrice);
    // var cnt = db.Products.Count();

    // label8.Text = top.ToString();
}

private void txtAra_TextChanged(object sender, EventArgs e)
{
    Doldur();
}

private void btnEkle_Click(object sender, EventArgs e)
{
    NorthwindEntities db = new NorthwindEntities();
    Products p = new Products();
    p.ProductName = txtUrunAd.Text;
    p.UnitPrice = Convert.ToDecimal(txtFiyat.Text);
    p.UnitsInStock = Convert.ToInt16(txtStok.Text);
    p.QuantityPerUnit = txtMiktar.Text;
    p.SupplierID = (int)cmbTedarikci.SelectedValue;
    p.CategoryID = (int)cmbKategori.SelectedValue;
    db.Products.Add(p);
    db.SaveChanges();
    Doldur();
}

private void btnSil_Click(object sender, EventArgs e)
{
    Products p = db.Products.Find(secilenid);
    db.Products.Remove(p);
    db.SaveChanges();
    Doldur();
}

```



```

private void btnGuncelle_Click(object sender, EventArgs e)
{
    // Products p = db.Products.Where(x=> x.ProductID==
    secilenid).FirstOrDefault()
    Products p = db.Products.Find(secilenid);
    p.ProductName = txtUrunAd.Text;
    p.QuantityPerUnit = txtMiktar.Text;
    p.UnitPrice = Convert.ToDecimal(txtFiyat.Text);
    p.UnitsInStock = Convert.ToInt16(txtStok.Text);
    p.SupplierID = (int)cmbTedarikci.SelectedValue;
    p.CategoryID = (int)cmbKategori.SelectedValue;
    db.SaveChanges();
    Doldur();
}

private void dataGridView1_CellMouseClick(object sender,
DataGridViewCellMouseEventArgs e)
{
    secilenid = (int)dataGridView1.CurrentRow.Cells[0].Value;
    Products p = db.Products.Find(secilenid);
    txtUrunAd.Text = p.ProductName;
    txtStok.Text = p.UnitsInStock.ToString();
    txtFiyat.Text = p.UnitPrice.ToString();
    txtMiktar.Text = p.QuantityPerUnit;
    cmbKategori.SelectedValue = p.CategoryID;
    cmbTedarikci.SelectedValue = p.SupplierID;
}

```

Northwind dbFirst Employees Örneği:

The screenshot shows a Windows application window titled "Form1". It features a search bar labeled "Ara" with a text input field. Below the search bar is a large gray rectangular area, which is likely a data grid or list box. To the right of this area, there are several input fields: "Adı" (Name), "Soyadı" (Surname), "Giriş Tarihi" (Start Date) set to "18 Eylül 2020 Cuma", and "Yöneticisi" (Manager) with a dropdown arrow. At the bottom right, there are three buttons: "Ekle" (Add), "Güncelle" (Update), and "Sil" (Delete).

```

NorthwindEntities db = new NorthwindEntities();
int secilenId;

void comboDoldur()
{
    var emp = db.Employees.Select(x => new
    {
        x.EmployeeID,
        Adsoyad = x.FirstName + " " + x.LastName

    }).Where(x => x.EmployeeID == 2 || x.EmployeeID == 5).ToList();

    cmBoxYonetici.DisplayMember = "Adsoyad";
    cmBoxYonetici.ValueMember = "EmployeeID";
    cmBoxYonetici.DataSource = emp;
}

void doldur()
{
    var emp = db.Employees.Select(x => new
    {
        x.EmployeeID,
        x.FirstName,
        x.LastName,
        x.HireDate,

        patronAd= x.Employee1.FirstName + " " +
            x.Employee1.LastName

    }).Where(ara => ara.FirstName.Contains(txtAra.Text) ||
        ara.LastName.Contains(txtAra.Text)).ToList();

    dataGridView1.DataSource = emp;
}

private void Form1_Load(object sender, EventArgs e)
{
    doldur();
    comboDoldur();
}

private void txtAra_TextChanged(object sender, EventArgs e)
{
    doldur();
}

```

```

private void btnEkle_Click(object sender, EventArgs e)
{
    Employee em = new Employee();
    em.FirstName = txtBoxAd.Text;
    em.LastName = txtBoxSoyad.Text;
    em.HireDate = dtPickerGirisTarihi.Value;
    em.ReportsTo = (int)cmBoxYonetici.SelectedValue;
    db.Employees.Add(em);
    db.SaveChanges();
    doldur();
}

private void dataGridView1_CellMouseClick(object sender,
DataGridViewCellMouseEventArgs e)
{
    secilenId = (int)dataGridView1.CurrentRow.Cells[0].Value;
    try {
        Employee emp = db.Employees.Where(x => x.EmployeeID ==
            secilenId).FirstOrDefault();
        txtBoxAd.Text = emp.FirstName;
        txtBoxSoyad.Text = emp.LastName;
        dtPickerGirisTarihi.Value = emp.HireDate.Value;
        cmBoxYonetici.SelectedValue = emp.ReportsTo;
    }
    catch
    { }
}

```

```

private void btnGuncel_Click(object sender, EventArgs e)
{
    Employee empAl = db.Employees.Find(secilenId);
    empAl.FirstName = txtBoxAd.Text;
    empAl.LastName = txtBoxSoyad.Text;
    empAl.HireDate = dtPickerGirisTarihi.Value;
    empAl.ReportsTo = (int)cmBoxYonetici.SelectedValue;
    db.SaveChanges();
    doldur();
}

```

```
private void btnSil_Click(object sender, EventArgs e)
{
    try {
        Employee empAl = db.Employees.Find(secilenId);
        db.Employees.Remove(empAl);
        db.SaveChanges();
        MessageBox.Show("Silme işlemi başarılı");
        doldur();
    }
    catch(Exception ex)
    {
        MessageBox.Show("Silme işlemi başarısız Çünkü, " + " " +
            ex.InnerException.Message);
    }
}
```