



# Entity Framework

Doç.Dr. Haydar Ankişhan

# Entity Framework nedir?

- ▶ Entity Framework (EF) 2008 yılından beridir Microsoft tarafından geliştirilen bir ORM (Object to Relational Mapping) aracıdır.
- ▶ EF sayesinde yazılım geliştirici veritabanı ve SQL komutlarına ihtiyaç duymadan yazılım geliştirebilir.



# Neden Kullanılır?

- ▶ NET veya herhangi bir programlama dili ile veri tabanı uygulamaları yaparken ilk olarak veritabanı bağlantısı yapılır,
- ▶ Daha sonra SQL komutları ile veri tabanındaki veriler alınır.
- ▶ Alınan veriler programlama diline uygun veri yapılarında saklanarak işlem yapılır,
- ▶ Verilerin programlama diline uygun veri yapılarına dönüştürülmesi sırasında beklenmedik hatalar, sorunlar olabilir, bunun yanı sıra karmaşık veritabanı sorguları geliştirmeyi daha da zor hale getirir,
- ▶ ORM araçlarının temel kullanım sebebi bu ve bunun gibi sorunları ortadan kaldırmaktır,
- ▶ EF gibi ORM araçları farklı veri sağlayıcıları (SQL Server, SQLite vb.) için aynı komutları kullanarak işlem yapmaya imkan sunar.

# EF Kurulumu

- ▶ EF aracı bir NuGet paketi olduğu için VS geliştirme ortamında projeye sağ tıklayıp Manage NuGet Packages seçeneği ile kurulum sağlanır,
- ▶ Ayrıca Package Manager Console alanına
  - ▶ Install-Package EntityFramework
  - ▶ Yazarak da aynı işlem gerçekleştirilir

# Model oluřturma

- Database First, Model First ve Code First yaklařımları kullanılır



# Database First

- ▶ Bu yapıda daha önce hazırlanmış veritabanı tablolarına uygun nesneler Visual Studio veya komut satırı ile otomatik olarak oluşturulur,
- ▶ Bu işlem için VS IDE ile projeye ADO.NET Entity Data Model ve EF Designer from Database seçeneği ile veritabanı bağlantısı yapılır ve ilgili tablolar eklenir

# Model First

- Bu yapıda Database First yapısına benzer şekilde ADO.NET Entity Data Model ve EF Designer from Model seçeneği ile oluşturulur.
- Oluşturulan modele görsel olarak tablolar, tablo alanları eklenerek model oluşturulur

# Code First

- ▶ Yukarıda yer alan Database First ve Model First ile aslında yapılan görsel ara yüz ile veritabanı veya model uygun kodların oluşturulmasıdır.
- ▶ Code First yaklaşımında ilk olarak tabloya karşılık gelen sınıf ve özellikler yazılır

```
public class Person
{
    public int ID { get; set; }
    public string FirstName { get; set; }
    public string LastName { get; set; }
    public string Email { get; set; }
    public string FullName => $"{FirstName} {LastName}";
}
```



# Code First

- ▶ CF ile geliştirme sırasında modeli oluştururken Data Annotations veya Fluent API ile özellik sınırının belirlenmesi faydalı olacaktır.
- ▶ EF string türünde bir değer için veri tabanında Nvarchar (MAX) gibi bir tür belirleyecektir.
- ▶ Model içerisinde yer alan ID özelliği varsayılan olarak EF tarafından birincil anahtar olarak belirlenir, ID isimlendirmesinden farklı bir alan için yine nitelik (Key) veya Fluent API kullanılabilir,
- ▶ Ancak daha sonra olası değişiklik için ID kullanmak faydalı olacaktır.
- ▶ EF enum ile hazırlanan özellikler içinde varsayılan olarak veritabanında int türünü seçer,
- ▶ Enum türleri genellikle az boyutlu olduğundan byte gibi bir tür belirlemek faydalı olacaktır.

## Basit bir EF örneği

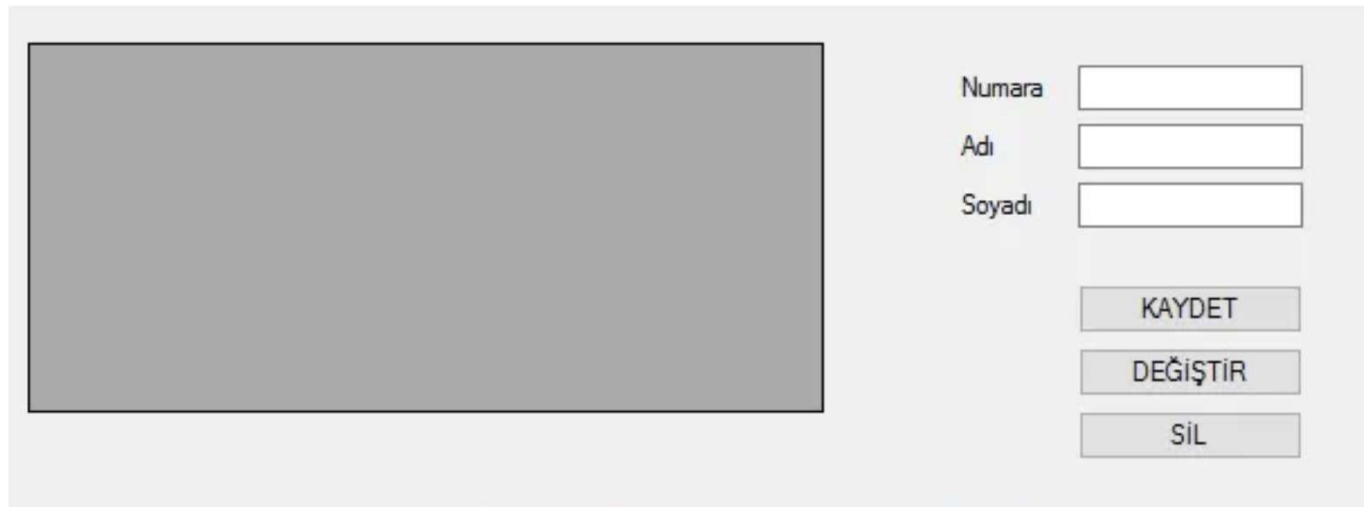
- Bu örneğimizde C# Windows Form kullanarak Entity Framework ile Select, Insert, Update ve Delete işlemlerini gerçekleştiren basit bir örnek yapacağız. Örneğimizde SQL Serverde bulunan dbOkul isimli veritabanını kullanacağız. Öncelikle veritabanımızı oluşturarak başlıyoruz.

CASPERAC.dbOkul - dbo.ogrenci			
	Column Name	Data Type	Allow Nulls
▶	Numara	int	<input type="checkbox"/>
	Ad	nvarchar(50)	<input checked="" type="checkbox"/>
	Soyad	nvarchar(50)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

CASPERAC.dbOkul - dbo.ogrenci			
	Numara	Ad	Soyad
▶	152	Ahmet	Cansever
	354	FatmaNur	Gül
	555	Selami	Tuna
*	NULL	NULL	NULL

## Basit bir EF örneđi

- Veri tabanımızı oluřturduktan sonra **Form** tasarımını ařađıdaki gibi oluřturuyoruz.



Numara

Adı

Soyadı

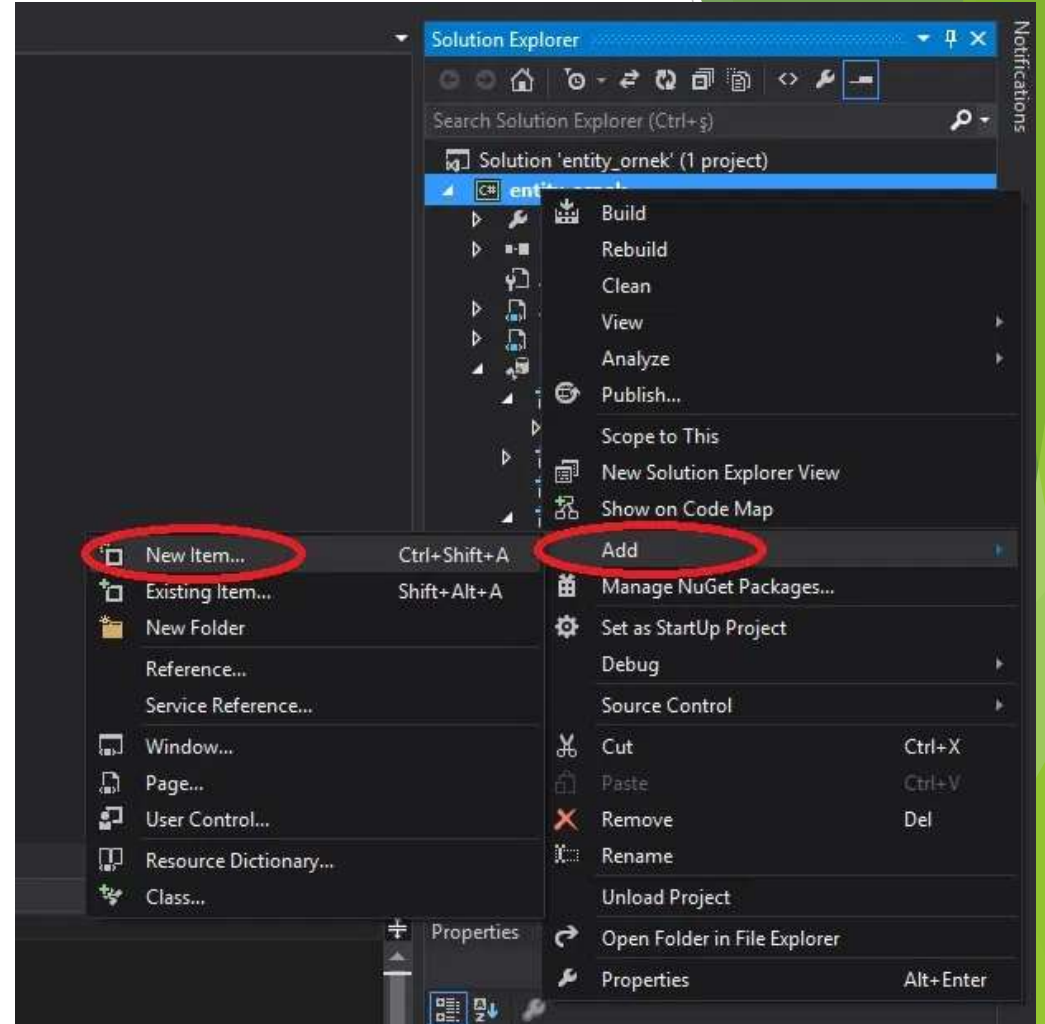
KAYDET

DEđİřTİR

SİL

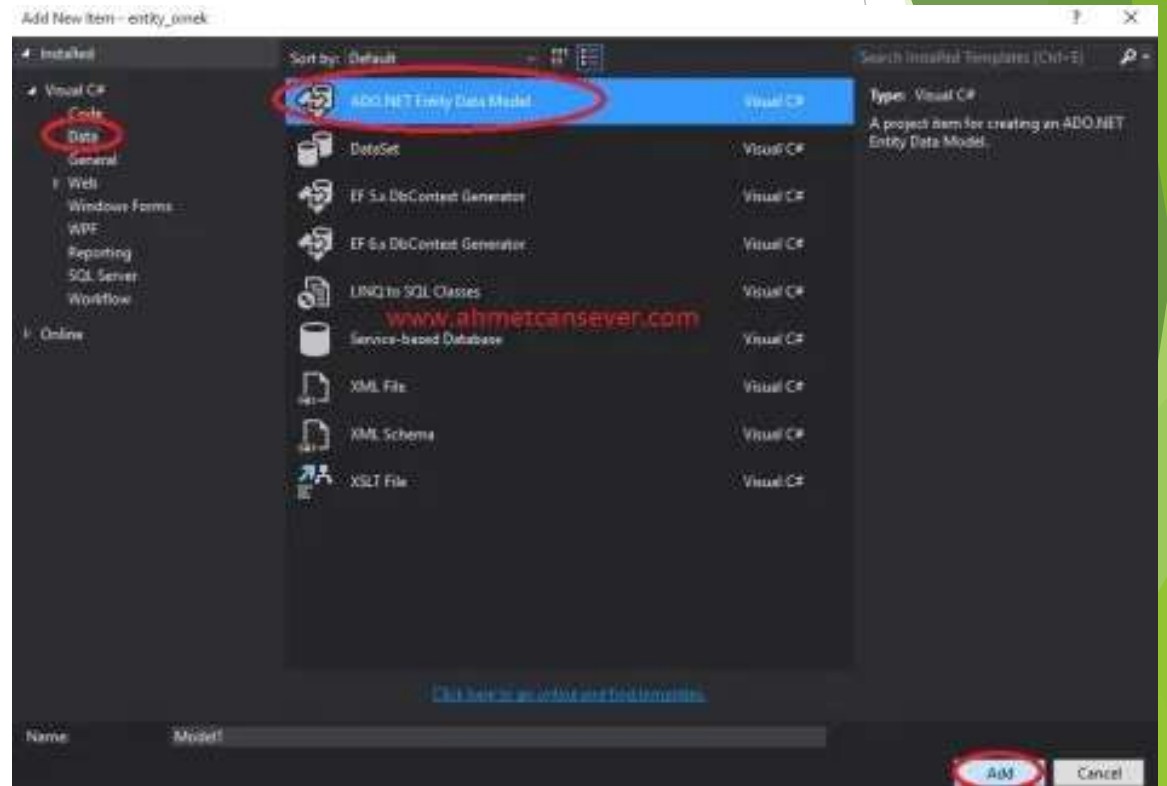
## Basit bir örnek

- Form tasarımını tamamladıktan sonra Entity Framework bağlantımızı aşağıdaki adımları takip ederek oluşturuyoruz.
- Öncelikle Solution Explorer Penceresinde projemizin üzerinde sağ tıklayarak Add-New Item tıklıyoruz.



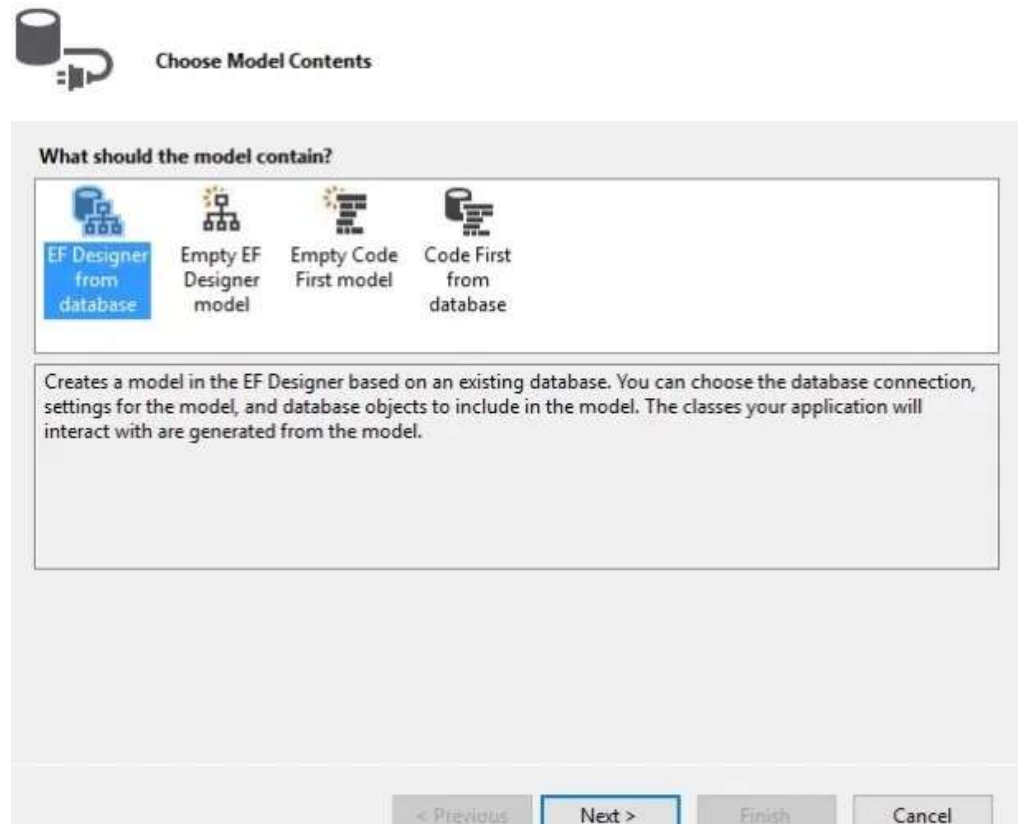
## Basit bir örnek

- Açılan pencereden ADO.NET Entity Data Model seçip Add butonuna basıyoruz.



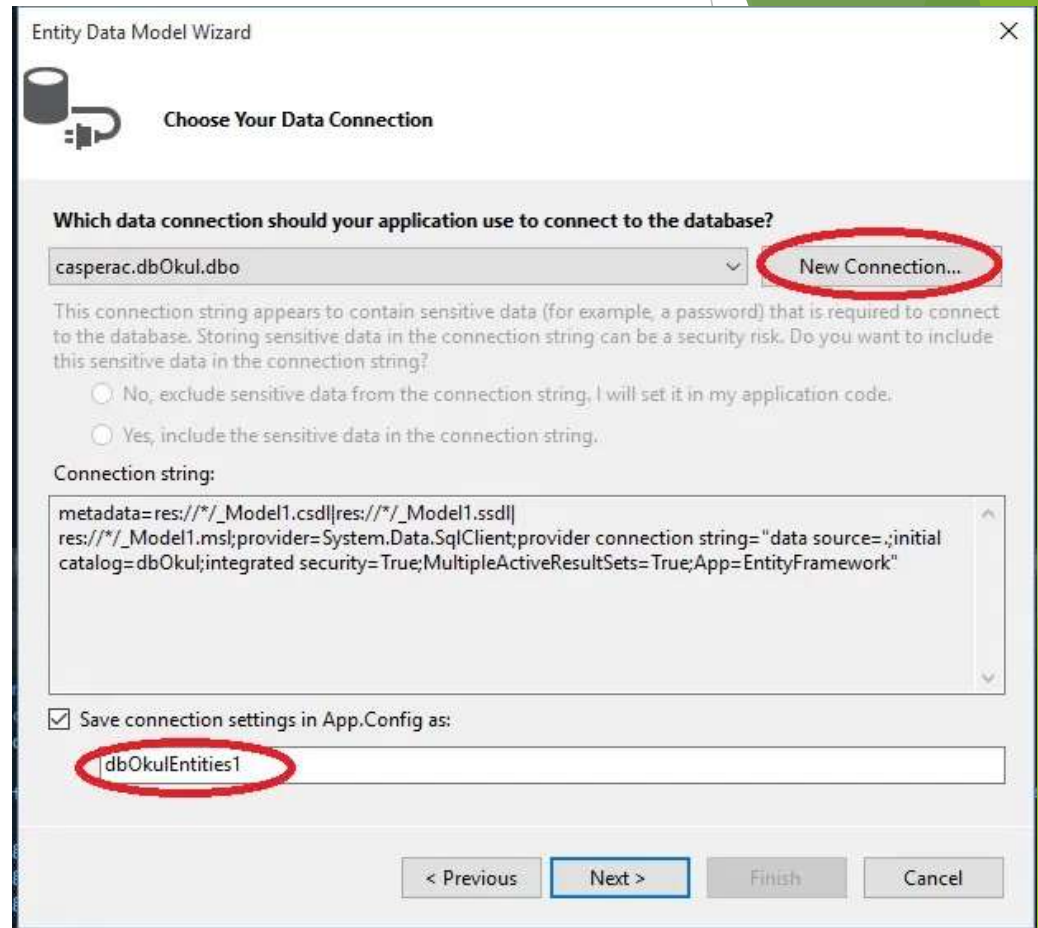
## Basit bir örnek

- Daha sonra açılan pencereden EF Designer from database seçeneğini seçiyoruz.



## Basit bir örnek

- Buradan New Connection diyerek SQL Server' da kullanacağımız database i seçiyoruz ve devam ediyoruz.



The image shows the 'Entity Data Model Wizard' dialog box, specifically the 'Choose Your Data Connection' step. The dialog has a title bar with a close button. Below the title bar is a section with a database icon and the text 'Choose Your Data Connection'. The main area contains a question: 'Which data connection should your application use to connect to the database?'. Below this is a dropdown menu showing 'casperac.dbOkul.dbo' and a 'New Connection...' button, which is circled in red. Below the dropdown is a warning message: '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?'. There are two radio buttons: 'No, exclude sensitive data from the connection string, I will set it in my application code.' and 'Yes, include the sensitive data in the connection string.'. Below this is a 'Connection string:' label and a text area containing a connection string. At the bottom, there is a checkbox 'Save connection settings in App.Config as:' which is checked, and a text box containing 'dbOkulEntities1', which is also circled in red. At the very bottom are four buttons: '< Previous', 'Next >', 'Finish', and 'Cancel'.

Entity Data Model Wizard

Choose Your Data Connection

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

casperac.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://\*/\_Model1.csdl|res://\*/\_Model1.ssdl|res://\*/\_Model1.msl;provider=System.Data.SqlClient;provider connection string="data source=.;initial catalog=dbOkul;integrated security=True;MultipleActiveResultSets=True;App=EntityFramework"

☒ Save connection settings in App.Config as:

dbOkulEntities1

< Previous Next > Finish Cancel

# Basit bir örnek

- Gelen pencerede Tables seçeneğini işaretliyoruz.

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:

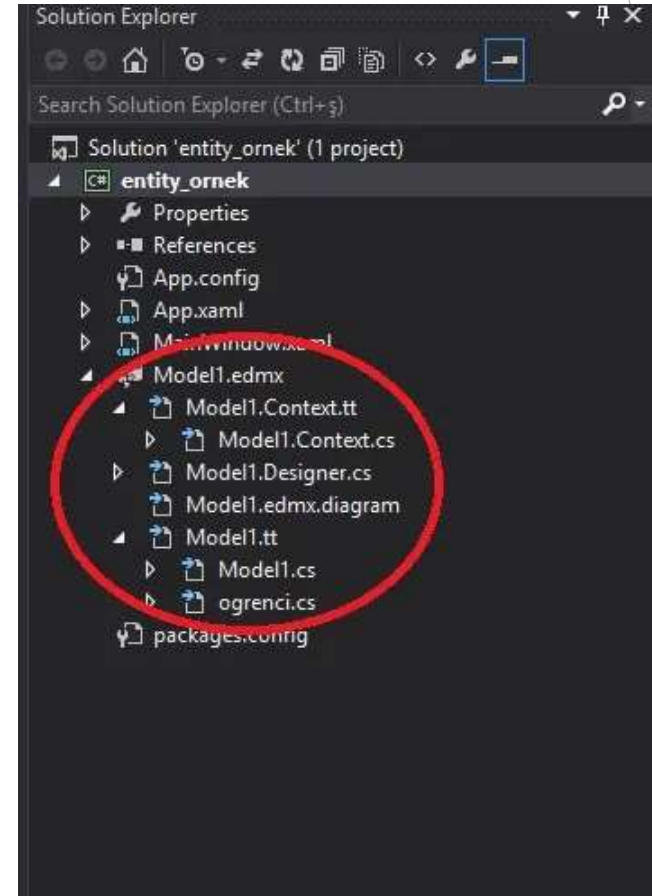
dbOkulModel1

< Previous Next > Finish Cancel



## Basit bir örnek

- Bu işlemlerden sonra oluşturulan Entity Data Model' imizin Solution Explorer penresine geldiğini göreceğiz.



## Basit bir örnek

- ▶ Bağlantı işlemlerini tamamladıktan sonra kod kısmına geçiyoruz.
- ▶ Veriler datagridview üzerinde Form açıldığında , Ekleme , Güncelleme ve Silme yapıldığında tazelenmesini sağlamak amacıyla Datagridview' e getirilme işlemini bir metot oluşturarak yapacağız.
- ▶ Öncelikle Public olarak nesnemizi aşağıdaki gibi oluşturuyoruz

# Basit bir örnek

- Öncelikle Public olarak nesnemizi aşağıdaki gibi oluşturuyoruz
- `dbOkulEntities db;`

ve Datagridview' e doldurma işlemini `doldur` isimli bir metot ile gerçekleştiriyoruz.

## Basit bir örnek

- ve Datagridview' e doldurma işlemini doldur isimli bir metot ile gerçekleştiriyoruz.

```
4 void doldur() //Verileri Çekme (Select)
5 {
6     db = new dbOkulEntities();
7     dataGridView1.DataSource = db.ogrencis.ToList();
}
```

## Basit bir örnek

- Bu işlemden sonra form açılır açılmaz datagridview'de verilerimizin görüntülenmesi için Form\_Load olayında

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

## Basit bir örnek

- ▶ “doldur” isimli metodumuzu çağırıyoruz.
- ▶ Daha sonra Kayıt Ekle butonuna çift tıklayarak aşağıdaki kodları yazıyoruz.

```
2
3
4
5     ogrenci yeniogrenci = new ogrenci();
6     yeniogrenci.Numara = Convert.ToInt32(textBox1.Text);
7     yeniogrenci.Ad = textBox2.Text;
8     yeniogrenci.Soyad = textBox3.Text;
9     db.ogrencis.Add(yeniogrenci);
10    db.SaveChanges();
11    doldur();
12 }
```

## Basit bir örnek

- Bu sayede textboxlara öğrenci bilgilerini girip Ekledikten sonra veri tabanına ekleme ( insert) işlemi yapılacak ve doldur metodu tekrar çağırılarak verilerin tekrar getirilmesi sağlanacaktır.
- Güncelle butonuna ise;

```
2     private void button2_Click(object sender, EventArgs e) //Güncelleme
3     {
4         int guncelle = Convert.ToInt32(textBox1.Text);
5         var guncellenecekogrenci = db.ogrencis.Where(w => w.Numara ==
6 guncelle).FirstOrDefault();
7         guncellenecekogrenci.Ad = textBox2.Text;
8         guncellenecekogrenci.Soyad = textBox3.Text;
9         db.SaveChanges();
10        doldur();
11    }
```

## Basit bir örnek

- kodlarını yazıyoruz. Burada öğrenci numarasına göre güncelleme işlemi yapıldığına dikkat edelim.
- Sıra Sil butonuna geldi. Yine silme işlemi de Öğrenci Numarasına göre yapılacağından kodlarımızı aşağıdaki şekilde oluşturuyoruz.

```
1 private void button3_Click(object sender, EventArgs e) //Silme
2 {
3     int silinecek = Convert.ToInt32(textBox1.Text);
4     var silinecekkisi = db.ogrencis.Where(w => w.Numara == silinecek).FirstOrDefault();
5     db.ogrencis.Remove(silinecekkisi);
6     db.SaveChanges();
7     doldur();
8 }
9
10
```