

## C# Windows Form Entity Framework Veritabanı Bağlantısı

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.

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"/>

	Numara	Ad	Soyad
▶	152	Ahmet	Cansever
	354	FatmaNur	Gül
	555	Selami	Tuna
*	NULL	NULL	NULL

Veri tabanımızı oluşturduktan sonra **Form** tasarımını aşağıdaki gibi oluşturuyoruz.

www.ahmetcansever.com

Numara

Adı

Soyadı

KAYDET

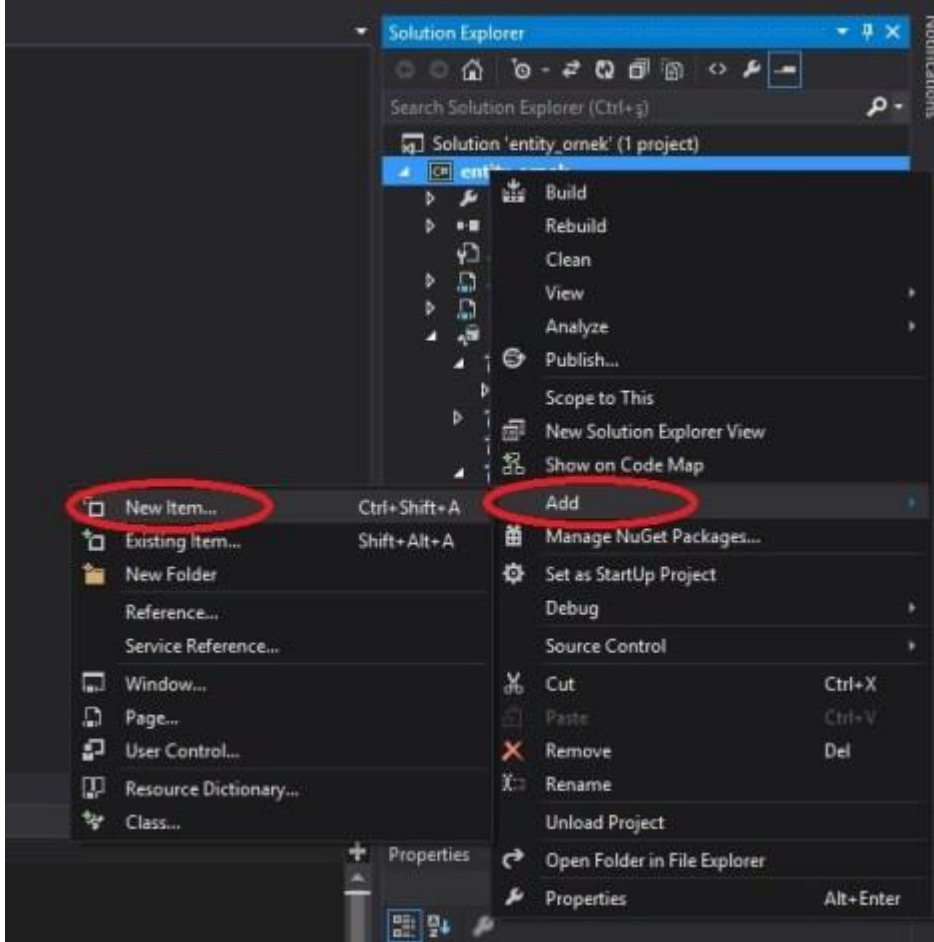
DEĞİŞTİR

SİL

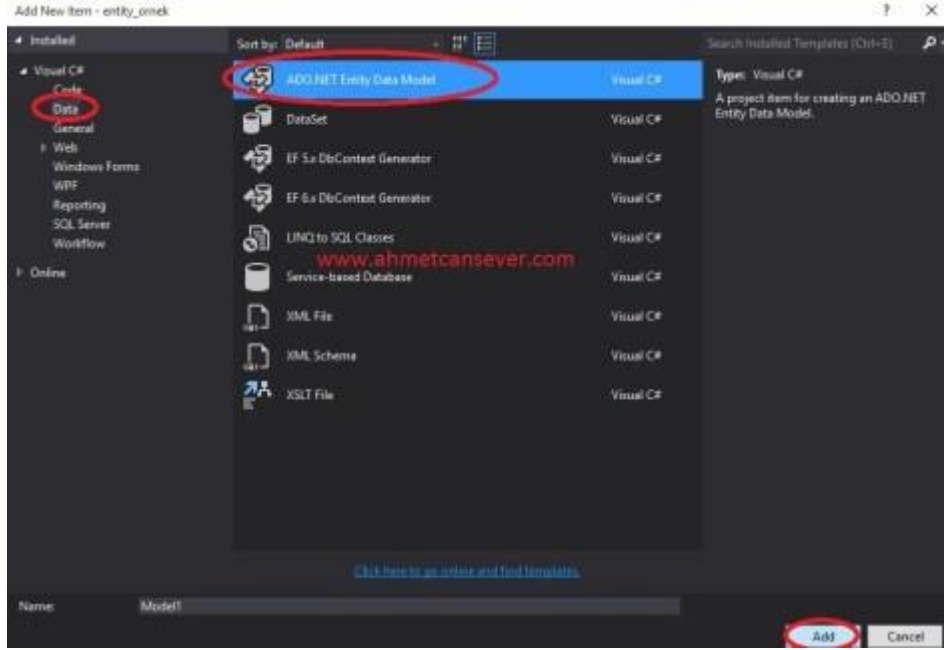
www.ahmetcansever.com

Form tasarımıını tamamladıktan sonra Entity Framework bağlantımızı aşağıdaki adımları takip ederek oluşturuyoruz.

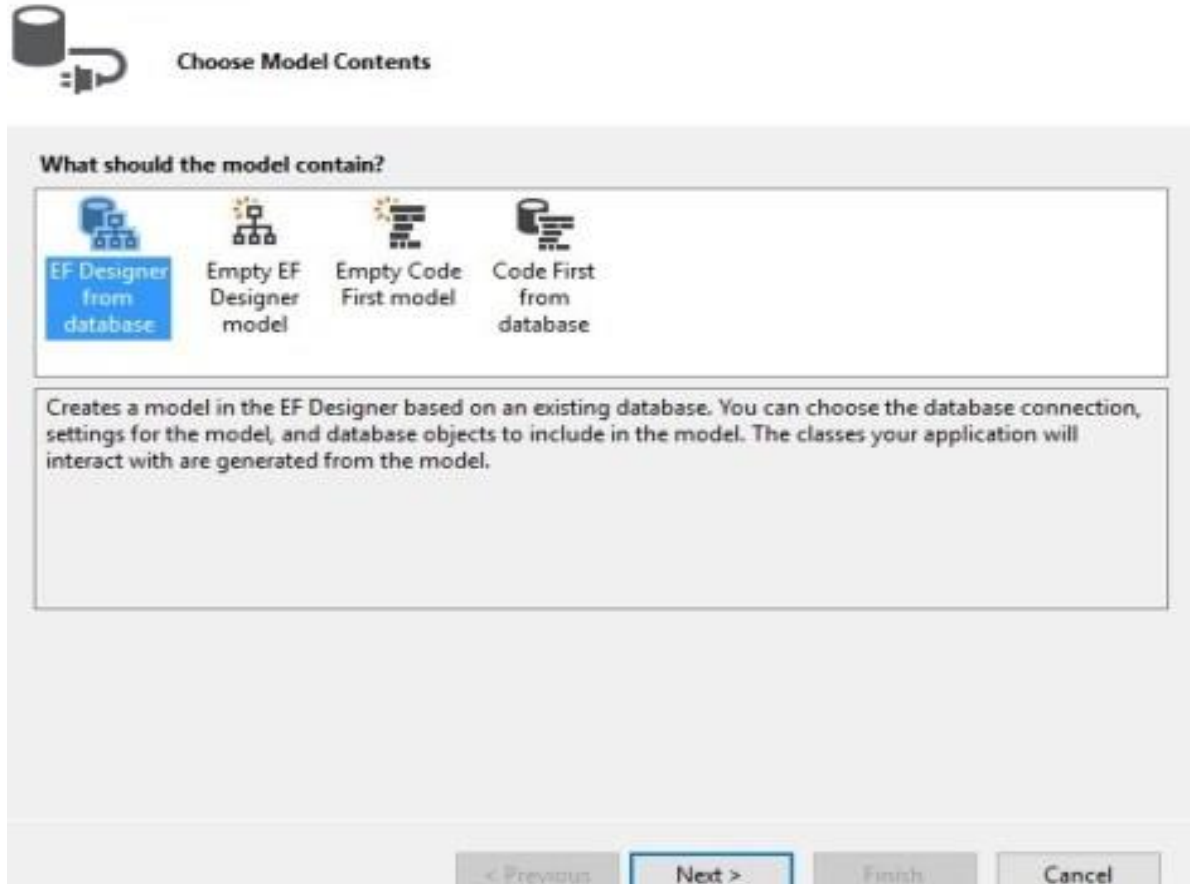
Öncelikle **Solution Explorer** Penceresinde projemizin üzerinde sağ tıklayarak **Add-NewItem** tıklıyoruz.



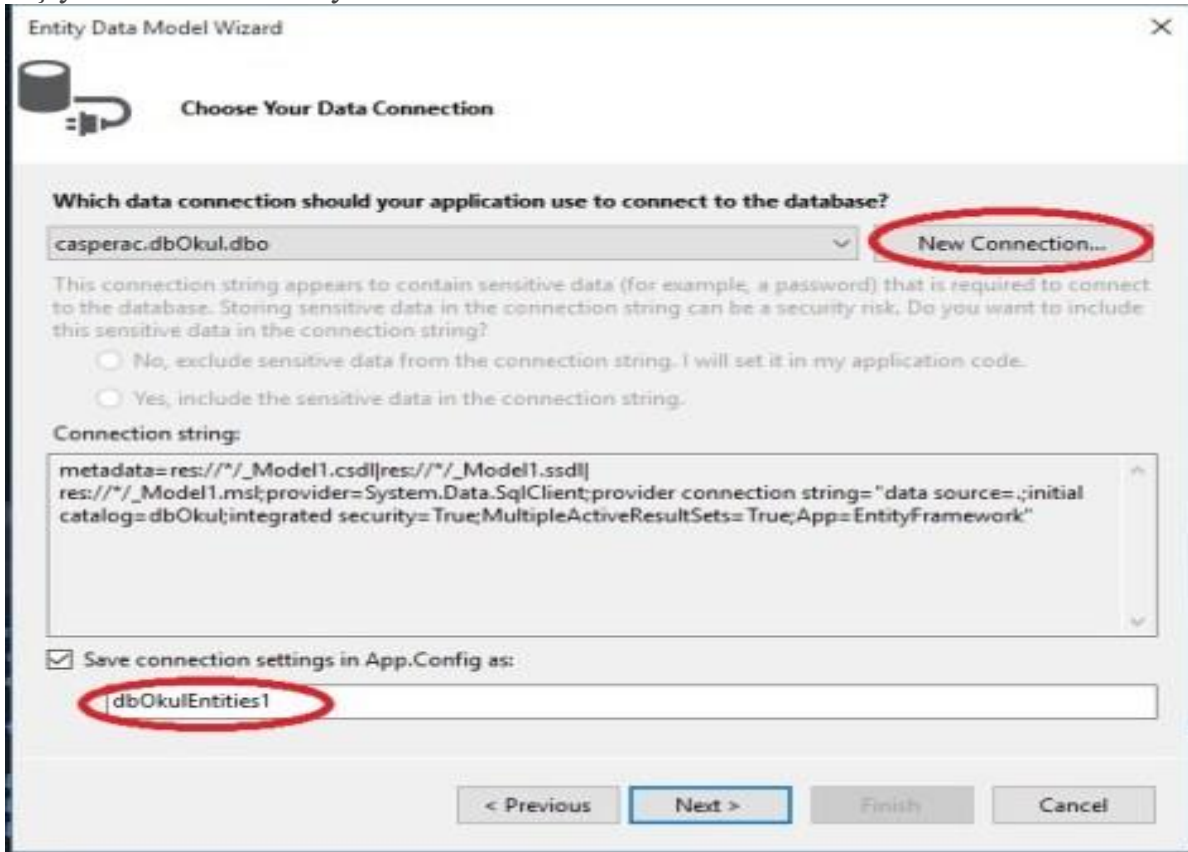
Açılan pencereden **ADO.NET Entity Data Model** seçip **Add** butonuna basıyoruz.



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

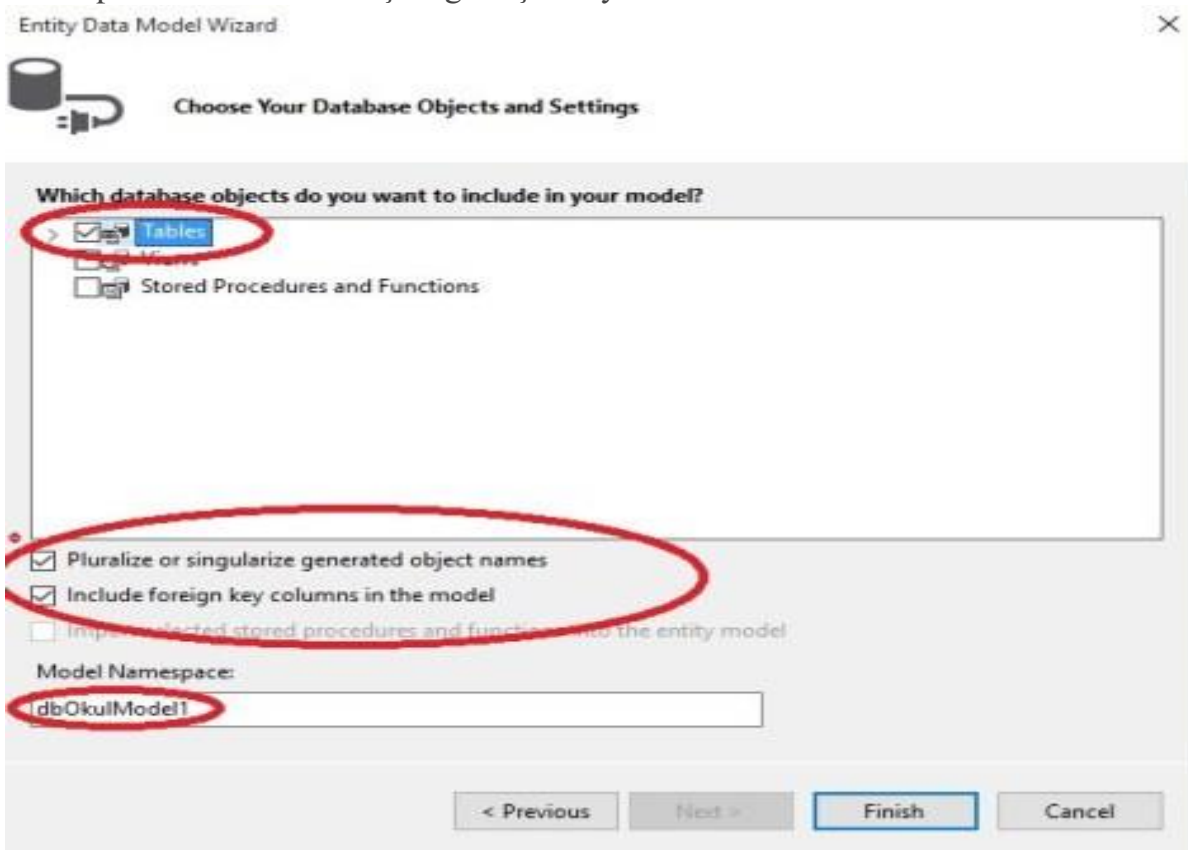


Buradan **New Connection** diyerek SQL Server' da kullanacağımız **database** i seçiyoruz ve devam ediyoruz.



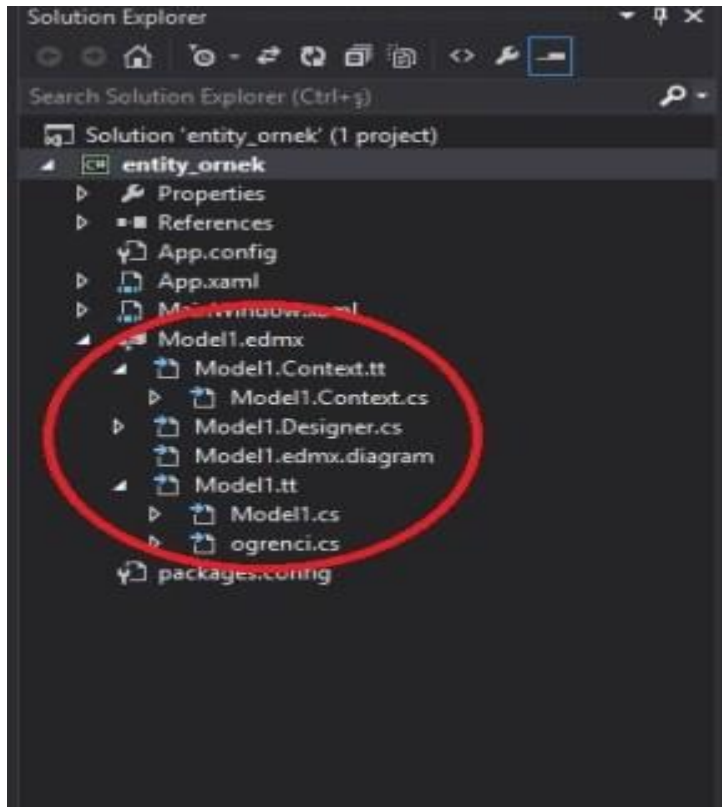
The screenshot shows the 'Entity Data Model Wizard' window, specifically the 'Choose Your Data Connection' step. The title bar reads 'Entity Data Model Wizard'. Below the title bar is a database icon and the text 'Choose Your Data Connection'. The main question is 'Which data connection should your application use to connect to the database?'. A dropdown menu shows 'casperac.dbOkul.dbo'. To the right of the dropdown is a button labeled 'New Connection...'. Below this, there 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 text box labeled 'Connection string:' containing the following text: '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"'. Below the text box is a checkbox labeled 'Save connection settings in App.Config as:' which is checked. Below the checkbox is a text box containing 'dbOkulEntities1'. At the bottom are four buttons: '< Previous', 'Next >', 'Finish', and 'Cancel'.

Gelen pencerede **Tables** seçeneğini işaretliyoruz.



The screenshot shows the 'Entity Data Model Wizard' window, specifically the 'Choose Your Database Objects and Settings' step. The title bar reads 'Entity Data Model Wizard'. Below the title bar is a database icon and the text 'Choose Your Database Objects and Settings'. The main question is 'Which database objects do you want to include in your model?'. There are three checkboxes: 'Tables' (checked), 'Views' (unchecked), and 'Stored Procedures and Functions' (unchecked). Below these is a section with three checkboxes: 'Pluralize or singularize generated object names' (checked), 'Include foreign key columns in the model' (checked), and 'Import selected stored procedures and functions into the entity model' (unchecked). Below this is a text box labeled 'Model Namespace:' containing 'dbOkulModel1'. At the bottom are four buttons: '< Previous', 'Next >', 'Finish', and 'Cancel'.

Bu işlemlerden sonra oluşturulan **Entity Data Model**’imizin **Solution Explorer**’e gelmesini göreceğiz.



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

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
```

```
namespace cwinentitty
{
    public partial class Form1 : Form
    {
        public Form1()
```

```

{
    InitializeComponent();
}
dbOkulEntities db;

void doldur() //Verileri Çekme (Select)
{
    db = new dbOkulEntities();
    dataGridView1.DataSource = db.ogrencis.ToList();
}
private void Form1_Load(object sender, EventArgs e)
{
    doldur();
}
private void button1_Click(object sender, EventArgs e) //Kayıt Ekle
{
    ogrenci yeniogrenci = new ogrenci();
    yeniogrenci.Numara = Convert.ToInt32(textBox1.Text);
    yeniogrenci.Ad = textBox2.Text;
    yeniogrenci.Soyad = textBox3.Text;
    db.ogrencis.Add(yeniogrenci);
    db.SaveChanges();
    doldur();
}
private void button2_Click(object sender, EventArgs e) //Güncelleme
{
    int guncelle = Convert.ToInt32(textBox1.Text);
    var guncellenecekogrenci = db.ogrencis.Where(w => w.Numara ==
guncelle).FirstOrDefault();
    guncellenecekogrenci.Ad = textBox2.Text;
    guncellenecekogrenci.Soyad = textBox3.Text;
    db.SaveChanges();
    doldur();
}
private void button3_Click(object sender, EventArgs e) //Silme
{
    int silinecek = Convert.ToInt32(textBox1.Text);
    var silinecekkisi = db.ogrencis.Where(w => w.Numara ==
silinecek).FirstOrDefault();
    db.ogrencis.Remove(silinecekkisi);
    db.SaveChanges();
    doldur();
}
}
}

```

---