

C# İLE DEVEXPRESS'DE SQL TABANLI

# TİCARİ OTOMASYON

---

MURAT YÜCEDAĞ

# İÇİNDEKİLER

<b>BÖLÜM 1: GİRİŞ</b>	<b>1</b>
Bizi Bekleyenler	2
Neler Öğrendik?	5
<b>BÖLÜM 2: VISUAL STUDIO KURULUMU</b>	<b>7</b>
Kurulum	8
Neler Öğrendik?	11
<b>BÖLÜM 3: SQL SERVER MANAGMENT STUDIO</b>	<b>13</b>
SQL Nedir?	14
Neler Öğrendik?	22
<b>BÖLÜM 4: DEVEXPRESS İNDİRME VE KURULUM</b>	<b>25</b>
DevExpress Nedir?	26
Neler Öğrendik?	29
<b>BÖLÜM 5: VERİTABANININ OLUŞTURULMASI</b>	<b>31</b>
Giriş	32
Veritabanının Oluşturulması	33
Kategoriler Tablosu	35
Departman Tablosu	39
Ürün Tablosu	40
Personel Tablosu	41
Cariler Tablosu	41
Firmalar Tablosu	42
Giderler Tablosu	43
Kasa Tablosu	44

İller ve İlçeler Tablosu	44
Satış Hareketleri Tablosu	46
Notlar Tablosu	48
Neler Öğrendik?	49
<b>BÖLÜM 6: İLİŞKİLERİN KURULMASI</b>	<b>51</b>
İlişki Nedir?	52
Neler Öğrendik?	56
<b>BÖLÜM 7: PROJENİN OLUŞTURULMASI</b>	<b>59</b>
Giriş	60
Neler Öğrendik?	66
<b>BÖLÜM 8: FORM ŞABLONU</b>	<b>69</b>
Mdi Form	70
Neler Öğrendik?	75
<b>BÖLÜM 9: ENTİTY FRAMEWORK</b>	<b>77</b>
Entity Framework Nedir?	78
Code First	79
Database First	79
Model First	79
Neler öğrendik?	80
<b>BÖLÜM 10: MODELİN OLUŞTURULMASI</b>	<b>83</b>
Giriş	84
Neler Öğrendik?	89

<b>BÖLÜM 11: KATEGORİ İŞLEMLERİ</b>	<b>91</b>
Giriş	92
Yeni Kategori Ekleme İşlemi	96
Xtra Messagebox Aracı	97
Mevcut Kategoride Silme İşlemi	99
Mevcut Kategoride Güncelleme İşlemi	100
Neler Öğrendik?	102
<b>BÖLÜM 12: ÜRÜN İŞLEMLERİ</b>	<b>105</b>
Giriş	106
LookupEdit Aracı	108
Kategorilerin LookupEdit Aracı Üzerinde Listelenmesi	109
Ürünlerin Listelenmesi	113
Yeni Ürün Ekleme İşlemi	115
Ürün Silme İşlemi	117
GridControl Aracından TextEdit Araçlarına Veri Aktarma	119
Ürün Güncelleme İşlemi	121
Neler Öğrendik?	124
<b>BÖLÜM 13: ÜRÜN İSTATİSTİKLERİ</b>	<b>127</b>
Giriş	128
Neler öğrendik?	141
<b>BÖLÜM 14: KATEGORİ İSTATİSTİKLERİ</b>	<b>143</b>
Neler Yapacağız?	144
Chart Control Kullanımı	147
GridControl Aracında Kategorilerin Listelenmesi	150
Panellere İstatistik Değerlerinin Çekilmesi	151
Chart Control Aracına Dinamik Olarak Veri Çekilmesi	152
Neler Öğrendik?	155

**BÖLÜM 15: ÜRÜN VE KATEGORİ EKLEME POPUP 157**

Yeni Kategori Ekleme Formu	158
Yeni Ürün Ekleme Formu	162
Neler öğrendik?	166

**BÖLÜM 16: CARİLER - FİRMALAR 169**

Cari Firma Menüleri	170
Cari Listesi Menüsü	170
İl - İlçe Scriptinin Yüklenmesi	174
Yeni Cari Kartı Tanımlama	186
Cari Hareketleri	195
Neler Öğrendik?	201

**BÖLÜM 17: PERSONEL VE DEPARTMAN İŞLEMLERİ 203**

Giriş	204
GridControl Üzerinden Veri Ekleme	209
GridControl Üzerinde Departman İşlemleri	212
GridControl Üzerinde Güncelleme İşlemi	215
GridControl Üzerinde Silme İşlemi	216
Departman Formu Tasarımı	218
GridControl Üzerinden Yeni Departman Ekleme İşlemi	220
Neler Öğrendik?	227

**BÖLÜM 18: ARAÇLAR 229**

Giriş	230
Kasa Modülü	230
Giderler Modülü	235
Döviz Kurları Modülü	238
Döviz Dönüşüm Modülü	240

Ajanda Modülü	242
Excel-Word Modülü	245
Gauge Modülü	247
YouTube Modülü	252
Yardım Modülü	253
Harita Modülü	254
Neler Öğrendik?	257

## **BÖLÜM 19: ANASAYFA FORMU** **259**

Giriş	260
Ürün - Stok Listesi	260
Son 5 Satış	263
Bugün Yapılacaklar	266
Personeller	267
Kurlar	268
Neler öğrendik?	269

## **BÖLÜM 20: QR CODE VE BARCODE** **271**

QR Kod Nedir?	272
QR Kod Oluşturma	272
QR Kod Çözme	275
Barkod Nedir?	276
Barkod Oluşturma	277
Neler Öğrendik?	278

## **BÖLÜM 21: PDF VE EXCEL RAPORLARI** **281**

Export to PDF	282
Export to Excel	286
Neler Öğrendik?	287

<b>BÖLÜM 22: SATIŞ İŞLEMLERİ</b>	<b>289</b>
Giriş	290
Trigger (Tetikleyici)	299
Neler Öğrendik?	306
<b>BÖLÜM 23: N KATMANLI MİMARİ YAPISI VE CODE FIRST İŞLEMLERİ</b>	<b>309</b>
Giriş	310
Yeni Bir Proje	310
Kurumsal Mimarinin Artıları	312
Kullanacağımız Katmanlar	312
Entity Layer	313
Data Access Layer	317
Migrations	322
<b>BÖLÜM 24: REPOSITORY DESIGN PATTERN</b>	<b>327</b>
Giriş	328
Interface	328
Neler Öğrendik?	335
<b>BÖLÜM 25: GENERIC REPOSITORY DESIGN PATTERN</b>	<b>337</b>
Amaç	338
Delete Metodu	340
Find Metodu	340
GetByID Metodu	341
Insert Metodu	341
List Metodu 1	341
List Metodu 2	342
Update Metodu	342
Neler Öğrendik?	343

**BÖLÜM 26: BUSINESS LAYER KATEGORİ CRUD 345**

Nedir?	346
UI Katmanı	348
Business Katmanı Yeni Veri Ekleme	351
Business Katmanı Veri Silme	353
Business Katmanı Veri Güncelleme	353
Neler öğrendik?	355

**BÖLÜM 27: PRODUCT SINIFI VE CRUD İŞLEMLERİ 357**

Giriş	358
Business Katmanı	359
Yeni Ürün Ekleme	362
Ürün Silme	365
Ürün Güncelleme	366
Ürün Getirme	367
Neler Öğrendik?	379

**BÖLÜM 28: RAPORLAR 381**

Giriş	382
DevExpress Rapor Sihirbazı	382
Neler Öğrendik?	392

**BÖLÜM 29: SESLİ KOMUTLAR 395**

Text to Speech	396
Speech to Text	398
Neler öğrendik?	399



<b>BÖLÜM 30: GİTHUB KULLANIMI</b>	<b>401</b>
GitHub Nedir?	402
Neler Öğrendik?	410
<b>BÖLÜM 31: VERİTABANLI GİRİŞ VE SQL YEDEĞİNİN ALINMASI</b>	<b>413</b>
Giriş	414
Login Formu	414
SQL Yedek Alma	419
Back-Up ile Yedek Alma	419
Log Dosyaları ile Yedek Alma	423
Neler Öğrendik?	425
<b>BÖLÜM 32: FINAL</b>	<b>427</b>
Proje Kodları	428
Form1	428
Ajanda Formu	435
Barkod Formu	436
Cari Form Listesi	437
Cari Hareketleri	438
Cari İstatistik	439
Cari Kartları	440
Cari Listesi	442
Cari Satış Listesi	446
Departman Formu	447
Döviz Dönüşüm Formu	449
Gauge Formu	450
Gider Formu	451
Harita Formu	452

İstatistik Formu	452
Kasa Listesi Formu	453
Kategori Ekleme Formu	454
Kategori İstatistik Formu	455
Kategoriler Formu	457
Kurlar Formu	459
Metni Seslendirme Formu	459
Personel Formu	460
QR Formu	461
Raporlar Formu	462
Speech To Text Formu	463
Ürün Ekleme Formu	464
Ürünler Formu	465
Ürün Rapor Formu	468
Yeni Not Formu	470
Yeni Satış Formu	471
YouTube Formu	473
Login Formu	474
Neler Öğrendik?	475

# 1

## GİRİŞ

### BU BÖLÜMDE

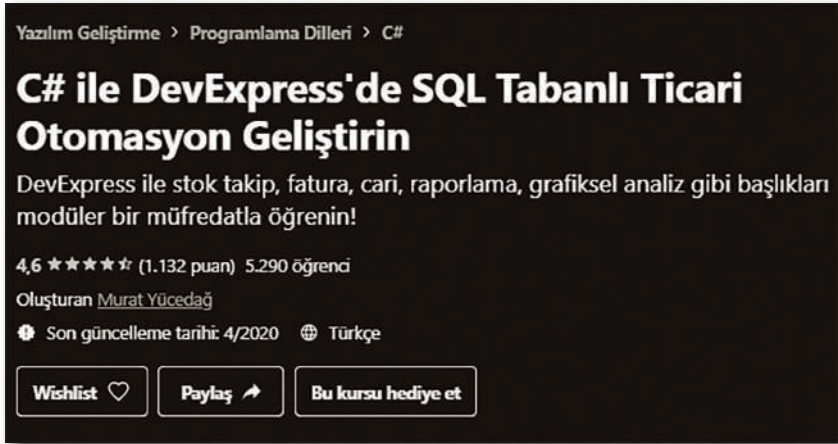
Bizi Bekleyenler	2
Neler Öğrendik?	5

Bu bölümde kitabımız boyunca yapacağımız uygulamanın detaylarını, bu uygulamada hangi geliştirme yaklaşımlarının kullanılacağını, uygulamamızın temelini oluşturan videolu eğitimlerin neler olduğunu öğreneceğiz.

## Bizi BEKLEYENLER

Bu benim 4. kitabım olacak ve ilk 3 kitabımda olduğu gibi her zaman en çok zorlandığım şey giriş cümleleri olmuştur. Giriş cümlelerinden sonra devamı çorap söküğü gibi geliyor. İlk kitabımı hazırlarken madem giriş kısmında zorlanıyorum o halde giriş kısmına zorlandığımı belirten cümleler yazarak giriş yapayım demiştim. Giriş biraz karmaşık oldu sanırım. :)

Uzun süredir aklımda olan bir kitaptı. Ben bu satırları yazarken takvimler 2021 yılının bahar aylarını gösteriyor. Türkiye'nin en çok satan ve eğitim formatında hazırladığım C# Eğitim Kitabı yaklaşık 475 sayfalık bir müfredattan oluşmuştu. Bu müfredatı bir proje ile daha kapsamlı başlıklar ile birleştirmek hedefindeydim. Kitabın çıkış noktası 2 tane Udemy 1 tane de YouTube serimiz oldu. Udemy'de yer alan C# ile DevExpress'de SQL Tabanlı Ticari Otomasyon kursumuz bu kitaba da ismini verdi. Her kitap benim için yeni bir bebek gibi.



Yazılım Geliştirme > Programlama Dilleri > C#

### C# ile DevExpress'de SQL Tabanlı Ticari Otomasyon Geliştirin

DevExpress ile stok takip, fatura, cari, raporlama, grafiksel analiz gibi başlıkları modüler bir müfredatla öğrenin!

4,6 ★★★★★ (1.132 puan) 5.290 öğrenci

Oluşturan [Murat Yücedağ](#)

🕒 Son güncelleme tarihi: 4/2020 🌐 Türkçe

Wishlist ❤️ Paylaş ↗️ Bu kursu hediye et

İlk olarak Ticari otomasyon kursumuzla başlayalım.

Kursta Adonet üzerinden C# programlama dili alt yapısıyla, SQL veritabanını kullanarak bir mağazanın ürün takip sistemi uygulamasını geliştirmiştik. Bu satırları yazdığım sırada kurs Türkiye'nin C# kategorisinde en fazla satın alınan 3. kursudur. (listenin ilk sırasında yine bana ait uygulama geliştirerek C# Öğrenin kursum yer alıyor)

# 3

## SQL SERVER MANAGMENT STUDIO

### BU BÖLÜMDE

SQL Nedir?	14
Neler Öğrendik?	22

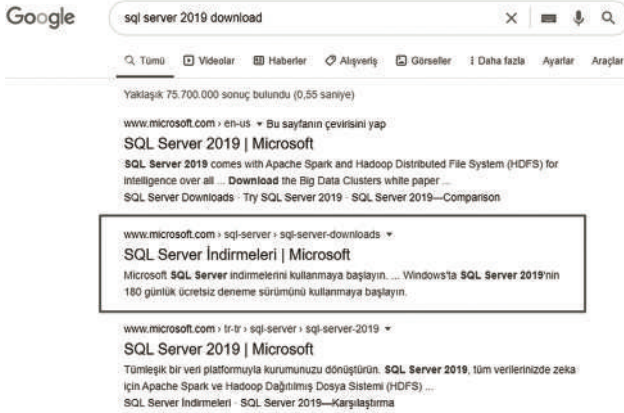
Bu bölümde, SQL kavramını, SQL dilini kullanan veritabanı yönetim sistemlerinin neler olduğunu ve SQL server Managment Studio programının indirme ve kurulum işleminin nasıl yapılacağını öğreneceğiz.

## SQL NEDİR?

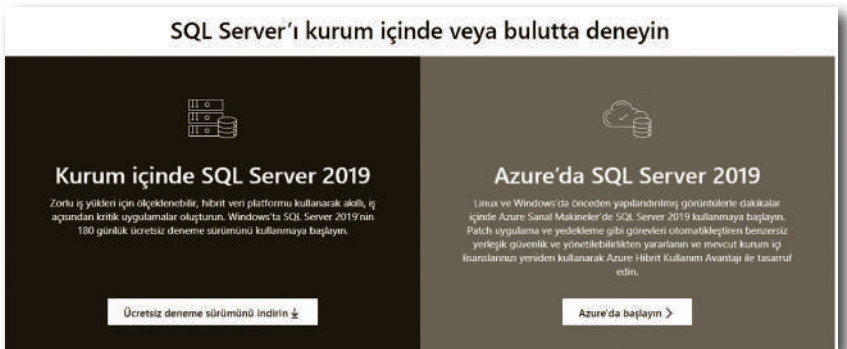
SQL ifadesi **Structured Query Language** yani Yapılandırılmış Sorgulama Dili kelimelerinin baş harfinden oluşan SQL, veritabanları ve tabloları kalıcı olarak saklayıp bunlar üzerinde işlem yapmakta kullanılan veritabanı yönetim sistemidir. Kitabımızda kullanacağımız Microsoft SQL Server ise Microsoft tarafından aktif bir şekilde geliştirilmesine devam edilen veritabanı yönetim sistemidir.

Tablolarımızı ve veritabanlarımızı barındıracak olan SQL Server Managment Studio programını indirip yavaş yavaş işin teknik boyutuna geçiş yapalım.

Google'da SQL Server 2019 download yazarak indirme işlemine başlayalım. Bu kitabın yazıldığı tarih olan Mart 2021'de SQL son sürüm olarak 19'u yayınlamıştı.



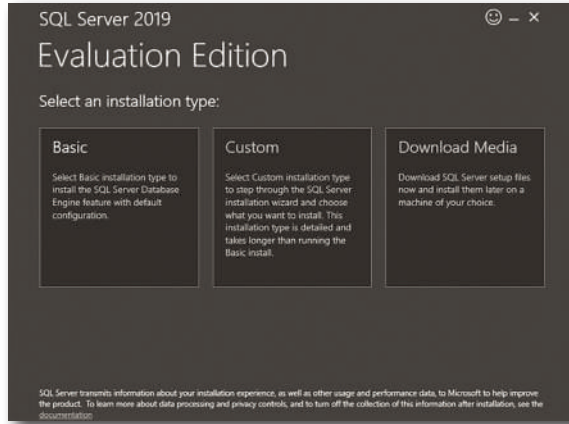
Açılan pencerede sol tarafta bulunan **Ücretsiz deneme sürümünü indirin** seçeneğine tıklayalım. Microsoft son kullanıcılar ve öğrencilerden herhangi bir ücret talep etmemektedir.



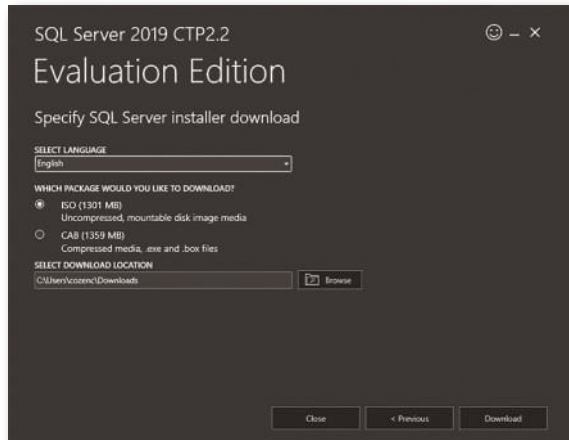
İndirmiş olduğumuz dosyayı çalıştıralım.



Çalıştırdığımız zaman karşımıza bu şekilde bir kurulum penceresi gelecektir. Kurulum penceremizden **Download Media** seçeneğine tıklalayalım.



Açılan pencerede SQL kurulumu için karşımıza indireceğimiz paket türünde 2 farklı paket çıkacaktır. Burada herhangi bir değişiklik yapmadan **ISO** formatında bırakarak **Download** seçeneğine tıklalayalım.



# 5

## VERİTABANININ OLUŐTURULMASI

### BU BÖLÜMDE

Giriő	32
Veritabanının Oluőturulması	33
Kategoriler Tablosu	35
Departman Tablosu	39
Ürün Tablosu	40
Personel Tablosu	41
Cariler Tablosu	41
Firmalar Tablosu	42
Giderler Tablosu	43
Kasa Tablosu	44
İller ve İlçeler Tablosu	44
Satıő Hareketleri Tablosu	46
Notlar Tablosu	48
Neler Öğrendik?	49

Bu bölümde, projemizin veritabanı ve tablolarının neler olduėunu, bu veritabanı ve tabloların nasıl hazırlanacaėını, tablo oluőturma iőlemleri sırasında en optimal veri türleri seçmenin neden önemli olduėunu, dıőarıdan alınan harici bir Script dosyasının SQL'e nasıl dahil edileceėini, tablo kopyalama iőlemini, tablo kaydetme iőlemini ve tablo düzenleme iőlemlerini öğreneceėiz.



## Giriş

Projemizin ne olduğundan, içeriğinde neler olacağından giriş bölümünde az çok bahsetmiştik. Bu bölümde projemizde yer alacak olan tablolarımızı oluşturacağız. Projemizde tablolara ek olarak prosedür, tetikleyici, view gibi yapılar da olacaktır. Ancak bunları şimdi değil ilerleyen bölümlerde veriyor olacağız. Bu bölümde ilk etap için yalnızca tablolara ve bu tablolar arasındaki bazı ilişkililere ihtiyacımız var. Projemize veritabanı adı olarak **DbTicariOtomasyon** ismini verebiliriz. Buradaki **Db** harfleri **Database** kelimesinin baş harflerini temsil etmektedir. Veritabanımızda yer alacak tablolar aşağıdaki gibidir;

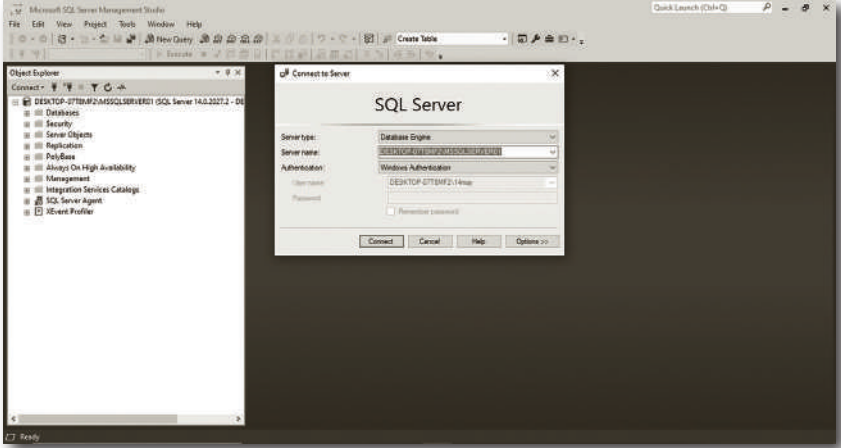
- » Kategoriler
- » Ürünler
- » Personeller
- » Departmanlar
- » Firmalar
- » Cariler (Müşteriler)
- » Giderler
- » Kasa
- » İller
- » İlçeler
- » Satış Hareketleri
- » Notlar

İlk etapta ihtiyacımız olan tablolar yukarıdaki gibidir. Bu tablolara ek olarak daha pek çok tablo kullanacağız fakat bunları ilerleyen bölümlerde ekliyoruz. Tablolarımızı oluşturmak için manuel yöntemi veya DDL sorguları üzerinden oluşturma yöntemini kullanabiliriz.

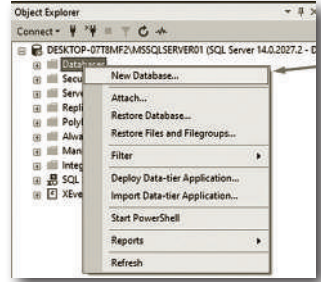
Kitapta olabildiğince farklı yapıları değinmek istediğim için her ikisini de sizlere anlatıyor olacağım. Tablolarımızı tek tek ele almadan önce veritabanımızı oluşturalım.

## VERİTABANININ OLUŞTURULMASI

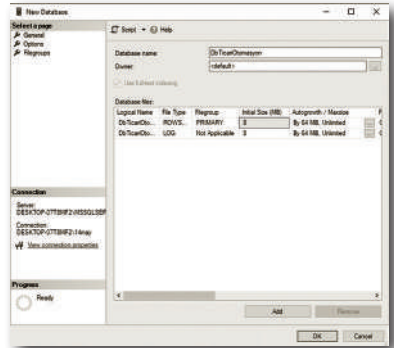
Tablolarda olduğu gibi veritabanı da DDL sorgular üzerinden oluşturulabilir. Bu konuya birazdan değineceğiz. İlk olarak SQL'i açalım. SQL Management Studio'yu çalıştırdığımız zaman karşımıza aşağıdaki gibi bir pencere ekranı gelecektir buradan Connect yani bağlan seçeneğine tıklayalım.



SQL'e bağlantımızı gerçekleştirdik. Yeni bir veritabanı oluşturmak için Databases klasörüne sağ tuşla tıklayalım ve New Database... seçeneğini seçelim.



Açılan pencerede veritabanı adı kısmına yandaki gibi DbTicariOtomasyon yazalım ve OK butonuna tıklayalım.



# 7

## PROJENİN OLUŐTURULMASI

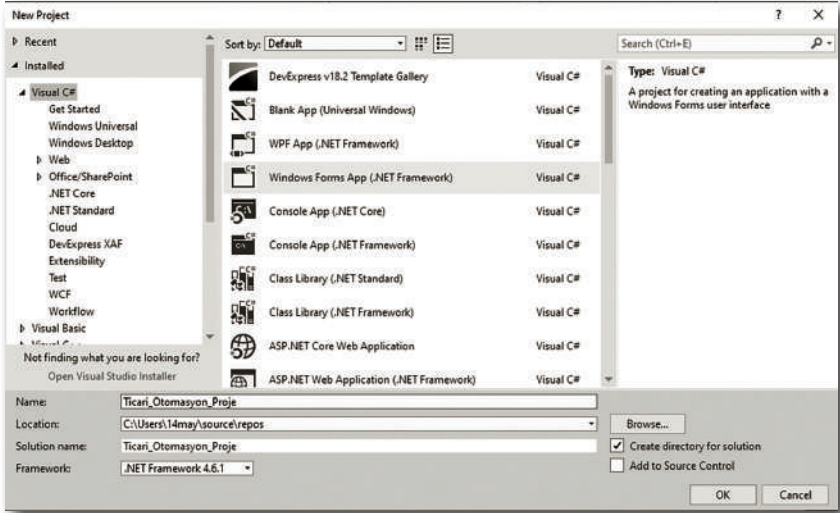
### BU BÖLÜMDE

Giriő	60
Neler Öđrendik?	66

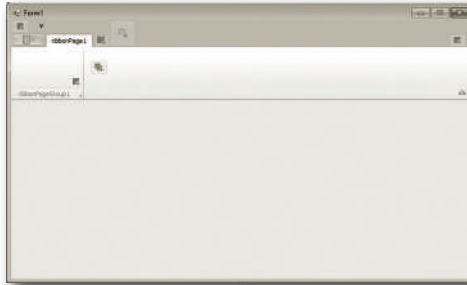
Bu bölümde, yeni bir C# form projesinin nasıl oluşturulacağını, ana form tasarımının nasıl yapılacağını, Ribbon Page kullanımını, Simple Button kullanımını, butonlara resim ikonları tanımlamayı, kategori, ürün, cari, departman, personel, araçlar, satış hareketleri ve diğer menüleri ve bu menülerin alt menülerinde neler olacağını uygulamalı bir şekilde öğreneceğiz.

## Giriş

Visual Studio'yu açalım ve yeni bir C# Form projesi oluşturalım.



Projemizin adını **Ticari\_Otomasyon\_Proje** olarak belirledik. Bundan sonraki kısımda artık hayal gücümüzün büyüklüğü ciddi bir artı sağlayacaktır. Karşımıza gelen form penceresine **DevExpress** araçlarından **Ribbon**'u ekleyelim. Ribbon Control aracı ile üst menü oluşturup formlar arası geçişleri bu menü üzerinden sağlıyor olacağız. Ribbon Control aracını formumuza eklediğimiz zaman karşımıza aşağıda gibi bir arayüz ekranı gelecektir.



Ribbon aracını ekledikten sonra üst kısmın tasarımını yapmaya başlayalım. **Ribbon Page1** yazan kısmın hemen yanında yer alan + (artı) seçeneğine tıklayıp yeni bir Ribbon sayfası ekleyelim. (bir sonraki görseli inceleyebilirsiniz)

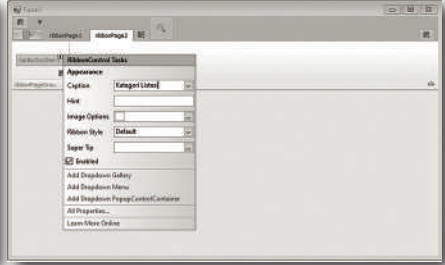
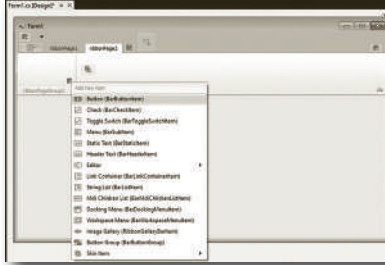


2. sayfamızı ekledikten sonra artık tasarıma tam anlamıyla başlayabiliriz.

İlk sayfayı ana sayfa olarak kullanacağımız ve ana sayfayı projenin sonunda yapacağımız için şimdilik bu kısmı daha sonra dönmek üzere atlıyoruz.

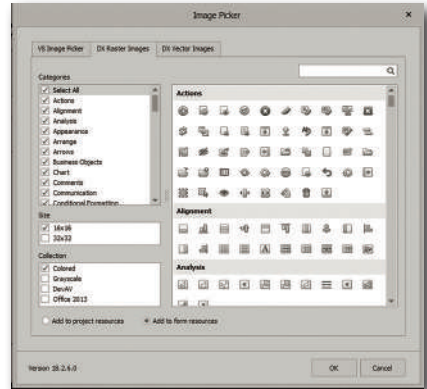
Aşağıdaki görselde (sağdaki) yer alan **RibbonPageGroup2** yazan kısma tıkladığımız karşımıza bu Ribbon üzerine ekleyebileceğimiz seçenekler gelmektedir.

Buradan bir tane **Button** ekleyelim. Butonumuzu ekledikten sonra butonumuzun üstünde yer alan ikona tıklayıp ismini aşağıdaki gibi kategori listesi olarak değiştirelim.



İsim değiştirme işlemini yaptıktan sonra sırada butonumuza görsel ekleme kısmı var.

Bunun için butonumuzun tekrar aynı kısmına gelip **Image Options** seçeneğine tıklatalım. Karşımıza yandaki gibi bir pencere gelecektir. DevExpress sürümleri değiştikçe bu pencereler arasında küçük değişiklikler olabilir ancak yapı aynıdır.



# 10

## MODELİN OLUŐTURULMASI

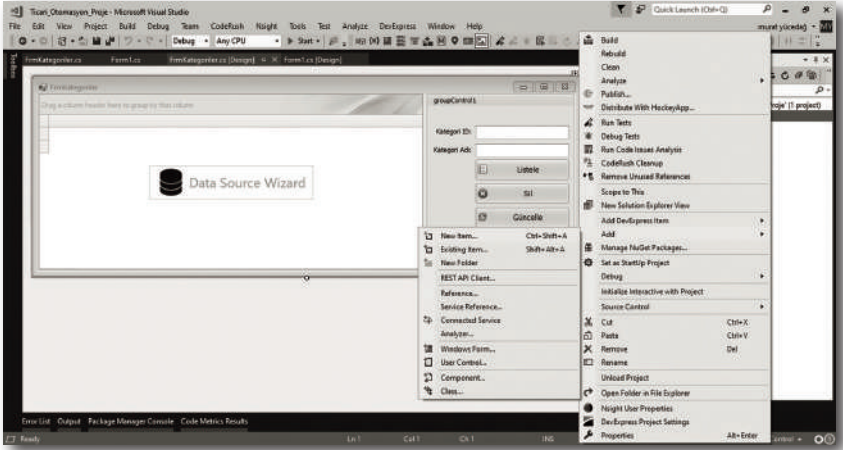
### BU BÖLÜMDE

Giriő	84
Neler Öđrendik?	89

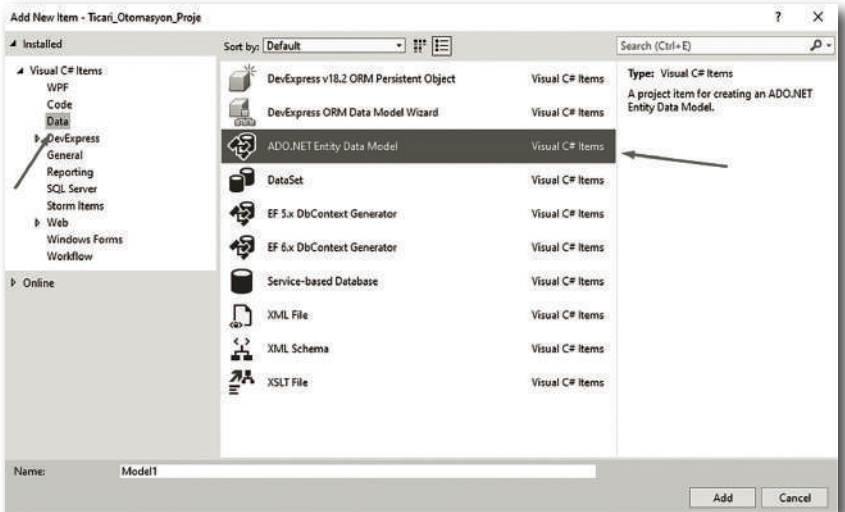
Bu bölümde, Entity framework DbFirst yaklaşımının nasıl kullanılacağını, projemize model ekleme işlemini, modele genel bakış yapıp modelin detaylarını, sınıfların nasıl tabloya ve propertylerin nasıl sütunlara dönüőtüđünü, SQL'de tanımladıđımız veri türlerinin C#da hangi deđişkenlere dönüőtüđünü öğreneceđiz.

## GİRİŞ

Geçtiğimiz bölümde entity framework kavramını detaylıca öğrenmiştik. Bu bölümde SQL üzerinde oluşturduğumuz modelimizi Visual Studio üzerine yansıtacağız. Bunun için aşağıdaki adımları takip edelim. İlk olarak projemizin üzerine sağ tuşla tıklayalım Add yani ekle kısmından New Item... yani yeni öğe seçeneğine gelelim.

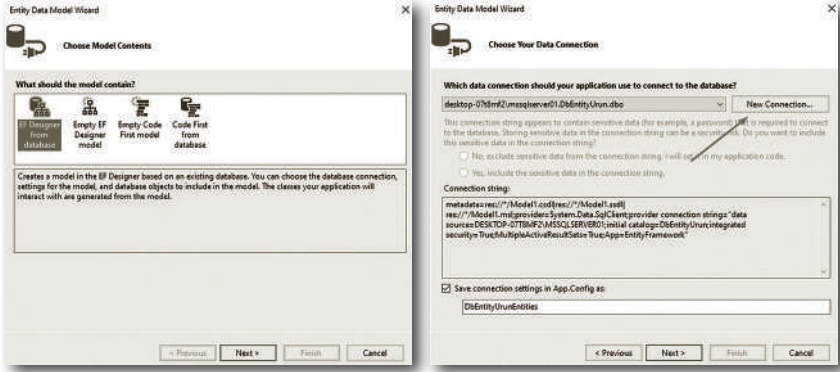


Açılan pencerede sol taraftan Data seçeneğini seçip orta kısımda yer alan ADO.NET Entity Data Model seçeneğini seçelim ve Add butonuna tıklayalım.



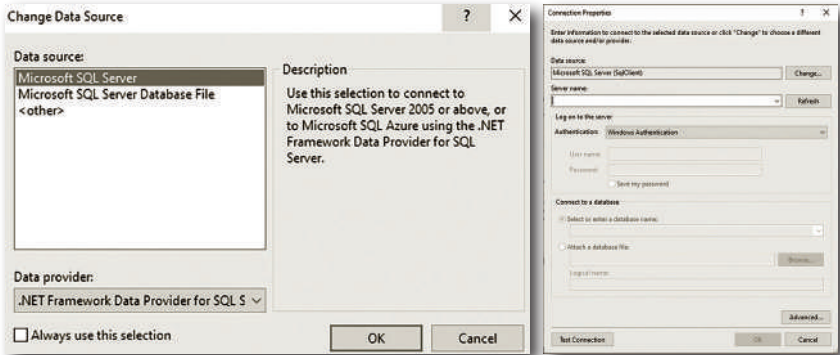
Visual Studio bize hangi entity framework yaklaşımı ile geliştirme yapacağımızı soruyor. Geliştirme ortamı olarak ilk sırada yer alan DbFirst yaklaşımını temsil eden alanı seçip Next butonuna tıklayalım.

Akabinde karşımıza yeni bağlantı ekranı geldi. Bu ekranda sağ tarafta görülen New Connection... yani yeni bağlantı butonuna tıklayalım.



Yeni bağlantı ekranına tıkladığımız zaman karşımıza bağlantının gerçekleşeceği veri kaynağı kısmı çıkacaktır. Buradan ilk sırada yer alan Microsoft SQL Server seçeneğini seçelim.

Microsoft SQL Server seçeneğini seçtikten sonra bu kez bağlantının yer alacağı veritabanının seçilmesi gerekecektir.





# 16

## CARİLER - FİRMALAR

### BU BÖLÜMDE

Cari Firma Menüleri	170
Neler Öğrendik?	201

Kitabımızın en kapsamlı bölümlerinden biri olan Cariler-Firmalar bölümünde cariler ve firmalarla ilgili listeleme, ekleme, silme ve güncelleme işlemlerinin nasıl yapılacağını, il ve ilçe script dosyasının SQL'e nasıl dahil edileceğini, sonradan SQL'e eklenen bir tablonun Visual Studio'da yer alan modelimize nasıl ekleneceğini, seçilen ile göre ilçelerin nasıl getirileceğini, cari hareketlerini, cari hareketi listesini, cari hareketi listesi için ayrı bir form oluşturma işlemini öğreneceğiz.

## CARİ FİRMA MENÜLERİ

Cari – firma bölümümüzde 8 tane menü bulunmaktadır. Bu menüler;

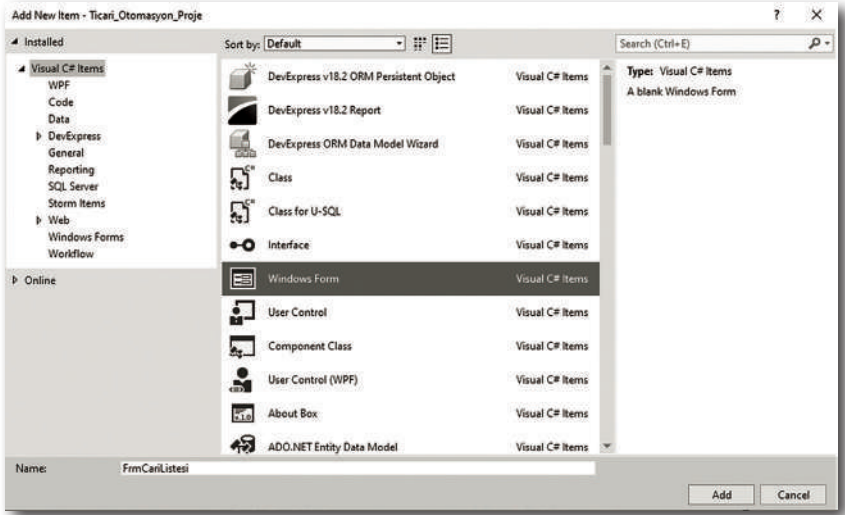
- » Cari Listesi
- » Yeni Cari
- » Firma Listesi
- » Yeni Firma
- » Cari İstatistik
- » Firma İstatistik
- » Cari Hareketleri
- » Firma Hareketleri

Her bir menü ögesini kendi başına ayrı bir başlık olarak ele alınacaktır.

Bu menüler arasında ilk olarak Cari Listesi ile başlayalım. Carileri listelerken aynı zamanda crud işlemlerini gerçekleştireceğiz.

### CARİ LİSTESİ MENÜSÜ

Cari listesinde sadece listeleme değil aynı zamanda crud işlemleri yapacağımız az önce belirtmiştik. Projemize yeni bir form ekleyip bu forma `frmcarilistesi` ismini vererek ilk adımı atalım.



Formumuzu oluşturduktan sonra form boyutunu diğer formlarımıza aynı size değerine getirelim.

The screenshot shows a Windows-style window titled "FrmCariListesi". It contains several input fields arranged in two columns. The left column has fields for "Cari ID:", "Ad:", "Soyadı:", "Telefon:", and "İl:". The right column has fields for "Mail:", "TC:", "Adres:", "Vergi Dairesi:", and "İlçe:". To the right of these fields is a vertical stack of buttons: "Listele", "Ekle", "Sil", "Güncelle", "Yeni Sayfada Cari Listesi", "Ara", "Carileri Yazdır", and "İptal". Below the input fields is a "Data Source Wizard" dialog box with a database icon and the text "Data Source Wizard".

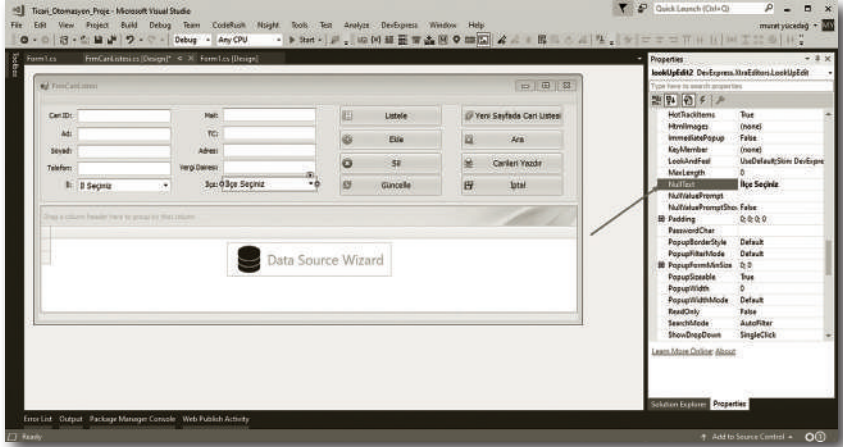
Cari listesi sayfasını yukarıdaki gibi tasarladık. Şimdi bu tasarımın detaylarını ele alalım.

Cariler tablomuzda 10 tane sütun bulunmaktadır. Bu sütunlara atanacak olan değerleri belirlediğimiz Label ve TextEdit araçlarını 5'erli 2 grup olarak ayırdık. Bu grupların sağ tarafında temel Ekleme, Silme, Güncelleme ve Listeleme işlemlerini tutacak olan butonlarımızı ekledik.

Akabinde bu 4 butonun sağ tarafına 4 buton daha ekledik. Eklediğimiz butonların bir tanesi carilerin listesini yeni formda göstermek için kullanacağız. Formumuzun size değeri cari listesi için eklediğimiz **GridControl** aracına göre küçük kaldığından dolayı burada yer alan carileri ayrı bir form penceresinde göstermeye ihtiyaç duyabiliriz. Bundan dolayı böyle bir araç kullanmayı uygun bulduk. **Ara** butonu carilerde ada, soyada veya belirlenen diğer kriterlere göre listeleme işlemi gerçekleştirecek. Yazdırma işlemlerinde carilerimiz için **print** yapısını kullanacağız. **İptal** butonunda ise herhangi bir işlem yapmadan sayfamızı kapatacağız. Gelelim iller ve ilçelere. İl ve ilçelere için eklediğimiz TextEdit araçlarını **LookUpEdit** araçlarına dönüştürelim.

This screenshot is identical to the previous one, but the "İl:" and "İlçe:" dropdown menus are now expanded to show the text "[EditValue is null]". The rest of the form, including the input fields and buttons, remains the same.

LookUpEdit araçlarımızın Properties kısmında yer alan NullText özelliğini İl Seçiniz ve İlçe Seçiniz olarak düzenleyelim.



Araçlarımızla ilgili tasarımsal düzenlemeleri tamamladığımızı göre kodlarımızı yazmaya başlayalım. İlk olarak listeleme işlemi ile başlayalım.

### Örnek Kod

```
DbTicariOtomasyonEntities db = new DbTicariOtomasyonEntities();
private void BtnListele_Click(object sender, EventArgs e)
{
    gridControl1.DataSource = (from x in db.TBLCARI
                               select new
                               {
                                   x.CARIID,
                                   x.AD,
                                   x.SOYAD,
                                   x.IL,
                                   x.ILCE,
                                   x.ADRES,
                                   x.MAIL,
                                   x.TC,
                                   x.TELEFON,
                                   x.VERGIDAIRESI
                               }).ToList();
}
```

## PERSONEL VE DEPARTMAN İŞLEMLERİ

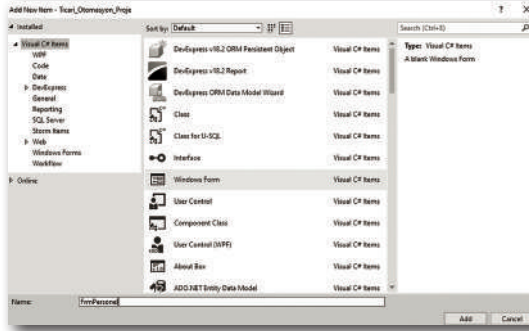
### BU BÖLÜMDE

Giriş	204
GridControl Üzerinden Veri Ekleme	209
GridControl Üzerinde Departman İşlemleri	212
GridControl Üzerinde Güncelleme İşlemi	215
GridControl Üzerinde Silme İşlemi	216
Departman Formu Tasarımı	218
GridControl Üzerinden Yeni Departman Ekleme İşlemi	220
Neler Öğrendik?	227

Kitabımızın bu bölümünde, departman tablosuna ait Ekleme, Silme, Güncelleme işlemlerini GridControl aracı üzerinden nasıl yapacağımızı, GridControl içinde bulunan GridView aracının nasıl kullanılacağını, ChartControl aracına dinamik olarak SQL üzerinden nasıl veri çekileceğini, Column Chart türünü ve Pie Chart türünü grafik aracımız üzerinde nasıl kullanacağımızı öğreneceğiz.

## Giriş

Projemizin bu bölümünde personel ve departmanlara ait işlemleri GridControl aracı üzerinden gerçekleştireceğiz. İlk olarak projemize yeni bir form ekleyerek başlayalım.

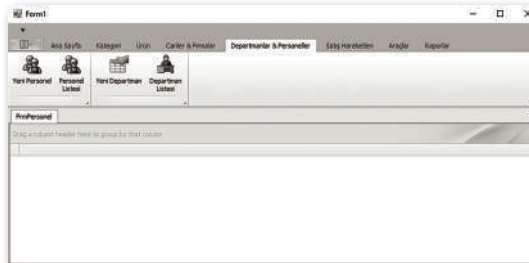


Formumuzu ekledikten sonra boyutunu diğer formlarımızla aynı ebata getirelim. İşlemlerimizi bu kez GridControl aracı üzerinden gerçekleştireceğiz.

Formumuza bir tane GridControl Aracı ekleyelim ve bunu **Dock in Parent Container** formatında ayarlayalım. Formumuza araç kutusundan bir tane **Binding Source** aracı ekleyelim. Bu aracımız üzerinden veritabanı işlemlerini gerçekleştireceğiz. Formumuzun Load kısmına aşağıdaki kodları yazalım.

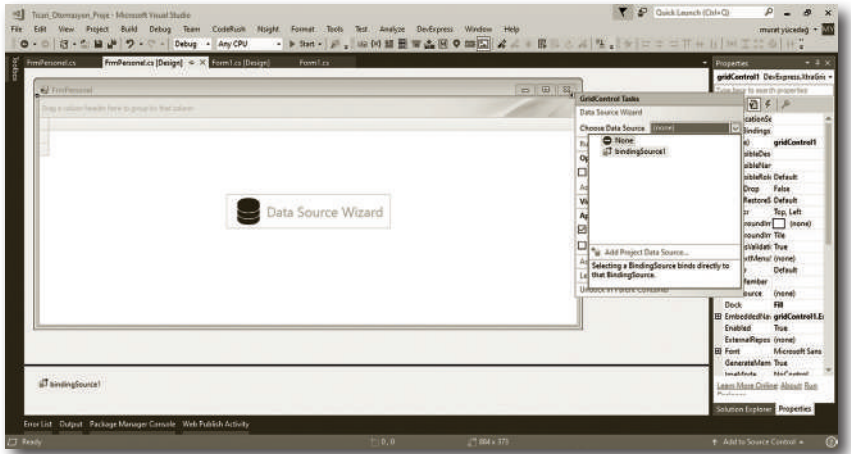
### Örnek Kod

```
DbTicariOtomasyonEntities db = new DbTicariOtomasyonEntities();
private void FrmPersonel_Load(object sender, EventArgs e)
{
    db.TBLPERSONEL.Load();
    bindingSource1.DataSource = db.TBLPERSONEL.Local;
}
```

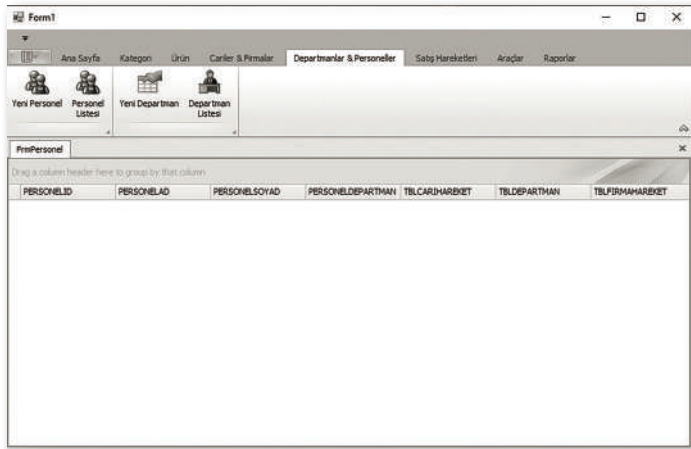


Kodlarımızı yazıp formumuzu çalıştırdığımız zaman karşımıza yukarıdaki gibi boş bir Grid ekranı gelecektir. Load metoduyla amacımız personel tablosunda yer alan verileri hafızaya alıp akabinde **Binding** aracına veri kaynağı olarak hafızaya alınan değerleri gönderebilmektir.

Load metodunu kullanabilmeniz için `using.system.data.entity` komutunu `using` kısmına eklemeyi unutmayın. Projemizi durdurup formumuza geri dönelim. GridControl aracımızın üzerinde yer alan **Play** şeklindeki sembole tıklayıp açılan GridControl görevleri penceresinde **Data Source** kısmına `bindingSource1` atamasını yapalım.



Projemizi yeniden çalıştıralım.



# 23

## N KATMANLI MİMARİ YAPISI VE CODE FIRST İŞLEMLERİ

### BU BÖLÜMDE

Giriş	310
Yeni Bir Proje	310
Kurumsal Mimarinin Artıları	312
Kullanacağımız Katmanlar	312
Migrations	322

N katmanlı mimari yapısının ne olduğunu, katmanlı mimarini artılarını, Entity Layer, Business Layer ve Data Access Layer yapısının ne olduğunu, Entity Framework Code First yapısını, Code First yapısı ile yeni bir tablo oluşturmayı, Context sınıfını ve migrations işlemlerini öğreneceğiz.



## Giriş

Projemizde yavaş yavaş sona gelirken kitaba eklemek istediğim yeni bir konunun perdelerini aralıyoruz.

Bu konu yazılım geliştirme sürecinde inanılmaz önemli bir yere sahip olan ve benim de burada atlamak istemediğim, sizlere sunmak istediğim konu olan **N Katmanlı Mimari** prensibidir.

N katmanlı mimari prensibinde amacımız projeyi **Repository Pattern** adı verilen mimari üzerinden geliştirmektir. Bu mimariyi ilerleyen sayfalarda detaylı olarak ele alacağız. Mimariyi anlatmadan önce yapacağımız işlemlerden bahsetmek istiyorum. Projemize kısa bir ara verip yeni bir proje oluşturacağız. Bu projemizin veritabanı kısmında sadece 2 tane tablo olacak. Bu tablolar birbiriyle ilişkili olan ürün ve kategori tabloları olacaktır. Burada amacım sizlere Entity Framework'un Code First yaklaşımı ile proje geliştirmenize olanak sağlayıp, kurumsal mimari yaklaşımlarını yeni projede uygulayabilmenizdir. Sizlere 2. projeyi anlattıktan sonra ilk projemizin eksik kısımlarına geri dönüp kalan son işlemleri yapacağız.

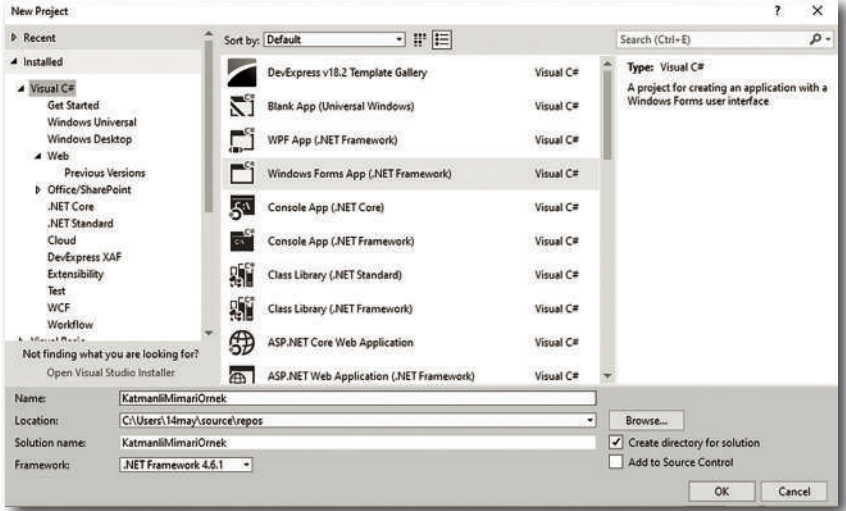
Burada şöyle bir soru akıllara gelebilir *"Murat hocam neden projeyi en baştan itibaren katmanlı mimaride yapmadık?"* başlangıçta amacım size kurumsal mimari öğretmek değildi. Entity Framework DbFirst yaklaşımını kullanıp SQL alt yapısında C# dilinde DevExpress bileşenleri ile ara yüzü oldukça güçlü bir proje çıkarmaktı. Sayfalar ilerledikçe, proje büyüdükçe kitabımızda kurumsal mimariye ve Code First yaklaşımına da yer verme ihtiyacı duydum. Size Code First yaklaşımını ve kurumsal mimari yapısını anlattıktan sonra eminim ki projeyi yeniden yapmaya başladığınız zaman kurumsal mimari prensibini kolaylıkla uygulayabileceksiniz. O halde yavaş yavaş başlayalım.

## Yeni Bir Proje

Mevcut projemizde epey yol aldığımız, Crud işlemlerini tamamladığımız, yaklaşım olarak DbFirst kullandığımız için orada yeniden mimari uygulamak pek sağlıklı olmayacaktır. Bunun yerine sıfırdan yeni bir form projesi oluşturup bu form projesi üzerinden ilerleyebiliriz.

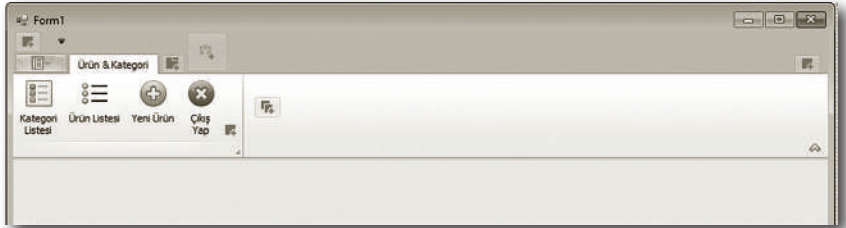
Projemizde 2 tane tablo yer alacak. Bunlar;

1. Ürünler
2. Kategoriler şeklinde olacaktır.



Katmanlı mimari örnek ismiyle yeni bir proje oluşturarak başlayalım. Projemizde tasarımı yine DevExpress araçları üzerinden yapacağız. Ancak biraz önce de belirttiğim gibi burada amaç yeni bir proje yapmak değil kurumsal mimari yapısını anlamak olacağı için tasarımda çok uzun uzadıya kalmayacağız.

Projemizin karşımıza gelen ana formuna bir **Ribbon** aracı ekleyip aşağıdaki tasarımı yapalım.



Form boyutu 900;580 olan bir form tasarımı yaptık. Formumuza bir tane Ribbon aracı ekleyip Ribbon aracı üzerine gerekli olan butonları ekledik. Formumuza bir de mdi ekleyip tasarım kısmına kısa bir mola verelim. Mdi aracını ekledikten sonra projemizin **Solution Explorer** penceresini açalım.

# 25

## GENERIC REPOSITORY DESIGN PATTERN

### BU BÖLÜMDE

Amaç	338
Delete Metodu	340
Find Metodu	340
GetByID Metodu	341
Insert Metodu	341
List Metodu 1	341
List Metodu 2	342
Update Metodu	342
Neler Öğrendik?	343

Kitabımızın bu bölümünde, N katmanlı mimari prensibine devam ederken bu kez Generic Repository yapısını, Repository sınıfı içinde tanımladığımız metotlara görev atama işlemlerini gerçekleştireceğiz.

## AMAÇ

Interfaceleri her bir Entity sınıfımız için ayrı ayrı tanımladıktan sonra tüm yapıyı tek bir sınıftan ve bu sınıf içinde yer alan IRepository'den miras aldığımız metotlarla beraber tanımlayacağımız Repository sınıfı için gerekli olan düzenlemeleri yapacağız. Geçen bölümde Repository sınıfımız içine Interfaceleri implemente ettikten sonra sınıfımızın kod bloğunun son hali aşağıdaki gibi olmuştu.

### Örnek Kod

```
using DataAccessLayer.Abstract;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Linq.Expressions;
using System.Text;
using System.Threading.Tasks;

namespace DataAccessLayer.Concrete
{
    public class Repository<T> : IRepository<T> where T : class
    {
        public int Delete(T p)
        {
            throw new NotImplementedException();
        }

        public T Find(Expression<Func<T, bool>> where)
        {
            throw new NotImplementedException();
        }

        public T GetByID(int id)
        {
            throw new NotImplementedException();
        }

        public int Insert(T p)
        {
            throw new NotImplementedException();
        }
    }
}
```

```

public List<T> List()
{
    throw new NotImplementedException();
}

public List<T> List(Expression<Func<T, bool>> filter)
{
    throw new NotImplementedException();
}

public int Update(T p)
{
    throw new NotImplementedException();
}
}
}

```

Sınıfımızın içinde Context sınıfından bir nesne türeterek yeni kodlarımızı yazmaya başlayalım. Türettiğimiz bu nesne aracılığıyla çağırdığımız tabloya / sınıfa ait işlemleri gerçekleştireceğiz. Sınıfımızın içine aşağıdaki kodu yazalım.

```

Context c = new Context();
    DbSet<T> _object;

public Repository()
{
    _object = c.Set<T>();
}

```

Context sınıfımızdan c isminde bir nesne türettik. C nesnesi aracılığıyla Context sınıfımızda yer alan SQL üzerinde oluşturduğumuz tüm sınıflara erişim sağlayabileceğiz. Hemen altında DbSet türünde T Entity değeri alan \_object isminde bir field tanımladık.

Bu field aracılığıyla crud işlemleri uygulanacak değerlere atamalar yapılacaktır. Yani bir nevi parametre görevi görecektir.

Public Repository ifadesiyle ilgili sınıfımıza ait yapıcı (constructor) bir metod tanımladık. Bu metodumuzda bir üst satırda oluşturduğumuz \_object isimli field değerimize Entity'den gelecek değerimizi atadık. Sınıfımıza Interface imp-

# 27

## PRODUCT SINIFI VE CRUD İŞLEMLERİ

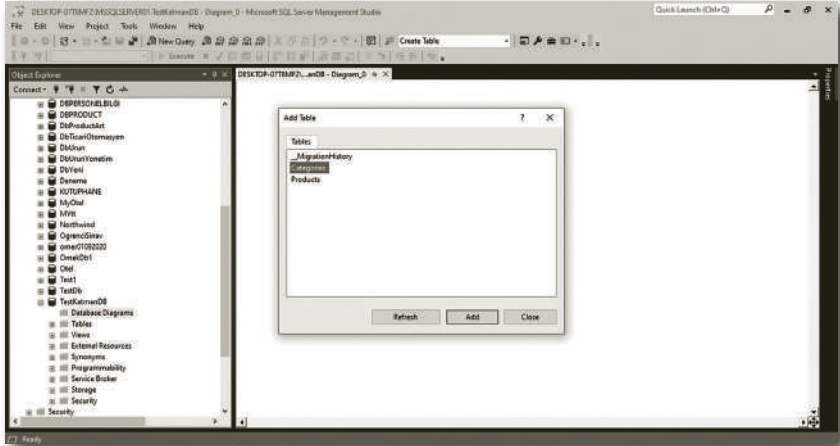
### BU BÖLÜMDE

Giriş	358
Business Katmanı	359
Yeni Ürün Ekleme	362
Ürün Silme	365
Ürün Güncelleme	366
Ürün Getirme	367
Neler Öğrendik?	379

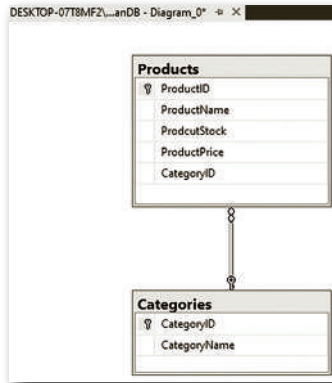
N Katmanlı mimari yapımızın bu kısmında ürün sınıfı üzerinde önce Business katmanına ait metotları akabinde UI katmanında bu metotlara bağlı olarak ekleme, silme, güncelleme, arama ve listeleme işlemlerinin nasıl yapılacağını öğreneceğiz.

## Giriş

Kategori sınıfımızla ilgili olan gerekli Crud komutlarını tüm katmanlarımız için gerçekleştirip ilgili işlemleri tamamlamıştık. Sırada aynı yapıyı ürün sınıfı üzerinden yapma işlemi var. Bunun için ilk önce SQL veritabanımızı açıp ilişkimizin oluşup oluşmadığını kontrol edelim. Diyagram üzerinde sağ tuşla tıklayıp yeni bir veritabanı diyagramını hazırlayalım.

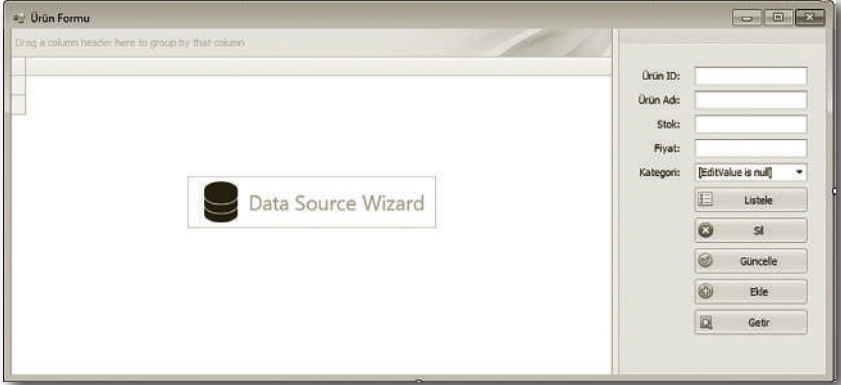


Karşımıza çıkan tablo seçeneklerinden **Kategori** ve **Ürün** tablolarını seçelim. Burada yer alan migration tablosu yaptığımız veritabanı güncellemelerine ait logları tutmaktadır.



Kategori ve ürün tablolarını seçtikten sonra karşımıza bire-çok ilişki yapısı sonsuz bir şekilde gelecektir.

Şimdi projemizin form kısmına gidip Yeni Bir Form ekleyelim. Formumuz ürün işlemlerine ait operasyonları gerçekleştirecektir. Formumuzun ismini **FrmProduct** olarak verebiliriz. Formumuzu formlar klasörüne oluşturduktan sonra boyutunu 900;412 olarak ayarlayıp akabinde tasarımımızı aşağıdaki gibi gerçekleştirelim.



Ürün tablomuzun SQL kısmına geçip birkaç tane örnek veri girişi yapalım.

ProductID	ProductName	ProdcutStock	ProductPrice	CateqoryID
1	Çamaşır Makinesi	100	1800,00	1
2	Bulaşık Mkinesi	150	1400,00	1
3	Buzdolabı	80	3500,00	1
4	Laptop	50	4000,00	3
5	Oturma Grubu	20	5500,00	6
»»	NULL	NULL	NULL	NULL

Tablomuz 5 tane örnek veri girişi yaptık. Artık listeleme işlemi için hazırız.

## BUSINESS KATMANI

Crud işlemlerini gerçekleştirebilmek için öncelikle Business tarafında istenilen şartların sağlanıp sağlanmadığını kontrol etmemiz gerekiyor. İlk olarak listeleme işlemi ile başlayalım.



# 30

## GİT HUB KULLANIMI

### BU BÖLÜMDE

Git Hub Nedir?	402
Neler Öğrendik?	410

Bu bölümde, Git Hub kavramının ne olduğunu, Git Hub'da nasıl hesap açıldığını, sisteme nasıl giriş yapıldığını, yeni bir projenin nasıl yükleneceğini, mevcut bir projenin gizliliğinin nasıl değiştirileceğini öğreneceğiz.

## GİTHUB NEDİR?

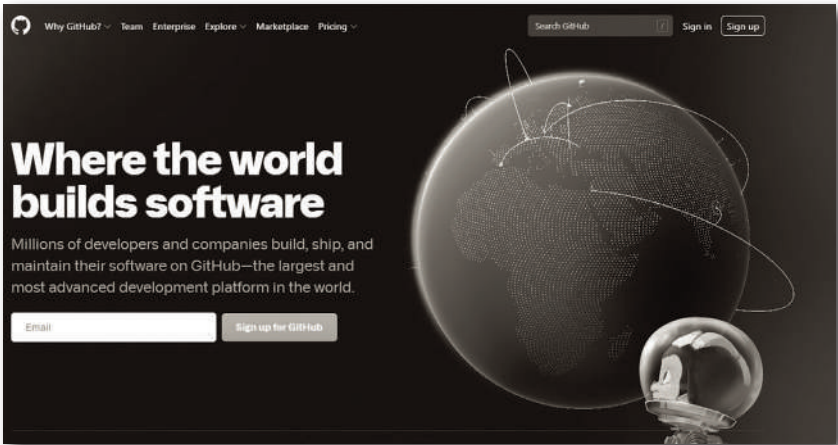
GitHub kavramını yazılım dünyasına giren her birey mutlaka duymuştur. 'Nedir?' sorusuna cevap olarak Vikipedi'den aldığım şöyle bir tanım var; sürüm kontrol sistemi olarak Git kullanan yazılım geliştirme projeleri için web tabanlı bir depolama servsidir.

8 Şubat 2008 yılında Tom Preston-Werner, Chris Wanstrath, P. J. Hyett, Scott Chacon tarafından kuruldu. GitHub özel depolar için ücretli üyelik seçenekleri sunarken, açık kaynaklı projeler için ücretsizdir. Mayıs 2011 itibarıyla GitHub açık kaynaklı projeler tarafından tercih edilen en popüler depolama servsidir.

4 Haziran 2018'de Microsoft, GitHub'ı 7.5 milyar dolar karşılığında satın alacağını açıklamıştır ve 26 Ekim 2018'de de satın alımın tamamlandığını açıklamıştır.

Yani özetleyecek olursak projelerin paylaşılmasına olanak sağlayan bir depolama ortamı olarak tanımlanabilir.

Resmi web sitesine <https://github.com/> gidelim.



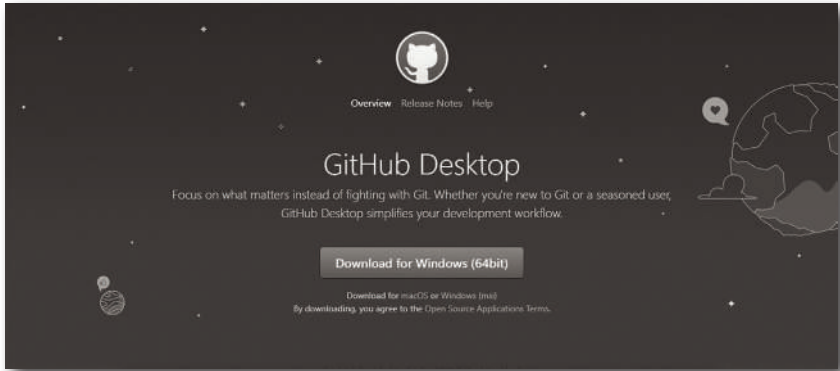
GitHub'da projelerimizi yüklemek için birden fazla yöntem bulunmaktadır. Biz bunlar arasında GitHub'ın masaüstü uygulamasını indirip buranın üzerinden yükleme işlemini yapacağız.

Bağlantıdan kendimize yeni bir GitHub hesabı oluşturalım.

<https://github.com/join?source=login>

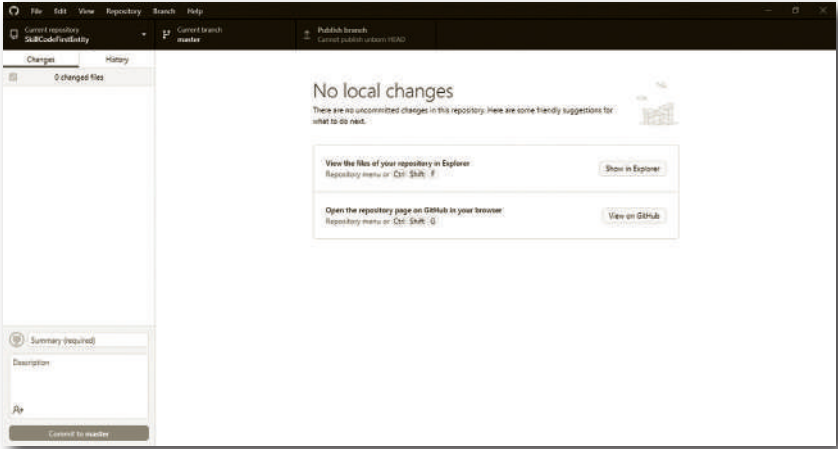
Hesabımızı oluşturduktan sonra GitHub'a giriş yapalım. Giriş yaptıktan sonra bağlantıdan Github masaüstü uygulamasını indirelim.

<https://desktop.github.com/>



İndirme işleminden sonra GitHub masaüstü programını kurup tekrar giriş yapmamız gerekecektir. Biraz önce oluşturduğumuz kullanıcı adı ve şifremizle sisteme masaüstü programı üzerinden de giriş yapalım. Kurulum işlemlerinden sonra karşımıza şöyle bir ekran gelecektir.

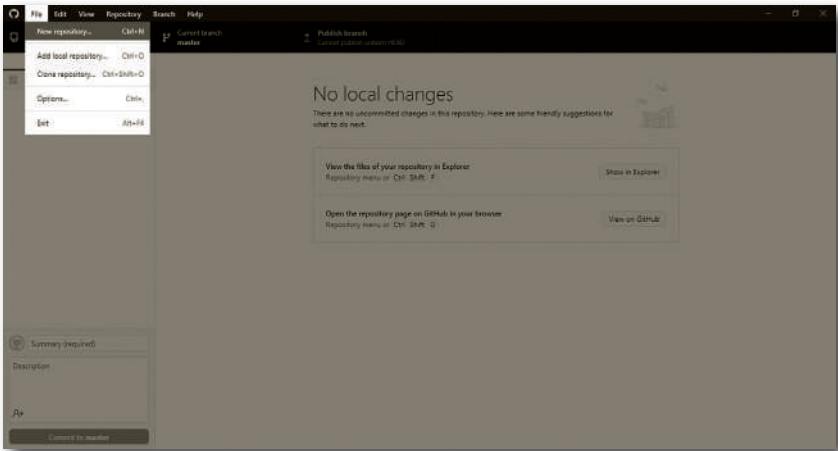
Bir sonraki görseli inceleyebilirsiniz.



Bu ekran projelerimizi yükleyeceğimiz **GitHub Desktop** uygulamasının ara yüzüdür. GitHub'da yüklenen projeler **Repository** olarak adlandırılır. Bizim yeni bir repo oluşturup yüklememiz gerekecektir. Ben sizlerle kitabımızın 2. projesi olan katmanlı mimari projesini paylaşacağım. Bunun için şu adımları takip edelim.

Öncelikle **Başlat** menüsüne **repos** yazıp **Visual Studio**'da oluşturduğumuz projelerin kaydedildiği **repos** isimli klasöre gidelim. Burada yer alan katmanlı mimari projemizi birazdan kopyalayacağız.

GitHub uygulamamızı açıp **File** kısmından **New repository...** seçeneğinin üzerine gelelim.



## MURAT YÜCEDAĞ

---

*[muratyucedag.com](http://muratyucedag.com)*



*[facebook.com/murattyucedag](https://facebook.com/murattyucedag)*



*[twitter.com/murattyucedag](https://twitter.com/murattyucedag)*



*[linkedin.com/in/murat-yücedağ-186933149](https://linkedin.com/in/murat-yücedağ-186933149)*



*[udemy.com/user/murat-yucedag-3](https://udemy.com/user/murat-yucedag-3)*



*[youtube.com/user/YazilimHerYerde](https://youtube.com/user/YazilimHerYerde)*



*[github.com/MuratYucedag](https://github.com/MuratYucedag)*



*[yucedagmurat23@gmail.com](mailto:yucedagmurat23@gmail.com)*

