

- **CRUD Açılımı:** Create → Oluştur / Read → Oku / Update → Güncelle / Delete → Sil
- Bir veritabanı bağlamında (DbContext) yapılmış olan ve bellekte duran değişikliklerin veritabanına uygulaması (kaydedilmesi) için çağırılan DbContext sınıfı metodu **db.SaveChanges();**
- Bir modelin verilerinde hata olup olmadığını, model durumunun (model state) geçerli olup olmadığını, bir aksiyon metodu içerisinde denetlemek için gereken ifade ModelState kullanılarak **if(ModelState.IsValid)** şeklinde yazılır.
- LINQ kısaltmasının açılımı **Language Integrated Query (Dil İlişkili Sorgu)** idir.
- Çoklu dil için kullanılan dosyaların uzantısı **.resx** idir.
- Bir projede “Yoneticiler” isminde bir rol mevcuttur. Bir görünümü (view), yalnızca “Yoneticiler” rolüne sahip kullanıcıların erişimine açmak için, ilgili aksiyon metodunun üst tarafına yazılması gereken veri notu (data annotation) **[Authorize(Roles = “Yoneticiler”)]** idir.
- Select \* FROM Personel Order by DogumTarihi, AdSoyad sorgusu LINQ kullanılarak bu şekilde yazılır.

**db.Personeller.OrderBy(x => x.DogumTarihi).ThenBy(x => x.AdSoyad);**

- Bölüm tablosunun ilk 5 sorgusunu elde eden tablo **db.Bolum.Take(5);**
- Personelleri doğum tarihlerine göre büyükten küçüğe olacak şekilde sıralayacak sorgu **db.Personeller.OrderByDescending(x => x.DogumTarihi);**
- Select \* FROM Bolum WHERE ID NOT IN (1,5,6) sorgusu LINQ kullanılarak bu şekilde yazılır.

**db.Bolum.Where(x => x.ID != 1 && x.ID != 5 && x.ID != 6);**

- App\_Data : Proje içerisinde ki veritabanı vb. veri kaynaklarının, dosyalar halinde eklendiği klasördür.
- App\_Start : Projenin ilk açılışı esnasında (sunucunun ilk başlatılışı ya da yeniden başlatılması), işlem yapması istenilen sınıfların, filtrelerin, metodların bulunduğu klasördür. Örneğin Route, yani sayfa yönlendirme dosyasının ya da projeye eklediğimiz JavaScript, Css gibi dosyaların sıkıştırılarak istemciye gönderilmesi mimarisi olan BundleConfig gibi dosyaları burada barındırırız.
- Content : Bu klasör, zorunlu olmamakla birlikte genelde css dosyaları bu klasörde barındırılabilir.
- Controllers : Controller dosyalarının barındırıldığı klasördür.
- fonts : Asp.Net Mvc 5 sürümü ile birlikte, örnek projeye eklenen (bu klasörü kullanmak zorunda değilsiniz), font dosyalarını barındırabileceğiniz alandır.
- Models : Model dosyalarının barındırıldığı klasördür.
- Scripts : Javascript tarzı script dosyalarını barındırabileceğiniz alandır.
- Views : View dosyalarının barındırıldığı klasördür.
- Görünüm (view) dosyalarına yazılan Razor Görünüm Motoru (Razor View Engine) komutlarının başına @ gelir.
- Bir kontrollerden (controller) , görünüm (view) dosyalarına bilgi aktarmak için kullanılan dinamik nesneye **Viewbag** denir.
- ASP.NET MVC projeleri için Visual Studio 2013 tarafından sunulan kimlik doğrulama metodları şu şekildedir.
  - Bireysel Kullanıcı Hesapları (Individual User Accounts)
  - Organizasyon Hesapları (Organizational Accounts)
  - Windows Kimlik Doğrulama (Windows Authentication)
  - Kimlik Doğrulama Seçeneği Olmayan (No Authentication)
- Projeye bootstrap vb. kütüphaneleri eklemek ve yönetmek için kullanılan paket yöneticisi **nuget** idir.

- Lambda ifadesinin sembolü => şeklindedir.
- MVC projelerinde veritabanı ayarları (bağlantı karakter dizini) **Web.config** yapılandırma dosyasında gerçekleşir.
- MVC projelerinde yönlendirmeleri düzenlemek için **RouteConfig.cs** dosyası kullanılır.
- Visual Studio 2013 ile varsayılan olarak kurulan SQL Server Express Veritabanı motorunun hafif sürümüne **LocalDB** adı verilir.
- El ile yazılan model ve veritabanı bağlam (dbcontext) dosyaları yardımı ile veritabanı , tablolar ve arasındaki tüm ilişkilerin entity framework tarafından otomatik olarak oluşturulmasını sağlayan geliştirme yaklaşımına **Önce Kod (Code First)** adı verilir.
- Asp.Net Mvc'de veri doğrulama için kullanılan ve köşeli parantezler içerisine yazılan ifadeler (örn, [Required]) **Veri Açıklama Notu (Data Annotation)** adı verilir.
- ASP:NET MVC çatısında görünümünün (view) kaynak kodların saklandığı dosyaların uzantısı **.cshtml** idir.
- ASP:NET MVC çatısında bir aksiyon (action) metoduna bağlı <a href= ""> oluşturmak için kullanılan Razor komutu **Html.ActionLink**
- ASP.NET MVC çatısında html formları oluşturmaya başlamak için kullanılan razor komutu **Html.BeginForm**
- Bir nesnenin modelin özellikleri için hmtl biçimlendirmeye yapılarını döndüren (Şablon ID veri) razor komutu **Html.DisplayFor**
- Çapraz site istek sahteciliği (Cross site saldırısı) ataklarından korunmak için controller içerisinde ilgili aksiyon metodlarının başında **ValidateAntiForgeryToken** ifadesi yer alır.
- Controller içinde veritabanı ile işlem sonlandığında veritabanı bağlantısının kapatılması için kullanılan metod **dispose**
- Güçlü tipli görünüm (Strong Type View) dosyasındaki kaynak kodlar **Model** anahtar kelimesi ile başlar.
- Yorum satırı yazmak için kullanılan razor sembolleri @\* \*@ şeklindedir.
- Yerel veritabanı kullanılırken oluşturulan .mdf veritabanı dosyası Asp.Net MVC projesinde **App\_Data** içerisinde yer alır.
- .Net MVC uygulamasında web tarayıcıları adres çubuğuna adres girildiğinde çağırılan controller action metodlarına verilen genel isim **Aksiyon metodu**
- Çoğu SQL komutlarına benzer ya da aynı olan C# anahtar kelimelerinden oluşan ifadeler ile .NET koleksiyonlarına LINQ sorguları yazılır. Bu anahtar kelimeler, **LINQ Standart Sorgu Operatörleri (LINQ Standart Query Operators)** olarak anılır. Bu operatörler **System.Linq** isim alanı altında yer alan **Enumerable** sınıf tarafından tanımlanır.

### **LINQ OPERATÖRLERİ**

- **Aggregate** metodu, aldığı listenin elemanları üzerinde tek tek işlem yaparak, bir önceki işlem sonucunu bir sonraki elemanın işleme sokulduğu noktaya taşır. Lambda ile beraber kullanılır.
- **All** operatörü belirli bir şartı tüm elemanların sağlayıp sağlamadığını kontrol eden bir LINQ operatörüdür.
- **Average** operatörü sayısal dizilerin ortalamasını bulmaya yarayan LINQ operatörüdür.
- **Concat** operatörü iki diziyi birleştirmeye yarayan LINQ operatörüdür. Yalnız iki dizide aynı eleman varsa o elemanı iki defa oluşturur.
- **Contains** operatörü bir dizinin, nesnenin, koleksiyonun belirtilen öğeyi içerip içermediğini görmeye yarayan LINQ operatörüdür.

- **Count** operatörü bir dizideki öğelerin sayısını döndüren LINQ operatörüdür.
- **DefaultIfEmpty** operatörü eğer null (boş) değer taşıyan bir değer varsa onun varsayılan değerini yazdırmaya yarayan LINQ operatörüdür.
- **Distinct** operatörü bir dizi vb. aynı elemandan birden fazla içeriyorsa onlardan sadece bir tanesini döndürmeye yarayan LINQ operatörüdür.
- **Except** operatörü elemanı hariç tutmaya yarayan LINQ operatörüdür. Örneğin iki dizi aynı elemana sahiptir ve her iki dizi de yazdırılacaktır. İlk dizi de o elemanı yazdırır ancak ikinci dizide o elemanı yazdırmaz.
- **First** operatörü belirtilen koşula uygun ilk öğeyi döndüren LINQ operatörüdür.
- **FirstOrDefault** operatörü belirtilen koşula uygun öğe varsa onun ilk öğesini, yoksa varsayılan değeri döndürmeye yarayan LINQ operatörüdür.
- **GroupBy** operatörü belirtilen koşula göre gruplandırmayı sağlayan LINQ operatörüdür.
- **Intersect** operatörü iki dizinin kesişimini bulan LINQ operatörüdür.
- **Join** operatörü iki dizi arasında birleştirme işlemi yapan LINQ operatörüdür. Lambda kullanılır.
- **Last** operatörü belirtilen koşula uygun son değeri döndüren LINQ operatörüdür.
- **LastOrDefault** operatörü belirtilen koşula uygun öğe varsa onun son öğesini, yoksa varsayılan değeri döndürmeye yarayan LINQ operatörüdür.
- **Max** operatörü grubun ya da dizinin en büyük elemanını bulmaya yarayan LINQ operatörüdür.
- **Min** operatörü grubun ya da dizinin en küçük elemanını bulmaya yarayan LINQ operatörüdür.
- **OrderBy** operatörü belirtilen koşula göre en küçükten en büyüğe doğru sıralamaya yarayan LINQ operatörüdür. Bu operatörün içerisinde sıralama için **OrderByDescending** kullanılırsa en büyükten en küçüğe doğru sıralar.
- **Range** operatörü belirli bir aralıktaki sayıları almaya yarayan LINQ operatörüdür.
- **Repeat** operatörü tekrar işlemleri yapmak için kullanılan LINQ operatörüdür.
- **Reverse** operatörü tersine çevirmeye yarayan LINQ operatörüdür.
- **Select** operatörü seçip göstermeye yarayan LINQ operatörüdür.
- **SelectMany** operatörü iç içe seçim yapıp göstermeye yarayan LINQ operatörüdür.
- **Skip** operatörü seçilen elemanları atlamaya yarayan LINQ operatörüdür.
- **SkipWhile** operatörü koşul sağlandığı sürece elemanları atlamaya yarayan LINQ operatörüdür.
- **Sum** operatörü toplam bulan LINQ operatörüdür.

- **Take** operatörü seçilen elemanları almaya yarayan LINQ operatörüdür.
- **ThenBy** operatörü sıralama içerisinde sıralama yapmaya yarayan LINQ operatörüdür, küçükten büyüğe sıralar, **ThenByDescending** ise büyükten küçüğe sıralama yapar.
- **Union** operatörü dizileri birleştirmeye yarayan LINQ operatörüdür. Concat operatöründen farkı ise Union aynı elemanları sadece bir kere gösterir.
- **Where** operatörü dizi, grup vb. içerisinde eleman bulma koşulu oluşturan LINQ operatörüdür.

### **HTML HELPER METODLARI**

- **[HttpPost]** ve **[HttpGet]** metodlarından biri yukarıda belirtilmediyse varsayılan olarak **[HttpGet]** metodu seçilir. Arama, detay, listeleme gibi konularda **[HttpGet]** doğru kullanımdır. Get metodu salt okuma işlemler için kullanılır. **[HttpPost]** metodu ise kullanıcı girdileriyle sunucularda bir şeyleri değiştirmek istendiğinde kullanılır. Kısacası web uygulamalarında okuma işlemleri için Get, yazma işlemlerinde Post metodu kullanılır.
- **Html.TextBox** → Bilgi giriş kutusu. **Html.TextArea** → Çoklu satır veri girişi için kullanılır.
- **Html.DropDownList/Html.ListBox** → Listedeki eleman seçmeye yarayan helperlardır. Farkları ise DropDownList tek eleman seçimine izin verirken Listbox birden fazla eleman seçimine izin verir.
- **Html.Label** etiket kullanımıdır. **Html.ValidationMessage** ise ModelStateDictionary içerisinde herhangi bir yerde hata varsa bu hatayı göstermek için kullanılır.
- **Html.RadioButton** birden çok seçenekten sadece birini seçmek için kullanılan helperdır. **Html.TextBox** yine seçmeye yarar, ancak birden fazla seçenek seçilebilir.
- Uzantıya **For** geldiğinde güçlü tipli (Strongly-Typed) olmaktadır. For olmayanlar zayıf tiplidir. Güçlü tipli veri tipi kesin olan anlamına gelmektedir.