**DATAGRİDVİEW SET İŞLEMİ**

GridView Doldurma işlemi için DbContext classımızda herhangi farklı bir işlem yapmamız gerekmemektedir.

Listeleme için ilgili Database tablosuna referans olarak DbSet Kod satırımız DbContext içerisinde yazılı olmalı.

GridView Doldurma işlemini Normal olarak Form\_Load olayında gerçekleştirilebilir. Fakat isteğe göre eğer çekilecek veri fazla ise Programın ilk açılma aşamasında yüklenme olabilir buda Performans zayıflamasına hatta programın çökmesine sebep olabilir buna alternatif çözüm olarak filtreleme gibi işlemler uygulanarak sadece istenilen veriler veyahut filtreleme ile tüm veriler çekilebilir.

Az Veri olduğunu varsayarak işlemimize devam edelim.

Database bağlantımız AppDbContext olduğu için using ile nesnemizi tanımlayıp işlemlerimizi bunun içerisinde yapmaya devam ediyoruz. GridView doldurma işleminde genelde çekilen veriler içerisinde Foreign Key varolan tablo bulunmaktadır. Ek olarak Null değer gelmesi muhtemel olucak Kolonlarda bulunabilir , null değer döndürmesi muhtemel olan bu kolonlar belirlenir ve Public olarak tanımladığımız EntityFramework Classlarımız içerisindeki bu değişkenler null değer çekmesi muhtemel olabilir şeklinde ayarlanmalı bu işlemi şu şekilde yapabiliriz.

Public int? TonerID {get; set;}

Yukarıdaki kod satırında bulunan “?” simgesi belirtilen değişkenin null değer döndürebilir bilgisini belirtmekte. Ayrıca bu işlem null değerden dolayı programın hata vermesini absorbe eder.

Gelelilim Tekrar Formun .cs kısmına;

Using (var context = new AppDbContext()){

var query = context.Printer.Include(p => p.Toner).Select(p.new{ p.ID, p.Marka, p.Model, Toner\_Kodu = p.Toner.TonerKodu}).ToList();

gwYazici.DataSource = query; gwYazici.Colums[0].IsVisible = false;

Yukarıdaki kod bloğunda DbContext using nesnemizin içerisinde işlem yapmaktayız bu işlemde öncelikle veritabanından query işlemi yapacağımızdan dolayı name bilgisine query dediğim bur var nesnesi oluşturuyoruz. Context içerisindeki Printer Tablo bağlantımızı kullanarak Include Methodunu kullanıyoruz. Include Methodu işlem yapmak istediğiniz sorgudaki nesnenin ilişkili olduğu bütün nesneleri kullanmanıza olanak tanır bu işlem EntityFramework ve Linq nun birlikte özdeşleşmiş bir halini sunar. Include içerisine p adında nesnemizi tanımlayıp Printer ın ilişkili olduğu Toner Tablosunuda işin içine dahil ediyoruz. Ve bu p nesnemizi new ile oluşturmaya başlıyoruz. Öncesinde Include ile birleştirdiğimiz bu 2 tablodaki Kolonları kullanarak GridView a eklemek istediğimiz kolonları araya virgül koayarak yazıyoruz.

“Toner\_Kodu” ile başlayan Kod satırımıza gelmeden içerisinde TonerID nin bulunduğu EntityFramework olarak kullandığımız clsPrinter classımıza geliyoruz. Eğer TonerID olarak GridView a ekleme işlemini yaparsak karşımıza Foreign Key olarak alınan ID değerleri çıkar fakat kullanıcıların ID işelmleri ile ilişkili olmaması gerekli bunun için “Navigation Property” denilen işlemi uyguluyoruz. Bu işlem Printer tablosundaki TonerID kolonuna denk olan Toner Tablosundaki TonerKodu text bilgisini ekrana getirmektir. clsPrinter classımıza public olarak bir değişken daha ekliyoruz.

Public clsToner Toner { get; set;}

Bu işlem yukarıdaki quer kısmında dahil ettiğimiz Tablodanda veri çekmemizi sağlamakta ve görüldüğü üzere veritipi kısmına clsToner yazmaktayız.

Navigasyon işlemimiz bittiğine göre tekrar .cs kısmına gelebiliri. Toner\_Kodu olarak başladığımız Kod satırında öncelikle GridView Kontrolünün Sütun isminde görünmesini istedğimiz şekilde name veriyoruz. İçerisinde iki tabloyu birleştirdiğimiz p nesnesini yazıp Toner Tablosunu belirtip yanınada TonerKodu yazarak TonerID ye karşılık gelen text bilgisini çekiyoruz. Bu işlemden sonra listeleme için ToList() ile bitiriyoruz son olarak GridViewımızın datasını DataSource ile oluşturduğumuz query nesnesi olarak belirleyip Printer Tablosundaki ID değerimizi gizledikten sonra işlemi bitiriyoruz.