**ENTİTY FRAMEWORK & LİNQ İLE UPDATE VE DELETE İŞLEMLERİ**

Önceki yazılarımızda İnsert İşlemini gerçekleştirmiştik. Şimdi Update işlemini anlatıyorum.

Update Yapılması gereken Button Click eventine geliyoruz.

Using (var context = new AppDbContext())

{

İnt selectedValue = Convert.ToInt32(cmbSecToner.SelectedValue);

clsPrinter p = new clsPrinter

{

ID = Convert.ToInt32(txtSecID.Text),

Marka = txtSecMarka.Text,

Model = txtSecModel.Text,

TonerID = selecetedValue

};

Context.Printer.Update(p);

Context.SaveChanges();

MessageBox.Show(“Update Successful”, “Update Process”, MessageBoxButtons.OK, MessageBoxIcon.Information);

checkSecUpdate.Checked = false;

}

Database işlemlerini gerçekleştirmek için oluşturduğumuz using içerisinde GridView ın Satır seçiminde verileri aktardığımız kontrollerden öncelikle Combobox’ın Value değerini int tipindeki bir değişkene alıyoruz bunun sebebi okunabilirliği arttırmaktır. Daha sonra işlem yapıcağımız değişkenlerimizin bulunduğu Entity Classlarımızı nesne olarak tanımlayıp içerisine classlardaki değişkenleri ve karşılığı olarak Formumuzdan kaydedilecek verileri gönderen kontrolleri yazıyoruz. ID kolonu bu noktada önemlidir sebebi ise hangi kaydı Update edeceğini belirler.

Class içerisinde değişken eşitleme işlemlerimiz bittikten sonra Update Methodunu kullanarak nesne olarak tanımladığımı “p” classını parantez içerisinde belirtiyoruz.

Kullanıcıya Update işleminin başarılı olduğuna dair MessageBox ile bilgi verdikten sonra son olarak önceki metinlerimizde belirttiğimiz Update işleminin güvenliğini amaçlayan Checkbox kontrolünüde False hale getirip Güvenliği sağladıktan sonra Update ile ilgili olan işlemlerimizi bitiriyoruz.

Delete işlemindede İnsert ve Update işlemlerinin aynısı yapılır. Eğer Sql Server üzerinde Sorgu yazmayı biliyorsanız Update işleminde Hem güncellenmesi gereken veriler hemde güncellenmesi gereken kaydın ID değeri gerektiğini bilirsiniz. Delete işleminde ise ID haricinde hiçbir bilgiye gerek olmadığınıda bilirsiniz.

Using(var context = new AppDbContext())

{

clsPrinter p = new clsPrinter

{

ID = Convert.ToInt32(txtSecID.Text)

};

Context.Printer.Remove(p);

Context.SaveChanges();

MessageBox.Show(“Delete Successful”, “Delete Process”, MessageBoxButtons.OK, MessageBoxIcon.Information);

gwYazici.Rows.Clear();

GwYazici();

}

Delete işleminin yapılacağı Buttonun Click eventine geliyoruz. İçerisinde database işlemi yapmamiz için DbContext nesnesi oluşturuyoruz. İşlem yapılacak Tablonun Entity Classının içerisinde Class içindeki ID değişkenine Formdaki ID değerimizi gönderiyoruz bu işlemde ID direkt olarak GridView içerisinden değil. GridView da seçili olan satırın verilerini aktardığımız kontrollerden alıyoruz.

Daha sonra delete işlemi için context nesnesinde Printer tablosunun Remove Methodunu kullanıp içerisine ID değeri çektiğimiz Entity Classımızı belirtiyoruz.

Delete işleminin yapıldığına dair bilgi veren bir messagebox ile kullanıcıya bilgi veriyoruz ve GridView içeriğini restart edip işlemi bitiriyoruz.