6. Gun

New file kismindan “Service-based Database” secenegine tiklayip veritabanini programimiza ekliyoruz.

.mdf = Microsoft database file

Anahtar olan satirin otomatik artan olmasi icin “Identity Specification” kismini “yes” yapacaksin.

Tablodaki verilerin default degerlerini verebiliyoruz. (False icin 0 da kullanilir)

GetDate() tarihi uretir default kismina eklenebilir.

Database diagramin icine tablolarin ikisini de ekliyoruz ve tablolar onumuzda gozukuyor.

Iki tabloyu iliskilendirirken bir taraftan ok cikarip mouse ile obur tarafa surukleyip birakabiliriz.

INSERT AND UPDATE Specific = burdan ayarlama yapabiliriz su an bir degisiklik yapmiyoruz.

Diagrami da kaydediyoruz.

Direkt olarak diagramin icinden de yeni tablo eklenebilir.

Table view kismindan gorunum degistirilebilir diagram icinde.

Modify Type kismindan tabloda gosterilecek verileri de degistirebiliriz. Artan mi, primary key mi vs…

Iki tabloyu iliskilendirdigimiz icin satis kismina bilgi giriyoruz musteriID satiri ile ve musterinin satis bilgileri ekleniyor oraya.

Iki tabloda ortak bir satir var!!!

Normalizasyon = Kisisel bigiler farkli tabloda, ornegin satislar farkli tabloda, kargolar farkli tabloda vs…

Normalizasyon yapmak daha iyi.

Tablolari olabildigince ve mantikli olarak parcalamamiz bize avantaj saglar.

Herhangi bir tabloya sag tiklayip “new query” degerini seciyoruz. (yeni sorgu)

Gelen sorguda tum tablolari sorgu kismina ekleyebiliriz.

New Query icinde yeni bir tablo ekleyebiliriz.

Su an sadece bir tane tablo uzerinde sorgularimizi gerceklestirecegiz.

All column kismina tikliyoruz.

Sorgu ekranina sag tiklayip “execute sql” e tiklayip tablodaki degerleri gorebiliriz.

‘Sort order’ kisminda ‘ascending’ a dan z ye siralar. ‘Descending’ ise z den a ya siralar.

Kredi limiti belli bir sinirin ustunde olan degerleri bulmak icin ‘filter’ kismina gelip sinir sayiyi oraya girebiliriz. Basindaki isareti “<” veya “>” yapabiliriz. Alt sinir ve ust siniri bulabiliriz bu sayede. ‘=’ de kullanabiliriz. Olmayan bir deger girilirse ‘null’ uretecektir cunku oyle bir musteri yok tabloda mesela.

Gozukmemesi icin output kismindaki checkbox i isaretsiz hale getiririz.

SELECT Adiniz,Soyadiniz

FROM Musteriler

WHERE (MusteriID = 1) mesela yani …

Yukarida ‘Change Type’ kismi var oraya da bakabilirsin. SELECT, UPDATE and DELETE vs…

“1 row affected by last query“(UPDATE ettik)

Delete ederken uyari aldik foreign key den dolayi izin almamiz gerekir.

Veritabanina sol tarafta sag tiklayip ‘close connection’ dersek baglantiyi kesmis oluruz tekrar acmak icin ‘refresh’ butonuna basabiliriz.

Bir veri tabanina baglanirken ‘database explorer’ kisminda en sagdaki butona tiklariz. Veritabani onceden olusturulmus ve bilgisayarinizda bulunmasi gereklidir.

7. Gun

1

Sql server management studio ya gir daha once visual studio da olusturdugun veritabanini ‘object explorer’ – ‘databases’ kismindan eris daha sonra tabloya sag tikla ve ‘select top 1000’ kismina tikla.

3 tane tablo var su an veritabanimizda (musteriler, satilanurunler, satislar)

Script = senaryo – script file = senaryo dosyalari, execute ettigimizde bu senaryolar gerceklestirilir.

‘Data explorer’ da kendi veritabanimiza sag tiklayip ‘Tasks’ kismindan ‘Generate Scripts’ degerini seciyoruz. Next -> Next -> Next -> Tables -> Next -> Hepsi -> Next -> Script to file -> Next -> Finish 🡪

Verilerin kaydedilebilmesi icin yukleme sirasinda ‘Script Data’ kismini true yapmamiz gerek. 🡪 Bu sayede veritabanindaki tum veriler script dosyaya kaydedilmis olur.

Script dosyasi baska bilgisayara kopyalandiginda da calisir.

2

Sql server management studio ya gir daha once visual studio da olusturdugun veritabanini ‘object explorer’ – ‘databases’ kismindan eris daha sonra tabloya sag tikla ve ‘select top 1000’ kismina tikla.

Varsayalim ki veritabanimiza bir zarar geldi silindi, kayboldu veya atiyorum biz yanlislikla bir tabloyu sildik… Peki biz bu tabloyu veya belgeleri nerden bulacagiz. Iste burada ise yarayan sey daha once kaydettigimiz script dosyasidir. Script dosyasina tikliyoruz ve daha sonra ‘obejct explorer’ da refresh dedigimizde bu tablolarimiz ve kayip veriler geri gelmis olacak.

Script dosyayi kaydederken w sekilde kaydet!!! birincide veriler olmasin ikincide olsun. (Bu benim kendi fikrim)

Direkt ‘musteriler tablosu’ silinemez cunku musterilerin diger tablolarla iliskisi var o yuzden once iliskisinin oldugu tablolar silinir daha sonra ‘musteriler tablosu’ silinebilir

3 (TRANSACT SQL - SELECT)

Sql server management studio ya gir daha once visual studio da olusturdugun veritabanini ‘object explorer’ – ‘databases’ kismindan eris daha sonra veritabanina sag tikla ve new query yi tikla.

Simdi tum kodlari elimizle yazacagiz.

SELECT \* FROM Musteriler 🡪 Musteriler bir tablodur. Daha sonra execute ediyoruz.

SELECT Adiniz, Soyadiniz FROM Musteriler 🡪 Sadece ad ve soyad degerleri tablodan cekilecek.

SELECT Adiniz AS FirstName, Soyadiniz AS LastName FROM Musteriler 🡪 isimler tabloda AS ile degistiriliyor. Ama sadece goruntu olarak degisiyor gercekte yine adiniz ve soyadiniz degerleri kaliyor.

SELECT Adiniz + ‘ ’ + Soyadiniz AS AdveSoyad FROM Musteriler 🡪 adiniz ve soyadiniz sutunlarini birlestirip AdveSoyad adinda yeni bir sutun olusturmus oluruz.

SELECT Adiniz + ‘ ’ + Soyadiniz AS AdveSoyad FROM Musteriler

WHERE Sehir = ‘KONYA’ 🡪 Konya ilinde bulunan ad soyad olarak tek sutundaki kayitlari getirecek sadece

SELECT Adiniz + ‘ ’ + Soyadiniz AS AdveSoyad FROM Musteriler

WHERE Adiniz = ‘ILKER’ 🡪 ILKER adinda bulunan ad soyad olarak tek sutundaki kayitlari getirecek sadece

!!!HEPSINDEN SONRA EXECUTE EDIYORUZ!!!

4 (TRANSACT SQL - INSERT)

Tablolarin icine veri eklenirken kullanilir.

Yine bir new query aciyoruz.

INSERT Musteriler (Adiniz, Soyadiniz, Email, KrediLimiti, KayitTarihi)

VALUES (‘Sevket’, ‘TIRYAKI’, ‘abc@mail.com’, 100, 20/01/2000) 🡪 EXECUTE EDIYORUZ HER SEFERINDE

SELECT \* FROM Musteriler

WHERE Soyadiniz = ‘TIRYAKI’ 🡪 Bu kodu yazip mouse ile sadece bu kodu secerek ‘parse’ ye tikliyoruz. Yukarida tik isareti var o iste. Daha sonra execute edilirse sadece bu kod calisacak ve soyadi TIRYAKI olan kimseyi bulamadigi icin NULL olacak ama tum kodu calistirirsak bu sefer bize bulacak.

INSERT Musteriler (Adiniz, Soyadiniz, Email, KrediLimiti, KayitTarihi)

VALUES (‘Mucahit, ‘AKTEPE, ‘mma@mail.com’, 200, 20-07-2005) 🡪 Onceki kodun altina yazmadik direct ustune degistirdik. Tarih kismini string olarak girilemez!!!

SELECT \* FROM Musteriler

WHERE Soyadiniz = ‘AKTEPE’

Ortaklar adinda bir tablo olusturduk sadece ad, soyad e tarih iceriyor.

INSERT INTO Ortaklar (Adi, Soyadi, Tarih)

SELECT Adiniz, Soyadiniz, GETDATE () FROM Musteriler 🡪 GETDATE () = Tarihi dondurur KayitTarihi yerine kullandik.

WHERE Soyadiniz = ‘TIRYAKI’ 🡪 Parse ile bu satirlari secip control ediyoruz ve hata yok!

Simdi execute yapiyoruz.

SELECT \* FROM Ortaklar 🡪 execute ediyoruz ve eklendigini goruyoruz.

5 (TRANSACT SQL - UPDATE)

UPDATE verilerin daha sonra guncellenmesi veya degistirilmesi icin kullanilir.

Yine bir new query aciyoruz.

SELECT \* FROM Musteriler WHERE MusteriID = 1006 🡪 Sonuc Sevket tiryaki simdi bunu degistirecez.

UPDATE Musteriler

SET Adiniz = ‘Ahmet’,

Soyadiniz = ‘YORULMAZ’,

Email = ‘ahmet@yorulmaz.com’

WHERE MusteriID = 1006

// Yorum satiri haline getirmek icin ‘- -‘ kullaniriz. Cift tik. Veya /\* \*/ bunu da kullanabiliriz.

Yorum satirini secip execute etsek calisir yorum satiri olmasi biseyi degistirmez.

WHERE deyimini kullanmayip bir sart olusturmazsak tum tablo Ahmet yorulmaz seklinde update edilecek

6 (TRANSACT SQL - DELETE)

SIlmek icin kullanilir.

SELECT \* FROM Musteriler where MusteriID = 444 🡪 execute ediyoruz.

DELETE FROM Musteriler where MusteriID = 444 🡪 Bu kaydi siliyoruz. Bunu silemedik cunku oburleriyle iliskisi var o yuzden bunu silemiyoruz. Cunku bunu kaldirirsak Satislar tablosundaki deger ortada kalacak.

SELECT \* FROM Satislar where MusteriID = 444 🡪 execute ediyoruz. (Bu deger ortada kalacak yani)

‘Object explorer’ kismindan Database diagramini aciyoruz. Properties kisminda INSERT And UPDATE kismina giriyoruz ve bir very silindiginde veya guncellendiginde iliskili tum verileri de silmek veya guncellemek icin burdaki degerleri No actiondan cikarip Cascade e ceviriyoruz. Ikisi arasindaki iliskiyi bozmaz. Onceden bir deger verdiysek set default u secersek default deger olur veri veya da Null yapariz (Set Null ile birlikte)

DELETE FROM Musteriler where MusteriID = 444 🡪 simdi sadece bu degeri secip execute ettigimizde silecek ve control icin tekrardan

SELECT \* FROM Satislar where MusteriID = 444 bu kodu cagirdigimizda silinmis oldugunu gorecegiz (sadece bu kodu secip bunu calistiriyoruz)

SELECT \* FROM Musteriler where MusteriID = 444 bu kodu secip bunu calistirirsak yine degerin silindigini gorecegiz.

Sonra ayni sekilde 666 yi da siliyoruz.

DELETE FROM Ortaklar 🡪 Ortaklarin hepsi silinir herhangi bir ayar da yapmadik cunku bir tabloyla iliskisi veya primary key yok.

7 (TRANSACT SQL - WHERE)

SELECT \* FROM Musteriler WHERE Sehir != ‘ANKARA’ 🡪 Ankara haricinde oturanlari bulur (!=)

SELECT \* FROM Musteriler WHERE Sehir <> ‘ANKARA’ 🡪 Ankara haricinde oturanlari bulur (<>) (Ayni)

SELECT \* FROM Musteriler WHERE KrediLimiti > 4500 🡪 4500 den buyuk kredi limitleri listelenir. (<)

SELECT \* FROM Musteriler WHERE (KrediLimiti/2) > 2000 🡪 Bu islem de legal

SELECT \* FROM Musteriler WHERE KrediLimiti > 4500 AND KrediLimiti < 4600 🡪 Aradaki degerleri bulur.

SELECT \* FROM Musteriler WHERE KrediLimiti BETWEEN 4500 AND 4600 - Aradaki degerleri bulur (Ayni)

SELECT \* FROM Musteriler WHERE KayitTarihi > ’01/01/2000’ AND KayitTarihi < ‘01/01/2006

SELECT \* FROM Musteriler WHERE Adiniz = ‘Ilker’ OR Soyadiniz = ‘YORULMAZ’

SELECT \* FROM Musteriler WHERE Sehir = ‘KONYA’ OR Sehir = ‘ANKARA’ OR Sehir = ‘BURSA

SELECT \* FROM Musteriler WHERE Sehir IN (‘KONYA’, ‘ANKARA’, ‘BURSA’) 🡪 Aynisi usttekinin

SELECT \* FROM Musteriler WHERE Soyadiniz like ‘AL%’ 🡪 AL basharfli soyadli musterileri bulur.

SELECT \* FROM Musteriler WHERE Soyadiniz like ‘%PE’ 🡪 AL sonharfli soyadli musterileri bulur.

SELECT \* FROM Musteriler WHERE Soyadiniz like ‘K%E’ 🡪 K basharf, E ise son harf arasi bilinmiyor.

8 (TRANSACT SQL – ORDER BY - TOP)

ORDER BY siralama yapmak icin kullanilir.

TOP lislelenen verilerin ilk 5 inin 10 unun ne sayi verirsek o kadarini gosterir.

SELECT \* FROM Musteriler

ORDER BY Soyadiniz 🡪 a’dan z’ye siralar (default olarak a’dan z’ye siralar)

ORDER BY Soyadiniz desc 🡪 z’den a’ya siralar

ORDER BY Soyadiniz asc 🡪 a’dan z’ye siralar

ORDER BY Soyadiniz, Adiniz

ORDER BY Soyadiniz desc, Adiniz

ORDER BY KrediLimiti desc 🡪 Buyukten kucuge siralar

ORDER BY KrediLimiti asc 🡪 Kucukten buyuge siralar

ORDER BY KayitTarihi desc 🡪 en yeniden en eskiye siralar.

ORDER BY KayitTarihi asc 🡪 en eskiden en yeniye siralar.

SELECT TOP 5 \* FROM Musteriler

ORDER BY KrediLimiti desc 🡪 En yuksek kredi limitine sahip 5 degeri getirir.

SELECT TOP 10 \* FROM Musteriler

WHERE Sehir = ‘KONYA’

ORDER BY KrediLimiti desc 🡪 Konya’da bulunan en yuksek kredi limitine sahip 10 degeri getirir.

9 (TRANSACT SQL – INNER JOIN)

Yine bir new query aciyoruz.

SELECT Musteriler.MusteriID, Adiniz, Soyadiniz, SatisTarihi FROM Musteriler 🡪 SatisTarihi bu tablomuzun icinde bulunmuyor dolayisiyla

INNER JOIN Satislar on Musteriler.MusteriID = Satislar.MusteriID

SELECT m.MusteriID, m.Adiniz, m.Soyadiniz, s.SatisTarihi FROM Musteriler m

INNER JOIN Satislar s on m.MusteriID = s.MusteriID 🡪 Musteriler m ile Satislar s ile ifade edilecek.

WHERE m.MusteriID = 74 🡪 Iki tabloyu birlestirerek islem yapmis olduk execute et gor!

SELECT m.MusteriID, m.Adiniz, m.Soyadiniz, s.SatisTarihi, su.SatisID FROM Musteriler m

INNER JOIN Satislar s on m.MusteriID = s.MusteriID

INNER JOIN SatilanUrunler su on s.SatisID = su.SatisID

WHERE m.MusteriID = 74 🡪 execute edersek tablolari birlestirdigimizi yine gorebiliriz.

8. GUN

1 (STORED PROCEDURES (C# Method))

Derleme sonrasinda ‘stored procedures’ kisminda saklanir.

SQL Server uzerinde barindirilan, T-SQL Komutlari ile hazirladigimiz sorgularin derlenmesi aninda otomatik olarak olusan SQL Server bilesenleridir.

Sakli Yordamlar T-SQL komutlarinda oldugu gibi sadece bir defa kullanilmaz. Sadece bir defa derlenir ve sinirsiz sayida kullanilir ve cagirildigi her anda islemlerine devam eder.

Veri tabani tasinirken Stored Procedures birlikte tasinir. (SQL Serverin icinde saklanir bu prosedurler)

Stored Procedures sorgu hizini arttirir. Cunku Stored Procedures SQL Server uzerine barindirilir ve bundan dolayida islemci gorevini SQL Server ustlenir.

Guvenlik ve performansi arttirir.

Ag uzerindeki veri akisini azaltir. Farkli bilgisayarlarda bulunan veri tabanlarina baglanirken ve sorgu gonderirken very iletisiminde ag trafigi artar ancak Stored Proceures kullanildigi taktirde sadece metot calisir ve sorgu akisi olmaz ve bu trafigin yukunu SQL Server azaltmis olur.

Stored Procedures SQL Server tarafinda, Bilgisayarda, Farkli Bilgisayarlarda ve DLL’ lerde bulundurulabiliyor.

2 (STORED PROCEDURES DEFINITION)

New Query aciyoruz.

CREATE PROCEDURE MusteriListesi

AS

BEGIN

SELECT \* FROM Musteriler

END

‘Object Explorer’ da Programmability kisminda ‘Stored Procedures’ dosyasi altinda bu yazdigimiz Procedure saklanir. Gozukuyorsa refresh et!

Yeni bir sorgu ekrani aciyoruz. (Diger sorguda olusturdugumuz procedure u denemek icin)

EXEC MusteriListesi 🡪 execute et! (alti kirmizi renkli olacak)

SQL server I tamamen kapat ve yeniden ac!

EXEC MusteriListesi 🡪 execute

“Stored Procedures” sag tiklayip yeni olusturup yazabiliriz.

Daha Sonra

Olusturdugumuz stored procedure e yani MusteriListesine sag tiklayip “Script stored procedure as” den

“Alter To” yu ordan da “New Query editor Window” u tikliyoruz.

Create to = Procedure dosyasini olusturmak icin kullanilir.

Alter to = Olusturulmus procedure u duzeltmek icin kullanilir.

Drop to = Olusturulmus procedure leri silmek icin kullanilir.

Bir ekran acilacak farkli gormedigin use, go tarsi seyler olacak altinda 🡪

…….

ALTER PROCEDURE [dbo]. [MusteriListesi]

AS

BEGIN

SELECT \* FROM Musteriler WHERE Sehir = ‘KONYA’

END

GO

……..

Yeni sorgu ekrani acip Stored procedure u cagiriyoruz ve bakiyoruz olmus mu diye.

EXEC MusteriListesi

Olusturdugumuz stored procedure e yani MusteriListesine sag tiklayip “Script stored procedure as” den

“Drop To” yu ordan da “New Query editor Window” u tikliyoruz. (Silecegiz simdi)

Execute ediyoruz cikan kodu.

Ve sol tarafta bakarsak silindigini gorebiliriz. Tekrar silmeye kalkarsak hata vermez cunku if else yapisi var yani dosya bulunuyorsa siliyor yoksa silmiyor.

CREATE PROCEDURE sp\_MusteriListesi 🡪 sp koyulmasi onerilir, usp de olabilir. (user)

AS

BEGIN

SELECT \* FROM Musteriler WHERE Sehir = ‘KONYA’

END

Yeni sorgu ac

EXEC xp\_logininfo 🡪 Hazir procedure C# taki collectionlar gibi. 🡪 execute et!

Olusturdugumuz stored procedure e yani MusteriListesine sag tiklayip “Script stored procedure as” den

“Create To” yu ordan da “File” i tikliyoruz. Daha sonra kaydedebiliyoruz.

Olusturdugumuz stored procedure e yani MusteriListesine sag tiklayip “Script stored procedure as” den

“Alter To” yu ordan da “File” i tikliyoruz. Daha sonra kaydedebiliyoruz.

Olusturdugumuz stored procedure e yani MusteriListesine sag tiklayip “Script stored procedure as” den

“Drop To” yu ordan da “File” i tikliyoruz. Daha sonra kaydedebiliyoruz. (farkli adlarla)

3 (STORED PROCEDURE INPUT - OUTPUT)

Yeni bir sorgu ekrani aciyoruz.

SELECT \* FROM Musteriler WHERE Sehir = ‘Ankara’

81 il icin farkli farkli prosedurler olusturmak yerine hepsi icin tek bir tane dinamik prosedur olusturacagiz

SELECT \* FROM Musteriler WHERE Sehir = @sehir 🡪 @sehir bir degisken

Herseyi kaldirdik yeni bir store procedure yazacagiz.

CREATE PROCEDURE SehireGoreMusteri

@sehir nvarchar(50) 🡪 Tablomuzdaki deger neyse onun aynisini yaziyoruz. (nvarchar)

AS

BEGIN

SELECT \* FROM Musteriler WHERE Sehir = @sehir

END 🡪 execute et control et sonra dosyalar kismindan ;) 🡪 yeni sorgu ac

DECLARE @deger int

EXEC @deger = SehireGoreMusteriler

@sehir = N’KONYA’

SELECT ‘GELEN DEGER’ = @deger 🡪 execute et gor! (return var burda geriye deger donuyor)

Soldan procedure muzun ustune sag tikla EXECUTE stored procuders e tikla Value kismina Bursa yaz OK tikla SQL server otomatik olarak bir kod yazacaktir. Tekrar execute et gor!

Herseyi sildik!

EXEC SehireGoreMusteri @sehir = ‘ANKARA’ execute et gor! Yeni sorgu ac!

CREATE PROCEDURE MusteriEkle

@adi nvarchar(50),

@soyadi nvarchar(50),

@adres nvarchar(150),

@email nvarchar(100),

@krediLimit smallmoney,

@kayitTarihi datetime

AS

BEGIN

INSERT Musteriler (Adiniz, Soyadiniz, Adres, Email, KrediLimiti, KayitTarihi)

VALUES (@adi, @soyadi, @adres, @email, @krediLimit, @kayitTarihi)

END 🡪 execute et! Procedure eklenmis mi control et! Refresh unutma! Intellisans da gormek icin on,off

Yeni sorgu ac

EXEC MusteriEkle ‘Toprak Su’, ‘ISIK’, ‘Gazi Osman Pasa Mh.’, ‘toprak@su.com’, 25000, ‘01/01/2011’

--- STORED PROCEDURE LER HER ZAMAN GERIYE INTEGER DONDURUR HER ZAMAN ---

SELECT \* FROM Musteriler WHERE Adiniz = ‘Toprak Su’ 🡪 execute et ve gor degeri.

Soldan bizim stored procedurumuze tiklayip modify diyoruz. (ALTER edecez yani)

Cikan sayfada @musteriID int OUTPUT, degiskenini ekle.

BEGIN kisminin altindada VALUES den sonra

SELECT @musteriID = @@IDENTITY; 🡪 numeric degree sahip global bir degisken

Execute et ve yeni bir sorgu ac

DECLARE @Output AS int;

EXEC MusteriEkle @Output output, ‘FATIH’, ‘GUNES’, ‘Gazi Osman Pasa Mh.’, ‘toprak@su.com’, 25000, ‘01/01/2011’

SELECT ‘Eklenen Numara’ = @Output; 🡪 execute et burdan oncesini secip.

Eklenen Numara 🡪 Yeni bir sutun oluyor.

SELECT \* FROM Musteriler WHERE MusteriID = 1013 🡪 sadece bunu execute et!

SELECT \* FROM Musteriler WHERE Adiniz = ‘Toprak Su’

4 (Express ve Compact arasindaki farklar)

* Mobil uygulamalarda Compact Edition (Local Database) kullanilir. Cunku boyut olarak kucuktur ve islemcisi masaustu bilgisayardan dusuk olan Mobile, Tablet, pc. Icin daha verimli olacaktir.
* Yazilan uygulama sirket, kafe tarzi ag kurulu bilgisayarlarda kullanilacak ise Express Edition kullanilir cunku compact edition bu tarz uygulamalarda yetersiz kalacaktir. (Boyutlar yuksek)

Ikisi arasindaki farklari tablolar halinde anlatiyor yazamadim cok fazla!

5 (.Net te veritabanindan veri cekme)

Uygulamalarda guncelleme en kritik meseledir cunku her sene yeni teknoloji ortaya cikiyor.

(Eskisini oneriyor video da eger eskisinde rahat yaziyorsan)

Veri tabanlari, sirket uygulamalarinda daha detayli kullanilir.

Eger program ile veritabani baglantili olmazsa herkes veritabani dilini bilmek zorunda kalacakti ve cok ugrastirici ve maliyetli olacakti.

Eger baglanti olmazsa sirketteki herkes veritabanina ulasabilecek sirket gizliligi ve is bolumu ortadan kalkacaktir.

Aradaki baglantiyi ADO.NET ile veya LINQ ile yapabiliriz.

Entity framework veritabani iliski modelidir. OOP icerisinde LINQ olarak bulunmaktadir. Sorgu sistemi ADO.NET e gore daha farkli.

Kod yazmadan Databindin ile sorgular gonderilir ve veriler cagrilabilir.

Entity framework ile LINQ sayesinde veritabanindan veriler alinabilir.

6 (ADO.NET ile veritabanina baglama)

Visual studio yu aciyoruz ve yeni bir proje aciyoruz. (Console Application)

Solution explorer da projeye sag tiklayip yeni bir item ekliyoruz. (Local Database)

Cikan pencerede cancel de kapat.

Veritabaninda create table tikliyoruz ve yeni bir tablo ekleyecegiz. (Musteriler Tablosu)

MusteriID int primary key idendity true

Adi nvarchar

Soyadi nvarchar

Sehir nvarchar

Daha sonra tablomuza sag tiklayip show table data ya tikliyoruz. Simdi bazi musteriler ekleyecegiz. (2)

Simdi projemize geciyoruz. C# uygulamamiza. References kismina sag tikla ve add reference diyoruz.

System.Xml DLL dosyasini kaldirabiliriz biz kendi DLL dosyamizi eklemek icin cunku bunu kullanmiyoruz su an isimize yaramiyor ve yer sinirli yer kapliyor yer acmak icin silebiliriz. (- eklemeyecez)

.NET icerisinde System.Data.SqlServer.Ce yi ekliyoruz.

Using System.Data.SqlServerCe; using blogunu projemize ekliyoruz.

Simdi main kismimiza kod yaziyoruz.

SqlCeConnection myCon = new SqlCeConnection(“Data Source = Database1.sdf”); (Bu bir Class) Constructor iki kere overload edilmis. Bos da birakabiliriz veya baglanmak istedigimiz veritabaninin ismini de girebiliriz.

// myCon.ConnectionString = “Data Source = Database1.sdf”; // boyle de girebilirdik bos birakip

myCon.Open();

// Adim 1: Veritabanina baglandik ve baglantiyi actik.

// Adim 2 : Database sorgu gonder

SqlCeCommand myCom = new SqlCeCommand();

myCom = myCon.CreateCommand();

myCom.CommandText = “Musteriler”; 🡪 Tabloya baglanacaz.

myCom.CommandType = System.Data.CommandType.TableDirect;

// Adim 3 : Veri Oku ve Yazdir

SqlCeDataReader myDataReader = myCom.ExecuteReader(); 🡪 Musteriler tablosu myDataReader nesnesinin icinde artik.

While(myDataReader.Read())

(

Console.WriteLine(myDataReader[1] + “ “ + myDataReader[2]); Musteriler[0] = MusteriID sutun

)

// Adim 4 : Baglantiyi kapat

myCon.Close();

// Musteriler[1] ve [2] ile veritabanindan sadece ad ve soyad sutunlarini okuduk ve yazdirdik calistir gor.

7 (ADO.NET ile veritabanina baglama)

Visual studio yu aciyoruz. Yeni proje baslatiyoruz. (Console)

Projeye yeni item ekliyoruz. (Service – based Database) Cancel’a tikla acilinca

Yeni bir tablo ekliyoruz. (Musteriler)

MusteriID int primary key ve daha sonra identify yi turu yapiyoruz.

Adin varchar(50)

Soyadi nvarchar(50)

Sehir nvarchar(50)

Tabloya sag tikla show table data ya tikla very gir. (2 tane)

Programimiza giriyoruz.

// ADIM 1 : Baglanti islemini gerceklestir

Using System.Data.SqlClient i using bloguna ekliyoruz.

SqlConncection cnn = ne SqlConnection(); 🡪 2 kere overload edilmis yine biz alt satirda girecez.

String cnnStr = “Data Source = .\\SQLEXPRESS; AttachDbFileName = |DataDirectory|\Database1.mdf; integrated security = True; User Instance = True;”;

🡪 noktanin yerine ip adresi de yazilabilir. ;)

Cift \\ kacis karakteri oldugu icin falan kullandik. Istersek stringin basina @ koyabiliriz hata cikmaz.

. yerine IP adresi girersek o IP adresindeki veritabanina baglaniriz.

Cnn.ConnectionString = cnnStr

Cnn.open();

//ADIM 2 : Sorgu gonder

SqlCommand com = cnn.CreateCommand();

Com.CommandText = “Select \* from Musteriler where MusteriID = 1”;

Com.CommandType = System.Data.CommandType.Text; 🡪 TableDirect Hata aldik.

//ADIM 3 : Veri oku

SqlDataReader oku = com.ExecuteReader();

ADIM 4 : Ekrana yazdir

While(oku.Read())

(

Console.WriteLine(oku[1] + “ “ + oku[2]);

)

ADIM 5 : KAPAT

Cnn.Close();

Yeni bir stored procedure ekliyoruz. (MusteriOku) 🡪 Object explorer’ dan

CREATE PROCEDURE dbo.MusteriOku

AS

SELECT \* FROM Musteriler

RETURN

Daha sonra projemize donup Adim 2 nin ilk adimi haric yorum satiri haline getirip yeni sorgu olusturacaz.

Com.CommandText = “MusteriOku”;

Com.CommandType = System.Data.CommandType.StoredProcedure; calistir gor gerceklesiyor.

8 (Baglanti kapatildiktan sonra ADO.NET 2 ile verileri yonetme)

Yeni bir proje baslatiyoruz. (Visual Studio) (Console)

Projemizin icine exist item kismindan var olan bir veritabanini ekliyoruz. (Database1.mdf)

// 1. Uygulama

Using System.Data.SqlClient; (ekliyoruz)

SqlConnection myConnection = new SqlConnection(“Data Source = .\\Sqlexpress; AttachDbFileName = |DatDirectory|\Database1.mdf; Integradet Security = True; User Instance = True;”);

myConnection.Open();

SqlCommand myCommand = myConnection.CreateCommand();

myCommand.CommandText = “Select \* From Musteriler”;

myCommand.CommandType = System.Data.CommandType.Text;

SqlDataAdapter myDataAdapter = new SqlDataAdapter(myCommand); 🡪 Datareader yerine bunu kullanacaz.

Using System.Data; (ekle)

DataSet myDataSet = new DataSet();

myDataAdapter.Fill(myDataSet);

myConnection.Close();

Bu veriler artik RAM de tutulmaktadir.

Foreach(DataROw myRow in myDataSet.Tables[0].Rows) 🡪 Tables[0], Musteriler tablosuna isaret eder.

(

Console.WriteLine(myRow[0] + “ “ + myRow[“Adi”] + “ “ + myRow[“Soyadi”]); 🡪 adi soyadi yerine 1 ve 2 yazabilirdik. Hatirla daha once yaptik.

)

Calistir ve gor.

Hepsini secip ctrl + ks tusuna bastigimizda bir region olusacak ve kapatip acabilecez rahatlikla. Blok gibi dusun.

// 2. Uygulama (Region olustur)

String cnnStr = @”(“Data Source = .\Sqlexpress; AttachDbFileName = |DatDirectory|\Database1.mdf; Integradet Security = True; User Instance = True;” ;

SqlConnection myConnection = new SqlConnection(cnnStr);

Veritabanina sag tiklayip object explorer kisminda properties diyoruz. En yukarida veritabanin tam konumu yaziyor ve altta da baglanti stringi yaziyor. Onu kopyalayip tekrar girebiliriz ismi.

myConnection.Open();

SqlDataAdapter myAdapter = new SqlDataAdapter();

// Select sorgusu olustur

myAdapter.SelectCommand = myConnection.CreateCommand();

myAdapter.SelectCommand.CommandText = “Select \* From Musteriler”;

// Ekleme islemi insert into

myAdapter.InsertCommand = myConnection.CreateCommand();

myAdapter.InsertCommand.CommandText = “Insert Into Musteriler (Adi, Soyadi, Sehir) values (@Adi, @Soyadi, @Sehir)

[”;](mailto:\”;myAdapter.InsertCommand.Paramaters.Add(\“@Adi)

[myAdapter.InsertCommand.Paramaters.Add(“@Adi](mailto:\”;myAdapter.InsertCommand.Paramaters.Add(\“@Adi)”, SqlDbType.NVarChar, 50, “Adi”);

[myAdapter.InsertCommand.Paramaters.Add(“@Soyadi](mailto:\”;myAdapter.InsertCommand.Paramaters.Add(\“@Adi)”, SqlDbType.NVarChar, 50, “Soyadi”);

[myAdapter.InsertCommand.Paramaters.Add(“@Sehir](mailto:\”;myAdapter.InsertCommand.Paramaters.Add(\“@Adi)”, SqlDbType.NVarChar, 50, “Sehir”);

// Guncelleme islemleri UPDATE

myAdapter.UpdateCommand = myConnection.CreateCommand();

myAdapter.UpdateCommand.CommandText = “UPDATE Musteriler SET Adi = @Adi, Soyadi = @Soyadi, Sehir = @Sehir WHERE MusteriID = @MusteriID”;

[myAdapter.UpdateCommand.Paramaters.Add(“@MusteriID](mailto:\”;myAdapter.InsertCommand.Paramaters.Add(\“@Adi)”, SqlDbType.int, 4, “MusteriID”);

[myAdapter.UpdateCommand.Paramaters.Add(“@Adi](mailto:\”;myAdapter.UpdateCommand.Paramaters.Add(\“@Adi)”, SqlDbType.NVarChar, 50, “Adi”);

[myAdapter.UpdateCommand.Paramaters.Add(“@Soyadi](mailto:\”;myAdapter.InsertCommand.Paramaters.Add(\“@Adi)”, SqlDbType.NVarChar, 50, “Soyadi”);

[myAdapter.UpdateCommand.Paramaters.Add(“@Sehir](mailto:\”;myAdapter.InsertCommand.Paramaters.Add(\“@Adi)”, SqlDbType.NVarChar, 50, “Sehir”);

// SIlme islemleri DELETE

myAdapter.DeleteCommand = myConnection.CreateCommand();

myAdapter.DeleteCommand.CommandText = “Delete from Musteriler where MusteriID = @MusteriID”;

[myAdapter.DeleteCommand.Paramaters.Add(“@MusteriID](mailto:\”;myAdapter.InsertCommand.Paramaters.Add(\“@Adi)”, SqlDbType.Int, 4, “MusteriID”);

// Veri Depolama islemleri

DataSet myDataSet = new DataSet();

myAdapter.Fill(myDataSet);

// Baglanti kapatma islemleri

myConnection.Close(); 🡪 Artik disconnect durumdayiz.

Console.WriteLine(“Veri Tabanindaki Tablolarin Icerisindeki Veriler 🡪 Once”);

Foreach(DataRow item in myDataSet.Tables[0].Rows)

(

Console.WriteLine(item[0] + “ “ + item[1] + “ “ + item[2] + “ “ + item[3])

)

Console.WriteLine(“Eski Verileri Alma islemleri tamamlandi…”);

Console.WriteLine();

// Veri Ekleme

Console.WriteLine(“Veri ekleme basladi…”);

DataRow myRow = myDataSet.Tables[0].NewRow();

myRow[“Adi”] = “Ahmet”;

myRow[“Soyadi”] = “KOYUNCU”;

myRow[3] = “ISPARTA”;

myDataSet.Tables[0].Rows.Add(myRow);

Console.WriteLine(“Ekleme islemi bitti…”);

//Silme islemi

Console.WriteLine(“Silme islemi basladi…”);

myDataSet.Tables[0].Rows[1].Delete();

Console.WriteLine(“Silme islemi tamamlandi…”);

Console.WriteLine();

//Guncelleme islemi

Console.WriteLine(“Guncelleme islemi basladi…”);

myDataSet.Tables[0].Rows[1][“Adi”] = “Ramazan”;

Console.WriteLine(“Guncelleme islemi tamamlandi…”);

Console.WriteLine();

// EN SON ISLEM TABLOYA KAYDET VE TABLOYU GORUNTULE

myAdapter.Update(myDataSet);

Console.WriteLine(“Islemler Tamam Tablonun Son hali….”);

Foeach(DataRow item in myDataSet.Tables[0].Rows)

(

Console.WriteLine(item[0] + “ “ + item[1] + “ “ + item[2] + “ “ + item[3]);

)

9 (DataBinding ve DataSet)

Visual studio yu aciyoruz yeni bir proje aciyoruz. (Console Application)

Yine projeye bir veritabani ekliyoruz. Dataset secip next diyoruz. Tum tablolari seciyoruz. Tum sutunlari.

Sonra finish diyoruz. (MusterilerDataSet ismini de giriyoruz)

MusterilerDataSet.xsd ye cift tikliyoruz. Ayni yere sag tiklayip open with diyoruz. XML text editor tiklaip Ok butonuna tikliyoruz.

Sagda Designer.cs ye tikla incele oncekileri de incele.

Baglanti saglamak icin extra connection degerine gerek duyulmaz bu yontemle.

App.config ac incele connection string orda yaziyor mesela.

Simdi projenin main kismindayiz.

MusterilerDataSetTableAdapters.MusterilerTableAdapter musteriAdapter = new MusterilerDataSetTableAdapters.MusterilerTableAdapter();

MusterilerDataSet mds = new MusterilerDataSet();

musteriAdapter.Fill(mds.Musteriler); 🡪 Musteriler tablosu dolduruldu.

Console.WriteLine(“Tablonun Olagan Hali….”);

Foreach(MusterilerDataSet.MusterilerRow musteri in mds.Musteriler.Rows)

(

Console.WriteLine(musteri.Adi + “ “ + musteri.Soyadi);

)

Console.WriteLine(“Islem Tamamlandi”);

Console.WriteLine();

Console.WriteLine(“Veri ekleme islemleri basliyor”);

MusterilerDataSet.MusterilerRow yeniMusteri = mds.Musteriler.NewMusterilerRow();

yeniMusteri.Adi = “Ahmet”;

yeniMusteri.Soyadi = “Koyuncu”;

yeniMusteri.Sehir = “Erzincan”;

mds.Musteriler.AddMusterilerRow(yeniMusteri);

Foreach(MusterilerDataSet.MusterilerRow musteri in mds.Musteriler.Rows)

(

Console.WriteLine(musteri.Adi + “ “ + musteri.Soyadi);

)

Console.WriteLine(“Ekleme islemi tamamlandi”);

Console.WriteLine(“Guncelleme islemi basliyor”);

Mds.Musteriler[0].Adi = “Mucahit”;

Foreach(MusterilerDataSet.MusterilerRow musteri in mds.Musteriler.Rows)

(

Console.WriteLine(musteri.Adi + “ “ + musteri.Soyadi);

)

Console.WriteLine(“Guncelleme islemi bitti”);

Console.WriteLine(“Silme islemine baslaniyor”);

Mds.Musteriler[1].Delete();

Console.WriteLine(“Silme islemi tamamlandi”);

musteriAdapter.Update(mds);

Console.WriteLine(“Ekleme, slime, guncelleme islemi, tamamlandi”);

Console.WriteLine(“Tablonun son hali”);

Foreach(MusterilerDataSet.MusterilerRow musteri in mds.Musteriler.Rows)

(

Console.WriteLine(musteri.Adi + “ “ + musteri.Soyadi);

)

Bunu kaydedip yeni bir proje aciyoruz. (simdi databinding i gorecez) (Form application)

Projemizin icine yine olan bir tane veritabani ekliyoruz. Local ;) Dataset next diyoruz Tables tikladik son.

DataGridView toolbaxtan penceremize ekliyoruz. Datasource = Musteriler olacak.

Hic birsey yapmayip calistirsak bile tablodaki degerler ekranda gozukecek calistir gor.

Direkt calistirdigin program icerisinden degisiklik yapsak bile kapatip actigimizda degismedigini goruyoruz cunku update etmedik.

Button ekliyoruz bir tane cift tiklayip buttonun icerigini yazacagiz.(buttonun ismi = KAYDET)

Form a sag tiklayip properties leri aciyoruz yukarida event penceresi var ona tiklayinca eventler gozukuyor. Dolu olan kisma cift tikla Load falan yaziyor.

Butonun icini yaziyoruz.

This.musterilerTableAdapter.Update(this.database1DataSet);

Calistiriyoruz programi ve form ekrani uzerinden bir degerleri degistirdigimizde kaydet butonuna tikliyoruz. Ve kapatip actigimizda goruyoruz ki degisme basarili sekilde gerceklesti. Kaydete basmazsak olmaz ! ! !

Formun Eventlerina giriyoruz. RowValidated eventini cift tiklayip calistiriyoruz ve icin dolduruyoruz.

This.musterilerTableAdapter.Update(this.database1DataSet)

Tekrar calistiriyoruz ve artik butona basmaya gerek kalmadan sadece degistirdigimizde degistigini kapatip actigimizda gorebiliriz.

10 (LINQ ile Verilere Erismek)

Visual studio da yeni bir proje aciyoruz. (Console Application)

Yine veritabanini ekliyoruz projemize. Entity Data Model diyip Next diyoruz.

Daha sonra generate from database diyoruz next diyoruz. 🡪 Next 🡪 Tables secip 🡪 finish

Model1.edmx e bakabilirsin. Sag tikla properties a gir ismi = Database1Entities

Kod yazmaya baskliyoruz projemize.

Database1Entities dbe = new Database1Entities();

Var musteriler = from m in dbe.Musteriler

Where m.MusteriID == 1

Select m;

// select \* from Musteriler m where m.MusteriID == 1; bununla ayni yukaridaki

Console.WriteLine(“Adi..:” + musteriler.First().Adi);

Calistir Kodu burda. MusteriID == 1 kismini 2 yap yine calistir

Yeni bir tablo olusturuyoruz. (Satislar)

SatisID int Unique Primary Identity

SatisTutari money 19

SatisTarihi datetime 8

MusteriID int 4 🡪 iliski kurmak icin yazdik.

Satislar tablosuna sag tikla ve properties kismina gir Add relation kismindan ekliyoruz. Isim = FK\_Musteriler\_Satislar no, no, musteriler, satislar, musteriID, musteriID

Add Relation diyip OK diyoruz.

Satislar Tablosuna sag tikliyoruz Show table data kismina tikliyoruz.

Deger giriyoruz. (5 adet) (musteriID = 1 icin 3 , 2 icin 2 kayit)’

Kodumuza geri donuyoruz.

Solution Explorerdan modeli siliyoruz sonra yukardan data kismina giriyoruz ve Add new data sourc kismina giriyoruz. Database tikliyoruz Entity Data Model Generate database next finish.

Modele girdiginde 2 tabloyu ve iliskilerini gorebilirsin.

Database1Entities1 dbe = new Database1Entities1();

Var musteriler = from m in dbe.Musteriler

Where m.Satislar.Any(s => s.SatisTutari > 275)

Select m;

Foreach(var item in musteriler)

(

Console.WriteLine(item.Adi + “ Satis Adedi…: “ + item.Satislar.Count );

Foreach(var satis in item.Satislar)

(

Console.WriteLine(satis.SatisTarihi.ToShortDateString() + “ – “ satis.SatisTutari);

)

)

Calistir kodu gor !