**T-SQL  ile  İlişkisel Veritabanı Oluşturmak**

Merhaba arkadaşlar bu yazımda sizlere **MS SQL Server** ‘de  **T-SQL** veri sorgulama dilini kullanarak ilişkisel veritabanı nasıl hazırlanır anlatmaya çalışacağım.

**Peki Nasıl Bir Veritabanı Hazırlayacağız?**

Hazırlayacağımız veritabanı Türkiye’de ki üniversitelerin isimlerini ve her üniversiteye bağlı olan fakülte ve bölüm adlarını tutan basit bir veritabanı, eğer isterseniz örnekten sonra veritabanını geliştirip bir programlama dili vasıtasıyla bir uygulama yazabilirsiniz.

İlk iş olarak **MS SQL Server 2008 Management Studio** açıyoruz. Ardından, boş bir sorgu sayfası açıyoruz. Sol üst köşedeki **“New Query”**tuşuna tıklayarak açabilirsiniz. Ardından hemen veritabanımızı oluşturalım. Aşağıdaki kod satırı ile veritabanımız oluşturulacaktır.

CREATE DATABASE UnivesitelerDB

GO

Daha sonra da tablolarımızı oluşturan kodları yazmaya başlayalım. Boş bir sorgu sayfası açıp aşağıdaki kodları yazıyoruz.

USE UnivesitelerDB

GO

CREATE TABLE Universiteler

(

    UniversiteID intidentity(1,1) not null,

    UniversiteAdi nvarchar(200) not null,

    Primary key(UniversiteID)

)

CREATE TABLE Fakulteler

(

    FakulteID intidentity(1,1) not null,

    FakulteAdi nvarchar(200) not null,

    Primary key(FakulteID)

)

CREATE TABLE Bolumler

(

    BolumID intidentity(1,1) not null,

    BolumAdi nvarchar(200) not null,

    Primary key(BolumID)

)

CREATE TABLE UniversiteFakulte

(

    UniFakulID intidentity(1,1) not null,

    UniversiteID intnot null,

    FakulteID intnot null,

    Primary key(UniFakulID),

    Foreign key(UniversiteID) referencesUniversiteler,

    Foreign key(FakulteID) referencesFakulteler

)

CREATE TABLE UniversiteFakulteBolum

(

    UFBID int identity(1,1) not null,

    UniFakulID intnot null,

    BolumID intnot null,

    Primary key(UFBID),

    Foreign key(UniFakulID) referencesUniversiteFakulte,

    Foreign key(BolumID) references Bolumler

)

Şimdi de yazdığımız kodları açıklayalım. Yukarıdan aşağıya doğru İlk üç tablomuz  arasında ilişki yok normal şekilde **primary key**’i olan ve belirli sütunları olan tablolar. Biliyorsunuz ki **“ identity(1,1) “**şeklinde bir bildirim indisin birden başlayıp birer birer artacağını söylüyordu. **“Primary key(UniversiteID)”**  gibi bildirim ise **“UniversiteID”** sütunun primary key özelliğine sahip olacağını belirtiyor.

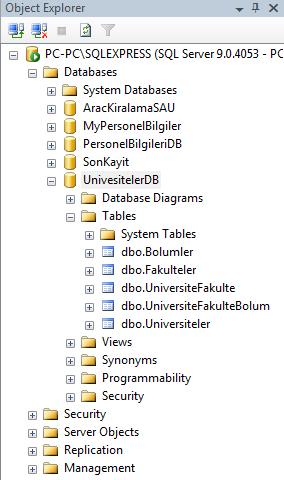
Yaptığımız tablolar arasında ilişki kurmaya çalıştığımızda görülüyor ki bazı tablolar arasında çoka-çok ilişki var. Ne demek çoka çok onu açıklayalım. Şimdi   **“Universiteler”** tablosunda  **A üniversitesi** olsun.Bu üniversite **“Fakulteler”**tablosundan  **Mühendislik,Tıp ve İktisadi İdari Bilimler**  fakultelerine sahip olsun. Fakat aynı şekilde **B üniversitesi**de bu fakültelere sahip olabilir. Tam tersinden baktığımızda bir Mühendislik Fakültesinin birden çok üniversite de  olma ihtimali de var. O halde bu iki tablo (Varlık) arasında çoka-çok ilişki vardır. Bu sebeple araya yeni bir tablo çıkar. Unutmayalım arkadaşlar eğer ki çoka-çok ilişkili bir tablo olursa ek olarak bir tablo daha meydana gelir. Yeni tablomuzun adını  **“UniversiteFakulte”**olarak belirledik. Bu tablo hem **"Universiteler"** tablosundan hem de **"Fakulteler"** tablosundan iki adet **Foreign key**  (Yabancıl Anahtar)  içerir.

Tamam üniversiteler ve fakülteler arasındaki yapıyı bitirdik. Şimdi ise bu yapıya bölümleri ekleyeceğiz.

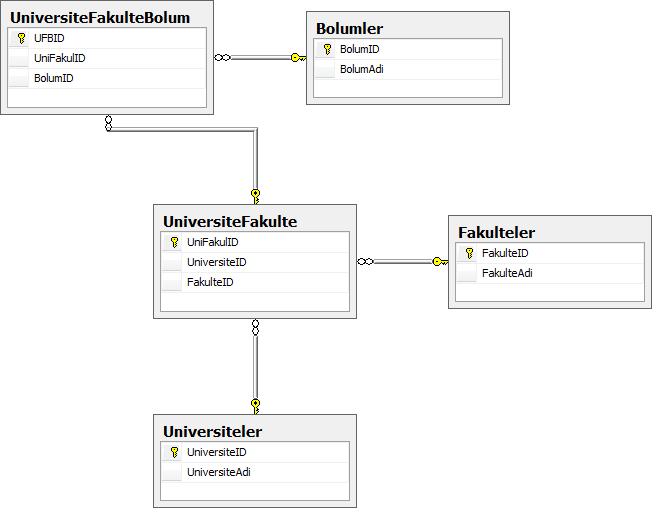
Bir bölüm **"UniversiteFakulteler"**  tablosunda birden çok yere ID verebilir mi? Evet verebilir. Yani **Endüstri Mühendisliği** bölümü hem  **Sakarya Üniversitesi’**nde, hem de**EgeÜniversitesi**’nde  olabilir. O halde**"UniversiteFakulte"** tablosuyla **"Bolumler"** tablosu arasında da çoka-çok ilişki vardır. Napıyoruz? Yeni bir tablo daha araya ekliyoruz.

Bir detaya değineyim. Dikkat ederseniz Foreign key komutunu içeren tabloları en son yazdım. Çünkü **Foreign key** komutunu kullanmadan önce ilişki kurduğum tablonun yukarda oluşturulmuş olması gerekir. Yoksa böyle bir tablo yok diyerek hata verecektir. Komutları satır satır işletir. İlk önce ilk üç tabloyu oluşturur, ardından diğerlerine geçer. Bu yüzden ilişki içeren tabloları her zaman en sona yazınız.

Foreign key anahtar sözcüğü de iki tablo arasında ilişki kurar. Ardından **“Execute”** komutuna basarak, sorguyu çalıştırın sol menüde veritabanı isminizin altında **“Tables”** klasörü aşağıdaki resimde ki gibi olmalı.



İlişkisel tablolarımızın diyagramı aşağıdaki gibidir.



Gördüğünüz gibi tablolarımız geldi. Eğer göremezseniz sağ tıklayıp **“Refresh”** deyin gelecektir.