**Veri Tipleri**

SQL Server, verileri çeşitli türleri ile saklamak için, veritabanı programcılarına ve kullanıcılarına çeşitli veri tipleri sunmaktadır. Bu veriler metinsel, tarihsel, sayısal ve parasal tipte olacağı gibi, görüntü, XML tabanlı ve kullanıcı tarafından oluşturulmuş(UDDT) tipte de olabilmektedir. Bu yazıda yukarıda sayılan SQL Server tarafından tanımlanmış veri tipleri anlatılacaktır.

**1)Kesin Sayısal Tipler**

**bit**: 1 **bayt** yer tutar. Alacağı değer 1 ya da 0 ‘dır. Yani true ya da false. C# ‘ta ki **bool** veri tipine karşılık gelir.

**(Eğitmen için) tinyint**: 1 bayt yer tutar. 0 ile 255 arasında ki tam sayıları tutar. C# ‘ta ki **byte** veri tipine karşılık gelir.

**(Eğitmen için) smallint**: 2 bayt yer tutar. -32.768 ile 32.767 arasında ki tam sayıları tutar. C# ‘ ta ki **short**tipine karşılık gelir.

**int** : 4 bayt yer tutar. -2.147.483.648 ile 2.147.483.647 arasında ki tam sayıları tutar. C# ‘ta ki kullanımında değişkenin adı gene aynıdır.

**(Eğitmen için) bigint**: 8 bayt yer tutar. -9.223.372.036.854.775.808 ile 9.223.372.036.854.775.807 arasında ki tam sayıları tutar. C# ‘ ta ki **long** tipine karşılık gelir.

**2) Yaklaşık Sayısal Değişkenleri ( Approximate Numerics )**

**(Eğitmen için) float**: ihtiyacınız olan alana göre bayt olarak kendi değer alır. Yani kaplayacağı alanı siz belirlersiniz. Kullanımı **: float(N)** burada ki N size ihtiyacınız olacak bayt boyutunu belli eder. float(5) derseniz 5 baytlık bir alan kaplar. Bu değişken içine aldığı sayıyı otomatik olarak yuvarlar böylece yaklaşık bir sayısal veri tutmuş olur. C# ‘ta ki kullanımı **double** ‘ dir.

**(Eğitmen için) real** **: 4 bayt yer tutar. Aynı zamanda float(4) ile aynı değerleri tutabilir.** **Dolayısıyla bazı yerlerde float olarak tanımlanması bellekte boş yer tutulmasının önüne geçebilir.** 3.40E+38 ile -1.18E-38, 0 ve 1.18E-38 ile 3.40E+38 arası ondalık sayıları tutabilirsiniz. C# ‘ta ki eşdeğeri**Single** ‘dır.

**Decimal(x,y):** Boyutu 5 ile 17 byte arasında değişir. Ondalıklı sayıları tutmaya yarar. İki parametre alır. İlk parametre (x) sayının kaç karakter uzunluğunda olacağını, ikinci parametre (y) ise bu karakterlerin kaçının ondalık kısım olduğunu ifade eder. Örneğin decimal(4,2) şeklinde tanımlanan bir değişken 12,34 değeri alabilir. (Money ve decimal için ondalik hane sayisi sabit)

**(Eğitmen için) Numeric(x,y):** Kullanımı ve özellikleri decimal ile aynıdır.

**3) Karakter Veri Tipleri**

**char : char(n) şeklinde kullanılır. N baytı temsil eder ve ihtiyacımıza göre genişletebiliriz. 1 ile 8000 arasında ki değerleri tutar.**

**varchar : char ile aynı yapıdadır. varchar(MAX) şeklinde tanımlanırsa maksimum 8000 karakter değil maksimum değer olan 2,147,483,647 karakter veri girilebilir.**

**(Eğitmen için) text** : Önceki versiyonları desteklemek amaçlı bulunan bir veri tipidir. varchar(MAX) ile aynı görevdedir. Uzun yazılarınızı bu veri tipinde barındırabilirsiniz.

**Unicode Karakter Değişkenleri ( Unicode Character Strings )**

**nchar** : char ile aynı yapıdadır. Tek farkı ise , char’ın aksine UniCode karakterleri dediğimiz karakteri de üzerinde barındırabilmesidir. (Unicode Karakterler) . 1-4000 arası değer tutar.

**nvarchar** : varchar ile kullanımı aynıdır. Tek farkı UniCode karakterleri dediğimiz karakteri de üzerinde barındırabilmesidir. (Unicode Karakterler) . 1-4000 arası değer tutar. **(**Unicode - **Her karakter için benzersiz bir numara kullanılarak platformlar arası karmaşalara çözüm getirildi. )**

Çeşitli yazılımlarda veya programlama yaparken bazen sayfa Türkçe dilini doğrudan desteklemiyorsa kullanmak istediğiniz karakterin Unicode değerini kullanarak karakteri meydana getirebilirsiniz. Bu açıdan bakarsak Unicode Türkçe karakterler tablosu her zaman size lazım olabilir.

“ç” harfinin unicode karşılığı “u00e7”, “Ç” harfinin unicode karşılığı “u00[c](https://wmaraci.com/nedir/cplusplus)7”,

“ğ” harfinin unicode karşılığı “u011f”, “Ğ” harfinin unicode karşılığı “u011e”,

“ı” harfinin unicode karşılığı “u0131”, “İ” harfinin unicode karşılığı “u0130”

“ö” harfinin unicode karşılığı “u00f6”, “Ö” harfinin unicode karşılığı “u00d6”

“ş” harfinin unicode karşılığı “u015f”, “Ş” harfinin unicode karşılığı “u015e”

“ü” harfinin unicode karşılığı “u00fc”, “Ü” harfinin unicode karşılığı “u00dc”

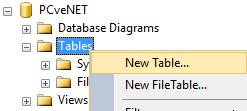
**(Eğitmen için) ntext** : Önceki versiyonları desteklemek amaçlı bulunan bir veri tipidir. nvarchar(MAX) ile aynı görevdedir.

|  |
| --- |
| **Not://** char ile varchar arasındaki fark; char(10) da 5 karakter girildiyse kalan kısmı bosluklarla doldurur. Fakat varchar(10) yazıp 5 karakterini doldurursak kalan kısmında boşluk tanımlamaz.  **Eğitmen Notu://** varchar veya char kullandığınızda her bir karakter için 1 byte yer kaplar, nchar veya nvarchar’da ise her bir karakter için 2 byte yer kaplar. Char ve varchar ASCII veri saklar, nchar ve nvarchar ise Unicode veri.  **4) Tarihsel(Date) Veri Tipleri**  **date** : 3 bayt yer tutar. YYYY-MM-DD yani YIL – AY – GÜN şeklinde tarihi tutar. 0001-01-01 ile 9999-12-31 arasında ki tarih bilgilerinizi sorunsuzca bu veri tipinde verebilirsiniz.  **datetime** : 8 bayt yer tutar. YYYY-MM-DD hh:mm:ss[.mmm] yani YIL – AY – GÜN SAAT:SANİYE:SALİSE.MİLİSALİSE şeklinde tarihi tutar. 1753-01-01 00:00:00.000 ile 9999-12-31 23:59:59.999 arası bir tarih değeri alır.  **datetime2** : DateTime veri tipinden tek farkı daha hassas veri tiplerini tutmasıdır. Yani bu tip ile daha detaylı veri tutarsınız. YYYY-MM-DD hh:mm:ss[.nnnnnnn]  **(Eğitmen için) time(7)** : Sadece saat bilgisini hh:mm:ss[.nnnnnnn] şeklinde tutar. 00:00:00.0000000 ile 23:59:59.9999999 arası değer alır. Dateteime2 gibi salise hassasiyeti maksimum 7 basamaktır ve kullanıcı tarafından değiştirilebilir. (3-5 byte)  **(Eğitmen için) datetimeoffset** : datetime2 ile yapısı aynıdır, tek farkı bu veri tipinin diğerinden farklı olarak **ülkelere göre tarihi tutmasıdır**. YYYY-MM-DD hh:mm:ss[. nnnnnnn] [{+|-}hh:mm] şeklinde tarihi tutar.  **(Eğitmen için) smalldatetime:** 4 byte yer kaplar. YYYY-MM-DD hh:mm:ss şeklinde saklanır. 1900-01-01 00:00:00 ile 2079-06-06 23:59:59 arası bir tarih değeri alır.  **(Eğitmen için) 5) Parasal(Money) Veri Tipleri**  Sql Server parasal türde veri saklamak için money ve smallmoney tiplerini sunmaktadır. Bu veriler, virgülden sonra 4 basamağa kadar duyarlı ondalık tipli verilerdir.  **smallmoney:** 4 byte uzunluğunda yaklaşık -214 000 ile 214 000 arasında parasal değerleri tutmak için kullanılır. Ondalık kısım duyarlılığı 4 basamaktır. 214,748.3648,yaklaşık 214 bin ile ,-214,748.3648 arası veriler için kullanılır. (Kurus kisminda 4 rakamin garanti)  **money:** 8 byte uzunluğunda yaklaşık -922 milyar ile 922 milyar arasındaki parasal değerleri tutmak için kullanılır. Ondalık kısım duyarlılığı 4 basamaktır. 99999999999999,9999,yaklaşık 99 Trilyon ile ,-99999999999999,9999 arası veriler için kullanılır. (Kurus kisminda 4 rakamin garanti)  **İkili(Binary) Veri Tipleri**  **Binary:** **Maksimum 8000 byte boyutunda, sabit uzunlukta Binary veri saklamak** için kullanılır.  **(Eğitmen için) varbinary:** **Değişken uzunlukta 8000 byte boyutunda binary değer** saklamak için kullanılır. (max) parametresiyle tutacağı maksimum değer 2^31 byte’a kadar yükseltilebilir.  **(Eğitmen için) image:** Önceki sql versiyonlarını desteklemek amacıyla kullanılmaktadır. Bu tip yerine varbinary (max) veri tipi tercih edilmelidir.  **Diğer Veri Tipleri**  Sql Server özel amaçlı, herhangi bir kategoriye girmeyen, dil destekli veri tiplerini de sunmaktadır. Bunlar, bit, sql\_variant, timestamp ve uniqueidentifier veri tipleridir.  Herbirinin görevi aşağıda açıklanmaktadır.  **Uniqueidentifier**: Bir satırı tek (aynı kopyası bulunmayan) yapmak için tanımlanan 16 bitlik kimliktir**(GUID - Global Unique Identifier ). Genellikle kayıt sayısının 4 milyarı geçeceği durumlarda veya çeşitli güvenlik sebeplerinden ötürü kullanılmaktadır**.  **GUID** dünya üzerinde kim, nerede, ne zaman oluşturulsa oluştursun bir benzeri olmayan 128 Bitlik(16-Byte) bir **Unique Identifier** (Benzersiz Kimlik) ‘dır. Küresel anlamda bir benzersizlik vaad eden **GUID** (Globally Unique Identifier) **INTEL** firması tarafından bulunmuştur. **SQL Server**‘da **GUID** değeri **NEWID()** fonksiyonu ile elde edebiliyoruz.**SQL Server 2008** ile birlikte gelen ve diğer sürümlerde de desteklenen **NewSequentialID()** fonksiyonu ise bize sıralı bir **GUID** değeri vermektedir. **NewSequentialID()** fonksiyonu her çalıştırıldığında sonra **GUID** değerinden daha büyük bir **GUID** değeri vereceğini taahhüt eder.   * **Bit**: Adı üzerinde 1 bitlik veri hücresidir. Birtakım verilerin işaret bilgisini, programcılıktaki bayrak kullanım yaklaşımlarında kullanılabilecek bir veri türüdür.   **(Eğitmen için) Xml:** Sql Server XML'i doğrudan desteklemektedir. **Genel itibariyle sık güncellenmeyen tablolarda xml veri türünü kullanmak tavsiye edilir.** Boyutu 2 gigabyte a kadar olan Xml dokümanlarını tutmaya yarar.  **(Eğitmen için) geography:** Coğrafi koordinatları ve gps verilerini tutmak için kullanılır.  **(Eğitmen için) geometry:** Öklid koordinat sistemine ait verileri tutmak için kullanılır. Geometrik şekillerin en, boy, yükseklik verilerini saklar.  **(Eğitmen için) hierarchyid:** Ağaç veri modeli veya hiyerarşik olarak sınıflandırılmış verileri saklamak için kullanılır.  http://1.bp.blogspot.com/_ULXttrvlFVQ/TA4oxrb-KfI/AAAAAAAABqU/iIxj0IqZkvc/s1600/h1.JPG  **(Eğitmen için) Sql\_variant: Girilecek olan verinin tipi belli olmadığında kullanılır. Kısacası tüm türleri kapsayan genel veri türüdür**.  **NULL ve NOT NULL**  Bir alanın NULL olarak tanımlanması demek bu alana veri girilmeden yeni bir satır oluşturulabileceği anlamına gelir. Örneğin: iki isimli insanları düşünerek 2. isim diye bir yer açtık tablomuzda. Fakat her insanın 2. ismi olmak zorunda değil. O zaman 2. isim özelliğini NULL yapabiliriz.  Fakat bazı veriler var ki bunları kesinlikle NULL yapamayız. Mesela TC kimlik numarası her insanda olmak zorundadır. O yüzden TC kimlik numarası değerini kesinlikle boş bırakılamaz.  **(Eğitmen için) Kullanıcı Tanımlı(User-Defined) Veri Tipleri (Eğitmenin Bilmesi Gereken)**  Sql Server tüm yerleşik veri tiplerinin dışında, yine bu tipleri isim değiştirerek kullanıcılarına farklı şekillerde kullandırabilmektedir. Örneğin sıklıkla birincil anahtar değeri olarak kullanacağınız int veri tipi ile oluşturulmuş bir ID isimli tip, tablo tasarımlarınızı daha okunaklı kılarak, onları daha kolay yönetebilmenizi sağlayabilmektedir.  Management studio ve T-Sql ifadesi ile 2 farklı şekilde kullanıcı tanımlı tipler(user-defined types) oluşturulabilmesi mümkündür. **CREATE TYPE type\_name FROM data\_type T-SQL ifadesi ile de bir kullanıcı tanımlı tip oluşturulabilir.**  Aşağıda MS Studio ara yüzünü kullanarak örnek bir kullanıcı tanımlı tip gösterilmektedir.   /Databases/Northwind/Programmability/Types/User-defined Data Types sekmesini sağa tıklayarak,      Mouse a sağ click yaparak açılan menüden New User-Defined Data Type seçiniz.      Karşınıza çıkan ekrandan tanımlayacağınız veri tipinin isim, büyüklük gibi bilgileri girdikten sonra OK'a tıklayınız.      Örnek olarak bir tablo sütunu oluşturulurken, daha evvelden tanımladığınız bu veri tipini ismi ile kullanabileceksiniz. |

**Creating a Table (Only Designer)**

**Object Explorer üzerinden görsel olarak oluşturma:**

Tablo oluşturmak istediğimiz veri tabanını seçiyoruz ve altında bulunanları görebilmek için yanındaki + işaretine tıklıyoruz.

Database Diagrams, **Tables**, Views gibi dizinler karşımıza çıkmaktadır. Burada bizi ilgilendiren kısım Tables kısmıdır. Tüm tablolarımız bu dizin altında gösterilmektedir. Şimdi Rehber adında bir tablo oluşturalım.  
  
Tablo oluşturma işlemini de tam olarak buradan yapacağız. Şimdi Tables’ın üzerinde **Sağ Tık > New Table** diyoruz. Karşımıza veriler için başlık (sütun bilgileri) ve veri tiplerini girmemizi sağlayacak olan bir ekran geliyor.

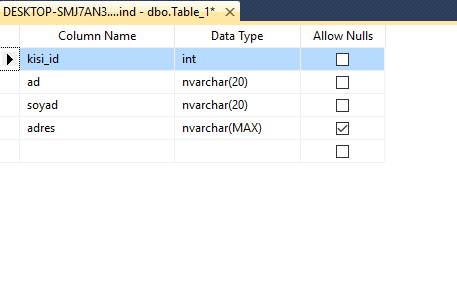
tablecreate

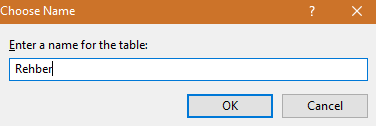
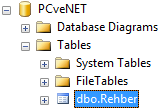
Bu ekrandaki sütunları açıklayacak olursak :

**Column Name**Buraya veri içerecek olan sütun isimleri gelecek. Örn : adsoyad, telefon\_no, adres gibi.

**Data Type :**Bu alanda Column Name alanının veri türünü belirleyeceğimiz alan. Bu alana tıkladığınızda birçok veri tipinin olduğunu göreceksiniz. Veri tipleri ile ilgili bilgi için bir önceki konu olan Tablolar ve Veri Tipleri ‘ne bakmalısınız.

**Allow Nulls:** Null olabilir anlamına gelir. null boş değer için kullanılır. Mesela bir üyelik formu oluşturdunuz ve formda email, şifre, şehir alanı var. email ve şifre zorunlu olmalıdır bu yüzden allow Nulls seçeneğini işaretlemiyoruz. Kullanıcı şehrini girmek istemezse diye bura için allow nulls seçeneğini işaretleyebiliriz.

Aşağıdaki gibi tablomun sutunlarını oluşturdum.

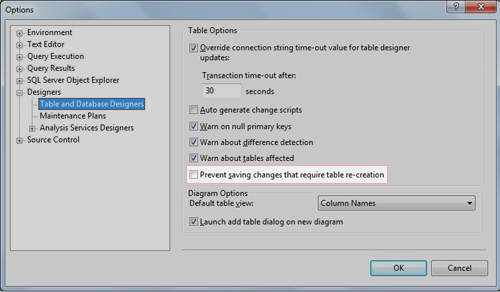
Bu bilgileri girdikten sonra Ctrl+S ile veya sekme üzerindeki x işaretine bastığınızda karşınıza **Choose Name**adında bir giriş penceresi gelecek. Burada tablonuzun ismini ne olarak kaydetmek istediğinizi belirterek OK tuşuna bastığınızda tablonuz oluşturulmuş olacak.  
  
Tablonuzu Tables altında görebilmek için Tables’ı seçerek F5 basın. Resimdeki şekilde dbo.Rehber isimli Rehber tablonuz hazırdır.  


**TABLO SİLME VE DİĞER İŞLEMLER**

Ayrıca tablo üzerinde 4 farklı işlem yapabilirsiniz. Tasarım, Seçme, Düzenleme ve Silme olarak.

**1. Tasarım (Design)**

Tablonuza Sağ Tık > Design diyerek, tablonuzu ilk oluşturmaya başladığınızda karşılaştığınız ekranı göreceksiniz. Burada sütun adları üzerinde değişiklikler yapabilirsiniz. **Ancak veri tipinde değişiklik yaparken çok dikkatli olmalısınız. Eğer tablonuz veri içeriyorsa veri tipini değiştirmeniz veri kayıplarına yol açabilir.**

Ancak burada **Saving changes is not permitted**adında bir hata ile karşılaşacaksınız. Karşılaşmadıysanız sorun yok. Bu hatayı gidermek çok basit. Hemen SQL Server Management üzerinde Tools > Options.. > Designers > Table and Database Designers sekmesinden **Prevent saving changes that require table re-creation**seçeneğini göreceksiniz. Hata aldığınıza göre bu seçenek seçili değil. Hemen bu seçeneği seçili hale getirin ve OK diyerek kaydedin. Şimdi düzenleme işlemini rahatlıkla yapabilirsiniz.

**2. Seçme ( Select Top 1000 Rows)**

Tablonuza Sağ Tık > Select Top 1000 Rows seçeneğini tıkladığınızda tablonuzdan 1000 adet veriyi listelemenizi sağlar.

**3. Düzenleme ( Edit Top 200 Rows)**

Tablonuza Sağ Tık > Edit Top 200 Rows seçeneğini tıkladığınızda tablonuzdan 200 adet veriyi listeler ve veriler üzerinde değişiklikler yapmanıza olanak sağlar.

**4. Silme (Delete)**

Bir tabloyu silmek istediğinizde ise; silmek istediğiniz tabloya Sağ Tık > Delete > OK’u seçmeniz yeterlidir.

**Not://** Edit veya Select de listelenecek olan dataların sayıları varsayılan olarak sırasıyla 200 ve 1000 dir. Bunu değiştirmek isterseniz, **Tools menüsünde Options seçeneğine** tıkladığımızda aşağıdaki pencere açılır. Bu pencerede sol taraftan **SQL Server Object Explorer** seçeneğine gelirseniz, varsayılan değerler görülür. Burada ki değerleri istediğiniz şekilde değiştirebilirsiniz.

