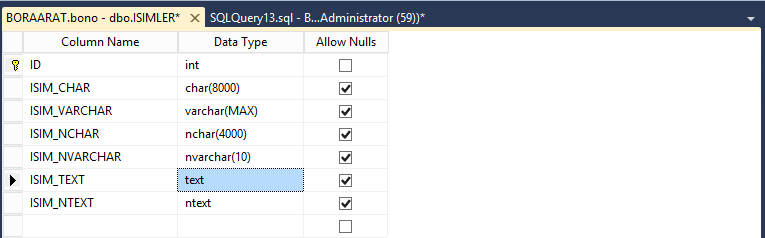
# MSSQL Veri Tipleri

## String Veri Tipleri

* **CHAR** ve **VARCHAR** arasindaki fark aslinda VAR kelimseinde gecmektedir. CHAR 10 dediğinizde biz kaç karakter girersek girelim, geri planda SQL’de bu bilgi 10 karakter olarak tutulmaktadır.  Dolayısı ile isim veri tabanı kullanacaksanız ve CHAR yaptıysanız bunu en uzun isme göre planlamalısınız. Diğer kısa karakterli olanlarda en az 10 karakterlik yer tutmaktadır, bu isim için pek doğru bir seçim olmayacaktır.
* **VARCHAR’da** ise, maksimum uzunluğu siz belirlersiniz ama arka planda sistem gerçek uzunluk kadar hafızada yer tutmaktadır.
* **NCHAR’ın** farkı ise unicode’a izin vermesi, çince, japanca vb global bir DB oluşturuyorsanız bunu nchar ya da nvarchar yapmalısınız. Aradaki fark ise, char türünde yaptığımız veri tipi her bir karakter için 1 byte’lık yer tutmaktadır, nchar ve nvarchar ise 2byte lık veri tutmaktadır.
* Yine **CHAR** tipinde karakte uzunluğu 8000 olaiblir ve bunu geçemez. NCHAR’da ise bu sayı (her karakter 2byte olduğu için) 4000 karakterdir.
* **TEXT** alanı ise 8000’den uzun bir datanın tutulması adına oluşturtulmuş bir veri tipidir. Sadece geçmiş sürümlere destek olması amacı ile bulunmaktadır. **Microsoft’un** önerisi ise **VARCHAR MAX** ya da **NVARCHAR MAX** kullanılmasıdır.

**NOT:** CHAR türündeki verilerde arama hızımız VARCHAR’a göre daha hızlıdır. Fakat diskteki yer büyüklüğü önemsenecek bir değer olduğu için CHAR türünü kullanmak bizim için dez avantaj olabilir. Peki nerelerde CHAR kullanılmalı? Sabit uzunluğu belli telefon TC kimlik gibi verilerede CHAR tipinin kullanılması, VARCHAR’a göre daha efektif olacaktır.

[](https://boraarat.com/wp-content/uploads/2018/08/mssql-string-tipleri-ve-farkl%C4%B1l%C4%B1klar%C4%B1.png)

## Sayısal Veri Tipleri

* **bit:** Genellikle ture ve false değerler için kullandığımız bir byte uzunluğunda tamsayı veri tipidir. Boyutu aklınızı karıştırmasın. Tablodaki ilk bit, bir byte büyüklüğünde yer kaplar, sonraki yedi bit aynı byte’ı kullanabilir.
* **tinyint:** 1 byte büyüklüğünde 0 ile 255 arasında değer alabilen tamsayı veri tipidir.
* **smallint:** 2 byte büyüklüğünde ve -32768 ile +32767 arasında değer alabilen tamsayı veri tipidir.
* **int:** 4 byte büyüklüğünde yaklaşık -2 milyar ile +2 milyar arasında değer alabilen tamsayı veri tipidir.
* **bigint:** 8 byte yüyüklüğünde daha büyük rakamlar yani -2^63 ile +2^63 arasında değer alabilen tamsayı veri tipidir.
* **decimal ve numeric:** Decimal ve nümeric veri tipleri adları farklı olmasına rağmen kullanımları aynıdır. Boyutu değişken olmakla birlikte -10^38 ile +10^38 arasında ondalık ve tamsayı türünde verileri saklayabilir.
* **float:** Boyutu ve doğruluğu (ondalık kısım duyarlılığı) aldığı parametreye göre değişen kayan noktalı sayılar için kullanılır.

## [Sayısal Veri Tipleri](https://boraarat.com/wp-content/uploads/2018/08/MSSQL-say%C4%B1sal-veri-tipleri.png)

Sayısal veri tiplerini yukardaki değerlere göre belirleyebilirsiniz, integer sizin için yeterli geliyorsa, big integer yaparak 2 katı bir boyutla çalışmanıza gerek bulunmuyor.

## Tarih-Zaman Veri Tipleri

* **date:** Tarihleri YYYY-MM-DD şeklinde saklayan 3 byte uzunluğunda veri tipidir. 0001-01-01 ile 9999-12-31 tarihleri arasındaki tüm değerleri tutabilir.
* **smalldatetime:** Tarih ve zaman verilerini  YYYY-MM-DD hh:mm:ss şeklinde saklayan 4 byte uzunluğunda veri tipidir. 1900-01-01 00:00:00 ile 2079-06-06 23:59:59 arasındaki tarih ve zaman değerlerini saklar.
* **datetime:** YYYY-MM-DD hh:mm:ss:mmm şeklinde tarih ve zaman verilerini tutan 8 byte uzunluğunda veri tipidir. 1753-01-01 00:00:00.000 ile 9999-12-31 23:59:59.999 arası değerlerini saklar.
* **datetime2:** Yeni gelen bir veri tipi, datetime ile arasındaki fark salise hassasiyetinin daha yüksek olmasıdır.  7 basamağa kadar salise hassasiyeti kullanıcı tarafından belirlenebilir. Kapladığı alan salise hassasiyetine göre değişir. (6-8 byte)
* **time:** Sadece saat verilerini hh:mm:ss:nnnnnnn şeklinde saklayan veri tipidir. Dateteime2 gibi salise hassasiyeti maksimum 7 basamaktır ve kullanıcı tarafından değiştirilebilir. (3-5 byte)
* **datetimeoffset:** Ülkelere göre değişen zaman farkını tutmak için kullanılır.

## Diğer Veri Tipleri

* **Xml** türünde bir veri tipimiz var temel olarak sakladığımız ve sorgulanabilir text’leri tutmamıza yarıyor.  Dolayısıyla bu artık Sadece bir text veri olmaktan çıkmış olur kendi içerisinde sorgulama yapabilir bir hala Aslında yeniden çıkar
* **Uniqueidentifier**: Dünya üzerinde tek olan bir şey üretmek istiyorsak,  örneğin bizim sahadan veri toplayan sistemlerimiz var ve bu sistemler üzerinde biz sürekli veri tabanında bir şeyler kaydediyoruz. Çakışması muhtemel verileri unique’leştirmek istiyorsak bu veri tipini kullanabiliriz.