

HATIRLATMA !

SELECT
FROM
WHERE

GROUP BY
HAVING
ORDER BY

Bölüm 3 Komutlar (Veri Tipleri):

Veriler structured, semi-structured, unstructured olmak üzere tiplerden oluşur bunu biliyoruz.

Yapısal veriler de tablolar, sütunlar ve satırlardan oluşur.

Tabloları oluşturmak için sütunlara ihtiyaç vardır; sütunlar, sütun başlığı ve verilerden (yani satırlardan) oluşur.

İşte bu tablolardaki sütunların isimlerini ve içerisindeki veri türünü belirleme işlemlerine normalizasyon denir.

Örnek olarak:

Çalışanın doğum tarihi : DATE

Çalışanın departmanı: VARCHAR

Çalışanın aylık maaşı: DECIMAL(18,2)

Tablo oluşturma:

Her tablo bir primary key sahiptir. Primary key, oluşturulan tablonun satırını belirten tekil bir alandır. Genellikle primary key ID olarak verilecekse otomatik artan olarak verilmesi önerilir.

➤ Tam Sayı Veri Tipleri:

Bir sütuna girecek verilerin türü seçilirken, sayısal olarak int seçilecekse birkaç seçenek bulunur:

bigint	Minimum: -2^{63} (-9,223,372,036,854,775,808)	8 Byte
	Maksimum: $2^{63}-1$ (9,223,372,036,854,775,807)	
int	Minimum: -2^{31} (-2,147,483,648)	4 Byte
	Maksimum: $2^{31}-1$ (2,147,483,647)	
smallint	Minimum: -2^{15} (-32,768)	2 Byte
	Maksimum: $2^{15}-1$ (32,767)	
tinyint	Minimum: 0	1 Byte
	Maksimum: 255	
bit	0 ya da 1 değerini alır.	Eğer tabloda 8 ya da daha az bit kolonu varsa 1 byte, 8'den fazla ise 2 byte yer kaplar.

DEĞERLİ BİLGİ ! Sütunların alacağı veri tipini belirlemek önemlidir çünkü depolama bilinçsizce kullanılmamalıdır, veri tiplerini bilinçli kullanma depolama alanından tasarruf sağlar ve performansı iyileştirir.

Örnek :

Bir çalışan sütununun aşağıdaki satırları bulunsun ve olması gereken veri tipi :

Primary key ID : Kaç çalışan olduğuna bağlı olarak değişir.

Country ID: Dünyada 255 ten fazla ülke olmadığına göre, tinyint

City ID: Her ülkenin 100 şehri bulursa, 32.767 geçmez, smallint

Gender: İki çeşit, kadın ve erkek, bit

➤ Ondalık Sayı Veri Tipleri:

decimal/ numeric	Minimum: $-10^{38} + 1$	Hassasiyetine göre diskte kapladığı alan değişir.
	Maksimum: $10^{38} - 1$.	1'den 9'a kadar Hassasiyet için: 5 byte
		10'dan 19'a kadar Hassasiyet için: 9 byte
		20'den 28'a kadar Hassasiyet için: 13 byte
		29'dan 38'e kadar Hassasiyet için: 17 byte
money	Minimum: -922,337,203,685,477.5808 Maksimum: 922,337,203,685,477.5807	8 Byte
smallmoney	Minimum: -214,748.3648	4 Byte
	Maksimum: 214,748.3647	
float	-1.79308 ile -2.23308, 0	7 basamağa kadar 4 Byte
	2.23308 ile 1.79308	15 basamağa kadar 8 Byte
Real	-3.438 ile -1.1838, 0	4 Byte
	1.1838 ile 3.438	

Decimal(18,0)

Float

Money

➤ Metin Veri Tipleri:

Alfanumerik yapıda verileri içerisinde barındıran tiptir.

Metin verileri bir veri tabanında genellikle daha çok yer kaplayan veri tipleridir, çünkü örnek olarak char(10) veri tipini seçtinizde 10 harf veya rakam girebileceğiniz yani 10 byte lık bir alan kaplayacağınız anlamına gelir.

char
nchar

Bu iki veri tipi uzunluğunu nasıl belirlerseniz, içerisine girilen değerin sonuna boşluk ekleyerek satırı en başta belirlenen veri uzunluğuna kadar tamamlar, bu durum eğer sütununuzun her satırında sabit veri uzunluğu tutmayacaksanız (TC kimlik no) gibi mesela, gereksizdir ve performans düşüklüğü yaratır.

Fakat sürekli kısa isimler girip, nolur nolmaz uzun isim girerim diyerek char seçmeniz bir iki uzun isim gireceksiniz diye diğer satırlarda fazla alan kullanımına sebep olur.!

varchar
nvarchar

Varchar ise içerisine girilen veri uzunluğu kadar satırda yer kaplatır, bu hem byte olarak yer kaplama durumundan tasarruf sağlar. Fakat her satıra girilen metnin uzunluğunu bilmediğiniz için SQL arama yaparken bu satırlarda önce her satırın uzunluğunu belirler ve sonra arar, bu da performans düşüklüğüne sebep olur.

O yüzden sütunu kullanım amacınızı doğru belirlemelisiniz, kullanım amacına göre faydalarını göz önünde bulundurarak maksimum verim sağlayabilirsiniz.

nchar ve nvarchar veri tiplerinin char ve varchar veri tiplerinden farkı nedir ?

Eğer uluslararası bir veri tabanı oluştuyorsanız, latin harflerindeki ifadeler dışında arapça gibi korece gibi şekilli ifadeler de kullanacaksanız nchar ve nvarchar kullanmalısınız, çünkü char ve varchar veri tipinde bunlar algılanmaz ve satırlara eklediğinizde ? ile karşılaşrsınız.

Fakat n li veri tiplerinin dezavantajı şudur ki normal bir char veya varchar için her ifade 1 byte yer kaplarken, bu n li veri tipleri için 2 byte'a tekabül eder.!

char max
varchar max

2 GB 'a kadar karakter tutabiliyorsunuz.

Genellikle kitap gibi büyük veri uzunluğuna sahip bilgileri tutmak için kullanılır.

text
ntext

Önceden kullanılan veri türleridir, günümüzde kullanılmayıp bunlar yerine varchar kullanılmaktadır. Şu anda SQL içerisinde olması önceden oluşturulmuş veri tablolarındaki bilgiyi kaybetmemek adınadır.

➤ Tarih-Saat Veri Tipleri:

Datetime veri tipi 8 byte yer kaplarken, Date ve Time olarak ayrı iki veri tipinde verileri saklamak 9 byte yer tutar. Bu yüzden hem tarih hem saat verisi tutulacak ise Datetime kullanılarak veri depolamasından tasarruf edilebilir.

date	Minimum: 0001-01-01	4 Byte
	Maksimum: 9999-12-31	
smalldate	Minimum: 1900-01-01	3 Byte
	Maksimum: 2079-06-06	
datetime	Minimum: 1753-01-01 00:00:00.000	8 Byte
	Maksimum: 9999-12-31 23:59:59.997	
datetime2	Minimum: 0001-01-01 00:00:00.000000	1-2 Hassasiyet İçin = 6 Byte
	Maksimum: 9999-12-31 23:59:59.999999	3-4 Hassasiyet İçin = 7 Byte
		5-7 Hassasiyet İçin = 8 Byte
datetimeoffset	Minimum: 0001-01-01 00:00:00.000000	1-2 Hassasiyet İçin = 8 Byte
	Maksimum: 9999-12-31 23:59:59.999999	3-4 Hassasiyet İçin = 9 Byte
	Time zone offset Aralığı: -14:00 / +14:00	5-7 Hassasiyet İçin = 10 Byte
time	Minimum: 00:00:00.000000	5 Byte(Default olarak kullanılırsa)
	Maksimum: 23:59:59.999999	

➤ Diğer Veri Tipleri:

image		Maksimum değeri: 2 ³¹ -1 (2,147,483,647) Byte
binary	0 ile 8000 arasında	Tanımlandığı değer kadar Byte. Binary(10) -> 10 Byte
varbinary	0 ile 8000 arasında	Tanımlandığı değer + 2 Byte
varbinary(MAX)	0 ile 2 147 483 647 arasında	Tanımlandığı değer + 2 Byte Maksimum değeri: 2 ³¹ -1 (2,147,483,647) Byte
sql_variant		Bazı veri tiplerinin değerlerini saklamak için kullanılır. Aşağıdakiler hariç: varchar(max), varbinary(max), nvarchar(max), xml, text, ntext, image, rowversion(timestamp), sql_variant, geography, hierarchyid, geometry, User-defined types, datetimeoffset
Xml		Xml veriler için kullanılır.
Table		Sonradan kullanım amacıyla bir sonuç kümesini saklamak için kullanılır.

Image : Artık kullanılmamaktadır, bunun yerine varbinary(MAX) kullanılmaktadır.

uniqueidentifier	GUID(global olarak tekliği garanti eder) veriyi tutar.
	select NEWID() script'ini çalıştırdığınızda aşağıdaki gibi bir GUID veri oluşturur.
	A4C5DB26-7F18-4B4F-A898-E7DE26A8446A
	Bazen veritabanlarında tekliği sağlamak için kullanılır. Ama bu amaçla kullanıldığında genelde performansı düşürür.
hierarchyid	Hiyerarşik yapılarda, hiyerarşideki pozisyonları temsil etmek için kullanılır.
geography	Dünyadaki koordinat sistemini tutar. Dünyanın eğimlerini de hesaba katarak.
geometry	Euclidean (flat) sistemi ile koordinat sistemini tutar. Sadece 2 düzlem üzerinden hesaplanır. Dünyanın eğimlerini hesaba katmaz.