

PostgreSQL Veri Türleri Nelerdir?

Diğer veri tabanlarına göre postgresql farklı türlere sahiptir. Ben en azından MySQL, MSSQL ve NoSQL tarafında MongoDB ile çalıştım, Oracle tarafında çalışmadığım için böyle söylüyorum. Belki postgresql ile aynı türlere sahip databases de mevcuttur. İsterseniz çok da uzatmadan nelere sahibiz ona bir bakalım.

Boolean

Bu veri türümüz anlaşıldığı üzere **true** ya da **false** işlemlerinde kullanılıyor. Ancak yes, no, t ve f gibi kullanımları da kabul ediyor. Bu kullanıma basit bir örnek vermek gerekiyor ancak yazı gereksiz uzayacağı için özel bir başlıkta bunu belirteceğim. Burada biraz insan diline kayma da yaşanmış oluyor. Örneğin;

```
SELECT * FROM dogruluk WHERE dogrumu = 'yes';
```

Karakter Türleri

Bu türler diğer veri tabanlarından da bildiğiniz gibi metinsel ifadeleri saklayabilen türler. Tabi kendi aralarında limitlere sahip olabildikleri gibi herhangi bir limite sahip olmayanları da bulunmakta. Bu türler

- char(n)
- varchar(n)
- text

şeklinde açıklanabilirler. Yukarıdan anladığınız gibi char ve varchar n adet karakter alabilirken text bir limite sahip değil. Bu konuya daha detaylı olarak konu başlığını içeren yazıda değineceğim. MSSQL üzerinde yer alan nvarchar merak ediliyor ise eğer varchar burada o işi görüyor ayrıca text türü de. Detaylar parmaktan sonra 😊

Sayısal Türler

Sayısal türlerde bazı sürprizler bizi bekliyor olsa da çoğunlukla bize yabancı gelmeyen tiplerle karşı karşıyayız. Burada belki en ilginç gelecek olan şey auto increment işlemlerini halledebileceğiniz farklı türlerin olması. Birisi açıkcası ben ilkel bir türüm diye bağırıyorken diğeri tam da bu işi yapmak adına var. Vereceğim türlerin range aralıklarını belirtmeyeceğim ancak byte boyutlarını belirteceğim.

- smallint – 2 byte
- integer – 4 byte
- bigint – 8 byte
- decimal ve numeric variable türünden
- real – 4 byte
- double precision – 8 byte
- smallserial – 2 byte
- serial – 4 byte
- bigserial – 8 byte

Burada daha önce görmedim ben dediğiniz direkt göze çarpanlar serial türler olabilir. Bu tipler auto increment işlemlerinde kullanılabilir. Bunları kullanmam, ben int adamıyım diyorsanız integer da kullanabilirsiniz. Ama extra bir next value olayını baştan kabul edeceksiniz.

Geçici Türler

Bu türler direkt olarak kalıcılığa sahip olmayan türlerdir. Bu türlere yabancılık çekmeyeceksiniz. Çünkü birçok veri tabanı yönetim sistemi bu türleri zaten kullanmakta. Bunlar bildiğinizi varsayarak şunlardır;

- date
- time
- timestamp
- interval

UUID Türler

Bir yazıda hatırlarsanız eğer GUID generatordan [bahsetmiştim](#). Bu türde evrensel düzeyde benzersiz tanımlayıcıların olacağı varsayılır. [RFC4122](#) türünden standartlara sahip bu kolona benzersiz değerlerin atanması gerekmektedir. Gerek postgresql olsun gerekse diğer vtys olsun tamamında yer alan uzmanlar guid / uuid öneriyor. PostgreSQL uzmanları da aynı şekilde serial türüne göre bu türün kullanımının garanti benzersizlik sunacağı görüşündeler.

Array Türler

Diğer veri tabanlarında da bu veri tabanındaki gibi uzman değilim. Ancak hiçbir veri tabanında bu türü duymadım belki de işime yaramadığı için.

PostgreSQL metinsel, sayısal vb. veri türleri için dizi şeklinde veri tutmaya imkan tanıyor.

Dizi kabul edilen kolona ekleme yapılırken ve select query çekilirken bildiğimiz programlamada kullanılan yöntem kullanılıyor. Ayrıca JSON syntax ile ekleme imkanına da sahip oluyorsunuz. Bunu gerçekten ilginç buldum. Bir kolonun sadece 1. indexini getir gibi bir şey var.

```
SELECT KullaniciAdi, Telefon[1] FROM kullanicilar
```

Tabi bunun avantajı, joinlerle bağladığınız telefon ve kullanıcı tablolarını belki ortadan kaldırabilir olursunuz.

JSON Türler

Bu türe artık pek yabancı değiliz. Çoğu büyük firma olacak şekilde tüm şirketler RESTful servislerinde artık JSON kullanıyor. Aynı şekilde postgresql için de json türünde veri saklama imkanı var. Birçok önemli veri tabanı yönetim sistemi de json veri döndürüyor hatta bazıları direkt veri türü şeklinde desteğe de sahip. PostgreSQL bu işi bir parça daha ileriye götürüp tamamen programlamaya dökmüş durumda. Iterasyondan tutun da tipe kadar.

hstore Türler

Bu türler garip bir şekilde JSON türü ile benzer gösteriyor. Kafa karıştırıcı olabilirler. Çünkü where olsun, column property mapper olsun tamamen benzer. Ancak bunlar key-value pair yani anahtar-değer türünden çalışırlar. Yine bu türlerde de enumeration işlemlerini gerçekleştirebilirsiniz.

Çok Daha Fazlası

PostgreSQL Veri Türleri içerisinde yazdıklarımdan çok daha fazla veri türü barınmakta. Bunların hepsi tek bir yazının konusu olacak cinsten şeyler değil. Çünkü mac adresinden, inetine, bitine kadar burada. Ayrıca geometri de unutulmamış durumda.

Bu yazıda PostgreSQL Veri Türleri için ufak bilgiler görmüş olduk. Sonraki serilerde bu türlere tek tek değineceğim. Bu yazıda yer alan türler haricinde bazı türlere de değinebilirim ancak bir kesinlik bulunmamakta.