**SQL Serverda Hesaplanmış Kolon Oluşturmak (Computed Column)**

Bu makalemizde Sql Server da kullanılan hesaplanmış kolonları (computed column) anlatacağız.   Eğer değeri aynı tablodaki diğer kolonlardan bir veya birkaçına bağlı olarak değişen bir kolona ihtiyacınız varsa bu durumda Hesaplanmış Kolonları kullanabilirsiniz.

Örneğin Satış tablosunda birim fiyat ve adet kolonlarının çarpımı ile tutar kolonu oluşturmak istiyorsanız, Tutar kolonunu hesaplanmış kolon olarak oluşturmanız işinize yarayacaktır.

Hesaplanmış kolonlar kullanarak sql sorgularını yazarken zamandan ve kod uzunluğundan kurtulmuş olursunuz. Bu sayede her seferinde hesaplama yapmak için kod yazmak zorunda kalmazsınız.

Aşağıdaki scripte baktığınız zaman daha iyi anlayacaksınızdır.

|  |
| --- |
| create table Satis  (     UrunAdi varchar(20),     BirimFiyati decimal(6,2),     Adet int,     Tutar as (BirimFiyati \* Adet)  ) |

Hesaplanmış kolonları oluşturuken şunlara dikkat etmelisiniz:

         Hesaplanmış bir kolonu oluşturan kolonlar aynı tablo içinde olmalıdır, başka bir tablodaki kolonu kullanmak isterseniz bu kolona ancak bir user-defined functions üzerinden ulaşarak kullanabilirsiniz.

         Hesaplanmış bir kolon bir sql sorgusu içeremez, yani select yazarak değer alamazsınız.

         Hesaplanmış kolonlar diğer kolonların değerleri üzerinden oluşturulduğu için default anahtar kelimesini alamazlar, yani bir varsayılan değer atanamazlar.

         Hesaplanmış kolonu oluşturan kolonlardan herhangi biri başka bir hesaplanmış kolon olamaz.

         Hesaplanmış kolona insert veya update ile bir değer ataması yapamazsınız.

Hesaplanmış kolonlar fiziksek olarak yer kaplamazlar, sorgunun çalışması anında otpmatik olarak hesaplanırlar. Yani hesaplanmış kolonları sanal kolonlar olarak düşünebilirsiniz. Eğer hesaplanmış kolonlarında hesaplanarak fiziksel yer tutmasını istiyorsanız tabloyu oluştururken PERSISTED anahtar kelimesini kullanmanız gerekir. Bu sayede sanal bir kolon değil gerçek bir kolon gibi davranır. PERSISTED anahtar kelimesini kullandığını zaman hesaplanmış kolon üzerinde bir index tanımlamasıda yapabilirsiniz.

Şimdi bir kaç örnek vererek konuyu tamamlayalım.

|  |
| --- |
| create table Satis  (     UrunAdi varchar(20),     BirimFiyati decimal(6,2),     Adet int,     Tutar as (BirimFiyati \* Adet)    )  insert into Satis values('Urun1',10.2,23)  insert into Satis values('Urun1',1,10)    select \* from Satis |

Yukarıdaki sorgunu sonucu aşağıdaki gibi döner.

|  |  |  |  |
| --- | --- | --- | --- |
| **UrunAdi** | **BirimFiyati** | **Adet** | **Tutar** |
| Urun1 | 10.20 | 23.00 | 234.60 |
| Urun1 | 1.00 | 10.00 | 10.00 |

Gördüğünüz gibi Tutar kolonu otomatik olarak hesaplanarak oluşturuldu.

Hesaplanmış kolonları where cümleciği içinde de kullanabilliriz.

|  |
| --- |
| select \* from Satis where Tutar > 100 |

Varchar kolonlarıda hesaplanmış kolonlarda kullanabiliriz. Örneğin Ad ve soyad kolonlarını birleştirerek Kisiye ait AdSoyad alanını aşağıdaki gibi oluşturabiliriz.

|  |
| --- |
| create table Kisi  (         KisiId int,         Ad varchar(50),         Soyad varchar(50),         AdSoyad as Ad+Space(1)+Soyad   )   insert into Kisi values(1,'Sabri','Kunt')   insert into Kisi values(1,'Ali','Demir')   insert into Kisi values(1,'Ahmet','Tanrýverdi')   insert into Kisi values(1,'Seher','Kunt')     select \* from Kisi |

Sorgu sonucu :

|  |  |  |  |
| --- | --- | --- | --- |
| **KisiId** | **Ad** | **Soyad** | **AdSoyad** |
| 1 | Sabri | Kunt | Sabri Kunt |
| 1 | Ali | Demir | Ali Demir |
| 1 | Ahmet | Tanrıverdi | Ahmet Tanrıverdi |
| 1 | Seher | Kunt | Seher Kunt |

Hesaplanmış kolonlar hakkında yeterince bilgi verdiğimizi düşünüyorum, Kendinizde örnekler yaparak konuyu pekiştirmenizi tavsiye ederim.