

STORED PROCEDURE

Sql Server ile veritabanı üzerinde yapılan sorgular (select), veri girişleri (insert), veriler üzerinde yapılan güncellemeler (update) ve verilerin veritabanından kaldırılması (delete) gibi işlemleri her seferinde ilgili komutu yazarak işleme koymak Sql Server üzerinde yük oluşturur. Her yazılan sorgu komutu için Sql Server komutu yazım kuralları açısından (syntax) denetler ve sonraki aşamada ilgili tabloların ve alanların var olup olmadığının kontrolü yapılır ve son olarak en kısa ve en hızlı sorgunun nasıl yapılacağı belirlendikten sonra ilgili Sql scripti derlenir ve komut çalıştırılır. İşte biz Sql Server'a bu kadar işi her seferinde tekrarlatmamak için "Stored Procedure" (saklı yordam) adını verdiğimiz yapılardan faydalanırız. Stored Procedure (kısaca sp veya procedure) kullanılarak komutun derlenmesine kadar olan aşamaları bir kez tamamlanır ve procedure her çağırıldığında sadece ilgili komutun çalıştırılması sağlanır. Böylece Sql Server üzerinde daha performanslı sorgular ve komutlar çalıştırılmış olur.

Stored Procedure' ler hem sorgu komutlarında hem de insert, update ve delete işlemlerinde kullanılabilir. C# taki kullanılan metot (yordam) kavramına benzerler. Parametre alabilirler ve geriye değer döndürebilirler. Geriye değer döndürme işlemi "return" anahtar kelimesi ile yapılır.

Stored Procedure tanımlamasını iki şekilde göstereceğiz. İlk olarak parametresiz bir procedure yazımının şablonunu verelim:

```
create proc/procedure <procedure adı>  
as  
begin  
< yazılacak ilgili sorgu komutu>  
end
```

İkinci olarak ise parametrelili bir procedure yazımının şablonunu verelim:

```
create proc/procedure <procedure adı>  
(  
@prm1,  
@prm2  
)  
as  
begin  
< yazılacak ilgili sorgu komutu>  
end
```

Procedure' lere isim verilirken genellikle "usp" ön takısı verilerek isimlendirilirler. Ayrıca procedure ler de metotlar gibi iç içe çağırılabilirler. Procedure oluşturulurken "with encryption" ifadesi eklenirse procedure şifrelenmiş olur. Şimdi bir adet parametre almayan bir procedure oluşturalım:

```

1 use AdventureWorks
2 go
3
4 create proc spUrunleriGetir
5 as
6 begin
7     select ProductID,Name,ListPrice from Production.Product
8 end
9
10
11 exec spUrunleriGetir

```

Results		Messages	
	ProductID	Name	ListPrice
1	1	Adjustable Race	190,00
2	2	Bearing Ball	190,00
3	3	BB Ball Bearing	190,00
4	4	Headset Ball Bearings	190,00
5	316	Blade	190,00
6	317	LL Crankarm	190,00
7	318	ML Crankarm	190,00
8	319	HL Crankarm	190,00
9	320	Chainring Bolts	190,00
10	321	Chainring Nut	190,00
11	322	Chainring	190,00
12	323	Crown Race	190,00

Yukarıda görüldüğü gibi AdventureWorks veritabanı kullanıma alınmış ve spUrunleriGetir adlı bir procedure oluşturulmuştur. Burada dikkat edilmesi gereken nokta oluşturulan procedure çalıştırılmak istendiğinde “exec <procedure adı>” şeklinde çağırılması gerektiğidir. Aynı şekilde “execute <procedure adı>” şeklinde veya direkt olarak “<procedure adı>” yazılarak da ilgili procedure çalıştırılabilir. Son olarak bir de parametre alan ve geriye değer döndüren bir sp yazalım:

```
1 use AdventureWorks
2 go
3
4 create proc spAdetDondur
5 (
6 @ID int
7 )
8 as
9 begin
10     declare @adet int
11     select @adet=COUNT(ProductID) from Production.Product
12     where ProductSubcategoryID=@ID
13     return @adet
14 end
15
16
17 declare @gelen int
18 exec @gelen=spAdetDondur 3
19 print @gelen
```

Messages

22

Yukarıdaki

örneğimizde alınan “@ID” parametresine “3” değeri verilmiş olup geriye bu kategoriden 22 adet ürün olduğunu “return” anahtar kelimesi ile aldık ve procedure çağırıldığında dönüş değeri bir değişkende tutularak ekrana sonucunu yazdırmış olduk.