SQL – Stored Procedure

- SQL Server'daki Stored procedure'lar aynı diğer programlama dillerindeki procedure'lara benzer. SQL deyimlerini içeren komut doayaları hazırlanır ve sunucu üzerinde saklanır.
- Stored procedure aracılığıyla şu işlemler yapılablir:
- Input parametrelerini kabul ederek ve birçok değerin geri dönmesini sağlar.
- Database içindeki işlemleri yapmak için programlama deyimleri içerir.
- Stored procedure'lar genellikle rutin hale gelmiş işleri kolayca yapmak için geliştirilirler. SQL deyimleriyle yazılan stored procedure'lar sadece **ilk kez çalıştırıldıklarında derlenirler. Daha sonraki çalıştırma işlemlerinde derlenmezler** ve böylece hızlı bir biçimde çalışma sağlanmış olur.
- Veritabanımızda "**Programmability** -> **Store Procedure"** düğümünde yer alırlar.

by yselim

SQL – Stored Procedure

- Bir stored procedure "CREATE PROCEDURE" ya da kısaca "CREATE PROC" ifadesi ile yaratılır. Stored procedure'ümüzü oluşturmadan önce şu bilgileri edinmekte fayda var:
- -CREATE PROCEDURE ifadesinin altında "CREATE DEFAULT, CREATE RULE, CREATE TRIGGER, CREATE VIEW ve CREATE PROCEDURE ifadeleri kullanılamaz. Yani, bir stored procedure oluşturulurken, bu procedure'ün içinde DEFAULT, RULE, TRIGGER, VIEW ve başka bir PROCEDURE oluşturulamaz.
- Stored procedure oluşturabilmek için ;
  System Administrator (sysadmin)
  Database Owner (db_owner)
  Data Definiton Language Administrator (db_ddladmin) rollerine yada
  CREATE PROCEDURE izni verilmiş bir role sahip olunmalıdır.

by yselim

SQL – Stored Procedure

- Stored Procedure'lerin Diğer Stored Procedureleri Çağırması (**NESTING**) :

  Stored Procedure'ler, diğer stored procedure'leri çağırabilme özelliğine sahiptirler. Yani bir stored procedure çalışırken, aynı anda başka bir stored procedure'ün çalışmasını tetikleyebilir. Bu olaya "Nesting" denir.

- - Stored procedure'ler 32 kez diğer procedure'leri çağırabilirler. (32 level). Bu sayı 32'yi geçer ise stored procedure zincirinin kırılmasına ve yeni bir stored procedure'ün çağırılamamasına neden olur. Dolayısıyla "Nesting" olayı hiçbir zaman sonsuz döngüye girmez. Maksimum 32 kez "Nesting" olur.
  - *"@@Nestlevel"* sistem fonksiyonu ile, stored procedure'ün kaç kez nesting yaptığı araştırılabilir.

SQL – Stored Procedure

- Bir stored procedure ikinci bir stored procedure'ü çağırıyor ise; ikinci (çağırılan) stored procedure, birincinin (çağıran) stored procedure'ün yarattığı tüm objelere erişme hakkına sahiptir.
- Stored Procedure'ler aynı anda birbirlerini de çağırabilirler. Yani X stored procedure'ü Y stored procedure'ünü çağırırken, aynı anda Y stored procedure'ü de X stored procedure'ünü çağırabilir.

SQL – Stored Procedure

STORED PROCEDURE'Ü OLUŞTURMAK (CREATE) :

Parametre almayan prosedür oluştururken alttaki yapı kullanılır.

```
Create PROCEDURE <procedure adi>
As
Begin
<Çalıştırılacak Sql Komutları>
End
```

SQL – Stored Procedure

STORED PROCEDURE'Ü OLUŞTURMAK (CREATE) :

spUrunleriGetir prosedür örneği;

```
CREATE PROCEDURE spUrunleriGetir
    AS
    BEGIN
        SELECT Urunler.UrunID,Urunler.UrunAdi FROM Urunler
    END
--Proc çalıştırma
EXEC spUrunleriGetir
```

SQL – Stored Procedure

STORED PROCEDURE'Ü OLUŞTURMAK (CREATE) ve ÇALIŞTIRMAK (EXECUTE) :

Örnek;

```
use urunler
--sipariş edilen bir ürünün miktarını ürünler tablosundan düşen miktarı
--store proc ile yazalım
create procedure SiparisiUrunlerdenDusProc(params)
as
begin
select * from urunler
declare @mik int=10
declare @uid tinyint=3
insert into siparis values(8,@uid,@mik);
update urunler set miktar=miktar-@mik from urunler where urunId=@uid
select * from urunler
end

exec SiparisiUrunlerdenDusProc
```

SQL – Stored Procedure

STORED PROCEDURE'Ü DEĞİŞTİRMEK (ALTER PROCEDURE) :

Stored Procedure'leri değiştirme (düzenleme) işini , "ALTER PROCEDURE" ya da kısaca "ALTER PROC" ifadesi ile yapıyoruz. ALTER PROCEDURE ifadesi çalıştığında, SQL Server var olan stored procedure'ün tanımını yenisi ile değiştirir.

STORED PROCEDURE'Ü SİLMEK (DROP PROCEDURE) :

Stored Procedure'leri silme işini , "DROP PROCEDURE" ya da kısaca "DROP PROC" ifadesi ile yapıyoruz. Bir stored procedure'ü silmeden önce "**sp_depends** <object name>" sistem stored procedure'ünü kullanılarak, bu stored procedure'lerin hangi objelere dayandığını öğrenmek önemli olabilir.

SQL – Stored Procedure

STORED PROCEDURE'DE PARAMETRE KULLANIMI :

Parametre Kullanımı stored procedure'lere işlevsellik kazandırır. Stored Procedure'ler içerisine parametre alabilir (Input Parameters) ve dışarıya parametre ile bir değer döndürebilir (Output Parameters). Parametre ile işlevsellik kazandırılmış bir stored procedure sayesinde bir veritabanını, sadece parametre değerlerini değiştirerek çok amaçlı olarak kullanabiliriz.

a) INPUT PARAMETERS :

Bir stored procedure içine dışarıdan parametre çağrılabilir. Bu şekildeki parametreler "INPUT parametreler"dir. Stored procedure'e bir ya da daha çok parametre eklenebilir ve bu parametreler CREATE / ALTER PROCEDURE ifadesinin altında tanımlanır. Parametrelerin tanımlanması "@" ile olup, aşağıdaki şekildedir:

Syntax: @parameter data_type [=Default]

SQL – Stored Procedure

STORED PROCEDURE'DE PARAMETRE KULLANIMI :

Parametre Kullanımı stored procedure'lere işlevsellik kazandırır. Stored Procedure'ler içerisine parametre alabilir (Input Parameters) ve dışarıya parametre ile bir değer döndürebilir (Output Parameters). Parametre ile işlevsellik kazandırılmış bir stored procedure sayesinde bir veritabanını, sadece parametre değerlerini değiştirerek çok amaçlı olarak kullanabiliriz.

by yselim

SQL – Stored Procedure

STORED PROCEDURE'DE PARAMETRE KULLANIMI :

a) INPUT PARAMETERS :

Bir stored procedure içine dışarıdan parametre çağrılabilir. Bu şekildeki parametreler "INPUT parametreler"dir. Stored procedure'e bir ya da daha çok parametre eklenebilir ve bu parametreler CREATE / ALTER PROCEDURE ifadesinin altında tanımlanır. Parametrelerin tanımlanması "@" ile olup, aşağıdaki şekildedir:

b) OUTPUT PARAMETERS :

Stored Procedure'ler OUTPUT parametrelerle dışarıya değer döndürebilir. Dışarıya dönen bu değer çoğu zaman programcının diğer işlemlerinde kullanılabilmek ya da matematiksel hesaplamaların sonuçlarını öğrenmek amacıyla kullanılırlar. Parametre tanımlanmaları Input parametrelerdeki gibi olmakla beraber tek fazlası hem Stored Procedure'ün oluşturulma tarafında hem de Execute tarafında dışarıya değer döndürecek parametrenin yanına OUTPUT yazılmasıdır

by yselim

SQL – Stored Procedure

STORED PROCEDURE'DE PARAMETRE KULLANIMI :

Parametre alan bir fonksiyon yazmak istediğimizde ise prosedür adından sonra **parantez içinde parametreleri** veriyoruz. Fakat parametre verirken **@** işaretini kullanıyoruz.

```
Create PROCEDURE <procedure adi>
    (<parametre adi,parametre tipi> <varsa varsayılan değeri>)
    As
    Begin
    <Çalıştırılacak Sql Komutları>
    End
```

SQL – Stored Procedure

STORED PROCEDURE'DE PARAMETRE KULLANIMI :

Parametre alan bir fonksiyon yazmak istediğimizde ise prosedür adından sonra **parantez içinde parametreleri** veriyoruz. Fakat parametre verirken **@** işaretini kullanıyoruz.

```
CREATE PROCEDURE spUrunGetir
(
@ParamID INT=1
)
AS
BEGIN
    SELECT Urunler.UrunID,Urunler.UrunAdi FROM Urunler WHERE
Urunler.UrunID=@ParamID
END

--Bu prosedürün kullanımı ise
EXEC spUrunGetir 1
```

SQL – Stored Procedure

GERİYE DEĞER DÖNDÜREN PROSEDÜRLER

Geriye değer döndüren prosedürleri anlamak için birkaç terimi daha bilmemiz gerekiyor. Bunlardan birisi **out** veya **output** anahtar kelimesidir. Bu kelimeleri kullanarak, hangi parametrenin değer döndürdüğünü SQL SERVER'a tanıtıyoruz.

.

```
CREATE PROCEDURE spUrunAdetGetir
(
@UrunID INT,
@UrunAdet INT OUTPUT
)
AS
BEGIN
```

SQL – Stored Procedure

GERİYE DEĞER DÖNDÜREN PROSEDÜRLER

```
--Dönüş değeri olan Proc'lar
-- Bir ürünün verilen sipariş miktarları toplamını ve ürünün ismini
yazdıran stor proc
create procedure UrunSipMiktar(@uid tinyint,@urunAd nvarchar(50)
out,@mik int OUTPUT)
as
begin
select @urunAd=urunAd from urunler where urunId=@uid
select @mik=sum(satisMiktar) from siparis where siparisId=@uid
end

declare @Umik int=0
declare @Uadi nvarchar(50)
exec UrunSipMiktar 2,@Uadi out,@Umik out
if(@Umik>0)
print @Uadi+' ürününün toplam sipariş miktarı:'+ cast(@Umik as
nvarchar(4))
Else
raiserror('ürün yok',16,1)
```

by yselim

SQL – Stored Procedure -Örnek

- Aşağıdaki stored proc'ları hazırlayıp çalıştırınız.
- Sipariş miktar aralığı verilen ürünlerin listesini bulan stored proc?

Örn: sipariş 100,200 sipariş miktarı 100-200 arası stored proc.

- Verilen bir sipariş numarası için; siparişin siparişler listesi içinde var ise bilgilerini yazdırıp (ürün adı ile beraber) yoksa hata mesajını yazdıran stored proc yazınız**.**

(sipariş no ->in, sipariş Id,sipariş Miktarı ve Ürün adı, ->out)

- Girilen ürün adına göre ürünün stoktaki ederini (mik*bfiyat) hesaplattırıp eğer;

eder 4000 altında ise 'ederi düşük'

eder 4000-10000 arası ise 'ederi normal'

eder 10000 üzeri ise 'ederi yüksek' yazdırılacak stored proc yazınız

(ürün adı->in, «eder»->out, case ile «eder» kontrolü)

Örn:

Kalem 1500 , ederi düşük

Defter 2500, ederi normal

by yselim

SQL – Stored Procedure -Örnek

```
---KENDİ substring komutumuz

create proc strKopar(@metin varchar(50),@baslangic int ,@adet int)
as
begin

declare @uzunluk int, @sonuc varchar(50),@yenideger int, @sagdan
varchar(50)

 set @uzunluk = (select len(@metin))

set @yenideger = @uzunluk - @baslangic +1

set @sagdan = (select right(@metin,@yenideger))

set @sonuc = (select left(@sagdan,@adet))

print @sonuc
end


exec strKopar "kocaeli",3,2
```

SQL – Stored Procedure

EXTENDED STORED PROCEDURE'LER :

- Extended stored procedure'ler bir DLL içindeki fonksiyonlar olup, SQL Server'ın işlevselliğini artırmaktadırlar. Extended stored procedure'leri sadece MASTER veritabanında iken çalıştırılabilir.

```
Use Master
EXEC xp_cmdshell 'dir c:\'
--HATA oluştu..cmdshell enable yapılmalı

--cmdshell enable yapılıyor sp_configure ile
Use Master
EXEC sp_configure 'show advanced options', 1;
-- gelişmiş ayarlar.
RECONFIGURE;
-- özelliği açıyoruz
EXEC sp_configure 'xp_cmdshell', 1;
-- konfigürasyonu güncelle
RECONFIGURE;
--Artık cmdshell kullanılabilir
EXEC xp_cmdshell 'dir c:\windows'
```

SQL – Stored Procedure

EXTENDED STORED PROCEDURE'LER :

- Tüm xp'lerin listesi

```
SELECT *
   FROM master.sys.extended_procedures
   GO
```

SQL – Stored Procedure

DROP STORED PROCEDURE'LER :

- Stored Proc silmek için DROP PROCEDURE kullanılır

```
DROP PROCEDURE listeleProc
```