**VERİTABANI –II**

#  5.TRANSACT SQL (T-SQL)

## 5.1 SAKLI YORDAMLAR(STORER PROCEDURE-DAVAM)

### 5.1.1 Saklı Yordamı Silmek

 Var olan bir saklı yordamı silmek için DROP komutunu kullanmak gereklidir. DROP komutundan sonra saklı yordamın sahibinin adı ve saklı yordamın adı yazılmalıdır.

Genel Kullanımı

DROP PROC sahip.prosedür\_adı

Örnek

DROP PROC dbo.usp\_VeliOgrenciBilgileri

### 5.1.2 Değer Alan Alt Yordamlar

Stored Procedure’lerin daha etkin kullanılabilmesi ve işlevsel bir hale gelebilmesi için dışarıdan değer almalarına ihtiyaç duyulur. Bu nedenle girdi parametreleri (Input Parameter) kullanılır.

Stored Procedure’nin aldığı değer Query’den gelen değerdir. Gönderilen değeri karşılayacak bir değişken Stored Procedure’de tanımlanmalıdır

Örnek:

Daha önce oluşturduğumuz “usp\_VeliOgrenciListe” stored procedure’ünde veliId bilgisini kendimiz yazıyorduk. Şimdi ise veliId bilgisini dışarıdan veri olarak gönderelim.



veliId adında bir değişken tanımladık. Sorgumuzda karşılaştırma yaparken bu değeri kullandık. “usp\_VeliOgrenciListeInputValue” stored procedure’ümüzü çağıralım.

veliId değeri 1 verildiğinde;



Sonuç;



veliId değeri 2 verildiğinde;



Sonuç;



Stored Procedure’e gelen parametrelerin isteğe bağlı olması istenebilir. Bu gibi durumlarda Stored Procedure’de tanımlanan değişkene default değer atanması yapılır. Stored Procedure’e gelen parametreye değer atanmazsa, tanımlanan değişkene atanan default değer işleme tabi tutulur. Stored Procedure’debir default değer atayacaksanız bu değer, bir sabit olması gerekir

Örnek;

Bir sınıftaki öğrencilerin bilgilerinin tutulduğu bir tabloda öğrenci adlarının içerisinde “a” harfi geçen öğrencileri gösteren bir Stored Procedure şöyle yazılmalıdır





Stored Procedure’de tanımlanan “ara” değişkeni Query’den bir değer gelmese de NULL değerini alacak ve işlem gerçekleşecektir.

Query’den öğrenci adlarının içerisinde “a” harfi olan öğrenci leri görmek için Reaşağıdaki gibi kod satırını yazınız ve çalıştırınız.

 Ogr\_ara ‘a’

Ogr\_Ara ‘a’ şeklindeki yazım ile de Stored Procedure çalışacaktır.

Sonuçları Results penceresinde görebilirsiniz.



Stored Procedure’e değer gönderilmeseydi Messages penceresinde “ Command(s) completed successfully.” mesajını görecektiniz.

### 5.1.3 Stored Procedurlerde Çıkış Parametresi Kullanmak

Stored procedurler çalıştırıldıktan sonra geriye bir sonuç döndürebilmektedirler. Bu işlem return veya output ifadeleri kullanarak yapmaktadır.Output ifadesi ile geri değer döndürülmek istenildiğinde procedure parametre olarak verilen değişken tanımlanırken sonunda output ifadesi ile belirtilmelidir. Aşağıdaki örnekte bu parametrelerin kullanımı ayrıntılı bir şekilde anlatılmaktadır.

Örnek: Dışardan string tipinde aldığı ad ve soyad adındaki iki değişkeni birleştirip tek bir satır olarak geri döndüren strored procedurü yazalım.



Yukarıda birleştir adı ile oluşturulan yordam çalıştırılırken geriye değer döndürmeyen yordamlar gibi çalıştırılır. Fakat burada çıkış parametresinin döndüreceği değeri tutabilmek için yine output tipinde bir değişken tanımlanmalı ve procedure gönderildiği aynı sırada yine procedur çalıştırılırken gönderilmelidir. Aksi taktirde geri dönüş değerinin tutulacağı bir yer bulunamayacağı için ve eksik parametre gözlemleneceği için sistem hata verecektir.



Örnek: Sistemimizde bir anket sisteminin olduğunu düşünelim. Anket sisteminde her kayıt yapıldığında sisteme ekleme işlemini bir procedür kullanarak yapalım.



Yukarıda AnketEkle adında bir procedur oluşturulmaktadır. Bu procdurde dikkat edilmesi gereken bazı konular ayrıntılı bir şekilde aşağıda açıklanmaktadır.



Yukarı if koşul bloğunda not exists ifadesi ile anketi oylayan kullanıcının aynı ankete daha önce oylama yapıp yapmadığı kontrol edilmektedir. Eğer yapılmadıysa veritabanına kayıt yapılmaktadır eğer daha önce kayıt yapılmışsa sadece geriye 0 değeri döndürülmektedir. Burada dikkat edilmesi gereken diğer konuda geriye değer döndürme tarzıdır. Daha önce geriye değer döndürmek için output ifadesini kullanmıştık. Eğer output ifadesi kullanılmak istenmiyorsa yukarıdaki şekilde de geri dönüş sağlamak mümkündür.

Yukarıdaki procedürü çalıştırıp sonucunu görelim.



## 5.2 KULLANICI TANIMLI FONKSİYONLAR

SQL Server da programcıların hizmetine sunulan çok sayıda sistem tarafından tanımlanmış fonksiyonlar bulunmaktadır. Bu fonksiyonların amacı bazı temek işlemlerin programlama yapılırken tekrar tekrar oluşturulmasını engellemek ve programlamayı kolaylaştırmaktır. Sistem tarafından tanımlanan bu fonksiyonlara max,sum,count gibi hazır fonksiyonları örnek verebiliriz. SQL server içerisinde bulunan sistem tarafından tanımlı bazı hazır fonksiyonlar aşağıdaki gibidir.



Yukarıda belirtilen sistem tanımlı fonksiyonlar dışında kullanıcılar tarafından kullanılmak üzere fonksiyonlar oluşturmak mümkündür. Kullanıcı tanımlı fonksiyonlar Görünümler ve saklı yordamlara benzer şekilde oluşturulmaktadır.Fakat aralarında kullanım açısından bazı farklılıklar bulunmaktadır. Fonksiyonlar yordamlar gibi dışardan değer alabilirler ve geriye değer döndürebilirler .Fakar yordamlar çoğunlukla yalnız kullanılırlar yani herhangi bir sql programının içinde kullanılmazlar bunun aksine fonksiyonlar çoğunlukla kod bloklarının içinde sql ifadeleri ile birlikte kullanılmaktadırlar.

Kullanıcı tanımlı fonksiyon ulaşabilmek için management studio ekranından stored fonksiyonların da bulunduğu Programlanability sekmesi açılmalıdır.



Oluşturulacak fonksiyonlar tek değer döndüren yada birden fazla değer döndüren foksiyonlar olarak ayırılmaktadır. Eğer Management Studio kullanılarak uluşturulacaksa Scalar-valued Functions sekmesi sağ tıklanarak New Scalar Valued fonksiyon menusu tıklanarak yeni bir foksiyon oluştuma ekranı açılmalıdır.



Yapılan seçim sonrasında karşımıza düzenlenmek üzere aşağıdaki gibi bir ekran çıkacaktır.



Yukarıda belirtilen alanların nasıl ne amaçla düzenleneceği aşağıda maddeler halinde açıklanmaktadır.



Oluşturulacak fonksiyonun adı belirtilmelidir. Bir veritabanı içerisinde her fonksiyonun isminin kendine özel olması gerekmektedir. Yani aynı isimli iki farklı fonksiyon oluşturulmamaktadır.

 

Yukarıdaki bölüme saklı yordamlarda yapıldığı gibi parametreler veri tipleri ve @ işareti kullanılarak tanımlanmalıdır.



Returns ifadesi ile geri döndürülecek veri tipi belirlenmelidir. Burada belirlenen veri tipinden başka türde bir veri döndürülmek istenildiğinde hata ile karşılaşılacaktır. Tek değer döndüren fonksiyonlarda geri dönüş tipi olarak text, image, cursor ve ntext dışında diğer veri türleri kullanılabilir.



Begin ..End Bloğu arasına fonksiyonun yapması gereken sql kodları yazılır.