



VERİTABANI YÖNETİM SİSTEMLERİ

Dr. Öğr. üyesi
Ender Şahinaslan

BÖLÜM -11-

FONKSİYONLAR



GİRİŞ

Geçen haftaki derslerimizde Görünümleri (View) ve Stored Procedure (SP)'leri öğrendik.

Bu hafta ise fonksiyonlar konusunu ele alacağız:

- ✓ Kullanıcı Tanımlı Fonksiyonlar
- ✓ Skaler Kullanıcı Tanımlı Fonksiyonlar
- ✓ Tablo Döndüren Fonksiyonlar

11.1. KULLANICI TANIMLI FONKSİYONLAR

- Fonksiyonlar **belli bir sonucu geri döndürmek için tasarlanmış** bir veya birden fazla yerde kullanılan yapılardır.
- **Kullanıcı tanımlı fonksiyonlar**, tıpkı Stored Procedure'ler gibi **dışarıdan parametre alabilirler.**
- Aynı zamanda **IF...ELSE** gibi **T-SQL**'in diğer ifadelerinin de kullanılmasına imkan tanırırlar.

11.1. KULLANICI TANIMLI FONKSİYONLAR (DEVAM)

Kullanıcı tanımlı fonksiyonlar, SP' ler gibi derlendikten sonra tampon bir belleğe atanır ve daha sonra bu tampon bellekten çağrılarak kullanılabilirler.

Bir diğer özelliği de view (görünüm)'lar gibi bir tablo şeklinde kayıt döndürebilirler.

11.1. KULLANICI TANIMLI FONKSİYONLAR (DEVAM)

Kullanıcı tanımlı fonksiyonları, View ve SP'lerden ayıran ve özellikle kullanılması gereken durumlar şöyledir:

- Dışarıdan parametre alan **View** tanımlanamamaktadır.
 - ✓ Ancak, böyle bir durum fonksiyonlarda kullanılabilir.
- **SP**'ler bir sorgunun parçası olarak birkaç durum dışında çalışmazken, **fonksiyonlar sorguların içerisinde kullanılabilirler.**
 - ✓ Örneğin **sp_helpdb**, **sp_helptext** gibi **prosedürler tek başlarına çalıştırılabilirler**, fakat **bir sorgunun parçası olarak çalıştırılmazlar.**

11.2. SKALER KULLANICI TANIMLI FONKSİYONLAR

Skaler fonksiyonlar, bir işlem neticesinde sadece tek bir değer döndüren fonksiyonlardır.

SQL içerisinde tanımlı birçok skaler fonksiyon vardır.

- Matematiksel, karakter, tarih fonksiyonları vb.

Örneğin **SELECT GETDATE ()** komutu **FROM** ifadesi olmadan kullanılabilir. Bu tür yani **GETDATE ()** şeklindeki fonksiyonlar skaler fonksiyonlardır.

Bazı durumlarda kullanıcılarda bu tür fonksiyonlara ihtiyaç duyabilirler.

- Örneğin yapılan alışveriş neticesinde sepette kaç ürün olduğunu döndüren fonksiyon skaler bir fonksiyondur.

11.2. SKALER KULLANICI TANIMLI FONKSİYONLAR

Skaler fonksiyonların genel tanımı şöyledir:

```
CREATE FUNCTION [sahip-ismi] fonksiyon_adi
```

```
(
```

```
    @parametre_adi veri_türü,
```

```
    ....
```

```
)
```

```
RETURNS skaler_veri_tipi
```

```
AS
```

```
BEGIN
```

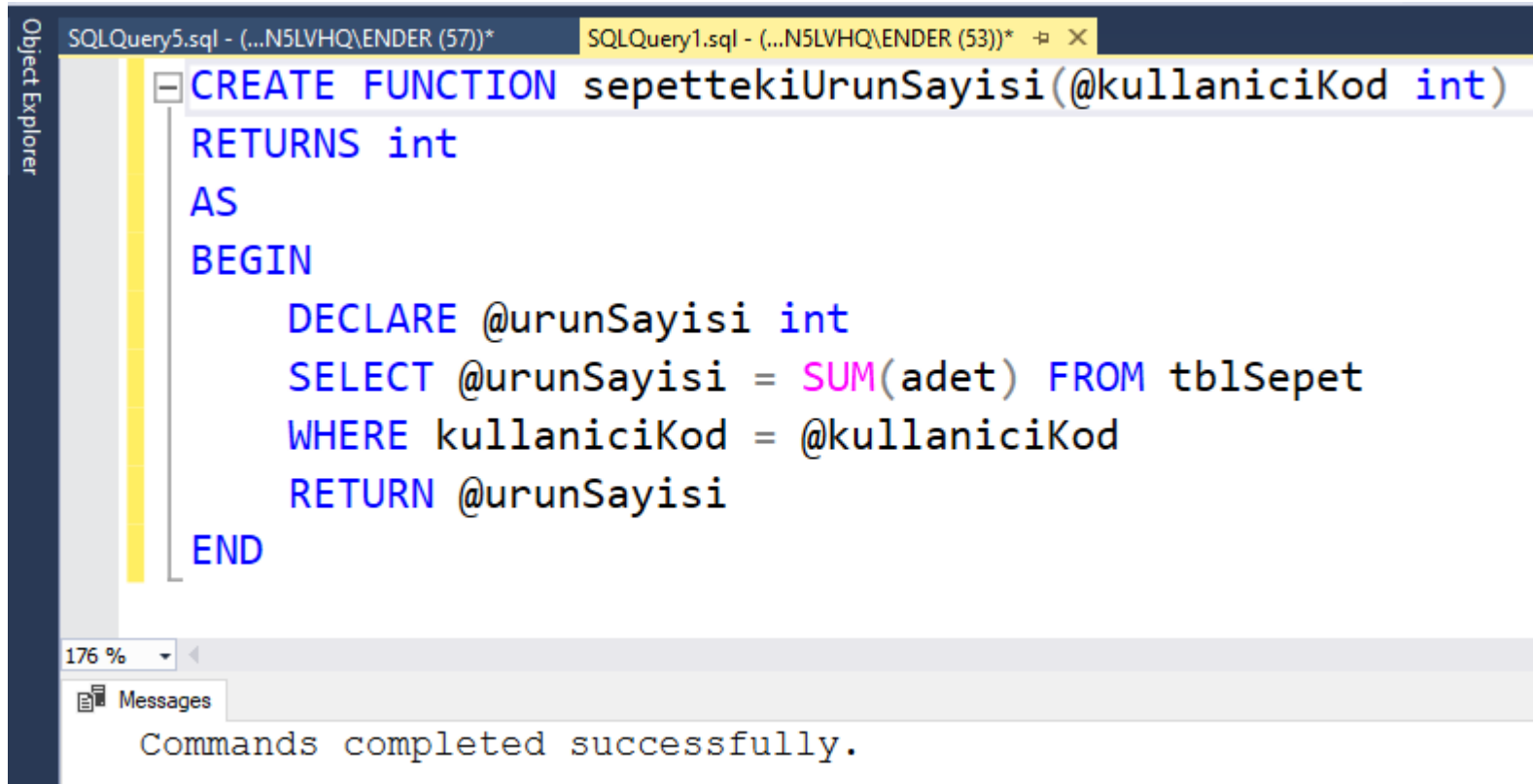
```
    SQL_ifadeleri
```

```
    RETURN skaler_deger veya değişken
```

```
END
```


11.2. SKALER KULLANICI TANIMLI FONKSİYONLAR

Örnek-1: Bir müşteri kodunu girerek müşterinin sepetinde toplam kaç adet ürün olduğunu bulan bir fonksiyon yazalım.



```
SQLQuery5.sql - (...N5LVHQ\ENDER (57))*  SQLQuery1.sql - (...N5LVHQ\ENDER (53))*
CREATE FUNCTION sepettekiUrunSayisi(@kullaniciKod int)
RETURNS int
AS
BEGIN
    DECLARE @urunSayisi int
    SELECT @urunSayisi = SUM(adet) FROM tblSepet
    WHERE kullaniciKod = @kullaniciKod
    RETURN @urunSayisi
END
```

176 %

Messages

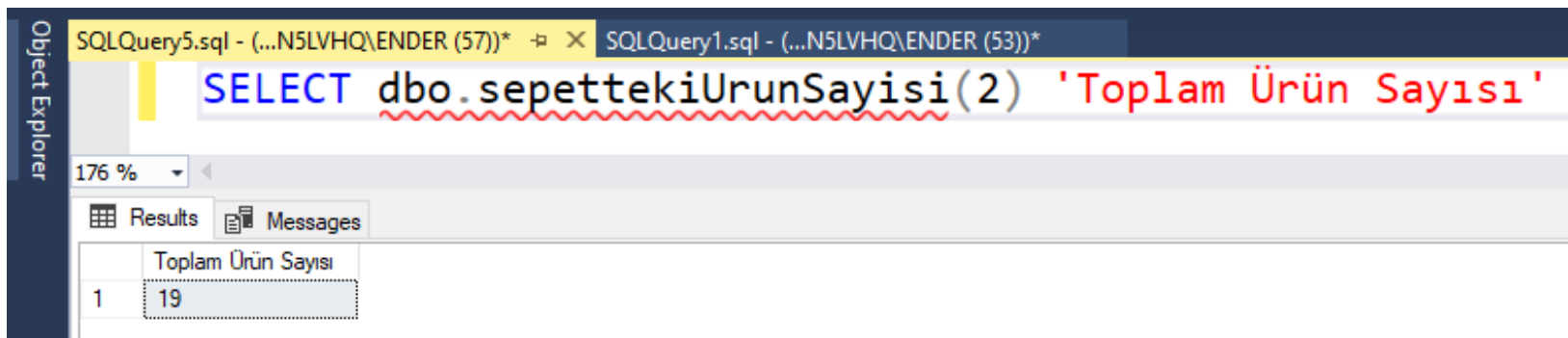
Commands completed successfully.

11.2. SKALER KULLANICI TANIMLI FONKSİYONLAR

- Yazmış olduğumuz fonksiyona dikkat ettiğimizde **RETURNS** deyimi ile **fonksiyonun integer tipte bir veri döndüreceği** anlaşılmaktadır.
- **@Urun_sayisi** olarak bir değişken tanımlandı.
- Daha sonra SQL cümlesinde dışarıdan parametre olarak verilen **kullanıcıKod** bilgisi ile veritabanındaki kullanıcıKod bilgisi karşılaştırılarak toplam ürün_sayısı hesaplandı ve **@Urun_sayisi** değişkenine aktarıldı.
- **RETURN** komutu ile elde edilen bu **ürün sayısı** dışarıya tek bir değer olarak gönderildi.

11.2. SKALER KULLANICI TANIMLI FONKSİYONLAR

Yazmış olduğumuz "sepettekiUrunSayisi" adlı fonksiyonun çalıştırılması aşağıda görülmektedir.

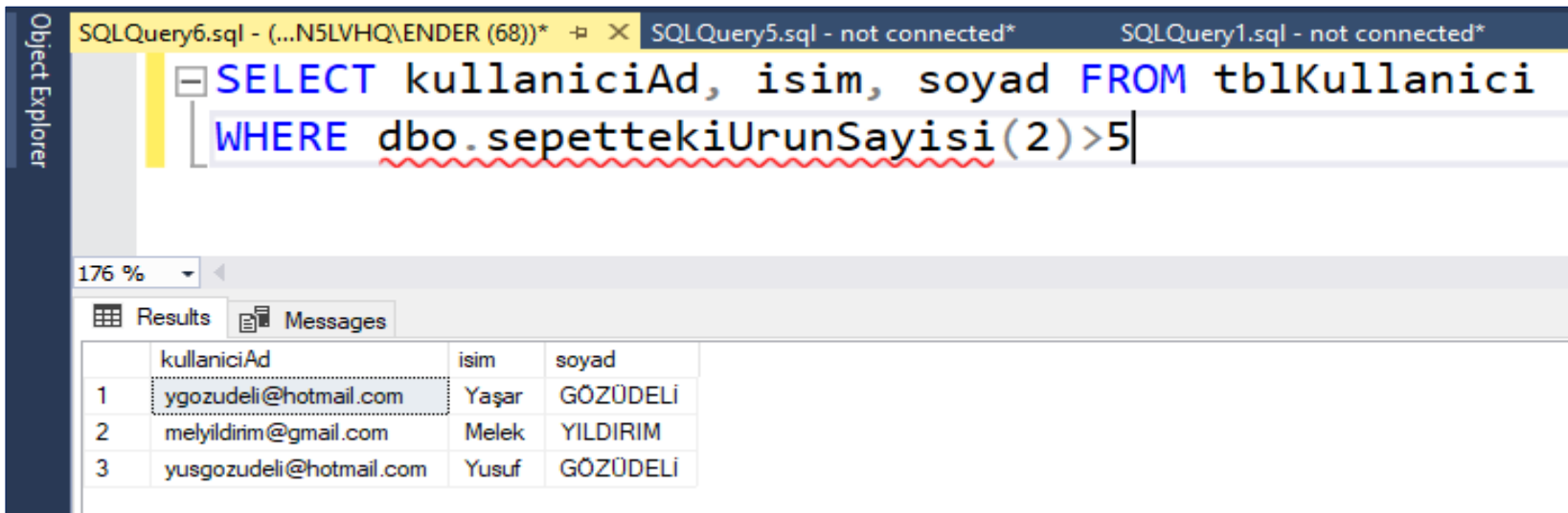


The screenshot shows the SQL Server Enterprise Manager interface. The top pane displays a query in the SQL Query window: `SELECT dbo.sepettekiUrunSayisi(2) 'Toplam Ürün Sayısı'`. The bottom pane shows the results of the query execution. The results are displayed in a table with one column, 'Toplam Ürün Sayısı', and one row with the value 19.

	Toplam Ürün Sayısı
1	19

11.2. SKALER KULLANICI TANIMLI FONKSİYONLAR

Yazmış olan "**sepettekiUrunSayisi**" adlı fonksiyonu bir sorgu içerisinde kullanmak isteyelim. **Kullanıcı kodu 2 olan ve sepetinde 5'ten fazla ürün olan** müşteriye bir sorgu yazarak bulalım. Fonksiyonun kullanımı ve SQL cümlesi şöyle olacaktır.



The screenshot shows the SQL Server Enterprise Manager interface. The top pane displays a SQL query in the 'SQLQuery6.sql' file. The query is:

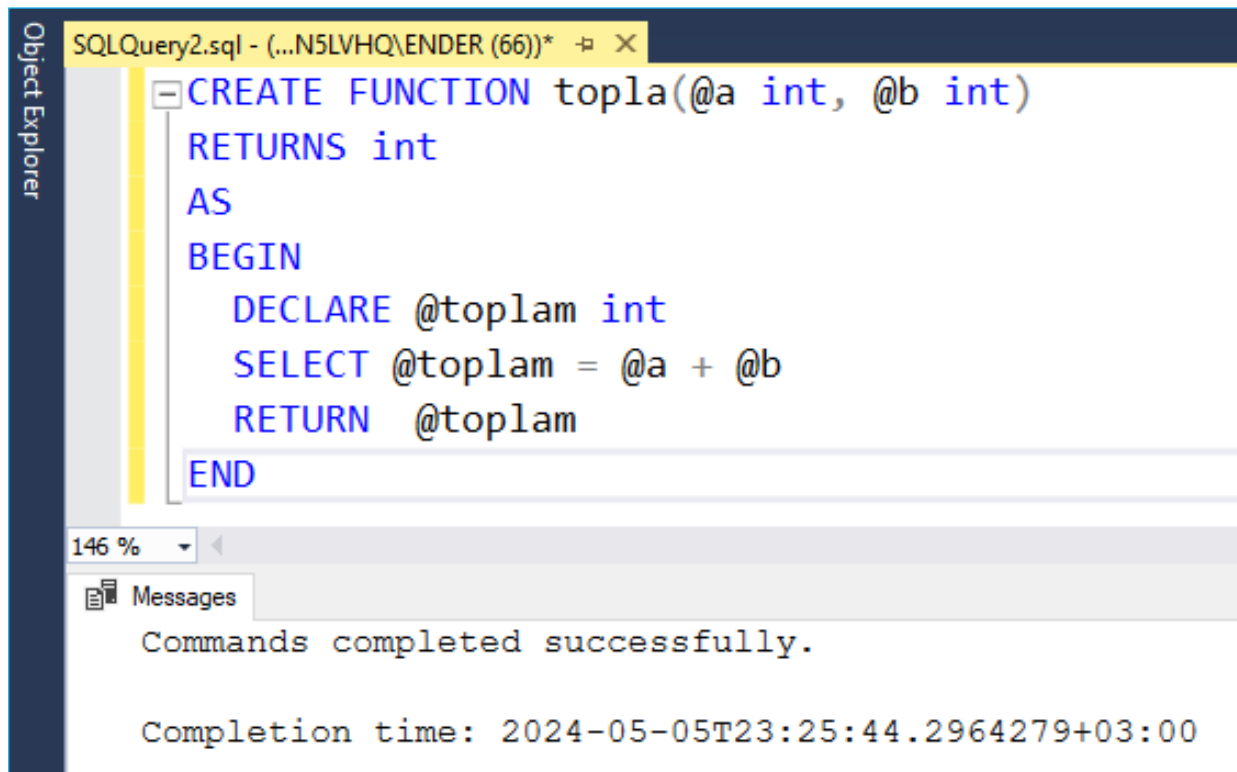
```
SELECT kullanıcıAd, isim, soyad FROM tblKullanici  
WHERE dbo.sepettekiUrunSayisi(2)>5
```

The bottom pane shows the 'Results' tab with a table of data. The table has four columns: 'kullanıcıAd', 'isim', and 'soyad'. The first column is highlighted. The data is as follows:

	kullanıcıAd	isim	soyad
1	ygozudeli@hotmail.com	Yaşar	GÖZÜDELİ
2	melyildirim@gmail.com	Melek	YILDIRIM
3	yusgozudeli@hotmail.com	Yusuf	GÖZÜDELİ

11.2. SKALER KULLANICI TANIMLI FONKSİYONLAR

- **Örnek-2:** Klavyeden girilen iki değerin toplamını gerçekleştiren bir fonksiyon yazınız?



The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'Object Explorer' pane is visible. The main window displays a SQL query in a text editor titled 'SQLQuery2.sql - (...N5LVHQ\ENDER (66))*'. The query is a T-SQL function definition for a scalar function named 'topla'. The function takes two integer parameters, '@a' and '@b', and returns an integer. The function body declares a local variable '@toplaml' of type 'int', assigns it the sum of '@a' and '@b', and then returns the value of '@toplaml'. Below the query editor, the 'Messages' pane shows the execution result: 'Commands completed successfully.' and the completion time: '2024-05-05T23:25:44.2964279+03:00'.

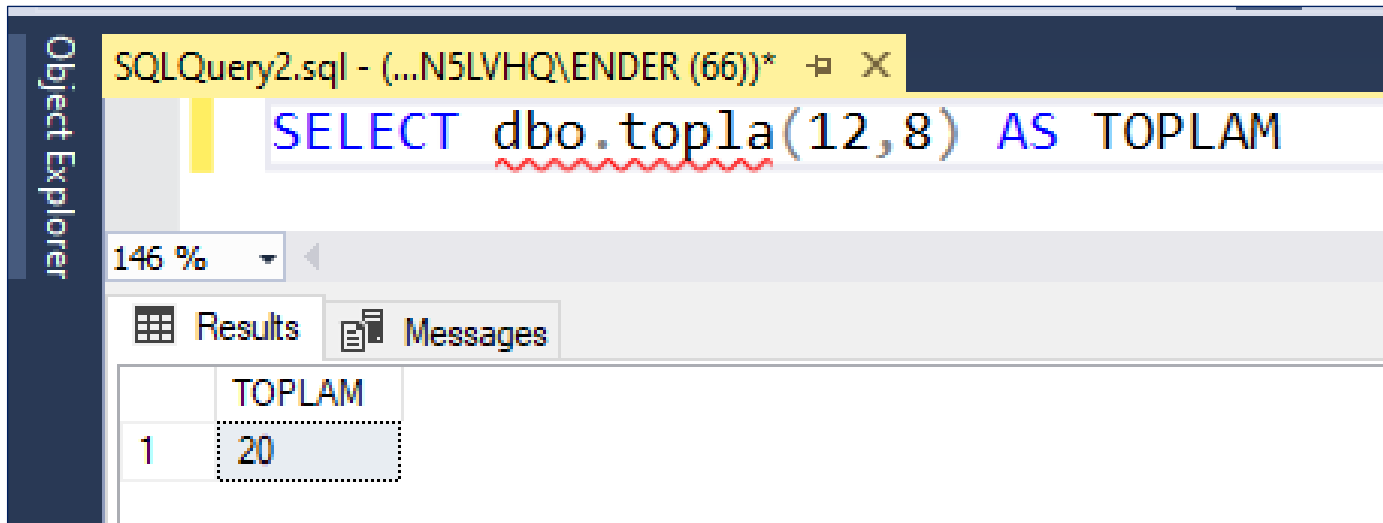
```
SQLQuery2.sql - (...N5LVHQ\ENDER (66))* X
CREATE FUNCTION topla(@a int, @b int)
RETURNS int
AS
BEGIN
    DECLARE @toplaml int
    SELECT @toplaml = @a + @b
    RETURN @toplaml
END

146 %
Messages
Commands completed successfully.

Completion time: 2024-05-05T23:25:44.2964279+03:00
```

11.2. SKALER KULLANICI TANIMLI FONKSİYONLAR

Bu fonksiyonu dışarıdan iki sayı girerek çalıştırdığımızda aşağıdaki sonuç elde edilecektir.



11.3. TABLO DÖNDÜREN FONKSİYONLAR

Bu tür fonksiyonlar çalışma prensibi bakımından view' lara benzerler, fakat farklı olarak dışarıdan parametre de alabilirler. **Tablo döndüren fonksiyonların genel kullanım biçimi şöyledir.**

```
CREATE FUNCTION [sahip-ismi]  
fonksiyon_adı ( [@Parametre_adı veri_tipi,  
.....] )
```

```
RETURNS TABLE
```

```
AS
```

```
RETURN (SELECT cümlesi)
```

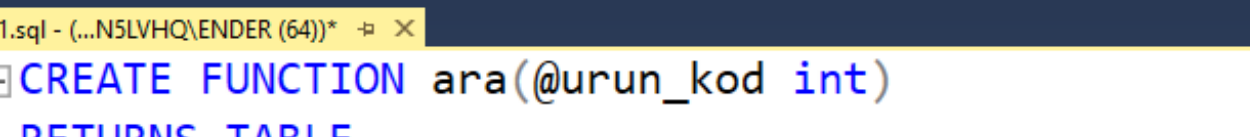
11.3. TABLO DÖNDÜREN FONKSİYONLAR (DEVAM...)

Örnek-1: Dışarıdan ürün kodu girilen bir ürünün bilgilerini tablo içerisinde arayarak ekrana listeleyen bir fonksiyon yazınız? tblUrun tablosu yan tarafta verilmiştir.

	Column Name	Data Type	Allow Nulls
▶	urunKod	int	<input type="checkbox"/>
	Barkod	varchar(32)	<input checked="" type="checkbox"/>
	bolgeKod	int	<input checked="" type="checkbox"/>
	markaKod	markaKod:int	<input checked="" type="checkbox"/>
	urunAd	varchar(255)	<input checked="" type="checkbox"/>
	bayiFiyat	float	<input checked="" type="checkbox"/>
	listeFiyat	float	<input checked="" type="checkbox"/>
	Indirim	float	<input checked="" type="checkbox"/>
	KDVoran	float	<input checked="" type="checkbox"/>
	urunResim	varchar(55)	<input checked="" type="checkbox"/>
	urunOzet	varchar(255)	<input checked="" type="checkbox"/>
	urunTanitim	varchar(4500)	<input checked="" type="checkbox"/>
	urunDurum	tinyint	<input type="checkbox"/>
	urunOncelik	tinyint	<input checked="" type="checkbox"/>
	stokDurum	int	<input checked="" type="checkbox"/>
	stokMinimum	int	<input checked="" type="checkbox"/>
	stokAmbarkod	tinyint	<input checked="" type="checkbox"/>
	garanti	varchar(50)	<input checked="" type="checkbox"/>
	dovizKod	char(10)	<input checked="" type="checkbox"/>
	reyonKod	bigint	<input checked="" type="checkbox"/>

11.3. TABLO DÖNDÜREN FONKSİYONLAR (DEVAM...)

Yazılması gereken fonksiyon aşağıdaki gibidir.



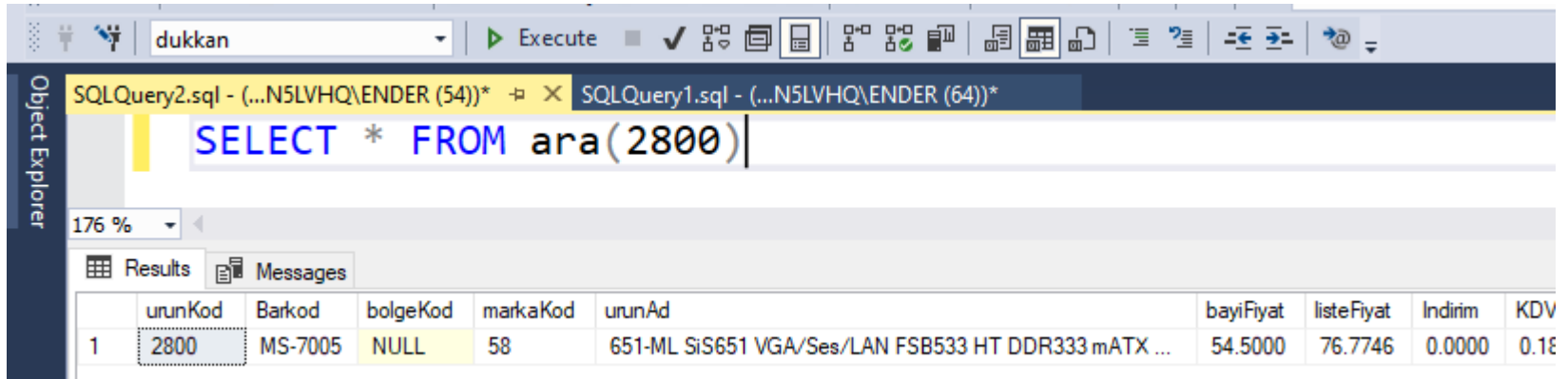
The screenshot shows the SQL Server Enterprise Manager interface. The top toolbar includes buttons for 'Execute' (a green play icon), 'Check for spelling' (a checkmark), 'Format' (a document with a ruler), 'Save' (a floppy disk), 'Print' (a printer), 'Zoom in' (a magnifying glass with a plus sign), 'Zoom out' (a magnifying glass with a minus sign), 'Full Screen' (a square with a circle), 'Help' (a question mark), 'Refresh' (a circular arrow), and 'Close' (an 'X'). The main window displays a T-SQL query in a text editor, with the file name 'SQLQuery1.sql' and the path '(...N5LVHQ\ENDER (64))*'. The query is as follows:

```
CREATE FUNCTION ara(@urun_kod int)
RETURNS TABLE
AS
RETURN SELECT * FROM tblUrun WHERE urunKod = @urun_kod
```

Below the query editor, the 'Messages' pane shows the result of the execution: 'Commands completed successfully.' The zoom level is set to 176%.

11.3. TABLO DÖNDÜREN FONKSİYONLAR (DEVAM...)

Yazılan fonksiyon incelendiğinde RETURNS TABLE ifadesi ile bir tablo döndürüleceği belirtilmiştir. Daha sonra RETURN ifadesinde yazılmış olan SELECT cümlesi ile kullanıcıya geri döndürülmesi gereken kayıtlar sorgulanarak gönderilmiştir. Fonksiyonun çalıştırılması şöyledir.



The screenshot shows the SQL Server Enterprise Manager interface. The top toolbar includes the 'Execute' button. The query editor displays the following SQL statement:

```
SELECT * FROM ara(2800)
```

Below the query editor, the 'Results' tab is active, showing a table with 10 columns and 1 row of data. The columns are: urunKod, Barkod, bolgeKod, markaKod, urunAd, bayiFiyat, listeFiyat, Indirim, and KDV. The first row contains the values: 2800, MS-7005, NULL, 58, 651-ML SiS651 VGA/Ses/LAN FSB533 HT DDR333 mATX ..., 54.5000, 76.7746, 0.0000, and 0.18.

	urunKod	Barkod	bolgeKod	markaKod	urunAd	bayiFiyat	listeFiyat	Indirim	KDV
1	2800	MS-7005	NULL	58	651-ML SiS651 VGA/Ses/LAN FSB533 HT DDR333 mATX ...	54.5000	76.7746	0.0000	0.18

11.3. TABLO DÖNDÜREN FONKSİYONLAR (DEVAM...)

- **Örnek-2:** Genellikle E-ticaret sitelerinde yer alan ve bir ürünü alanlar bunları da satın aldı şeklinde karşımıza çıkan ürünlerin listesi yer almaktadır. Aynı şekilde bir kullanıcı kodu kullanarak böyle bir liste oluşturacak bir fonksiyon yazalım. Bunun için gerekli olan tablolar şunlardır.
 - **tblUrun:** Ürün bilgilerinin yer aldığı tablodur.
 - **tblSiparisDetay:** Sipariş detaylarının yer aldığı tablodur.
 - **tblSiparis:** Siparişi verilen ürünlerin tutulduğu tablodur.

11.3. TABLO DÖNDÜREN FONKSİYONLAR (DEVAM...)

Bu tablolar aşağıda verilmiştir.

Table - dbo.tblUrun			
Column Name	Data Type	Allow Nulls	
urunKod	int	<input type="checkbox"/>	
Barkod	varchar(32)	<input checked="" type="checkbox"/>	
bolgeKod	int	<input checked="" type="checkbox"/>	
markaKod	markaKod:int	<input checked="" type="checkbox"/>	
urunAd	varchar(255)	<input checked="" type="checkbox"/>	
bayiFiyat	float	<input checked="" type="checkbox"/>	
listeFiyat	float	<input checked="" type="checkbox"/>	
Indirim	float	<input checked="" type="checkbox"/>	
KDVRan	float	<input checked="" type="checkbox"/>	
urunResim	varchar(55)	<input checked="" type="checkbox"/>	
urunOzet	varchar(255)	<input checked="" type="checkbox"/>	
urunTanitim	varchar(4500)	<input checked="" type="checkbox"/>	
urunDurum	tinyint	<input type="checkbox"/>	
urunOncelik	tinyint	<input checked="" type="checkbox"/>	
stokDurum	int	<input checked="" type="checkbox"/>	
stokMinimum	int	<input checked="" type="checkbox"/>	
stokAmbarkod	tinyint	<input checked="" type="checkbox"/>	
garanti	varchar(50)	<input checked="" type="checkbox"/>	
dovizKod	char(10)	<input checked="" type="checkbox"/>	
reyonKod	bigint	<input checked="" type="checkbox"/>	
kullanimKlavuzWEB	varchar(255)	<input checked="" type="checkbox"/>	
baStokDurum	bit	<input checked="" type="checkbox"/>	
teslimatSure	tinyint	<input checked="" type="checkbox"/>	
dagiticiKod	tinyint	<input checked="" type="checkbox"/>	
hacimDesi	float	<input checked="" type="checkbox"/>	
baKarisikKoli	bit	<input checked="" type="checkbox"/>	
birimKod	tinyint	<input checked="" type="checkbox"/>	

Table - dbo.tblSiparisDetay			
Column Name	Data Type	Allow Nulls	
SatirNo	int	<input type="checkbox"/>	
faturaKod	int	<input checked="" type="checkbox"/>	
urunKod	int	<input checked="" type="checkbox"/>	
fiyat	float	<input checked="" type="checkbox"/>	
KDV	float	<input checked="" type="checkbox"/>	
adet	smallint	<input checked="" type="checkbox"/>	
		<input type="checkbox"/>	

Table - dbo.tblSiparis			
Column Name	Data Type	Allow Nulls	
faturaKod	int	<input type="checkbox"/>	
kullaniciKod	int	<input checked="" type="checkbox"/>	
faturaAdresi	nvarchar(255)	<input checked="" type="checkbox"/>	
vergiDairesi	nvarchar(50)	<input checked="" type="checkbox"/>	
vergiNo	nvarchar(50)	<input checked="" type="checkbox"/>	
siparisDurumKod	int	<input checked="" type="checkbox"/>	
siparisTarih	datetime	<input checked="" type="checkbox"/>	

11.3. TABLO DÖNDÜREN FONKSİYONLAR (DEVAM...)

Bu işlem iki aşamalıdır. Yani iki ayrı fonksiyon yazılmalıdır. Bunlardan birincisi şöyledir:

```
CREATE FUNCTION FN_AlanlarinKodu(@urunKod int)
```

```
RETURNS TABLE
```

```
AS
```

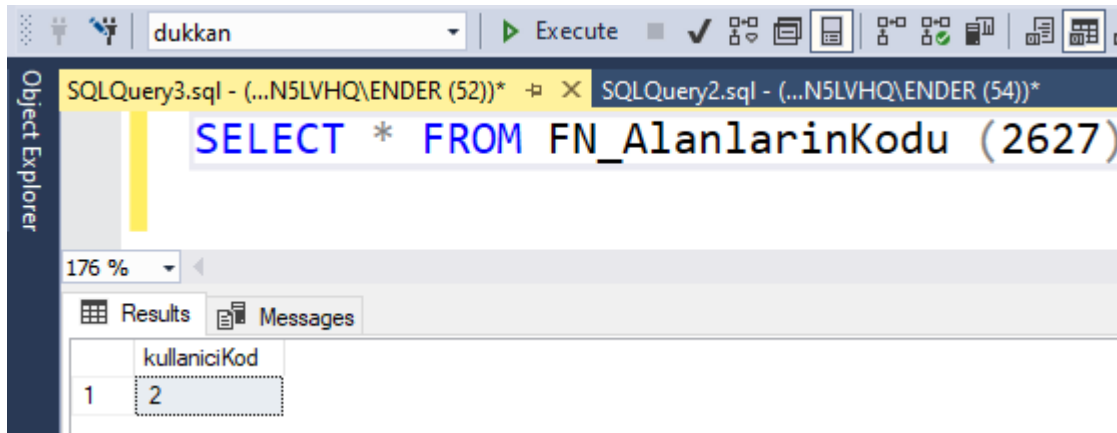
```
RETURN SELECT kullaniciKod FROM tblSiparisDetay SD
```

```
INNER JOIN tblSiparis S ON S.faturaKod=SD.faturaKod
```

```
WHERE urunKod=@urunKod
```

11.3. TABLO DÖNDÜREN FONKSİYONLAR (DEVAM...)

Bu fonksiyon belli bir ürünü satın alan kullanıcıya ait kullanıcı kodunun bulunmasını sağlamaktadır. Buradan elde edilecek olan kullanıcı kodu yazılacak olan ikinci fonksiyonda kullanılacaktır. Bu fonksiyon tek başına çalıştırılırsa aşağıdaki sonuç elde edilecektir.



11.3. TABLO DÖNDÜREN FONKSİYONLAR (DEVAM...)

İkinci yani asıl fonksiyonumuz ise şöyle olacaktır.

```
CREATE FUNCTION FN_BunuAlanBunlariAldi(@urunKod int)
RETURNS TABLE
AS
RETURN
SELECT U.urunKod, U.urunAd FROM tblUrun U
JOIN tblSiparisDetay SD ON SD.urunKod=U.urunKod
JOIN tblSiparis S ON S.faturaKod=SD.faturaKod
JOIN FN_AlanlariKodu(@urunKod) FNA ON
FNA.kullaniciKod=S.kullaniciKod
```

11.3. TABLO DÖNDÜREN FONKSİYONLAR (DEVAM...)

Bu fonksiyonda ise ürün kodu girilerek bu ürünü alan kişinin başka hangi ürünleri aldığı listelenmektedir.

Burada **tblUrun** ve **tblSiparisDetay** tabloları kullanılarak siparişi verilen ürünler tespit edilmiştir.

Ardından **tblSiparis** tablosu ve elde edilen bu liste **faturaKod** değerleri kullanılarak birleştirilmiştir.

Elde edilen yeni liste kullanıcı kodlarına bakılarak bu listede hangi kullanıcının hangi ürünü satın aldığı tespit edilmiştir.

11.3. TABLO DÖNDÜREN FONKSİYONLAR (DEVAM...)

Bu kodun çalıştırıldığında 2627 ürün koduna sahip ürünü alan kişinin başka hangi ürünleri satın aldığı listelenmektedir.

The screenshot shows the SQL Server Enterprise Manager interface. The top toolbar includes buttons for 'Execute', 'Save', 'Print', 'Refresh', 'Zoom', 'Grid', 'Messages', and 'Help'. The 'Object Explorer' pane on the left shows the 'dukkán' database. The 'SQLQuery5.sql' file is open, displaying the query: `SELECT * FROM FN_BunuAlanBunlariAldi(2627)`. The 'Results' pane at the bottom shows a table with 7 rows of data. The first row is highlighted, showing product code 2627 and its name '1.44MB Tekli Floppy Disket Sürücü.'.

	urunKod	urunAd
1	2627	1.44MB Tekli Floppy Disket Sürücü.
2	3265	FlyVideo3000 TV Karty+Radyo Philips7134, Capture, ...
3	3468	SIDEWINDER PRECISION 2 USB PORT (Bundle)
4	2720	IDE 20GB 7200RPM Caviar HDD
5	2629	7 Kart okuyuculu external FDD,CF, SD, SM, MMC, M...
6	2635	AMD 2400+ Kutulu Fanli Islemci
7	2657	Celeron 2.8Ghz SOCKET Kutulu Fanli 128K Islemci

11.3. TABLO DÖNDÜREN FONKSİYONLAR (DEVAM...)

- **Örnek-3:** Girilen fatura koduna göre bugün alınan siparişlerin durumunu gösteren “`siparisDurumunuGoster`” adında bir fonksiyon yazınız.
 - `tblSiparis` ve `tblSiparisDurum` tablolarını dikkate alınız.

11.3. TABLO DÖNDÜREN FONKSİYONLAR (DEVAM...)

- Girilen fatura koduna göre bugün alınan siparişlerin durumunu göstermesi amacıyla yazılan “siparisDurumunuGoster” fonksiyonu aşağıda görülmektedir.

```
SQLQuery3.sql - A...RTHUR\ARWEN (55))* X
CREATE FUNCTION siparisDurumunuGoster(@faturaKod int)
RETURNS TABLE
AS
RETURN SELECT S.faturaKod, Sd.aciklama
FROM tblSiparis S INNER JOIN tblSiparisDurum SD ON S.siparisDurumKod=SD.siparisDurumKod
WHERE DAY(S.siparisTarih)=DAY(GETDATE()) AND
      MONTH(S.siparisTarih)=MONTH(GETDATE()) AND
      YEAR(S.siparisTarih)=YEAR(GETDATE()) AND
      S.faturaKod=@faturaKod
```

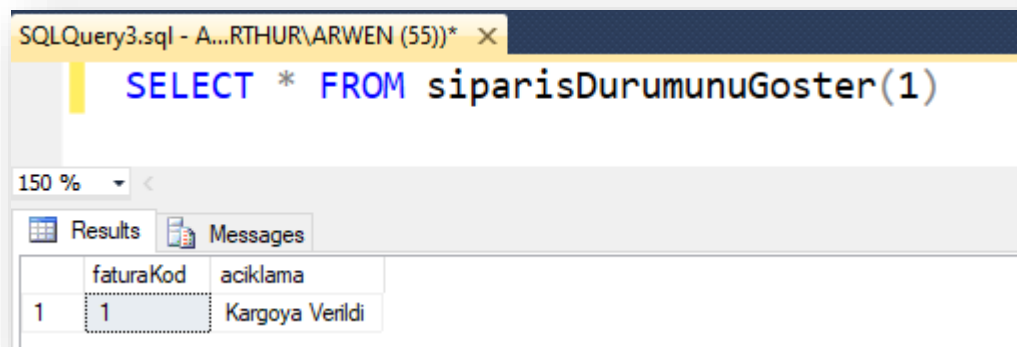
150 % <

Messages

Command(s) completed successfully.

11.3. TABLO DÖNDÜREN FONKSİYONLAR (DEVAM...)

- Girilen fatura koduna göre bugün alınan siparişlerin durumunu göstermesi amacıyla yazılan “**siparisDurumunuGoster**” fonksiyonunun kullanımı aşağıda görülmektedir.



The screenshot shows a SQL Server Enterprise Manager window titled "SQLQuery3.sql - A...RTHUR\ARWEN (55))*". The query editor contains the SQL statement: `SELECT * FROM siparisDurumunuGoster(1)`. Below the query editor, the "Results" tab is active, displaying a table with two columns: "faturaKod" and "aciklama". The table contains one row with the values "1" and "Kargoya Verildi".

	faturaKod	aciklama
1	1	Kargoya Verildi

11.3. TABLO DÖNDÜREN FONKSİYONLAR (DEVAM...)

- Örnek-4: Fatura kodu dışarıdan girilen bir ürünün adı, markası ve sipariş tarihini bulmaya yarayan "**siparisBilgisiGetir**" adında bir fonksiyon yazınız?

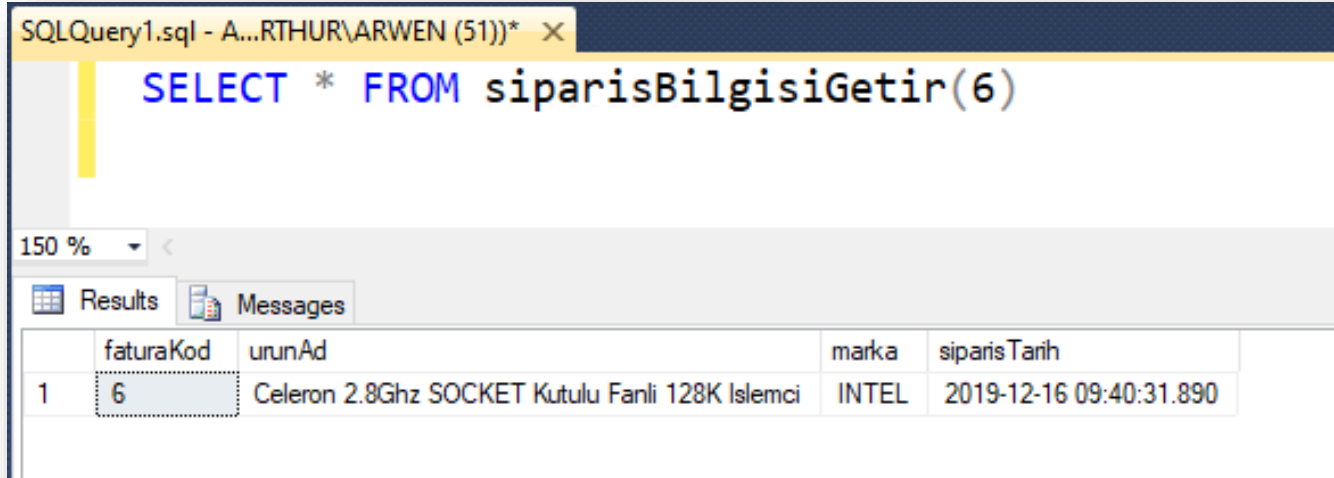
11.3. TABLO DÖNDÜREN FONKSİYONLAR (DEVAM...)

- Fatura kodu dışarıdan girilen bir ürünün adı, markası ve sipariş tarihinin bulunmasını sağlayan "**siparisBilgisiGetir**" fonksiyonu aşağıda görülmektedir.

```
SQLQuery1.sql - A...RTHUR\ARWEN (51)) * x
CREATE FUNCTION siparisBilgisiGetir(@faturaKod int)
RETURNS TABLE
AS
RETURN SELECT S.faturaKod,u.urunAd,M.marka,s.siparisTarih
FROM tblUrun U INNER JOIN tblsiparisDetay SD ON U.urunKod = SD.urunKod
INNER JOIN tblSiparis S ON S.faturaKod = SD.faturaKod
INNER JOIN tblMarka M ON M.markaKod = U.markaKod
WHERE S.faturaKod=@faturaKod
150 %
Messages
Command(s) completed successfully.
```

11.3. TABLO DÖNDÜREN FONKSİYONLAR (DEVAM...)

- Fatura kodu dışarıdan girilen bir ürünün adı, markası ve sipariş tarihinin bulunmasını sağlayan "**siparisBilgisiGetir**" fonksiyonunun kullanımı aşağıda görülmektedir.



The screenshot shows a SQL query window titled "SQLQuery1.sql - A...RTHUR\ARWEN (51))*". The query is `SELECT * FROM siparisBilgisiGetir(6)`. Below the query, the "Results" tab is active, displaying a table with 5 columns: faturaKod, urunAd, marka, and siparisTarih. The first row of data shows faturaKod as 6, urunAd as "Celeron 2.8Ghz SOCKET Kutulu Fanli 128K Islemci", marka as INTEL, and siparisTarih as 2019-12-16 09:40:31.890.

	faturaKod	urunAd	marka	siparisTarih
1	6	Celeron 2.8Ghz SOCKET Kutulu Fanli 128K Islemci	INTEL	2019-12-16 09:40:31.890

TEŞEKKÜRLER...

SORULARINIZ ?
