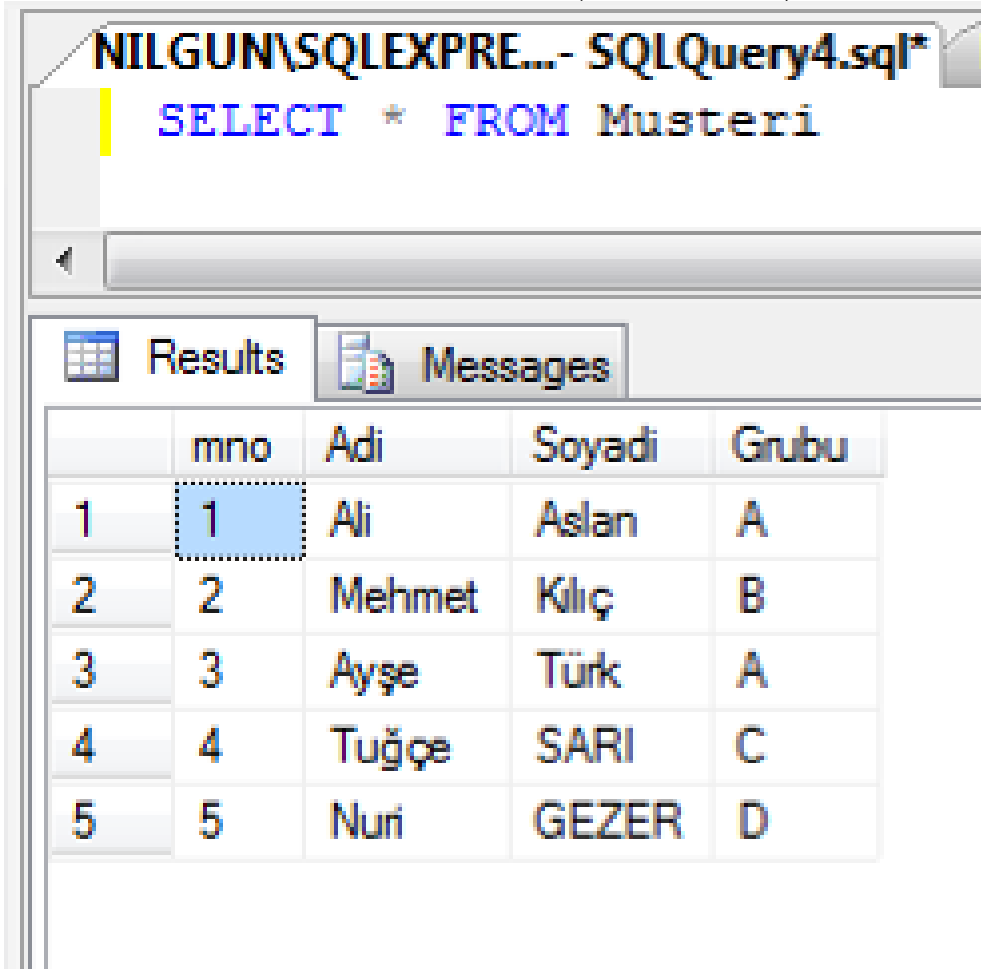


## KULLANICI TANIMLI FONKSİYONLAR (Devam)



The screenshot shows a SQL Server Enterprise Manager window titled "NILGUN\SQLEXPRES...- SQLQuery4.sql\*". The query window contains the SQL statement: `SELECT * FROM Musteri`. Below the query window, there are two tabs: "Results" and "Messages". The "Results" tab is active, displaying a table with the following data:

	mno	Adi	Soyadi	Grubu
1	1	Ali	Aslan	A
2	2	Mehmet	Kılıç	B
3	3	Ayşe	Türk	A
4	4	Tuğçe	SARI	C
5	5	Nuri	GEZER	D

Skaler değerli fonksiyon örneği:  
Parametre müşteri grubu olacak, eğer grubu parametresi değeri NULL olursa tüm müşteri sayısını, null değil ise verilen gruptaki müşteri sayısını veren bir fonksiyon oluşturunuz.

Çözüm:

```
CREATE FUNCTION fnMusteriSayisi
(@Grubu varchar(10) )
RETURNS int
AS
BEGIN
    DECLARE @msayi int
    IF @Grubu IS NULL
        SELECT @msayi=count(*) from
musteri
    ELSE
        SELECT @msayi=count(*) from
musteri WHERE Grubu=@Grubu
    RETURN @msayi
END
```

Yukarıdaki fonksiyondan,

-- Tüm müşteri sayısı

```
SELECT dbo.fnMusteriSayisi(NULL)
```

-- A grubundaki müşteri sayısı

```
SELECT dbo.fnMusteriSayisi('A')
```

-- C grubundaki müşteri sayısı

```
SELECT dbo.fnMusteriSayisi('C')
```

## TABLO DÖNDÜREN FONKSİYONLAR

Bu tip fonksiyonlar viewlere çok benzerler. Bir tek select ifadesi içerir. Bu ifadenin sonucunu gösterir. View'den farklı olarak dışarıdan parametre alır.

Genel ifade

```
CREATE FUNCTION fonksiyonAdi
```

```
(varsaParametreler)
```

```
RETURNS TABLE
```

```
AS
```

```
RETURN selectİfadesi
```

```
GO
```

Örnek:

Tüm müşteri listeleme için bir view oluşturunuz ve bu viewi sorgulayınız.

```
CREATE VIEW vwMusteri AS
```

```
SELECT * FROM musteriler
```

```
Go
```

Bu viewi sorgulayalım

```
SELECT * FROM vwMusteri
```

B2-30-HOCA\S...QLQuery4.sql\* B2-30-H

```
SELECT * FROM vwMusteri
```

Results Messages

	mno	Adi	Soyadi	Grubu
1	1	Ali	Aslan	A
2	2	Mehmet	Kılıç	B
3	3	Ayşe	Türk	A
4	4	Tuğçe	SARI	C
5	5	Nuri	GEZER	D

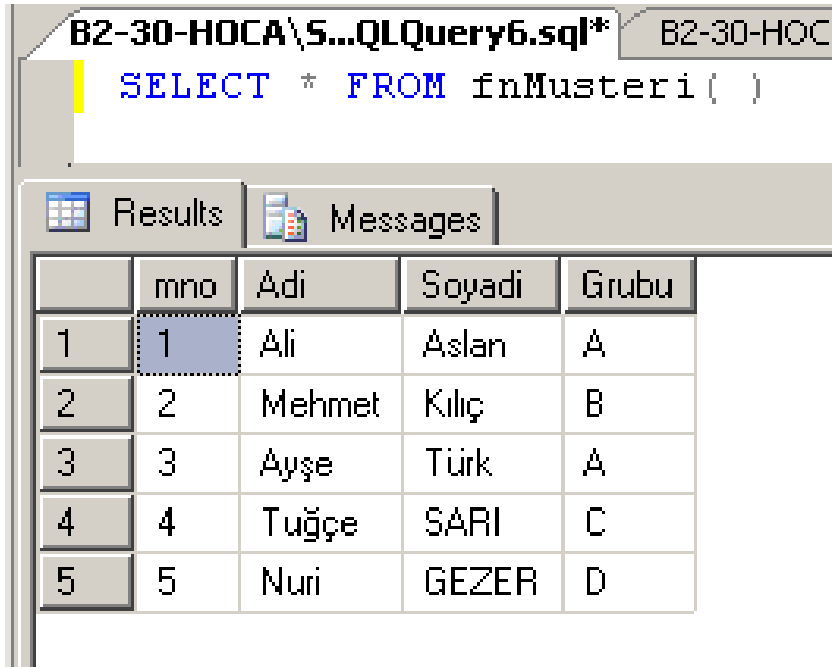
Tüm müşterileri listeleyen bir fonksiyon oluşturunuz ve bu fonksiyonu sorgulayınız.

```
CREATE FUNCTION fnMusteri (  
RETURNS TABLE  
AS  
RETURN select * from musteriler  
Go
```

Sorgulayalım

Tablo döndüren fonksiyonlarda dbo. yazma zorunluluğu yoktur.

```
SELECT * FROM fnMusteri( )
```



The screenshot shows a SQL Server Enterprise Manager window with a query window titled 'B2-30-HOCA\S...QLQuery6.sql\*' containing the SQL statement 'SELECT \* FROM fnMusteri( )'. Below the query window, the 'Results' tab is active, displaying a table with 5 rows and 5 columns: 'mno', 'Adi', 'Soyadi', and 'Grubu'. The first row is highlighted.

	mno	Adi	Soyadi	Grubu
1	1	Ali	Aslan	A
2	2	Mehmet	Kılıç	B
3	3	Ayşe	Türk	A
4	4	Tuğçe	SARI	C
5	5	Nuri	GEZER	D

Tüm müşterileri listeleyen bir stored procedure oluşturunuz ve bu prosedürü çalıştırınız

```
CREATE PROCEDURE spMusteri  
AS  
SELECT * FROM musteriler  
Go
```

Çalıştırılabilir  
EXEC spMusteri

B2-30-HOCA\S...QLQuery8.sql\* B2-30

```
EXEC spMusteri
```

Results Messages

	mno	Adi	Soyadi	Grubu
1	1	Ali	Aslan	A
2	2	Mehmet	Kılıç	B
3	3	Ayşe	Türk	A
4	4	Tuğçe	SARI	C
5	5	Nuri	GEZER	D

Yukarıdaki view, fonksiyon ve prosedürden faydalanarak A ve C grubundaki müşterileri ayrı ayrı listeleyiniz.

--vwMusteri viewinden A ve C grubundaki müşteriler

```
SELECT * FROM vwMusteri WHERE Grubu='A'
```

```
SELECT * FROM vwMusteri WHERE Grubu='C'
```

The screenshot shows a SQL query editor with two tabs: 'B2-30-HOCA\S...QLQuery9.sql\*' and 'B2-30-HOCA\S...QLQuery8.sql\*'. The first query is `SELECT * FROM vwMusteri WHERE Grubu='A'` and the second is `SELECT * FROM vwMusteri WHERE Grubu='C'`. Below the queries, there are two result grids. The first grid shows two rows for group 'A': (1, 1, Ali, Aslan, A) and (2, 3, Ayşe, Türk, A). The second grid shows one row for group 'C': (1, 4, Tuğçe, SARI, C).

```
SELECT * FROM vwMusteri WHERE Grubu='A'
```

```
SELECT * FROM vwMusteri WHERE Grubu='C'
```

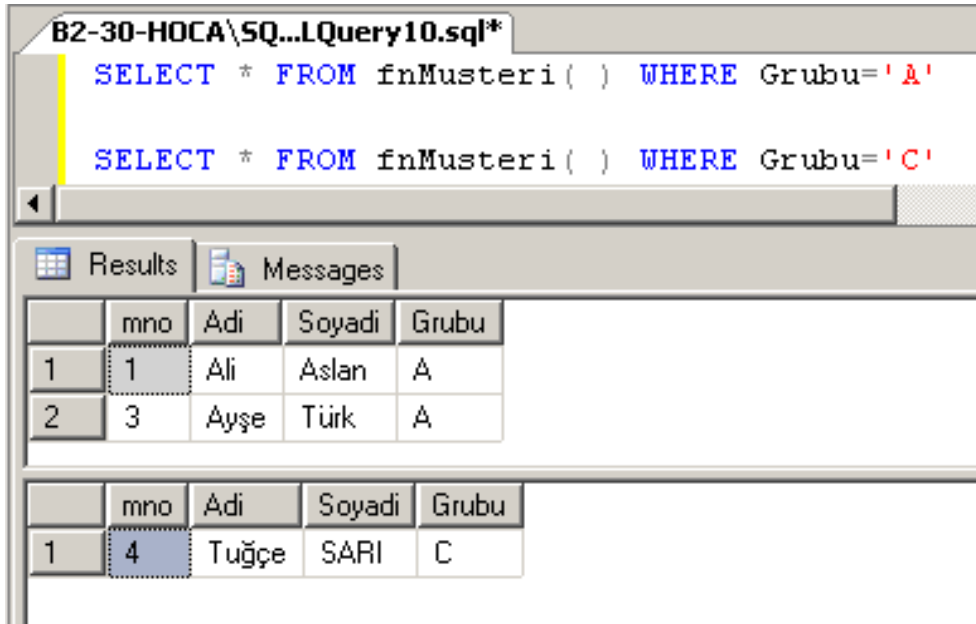
	mno	Adi	Soyadi	Grubu
1	1	Ali	Aslan	A
2	3	Ayşe	Türk	A

	mno	Adi	Soyadi	Grubu
1	4	Tuğçe	SARI	C

-- fnMusteri fonksiyonundan A ve C grubundaki müşteriler

```
SELECT * FROM fnMusteri() WHERE Grubu='A'
```

```
SELECT * FROM fnMusteri() WHERE Grubu='C'
```



-- spMusteri prosedüründen A ve C grubundaki müşteriler.

Bu prosedürden A ve C grubundaki müşteriler listelenemez. Çünkü prosedürler sorgulanamaz. Yani SELECT deyimi içinde prosedür bulunamaz.

Yukarıdaki view, fonksiyon ve prosedüre Grubu parametresi koyalım ve yeniden oluşturalım.

-- vwMusteri view'ine parametre tanımlayalım.

Viewe parametre tanımlanamaz.



-- fnMusteri fonksiyonuna parametre tanımlayalım.

```
ALTER FUNCTION fnMusteri (@Grubu  
varchar(10) )  
RETURNS TABLE  
AS  
RETURN select * from Musteri where  
Grubu=@Grubu  
Go
```

A ve C grubundaki müşteriler

```
SELECT * FROM fnMusteri('A')
```

```
SELECT * FROM fnMusteri('C')
```

```
B2-30-HOCA\5Q...LQuery12.sql*
SELECT * FROM fnMusteri('A')

SELECT * FROM fnMusteri('C')
```

Results Messages

	mno	Adi	Soyadi	Grubu
1	1	Ali	Aslan	A
2	3	Ayşe	Türk	A

---

	mno	Adi	Soyadi	Grubu
1	4	Tuğçe	SARI	C

--spMusteri prosedürüne parametre tanımlayalım.

```
ALTER PROCEDURE spMusteri @Grubu  
varchar(10)
```

```
AS
```

```
SELECT * FROM musteriler WHERE  
Grubu=@Grubu
```

```
Go
```

A ve C Grubundaki müşteriler  
EXEC spMusteri 'A'

## EXEC spMusteri 'C'

B2-30-HOCA\SQL...LQuery15.sql\*

```
EXEC spMusteri 'A'|  
EXEC spMusteri 'C'
```

Results Messages

	mno	Adi	Soyadi	Grubu
1	1	Ali	Aslan	A
2	3	Ayşe	Türk	A

	mno	Adi	Soyadi	Grubu
1	4	Tuğçe	SARI	C