

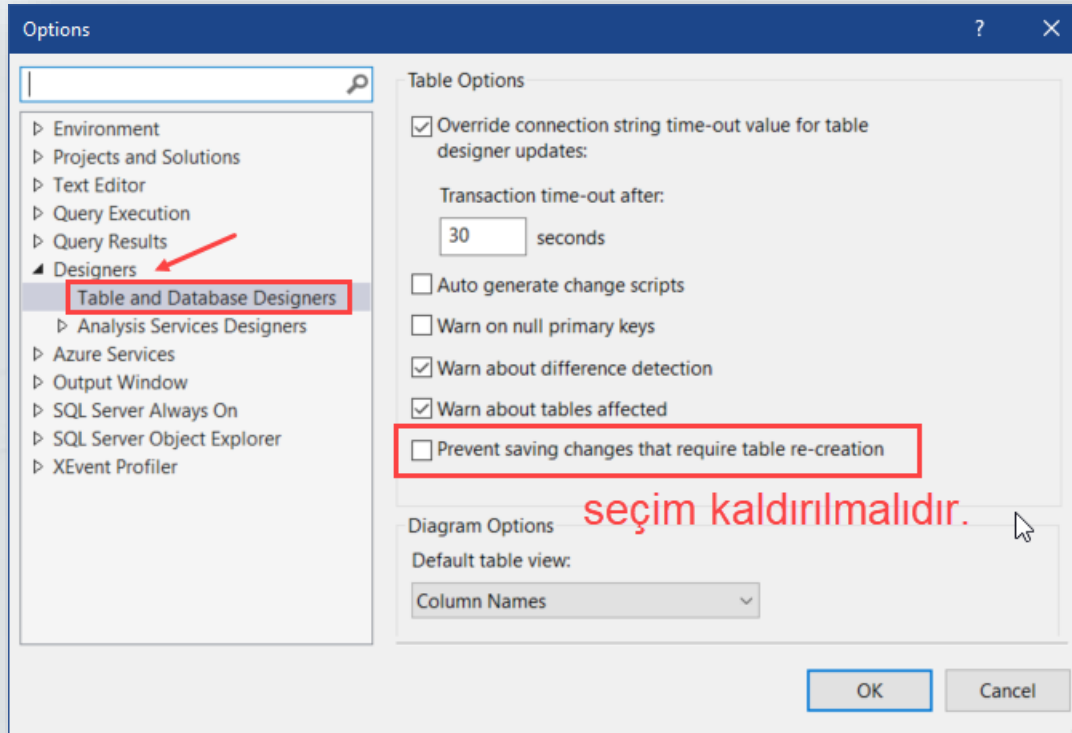
Aritmetiksel Operatörler ve Hazır Fonksiyonlar

Adem AKKUŞ

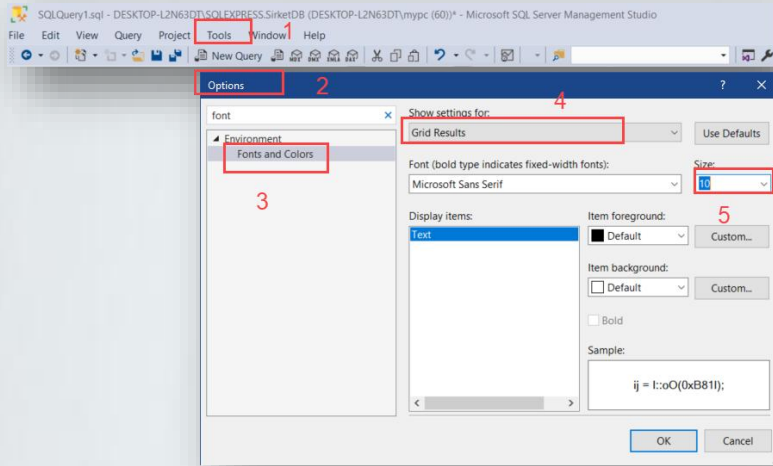
| Bilgisayar Mühendisi | Uzm. Bilişim Tekn. Öğrt. | Öğitmen

Tablo Deęiřiklik Yapma Ayarı

Tabloda deęiřiklik yapmak için, tabloyu yeniden oluřturma zorunluluęu ayarının kaldırılması



SSMS Font Ayarları



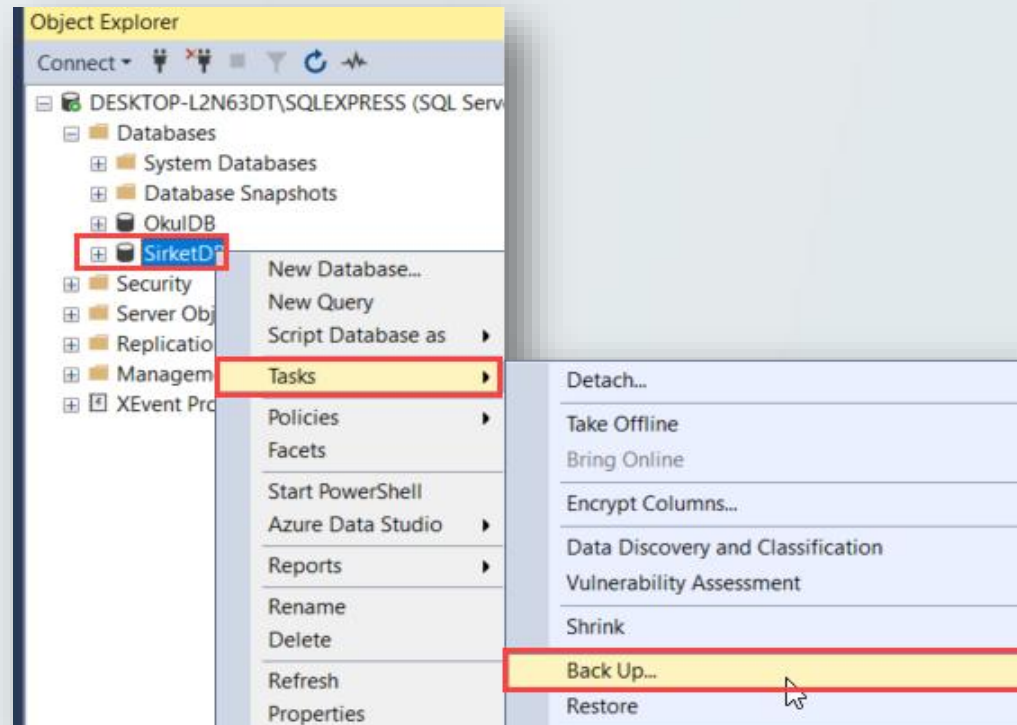
perid	adi	soyadi	departmani	gorevi	sehir	maas	cinsiyet
1	Adem	AKKUŞ	Yazılım	Bilgisayar Mühendisi	ANKARA	13000,00	ERKEK
2	Canberk	METİN	Pazarlama	Pazarlamacı	İSTANBUL	19500,00	ERKEK
3	Mehmet	ADSIZ	Muhasebe	Muhasebeci	ÇORUM	26000,00	ERKEK
4	Halime	DAŞDEMİR	Muhasebe	Muhasebeci	KASTAMONU	156000,00	Kadın
5	Alperen	YAVUZ	İhracat	Karavan	İZMİR	156000,00	ERKEK
6	Kevem	ALDEMİR	Dış Ticaret	Exper	MERSİN	169000,00	NULL
7	Güler	DUMAN	Müzik	Sanatçı	Adıyaman	20800,00	Kadın
8	Alperen	KILAVUZ	İhracat	Karavan	İZMİR	156000,00	ERKEK

6 MSSMS yeniden başlatıldığında ayarlar geçerli olur.

Veritabanı Backup Alma

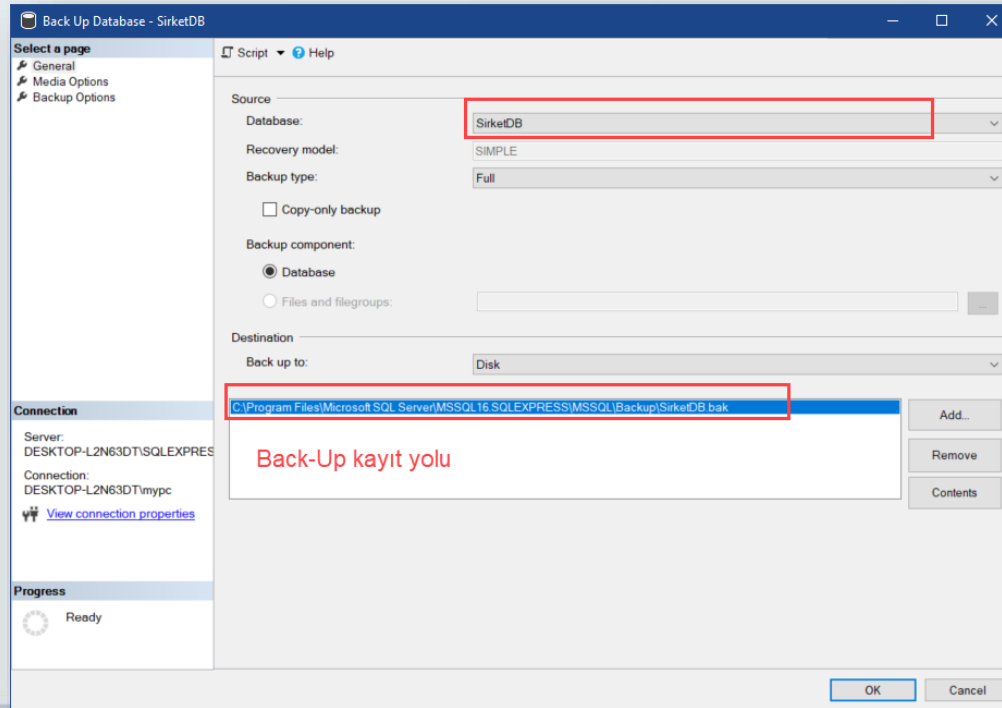
Veritabanınızın yedeğini almak için şu adımları izleyin:

1. SQL Server Management Studio'yu (SSMS) başlatın ve SQL Server örneğinize bağlanın.
2. **Object Explorer -> Databases** ile veritabanları düğümünü genişletin .
3. Veritabanına sağ tıklayın, **Tasks** (Görevler) üzerine gelin ve **Back Up...** (Yedekle) ögesini seçin .



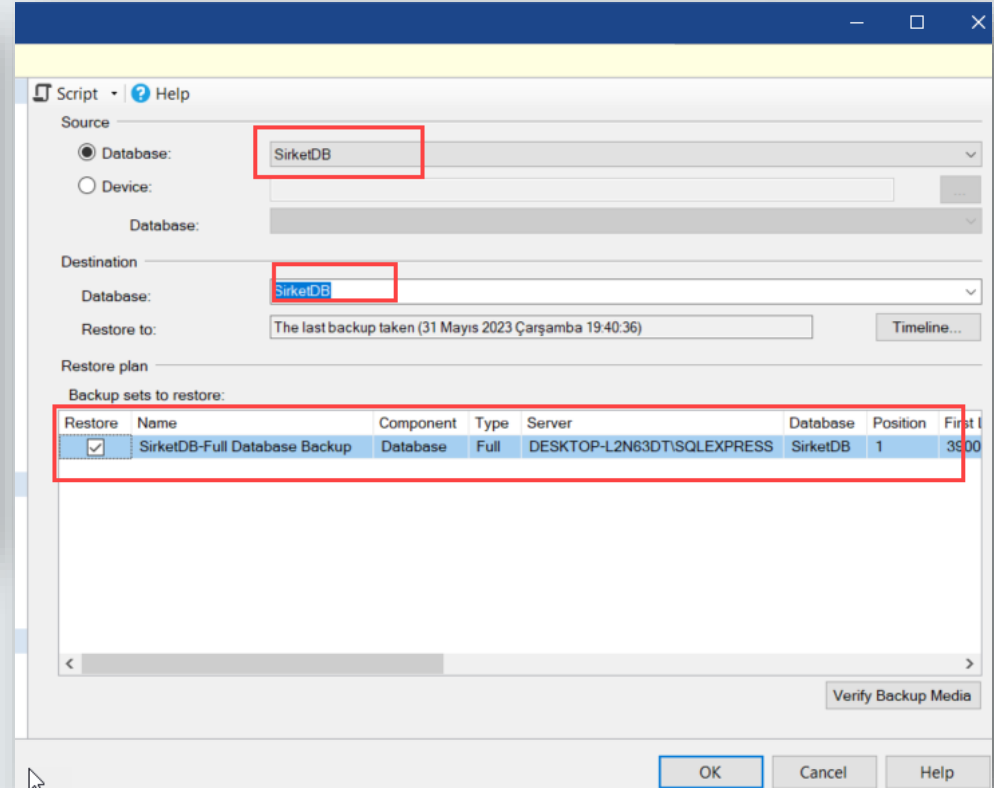
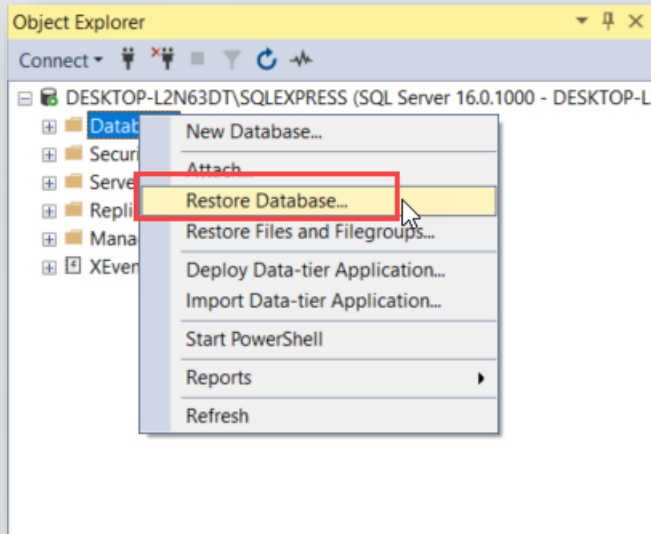
Veritabanı Backup Alma

4. **Destination** (Hedef) altında , yedeklemenizin yolunun doğru olduğunu onaylayın. Yolu değiştirmeniz gerekirse, mevcut yolu kaldırmak için **Remove** (Kaldır) seçin ve ardından yeni bir yol yazmak için **Add** (Ekle) seçin . Belirli bir dosyaya gitmek için üç noktayı kullanabilirsiniz.
5. Veritabanınızın yedeğini almak için **OK** (Tamam) seçin ..



Veritabanı Restore

1. **SQL Server Management Studio'yu** (SSMS) başlatın ve SQL Server örneğinize bağlanın.
2. **Object Explorer** veritabanları düğümüne sağ tıklayın ve veritabanını **Restore Database...** ögesini seçin .



TOP Komutu (limit, rownum)

- Belirli sayıda satır çekmek için SQL'de TOP ifadesi kullanılır.
- TOP n ifadesiyle tablonun baştan ilk n satırını çekebiliriz.
- TOP n veya TOP n PERCENT anahtar kelimeleri ORDER BY yan tümcesini ile birlikte kullanılmalıdır.
- Aksi halde listelenen verilerde WHERE ifadesi ile belirtilen koşula uyan kayıtlar rastgele dizileceğinden, istediğimiz sonucu elde edemeyebiliriz.

MySQL-> LIMIT, ORACLE->ROWNUM kullanır

```
SELECT TOP N COLUMN1, COLUMN2...  
FROM TABLOADI  
ORDER BY COLUMN1 ASC,COLUMN2 DESC
```


TOP Komutu (limit, rownum)

```
SQLQuery3.sql - (lo...Administrator (54))* SQLQuery2.sql - (lo...Administrator (59))* SQLQuery1.sql - (lo...Administrator (56))*  
SELECT TOP 5 * FROM CUSTOMER  
ORDER BY CUSTOMERNAME
```

235 %

Results Messages

	ID	CUSTOMERNAME	CITY	DISTRICT	BIRTHDATE	GENDER	AGE
1	3	İsmak TAHSİNOĞLU	Erzincan	Çayır	1940-06-23	KADIN	78
2	4	Yaşar SAVURGAN	İsparta	Gelendost	1991-08-07	ERKEK	27
3	5	Remzi ELYİĞİT	Giresun	Çanakçı	1953-09-14	ERKEK	65
4	6	Mehmet Akif POLAT	Bitlis	Mutki	1992-12-25	ERKEK	26
5	7	Sahil FAYDALI	Ordu	Çatalpınar	1996-08-03	ERKEK	22

```
SQLQuery3.sql - (lo...Administrator (54))* SQLQuery2.sql - (lo...Administrator (59))* SQLQuery1.sql - (lo...Administrator (56))*  
SELECT * FROM CUSTOMER  
WHERE CITY='İSTANBUL'  
ORDER BY CUSTOMERNAME
```

235 %

Results Messages

	ID	CUSTOMERNAME	CITY	DISTRICT	BIRTHDATE	GENDER	AGE
1	427	Ali Osman OZELÇAM	İstanbul	Beykoz	1990-09-08	ERKEK	28
2	493	Amin TUĞLU	İstanbul	Ümraniye	1992-10-31	KADIN	26
3	162	Can TAŞER	İstanbul	Bayrampaşa	1953-11-04	ERKEK	65
4	691	Ceren ÇALIŞKAN	İstanbul	Maltepe	1997-12-02	KADIN	21
5	549	Çınar BİTGEN	İstanbul	Ümraniye	1981-01-01	ERKEK	37
6	302	Eda TIC.LTD.STİ	İstanbul	Büyükçekmece	1941-09-18	KADIN	77
7	653	Feyza GÜNES	İstanbul	Ataşehir	1944-08-17	KADIN	74
8	33	Fikret DENİZALP	İstanbul	Ümraniye	1995-03-17	ERKEK	23
9	169	Gökçe KATI AMERİT	İstanbul	Ataşehir	1991-01-20	KADIN	27

TOP Komutu (limit, rownum)

--en üstteki 10 kaydı getir

```
SELECT TOP 10 * FROM CUSTOMER
```

----en üstteki 10 kaydı getir isme sırala

```
SELECT TOP 10 * FROM CUSTOMER  
ORDER BY CUSTOMERNAME ASC
```

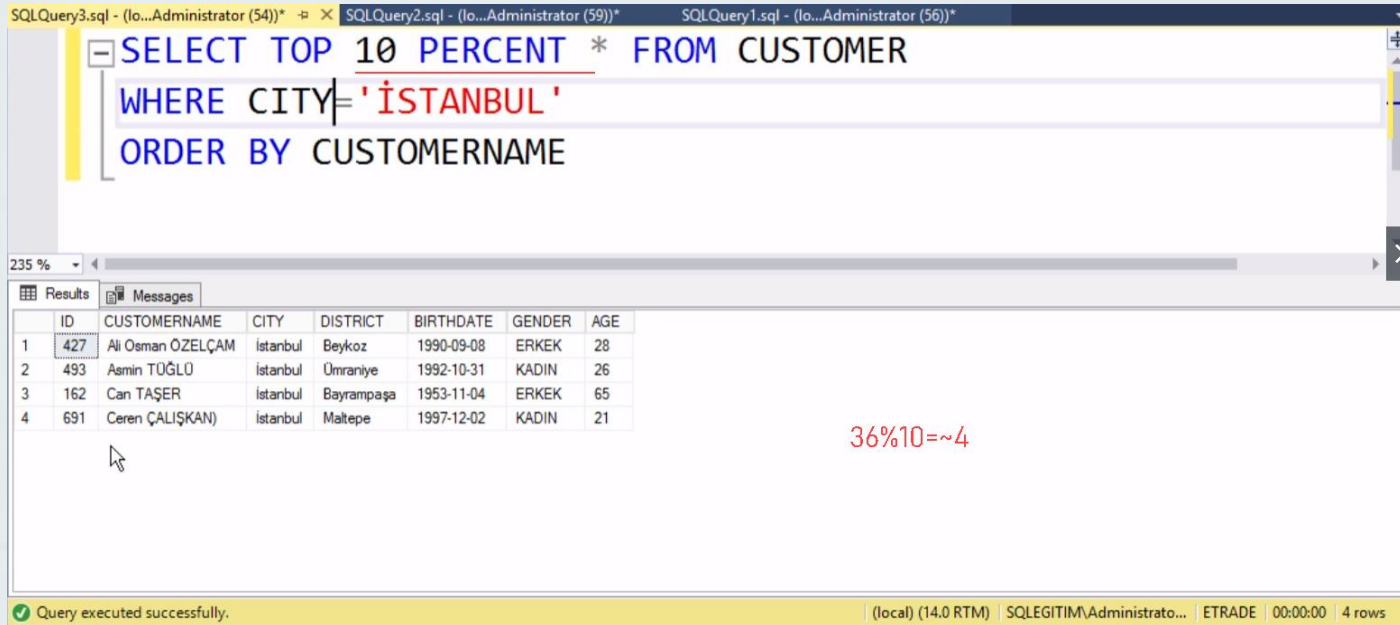
```
SELECT * FROM CUSTOMER
```

```
ORDER BY CUSTOMERNAME ASC |
```

TOP Komutu (limit, rownum)

```
--şehri Çorum olanları en üstteki 10 kişi
SELECT TOP 10 * FROM CUSTOMER
WHERE CITY='ÇORUM'
--
SELECT TOP 10 * FROM CUSTOMER
WHERE CITY='ÇORUM'
ORDER BY CUSTOMERNAME DESC
--en üstteki 25 kişinin ad soyadı ve şehir
SELECT TOP 25 CUSTOMERNAME,CITY FROM CUSTOMER
--en yaşlı 25 kişi
SELECT TOP 25 CUSTOMERNAME,AGE FROM CUSTOMER
ORDER BY AGE DESC
SELECT * FROM CUSTOMER
--en genç 25 kişi
SELECT TOP 25 CUSTOMERNAME,AGE FROM CUSTOMER
ORDER BY AGE ASC
```

TOP Komutu (limit, rownum)



The screenshot shows a SQL Server Enterprise Manager window with three tabs: SQLQuery3.sql, SQLQuery2.sql, and SQLQuery1.sql. The active tab, SQLQuery3.sql, contains the following SQL query:

```
SELECT TOP 10 PERCENT * FROM CUSTOMER  
WHERE CITY='İSTANBUL'  
ORDER BY CUSTOMERNAME
```

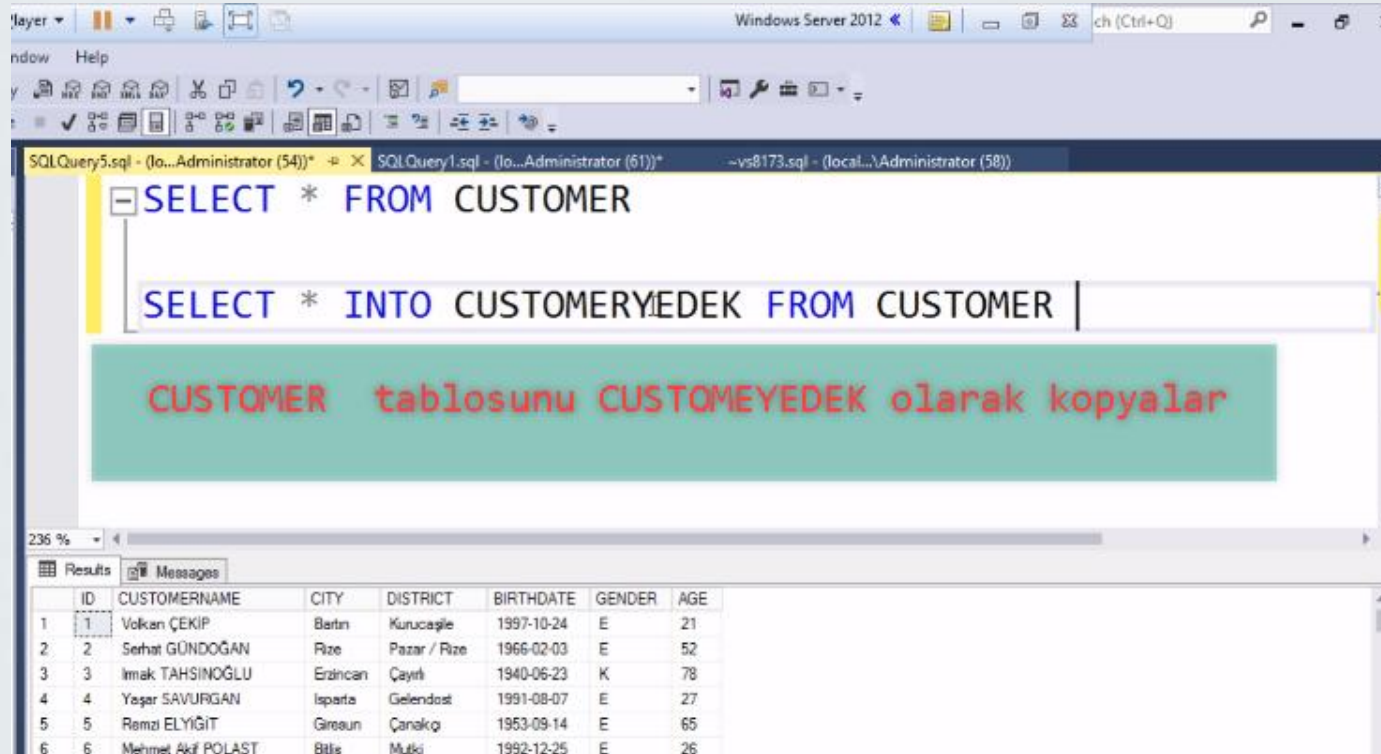
Below the query editor, the 'Results' pane displays the output of the query. The results are shown in a table with the following columns: ID, CUSTOMERNAME, CITY, DISTRICT, BIRTHDATE, GENDER, and AGE. The table contains 4 rows of data.

ID	CUSTOMERNAME	CITY	DISTRICT	BIRTHDATE	GENDER	AGE	
1	427	Ali Osman ÖZELÇAM	İstanbul	Beykoz	1990-09-08	ERKEK	28
2	493	Amin TUĞLU	İstanbul	Umraniye	1992-10-31	KADIN	26
3	162	Can TAŞER	İstanbul	Bayrampaşa	1953-11-04	ERKEK	65
4	691	Ceren ÇALIŞKAN	İstanbul	Maltepe	1997-12-02	KADIN	21

Below the table, a red handwritten note reads: $36\%10 \sim 4$.

At the bottom of the window, a status bar indicates: 'Query executed successfully. (local) (14.0 RTM) | SQLLEGITIM\Administrato... | ETRADE | 00:00:00 | 4 rows'.

Bir Tabloyu Kopyalama



The screenshot shows the SQL Server Enterprise Manager interface. The SQL Query window displays the following query:

```
SELECT * FROM CUSTOMER
```

Below the query, a green box contains the text: **CUSTOMER tablosunu CUSTOMEYEDEK olarak kopyalar**

The Results pane shows the data from the CUSTOMER table:

ID	CUSTOMERNAME	CITY	DISTRICT	BIRTHDATE	GENDER	AGE
1	Volkan ÇEKİP	Bartın	Kurucagile	1997-10-24	E	21
2	Serhat GÜNDOĞAN	Rize	Pazar / Rize	1966-02-03	E	52
3	İmrek TAHSİNOĞLU	Erdincan	Çayırh	1940-06-23	K	78
4	Yaşar SAVURGAN	İsparta	Gelendost	1991-08-07	E	27
5	Remzi ELYİĞİT	Giresun	Çanakçı	1953-09-14	E	65
6	Mehmet Akif POLAST	Bitlis	Mutki	1992-12-25	E	26

Aritmetiksel Operatörler

- **SELECT** adi,soyadi, maas + maas*10/100 zamliMaas
FROM tbl_personel
- **select** 3*5
- **select** 5+3
- **select** 3*5 ,3+5, 3/5
- **select** 3*5 carpım ,3+5 toplam

Fonksiyonlar

- Kullanıcıya hesaplama ve dönüşüm gibi işlemlerde yardımcı olması için yazılmış hazır fonksiyonlar sql de mevcuttur.
 - getdate() → bir fonksiyondur.
- Karakter Kümesi Fonksiyonları
- Sayısal Değerler İçin Fonksiyonlar
- Tarih ve Saat Fonksiyonları
- Dönüşüm Fonksiyonları

Fonksiyonlar

SQLQuery5.sql - D...L2N63DT\mypc (74))*

```
1  
2 SELECT 3+4  
3 SELECT 3*4  
4 SELECT 3*4,3+4,9/2  
5 --HAZIR FONKSİYONLAR---  
6 SELECT GETDATE()  
7 SELECT SYSDATETIME()
```

132 %

Results Messages

	(No column name)
1	2024-12-15 10:05:15.120

	(No column name)
1	2024-12-15 10:05:15.1217995

AGGREGATE FUNCTIONS (SUM,COUNT,MIN,MAX,AVG)

```
SELECT  
SUM(PRICE),COUNT(ID),MIN(PRICE),MAX(PRICE),  
AVG(PRICE)  
FROM TABLOADI
```

Sayısal Değerler İçin Fonksiyonlar -4

➤ Toplam Bulma – SUM()

- SUM() fonksiyonu bir sütun için toplam almakta kullanılır.
- Yanına bir sütun daha yazılabilmesi için ya SUM() benzeri bir fonksiyonla kullanılmalıdır yada GROUP BY deyimi ile gruplandırılmalıdır.
- `SELECT SUM(maas) FROM tbl_personel`
- `SELECT SUM(maas) FROM tbl_personel WHERE gorevi= 'Mühendis'`
- `SELECT adi,soyadi,maas, SUM(maas)`
`FROM tbl_personel` → şeklindeki kullanım hata verecektir.
- `SELECT SUM(borc) toplamBorc, SUM(alacak) toplamAlacak`
`FROM tbl_musteri`
- `SELECT SUM(borc) - SUM(alacak) toplamNet, SUM(borc-alacak) net`
`FROM tbl_musteri`

Sayısal Değerler İçin Fonksiyonlar -4

```
--HAZIR FONKSİYONLAR---
SELECT GETDATE()
SELECT SYSDATETIME()
-- AGGREGATE FUNCTIONS :
--SUM(): kayıtların toplamını verir.
SELECT * FROM CUSTOMER
-- yaşları toplamı
SELECT SUM(AGE) FROM CUSTOMER
--İstanbulluların yaşları toplamı
SELECT SUM(AGE) FROM CUSTOMER
WHERE CITY='İstanbul'
--cinsiyeti erkek olanların yaşları toplamı
SELECT SUM(AGE) FROM CUSTOMER
WHERE GENDER='Erkek'
--cinsiyeti kadın olanların yaşları toplamı
SELECT SUM(AGE) FROM CUSTOMER
WHERE GENDER='Kadın'
```

Sayısal Değerler İçin Fonksiyonlar -5

➤ Ortalama Değer Hesaplama – AVG()

- AVG() fonksiyonu bir sütun için ortalama değeri almak için kullanılır.
- Kullanım biçimi sum ile aynıdır.

➤ `SELECT AVG(maas) FROM tbl_personel`

➤ `SELECT AVG(maas) FROM tbl_personel WHERE gorevi='Mühendis'`

➤ `SELECT SUM(maas) toplamOdenen , AVG(maas) ortalamaMaas
FROM tbl_personel`

Sayısal Değerler İçin Fonksiyonlar -5

```
22 --AVG():ortalmasını verir
23 --yaşları ortalaması
24 SELECT AVG(AGE) FROM CUSTOMER
25 SELECT AVG(AGE) AS[Yaş Ortalaması] FROM CUSTOMER
26 SELECT SUM(AGE) AS[Toplam], AVG(AGE) AS[Ortalama] FROM CUSTOMER
```

% ▾

Results Messages

(No column name)

47

Yaş Ortalaması

47

Toplam Ortalama

47276

47

Sayısal Değerler İçin Fonksiyonlar -6

- **En Büyük ve En Küçük Değer Bulma – MAX() ve MIN()**
 - Bir sütunun içerdiği;
en büyük değeri bulmak için MAX() fonksiyonu ,
en küçük değeri bulmak için MIN() fonksiyonu kullanılır.
 - Sayısal veri tiplerinde en büyük yada en küçük değeri bulurlar. Text veri tipinde ascii koduna göre en son yada en ilk gelene göre hesaplar.
 - Kullanım biçimi sum ve avg ile aynıdır.
- `SELECT MAX(maas) FROM tbl_personel`
- `SELECT MIN(maas) FROM tbl_personel`
- `SELECT SUM(maas) toplamOdenen , AVG(maas) ortalamaMaas
 , MIN(maas) enDusukMaas, MAX(maas) enYuksekMaas
FROM tbl_personel`

Sayısal Değerler İçin Fonksiyonlar -6

```
33 SELECT COUNT(*) AS [Adet],  
34 SUM(AGE) AS [Toplam],  
35 AVG(AGE) AS [Ortalama],  
36 MIN(AGE) AS [Min],  
37 MAX(AGE) AS [Max]  
38 FROM CUSTOMER
```

120 %

Results Messages

(No column name)	
1	78

	Adet	Toplam	Ortalama	Min	Max
1	996	47276	47	19	78

AGGREGATE FONKSİYONLARI

BİRLEŞTİRME (GRUPLAMA) FONKSİYONLARI (AGGREGATE FUNCTIONS)

Tek satır fonksiyonları tablonun bir satırına uygulanıp buna karşılık gelen bir sonuç satırı elde ediliyordu. Bir grup satıra bir fonksiyonun uygulanmasının söz konusu olduğu durumlara; "**çoklu satır**" veya "**grup fonksiyonları**" adı verilir. Grup fonksiyonları tablonun tüm satırlarına uygulanabilmektedir.

AVG()	Bu fonksiyon, herhangi bir sütunun içerdiği sayısal değerlerin aritmetik ortalamasını hesaplamak amacıyla kullanılır. Fonksiyonun uygulandığı sütunun veri türü sayısal olmalıdır. Örnek: SELECT AVG(NOTLAR) AS ORTALAMA FROM OGRENCİ
SUM()	Sütunların içerdiği sayısal değerleri toplamak amacıyla kullanılan fonksiyondur. Örnek: SELECT SUM(NOTLAR) AS TOPLAM FROM OGRENCİ
STDEV()	Standart sapma, sayısal verilerin aritmetik ortalamalardan farklarının kareli ortalaması olarak bilinir. Bu hesaplamayı yapan SQL fonksiyonu ise; STDEV() 'dir. Örnek: SELECT STDEV(NOTLAR) AS "St SAPMA" FROM OGRENCİ
VARP()	Sorgunun belirtilen alanında bulunan değerler kümesiyle temsil edilen bir grubun tahmini varyansını gösterir. Varyans, sayısal değerler arasındaki değişkenliği ölçen bir kavramdır. Standart sapmanın karesi olarak bilinmektedir. Örnek: SELECT VARP(MAAŞ) AS VARYANS FROM PERSONEL
MAX() ve MIN()	Tablodaki değerler arasında en büyük olanı bulmak için MAX() , en küçük olanını bulmak içinse MIN() ve fonksiyonları kullanılır. Örnek: SELECT MAX(NOLAR) AS "EN YÜKSEK" FROM ÖĞRENCİ Örnek: SELECT MIN(NOTLAR) AS "EN DÜŞÜK" FROM ÖĞRENCİ
COUNT()	Bir tablodaki kayıtların sayılması amacıyla kullanılan fonksiyondur. COUNT(*) fonksiyonu, NULL değerleri de içeren tüm kayıtların sayılmasını sağlar. WHERE ile birlikte kullanılırsa, bu koşula uygun tüm kayıtları sayar. COUNT(sütun) biçiminde kullanılırsa, söz konusu sütunda NULL değerler içermeyen tüm kayıtların sayılmasına neden olur. Örnek: SELECT COUNT(*) AS "ÖĞRENCİ SAYISI" FROM ÖĞRENCİ

AGGREGATE FONKSİYONLARI

```
SELECT SUM(LINENET), COUNT(*), MIN(LINENET), MAX(LINENET)  
FROM SALES WHERE BRANCH='Kocaeli Subesi'
```

235 %

	(No column name)	(No column name)	(No column name)	(No column name)
1	53151.0199999995	14292	0.01	162.22

SQLQuery4.sql - (lo...Administrator (61))* SQLQuery3.sql - (lo...Administrator (54))* SQLQuery2.sql - (lo...Administrator (59))* SQLQuery1.sql - (lo...Administrator (56))*

```
SELECT SUM(LINENET) AS TOPLAMSATIS,  
COUNT(*) AS SATIRSAYISI, MIN(LINENET), MAX(LINENET)  
FROM SALES WHERE BRANCH='İSTANBUL Subesi'
```

235 %

	(No column name)	(No column name)	(No column name)	(No column name)
1	422170.479999999	113622	0.01	1219.66

AGGREGATE FONKSİYONLARI

```
- SELECT COUNT(*) AS [Toplam Kayıt Sayısı]  
  FROM MUSTERI  
  
- SELECT COUNT(DISTRICT) AS [Semt Kayıt Sayısı]  
  FROM MUSTERI
```

sults Messages

Toplam Kayıt Sayısı

996

tablodaki tüm satırlar

Semt Kayıt Sayısı

994

2 adet NULL olan var

AGGREGATE FONKSİYONLARI

```
39 --yaşı 45 ten büyük olanlarının sayısı
40 SELECT COUNT(*) AS [Yaşı 45 ten Büyük]
41 FROM CUSTOMER
42 WHERE AGE>45
```

120 %

Results Messages

	Yaşı 45 ten Büyük
1	534

SQLQuery1.sql - La...TMEN\ogrenci (52))* X

```
USE ETRADE
GO
SELECT COUNT(*) [Kadın Çalışan Sayısı]
FROM MUSTERI
WHERE GENDER<>'Erkek'
```

100 %

Results Messages

	Kadın Çalışan Sayısı
1	527

AGGREGATE FONKSİYONLARI

```
SELECT ID,CUSTOMERNAME  
FROM MUSTERI  
WHERE AGE >  
(SELECT AVG(AGE)  
FROM MUSTERI  
)
```

.21 %

Results Messages

	ID	CUSTOMERNAME
1	3	Irmak TAHSİNOĞLU
2	5	Remzi ELYİĞİT
3	8	Münevver AYAOKU
4	10	Nazlıcan ÖZSİMİTÇİ
5	11	Arya UNLUMAMULERİ
6	14	Soner ÜLGEN
7	15	Çetin BÖRKLÜ
8	18	Aykut SUYUR
9	21	Berkay PİRİNÇAL

```
SELECT ID,CUSTOMERNAME  
FROM MUSTERI  
WHERE AGE >  
(SELECT AVG(AGE)+10  
FROM MUSTERI  
)
```

121 %

Results Messages

	ID	CUSTOMERNAME
1	3	Irmak TAHSİNOĞLU
2	5	Remzi ELYİĞİT
3	8	Münevver AYAOKU
4	10	Nazlıcan ÖZSİMİTÇİ
5	11	Arya UNLUMAMULERİ
6	14	Soner ÜLGEN
7	15	Çetin BÖRKLÜ

AGGREGATE FONKSİYONLARI

```
53 --yaşı ortalamaya eşit olanları getir.
54 SELECT ID,CUSTOMERNAME,DISTRICT,AGE
55 FROM CUSTOMER
56 WHERE AGE=(SELECT AVG(AGE) FROM CUSTOMER)
57 --yaşı ortalamasının 10 üzerinde olanları getir.
58 SELECT ID,CUSTOMERNAME,DISTRICT,AGE
59 FROM CUSTOMER
60 WHERE AGE>(SELECT AVG(AGE) FROM CUSTOMER)+10
```

ults Messages

ID	CUSTOMERNAME	DISTRICT	AGE
70	Cuma KAZIKLI	Dalaman	47
77	Özcan KÜÇÜKİRAVUL	Amasya Merkez	47
142	Raziye ERGÜZER	Onikişubat	47
184	Doğan UZATMA	Kuyucak	47
297	Halil İbrahim BİMBİRDİREK	Sason	47
449	İrfan FAKOĞLU	Gölbaşı / Adıyaman	47
457	Şengül SABUNLARI	Şahinbey	47
469	Zeliha BADIL	Suşehri	47

ID	CUSTOMERNAME	DISTRICT	AGE
3	Irmak TAHSİNOĞLU	Çayırli	78
5	Remzi ELYİĞİT	Çanakçı	65
8	Münevver AYAOKU	Artvin Merkez	64
10	Nazlıcan ÖZSİMİTÇİ	Viranşehir	67
11	Arya UNLUMAMULERİ	Terme	61
14	Soner ÜLGİN	Alacakaya	78
15	Çetin BÖRKLÜ	Çifteler	72
22	Anıl GÜLDÜ	Gündoğmuş	77
25	Muzaffer AĞACKES	Fake	68

AGGREGATE FONKSİYONLARI

```
--SELECT SUM(AGE) AS [Yaşlar Toplamı]
--FROM MUSTERI
--
--COUNT(*) : satır sayısını verir
--SELECT COUNT(*) AS [Toplam Kayıt Sayısı]
--FROM MUSTERI
--
--SELECT COUNT(DISTRICT) AS [Semt Kayıt Sayısı]
--FROM MUSTERI
--
--AVG:AVERAGE:sütun ortalama verir
--yaşlar ortalamasını bulalım
--SELECT AVG(AGE) AS [Yaş Ortalaması]
--FROM MUSTERI
--
--SELECT MIN(AGE) FROM MUSTERI
--en küçük yaş değerini bulalım
--SELECT MAX(AGE) FROM MUSTERI
```

```
SELECT AVG(PRICE*AMOUNT) AS [Tutar Ortalama]
FROM SALES
```

100 % <

Results Messages

	Tutar Ortalama
1	4,05566654621863

AGGREGATE FONKSİYONLARI

GROUP BY, kullanarak belirtilen alan listesindeki benzer değerlere sahip kayıtları tek bir kayıt olarak birleştirmede kullanılır. Yani birden fazla sütun için de gruptama yapılabilir.

```
SELECT ITEMCODE, COUNT(*) AS [Satılan Adet Sayısı]  
FROM SALES  
WHERE STARTDATE BETWEEN '2017-01-03' AND '2017-01-04'  
GROUP BY ITEMCODE
```

121 %

Results

Messages

	ITEMCODE	Satılan Adet Say...
1	00000000011	2
2	00000000015	1
3	00000000019	1
4	00000000041	1
5	00000000218	3
6	00000000235	1

AGGREGATE FONKSİYONLARI

SQLQuery7.sql - D...L2N63DT\mypc (66))*

```
5 SELECT * FROM SALES
6 --
7 SELECT BRANCH, COUNT(*) AS[Satılan Adet]
8 FROM SALES
9 WHERE BRANCH LIKE '%istanbul%'
10 GROUP BY BRANCH
11 SELECT BRANCH, COUNT(*) AS[Satılan Adet]
12 FROM SALES
13 WHERE BRANCH LIKE '%çorum%'
14 GROUP BY BRANCH
```

120 %

Results Messages

	BRANCH	Satılan Adet
1	İstanbul Subesi	113622

	BRANCH	Satılan Adet
1	Çorum Subesi	4227

AGGREGATE FONKSİYONLARI

HAVING, grup işlemlerinin uygulanması sırasında birtakım sınırlamalar gerekebilir. Grup koşulları belirlenirken **HAVING** sözcüğü kullanılır. **HAVING** sözcüğü grupta yapıldıktan sonra kullanılır. Gruplandırmak istenilmeyen satırları dışarıda tutmak için **WHERE**, gruplandırılan kayıtlara filtre uygulamak için ise **HAVING** kullanılır.

Group By ifadesi Where ifadesinden sonra, Having ifadesinden önce kullanılır.

Order By ifadesi ise en son kullanılır. Kayıtlar **GROUP BY** ile gruplandırılır ve **HAVING** ile de hangi kayıtların görüntüleneceği gösterilir.

AGGREGATE FONKSİYONLARI

SQLQuery1.sql - (I...RT\LAB2-OGRT (57))* x

--Toplam satışı 50.000 den fazla olan mağazalar

```
SELECT BRANCH, SUM(LINENET) AS [TOPLAM SATIŞ]  
FROM SALES  
GROUP BY BRANCH  
HAVING SUM(LINENET) > 50000
```

100 %

Results Messages

	BRANCH	TOPLAM SATIŞ
1	Adana Subesi	59889,1199999998
2	Ankara Subesi	153768,4399999999
3	Antalya Subesi	69754,8399999998
4	Bursa Subesi	74875,2299999999
5	Diyarbakır Subesi	52091,2099999999
6	Gaziantep Subesi	52746,3799999999
7	İstanbul Subesi	422170,4799999998
8	İzmir Subesi	121346,4999999999
9	Kocaeli Subesi	53151,0199999999
10	Konya Subesi	58842,9599999998
11	Şanlıurfa Subesi	53578,5199999999
12	Zonguldak Subesi	53098,9899999999

AGGREGATE FONKSİYONLARI

SQLQuery9.sql - D...L2N63DT\mypc (67)* X Hafta14_group_by....2N63DT\mypc (66))

```
1 --toplam satışı 50.00 0 den fazla şubeler
2 SELECT * FROM SALES
3 SELECT BRANCH,SUM(LINET) AS [TOPLAM
4   SATIŞ]
5 FROM SALES
```

120 %

Results Messages

	BRANCH	TOPLAM SATIŞ
1	Batman Subesi	15619,77
2	Edirne Subesi	8837,730000000001
3	Adıyaman Subesi	16761,309999999999
4	Karabük Subesi	6589,23
5	Kastamonu Subesi	10146,03
6	Tunceli Subesi	2375,75
7	İzmir Subesi	121346,5000000001
8	Bingöl Subesi	7913,820000000001
9	Antalya Subesi	69754,84000000003
10	İstanbul Subesi	422170,4799999978
11	Düzce Subesi	8962,410000000002
12	Kayseri Subesi	40039,30999999997
13	Burdur Subesi	6872,030000000001
14	Kars Subesi	6999,050000000001
15	Konya Subesi	58842,95999999999

Query executed succ... | DESKTOP-L2N63DT (16.0 RTM) | DESKTOP-L2N63DT\mypc (67) | ETRADE | 00:00:00 | 81 rows

AGGREGATE FONKSİYONLARI

```
--Toplam satışı ortalamadan fazla olan mağazalar  
SELECT BRANCH, SUM(LINENET) AS [TOPLAM SATIŞ]  
FROM SALES  
GROUP BY BRANCH  
HAVING SUM(LINENET) > AVG(LINENET)
```

100 %

Results Messages

	BRANCH	TOPLAM SATIŞ
1	Adana Subesi	59889,1199999999
2	Adıyaman Subesi	16761,31
3	Afyonkarahisar Subesi	18123,09
4	Ağrı Subesi	16053,43
5	Aksaray Subesi	11372,24
6	Amasya Subesi	10372,69
7	Ankara Subesi	153768,4399999999
8	Antalya Subesi	69754,83999999996
9	Ardahan Subesi	2633,52

Karakter Kümesi Fonksiyonları

- **+** (Birleştirme Operatörü)
- **SUBSTRING()**
- **LEFT()** ve **RIGHT()**
- **LOWER()** ve **UPPER()**
- **LTRIM()** ve **RTRIM()**
- **LEN()**
- **REPLACE()**
- **CHARINDEX()**
- **REVERSE()**

Karakter Kümesi Fonksiyonları -1

- İki Karakter Kümesini Birleştirmek

- **SELECT** dersKod +' '+ dersAd +' '+ dersVeren dersBilgileri
FROM tbl_ders

- **SUBSTRING()** Verininin İstenilen Bölümünü Almak

- Karakter kümesi içerisinde verinin istenilen bölümünün alınması için kullanılır.

- Kullanımı Substring(veri_kümesi, başlangıç, karakter_sayısı)

- **SELECT** Substring('Ali Coşkun bilgisayar',5,6) → Coşkun

- **SELECT** ogrNo, ad, soyad , SUBSTRING(ad,1,1)+soyad eposta
FROM tbl_ogrenci

Karakter Kümesi Fonksiyonları -1

```
14 SELECT * FROM CUSTOMER
15 SELECT SUBSTRING(CUSTOMERNAME,1,3) FROM CUSTOMER
16 SELECT SUBSTRING(CITY,1,3)+'-'+SUBSTRING(DISTRICT,1,3)
17 FROM CUSTOMER
```

%

Results Messages

(No column name)

Erz-Çay

Isp-Gel

Gir-Çan

Bit-Mut

Ord-Çat

Art-Art

Ayd-Ger

Şan-Vir

Sam-Ter

Küt-Eme

Sam-Lad

Karakter Kümesi Fonksiyonları - 2

Verinin İstenilen Bölümünü Almak – LEFT() ve RIGHT()

- LEFT() başlangıcından itibaren kaç karakter alınacağını belirler.
- RIGHT() verinin sondan kaç karakter alınacağını belirler.
- Kullanımı
LEFT(bilginin_alınacağı_yer,karakter_sayısı)
RIGHT(bilginin_alınacağı_yer,karakter_sayısı)
- **SELECT** ogrNo, ad, soyad , LEFT(ad,1)+RIGHT(ad,1)
isminBasHarfiSonHarfi
FROM tbl_ogrenci

Karakter Kümesi Fonksiyonları - 3

- **Büyük-Küçük Harf Dönüşümü - LOWER() ve UPPER()**
 - LOWER() küçük harfe dönüştürür, UPPER() büyük harfe dönüştürür.
 - Kullanımı
LOWER(dönüştürülecek_bilgi)
UPPER(dönüştürülecek_bilgi)
- **SELECT** ad, soyad , LOWER(ad) adKucuk, UPPER(soyad) soyadBuyuk
FROM tbl_ogrenci

Karakter Kümesi Fonksiyonları -4

➤ Boşlukları Kaldırmak – LTRIM() ve RTRIM()

- LTRIM() verinin başındaki boşlukları kaldırır.
- RTRIM() verinin sonundaki boşlukları kaldırır.
- Kullanımı LTRIM(bilgi)

RTRIM (bilgi)

➤ `SELECT LTRIM(' Ali Coşkun '), RTRIM(' Ali Coşkun ')`

➤ `SELECT LTRIM(RTRIM(' Ali Coşkun '))` her iki yandaki boşluk

```
1
2 SELECT ' ADEM AKUŞ '
3 SELECT LTRIM(' Adem AKKUŞ ')
4 SELECT RTRIM(' Adem AKKUŞ ')
```

20 %

Results Messages

	(No column name)
1	ADEM AKUŞ
1	Adem AKKUŞ
1	Adem AKKUŞ

Karakter Kümesi Fonksiyonları -5

➤ Karakter Sayısını Bulmak – LEN()

- LEN() veride bulunan karakter sayısını döndürür.
- Kullanımı LEN(bilgi)

➤ `SELECT LEN('Ali Coşkun') → 10`

➤ `SELECT ogrNo, ad, soyad , LEN(ad) adUz, LEN(soyad) soyadUz
FROM tbl_ogrenci`

Karakter Kümesi Fonksiyonları -5

```
9 --en fazla karakterden azalana doğru
10 SELECT LEN(CUSTOMERNAME) AS[Karakter Sayısı]
11 FROM CUSTOMER
12 ORDER BY LEN(CUSTOMERNAME) DESC
```

120 %

Results Messages

	Karakter Sayısı
1	49
2	45
3	40
4	33
5	31
6	31
7	30
8	29
9	28
10	27
11	27

Karakter Kümesi Fonksiyonları -5

```
9  --en fazla karakterden azalana doğru
10 SELECT LEN(CUSTOMERNAME) AS[Karakter Sayısı]
11 FROM CUSTOMER
12 ORDER BY LEN(CUSTOMERNAME) DESC
13 --en azdan karakterden artana doğru
14 SELECT LEN(CUSTOMERNAME) AS[Karakter Sayısı]
15 FROM CUSTOMER
16 ORDER BY LEN(CUSTOMERNAME) ASC
17
```

Results Messages

Karakter Sayısı

8

9

9

9

9

9

9

9

9

9

9

9

9

9

9

Karakter Kümesi Fonksiyonları -6

- **İstenilen Bilgiyi Değiştirmek – REPLACE()**
 - REPLACE() verideki değiştirilecek değerin yerine yeni değeri yazar.
 - Kullanımı REPLACE(sutun_adı, değişecek_bilgi, yeni_bilgi)
- **SELECT REPLACE**('Ali Coşkun','Ali', 'Veli') → Veli Coşkun
- **SELECT** adi, soyadi ,**REPLACE**(gorevi,'Tekniker', 'Teknisyen')
unvan
FROM tbl_personel

Karakter Kümesi Fonksiyonları -7

➤ Karakter Aramak – CHARINDEX()

- **CHARINDEX()** text verinin içerisinde başka bir karakter kümesinin yerinin bulunması için kullanılır. Geriye başlangıç noktasını döndürür.
- Kullanımı **CHARINDEX(aranacak_bilgi, arama_yeri, başangıç_nokt)**
- **SELECT CHARINDEX('Coş','Ali Coşkun Coşkun',0) →5**

➤ Tersten Yazdırma – REVERSE()

- **REVERSE()** aldığı veriyi tersten yazdırır.
- Kullanımı **REVERSE(tersten_yazılacak_veri)**
- **SELECT REVERSE('Ali Coşkun') → nukşoC ilA**

Sayısal Değerler İçin Fonksiyonlar

- **ABS()**
- **POWER()**
- **SQRT()**
- Bölümden Kalanı Bulma (%)
- **ROUND()**
- **FLOOR()** ve **CEILING()**

Sayısal Değerler İçin Fonksiyonlar -1

➤ Bir Sayının Mutlak Değerini Alma– ABS()

➤ `SELECT ABS(-1.25) → 1.25`

➤ Bir Sayının Kuvvetini Alma– POWER()

➤ `POWER(x,y) --> X^Y`

➤ `SELECT POWER(2,3) → 2^3 → 8`

➤ Bir Sayının Karekökünü Alma– SQRT()

➤ `SELECT SQRT(4) → 2`

➤ Bir Sayının Başka Bir Sayıya Bölümünden Kalanı Bulma– %

➤ Kullanımı `X % Y`

➤ `SELECT 4 % 2 → 0`

➤ `SELECT 23 % 10 → 3`

Sayısal Değerler İçin Fonksiyonlar -2

➤ En Yakın Tamsayıya Yuvarlama– ROUND()

- ROUND() aldığı ondalık değeri aldığı basamak sayısına göre aşağı veya yukarı yuvarlar.
- Kullanımı ROUND(ondalık_değer, basamakSayısı)

- SELECT ROUND(2.5,0) →3
- SELECT ROUND (2.4,0) →2
- SELECT ROUND (2.47,1) →2.5
- SELECT ROUND (2.44,1) →2.4
- SELECT ROUND (2.018,2) →2.02
- SELECT ROUND (2.011,2) →2.01

Sayısal Değerler İçin Fonksiyonlar -3

- **Aşağı ve Yukarı Yuvarlama– FLOOR() ve CEILING()**
 - FLOOR() aldığı ondalık değeri her koşulda aşağı yuvarlar.
 - CEILING() aldığı ondalık değeri her koşulda yukarı yuvarlar
- **SELECT FLOOR(2.5) →2**
- **SELECT FLOOR(2.3) →2**
- **SELECT FLOOR(2.99) →2**
- **SELECT CEILING(2.5) →3**
- **SELECT CEILING (2.01) →3**
- **SELECT CEILING (2.9) →3**

Tarih ve Saat Fonksiyonları

- **GETDATE()**
- **DATEPART()**
- **DATENAME()**
- **DATEDIFF()**
- **DATEADD()**

Tarih ve Saat Fonksiyonları -1

- **Güncel Tarih ve Saat Bilgisini Alma– GETDATE()**
 - GETDATE() fonksiyonu veritabanının üzerinde bulunduğu serverin tarih ve saatini alır.
 - **SELECT GETDATE()**
- **Peki ya sadece saati yada tarihi almak istiyorsam.**

Tarih ve Saat Fonksiyonları -2

➤ Verilen Tarihten Tipine Göre Veri Döndürme– DATEPART()

- DATEPART() fonksiyonu verilen tarihten almak istediğimiz kısmı döndürür.
- Kullanımı;

DATEPART(alınacak_bilgi,tarih_saat_bilgisi)

Kısım Adı	Açıklama	Örnek SELECT GETDATE()	Dönen Değer 02.04.2012 13:45:20
YEAR	Yıl bilgisini döndürür	SELECT DATEPART(YEAR, GETDATE())	2012
MONTH	Ay bilgisini döndürür	SELECT DATEPART(MONTH, GETDATE())	04
DAY	Gün bilgisini döndürür	SELECT DATEPART(DAY, GETDATE())	02
HOURL	Saati döndürür.	SELECT DATEPART(HOUR, GETDATE())	13
MINUTE	Dakikayı döndürür.	SELECT DATEPART(MINUTE, GETDATE())	45
SECOND	Saniyeyi döndürür.	SELECT DATEPART(SECOND,GETDATE())	20
WEEK	Yılın kaçınıcı haftası olduğunu döndürür.	SELECT DATEPART(WEEK, GETDATE())	14
QUARTER	Yılın kaçınıcı çeyreğinde olduğunu döndürür	SELECT DATEPART(QUARTER,GETDATE())	2

Tarih ve Saat Fonksiyonları -2

```
--SELECT GETDATE()  
--DATEPART() fonksiyonu  
--YEAR: yıl değerini verir  
SELECT DATEPART(YEAR,GETDATE())  
--MONTH: ay değerini verir  
SELECT DATEPART(MONTH,GETDATE())  
--DAY: gün değerini verir  
SELECT DATEPART(DAY,GETDATE())  
--HOUR: saat değerini verir  
SELECT DATEPART(HOUR,GETDATE())  
--MINUTE: dakika değerini verir  
SELECT DATEPART(MINUTE,GETDATE())  
--SECOND: saniye değerini verir  
SELECT DATEPART(SECOND,GETDATE())  
--MILLISECOND: milisaniye değerini verir  
SELECT DATEPART(MILLISECOND,GETDATE())
```

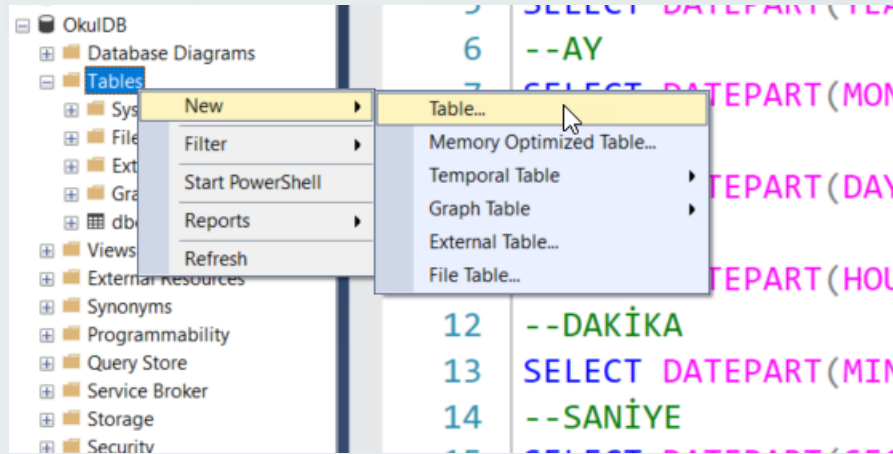
Tarih ve Saat Fonksiyonları -2

```
1  --TARİH FONKSİYONLARI
2  SELECT GETDATE() AS [SİSTEM SAAT]
3  --DATEPART()
4  --YIL
5  SELECT DATEPART(YEAR,GETDATE()) AS [YIL]
6  --AY
7  SELECT DATEPART(MONTH,GETDATE()) AS [AY]
8  --GÜN
9  SELECT DATEPART(DAY,GETDATE()) AS [GÜN]
0  --SAAT
1  SELECT DATEPART(HOUR,GETDATE()) AS [SAAT]
2  --DAKİKA
3  SELECT DATEPART(MINUTE,GETDATE()) AS [DAKİKA]
4  --SANİYE
5  SELECT DATEPART(SECOND,GETDATE()) AS [SANİYE]
6  --MİLİ SANİYE
7  SELECT DATEPART(MILLISECOND,GETDATE()) AS [MİLİ SANİYE]
```

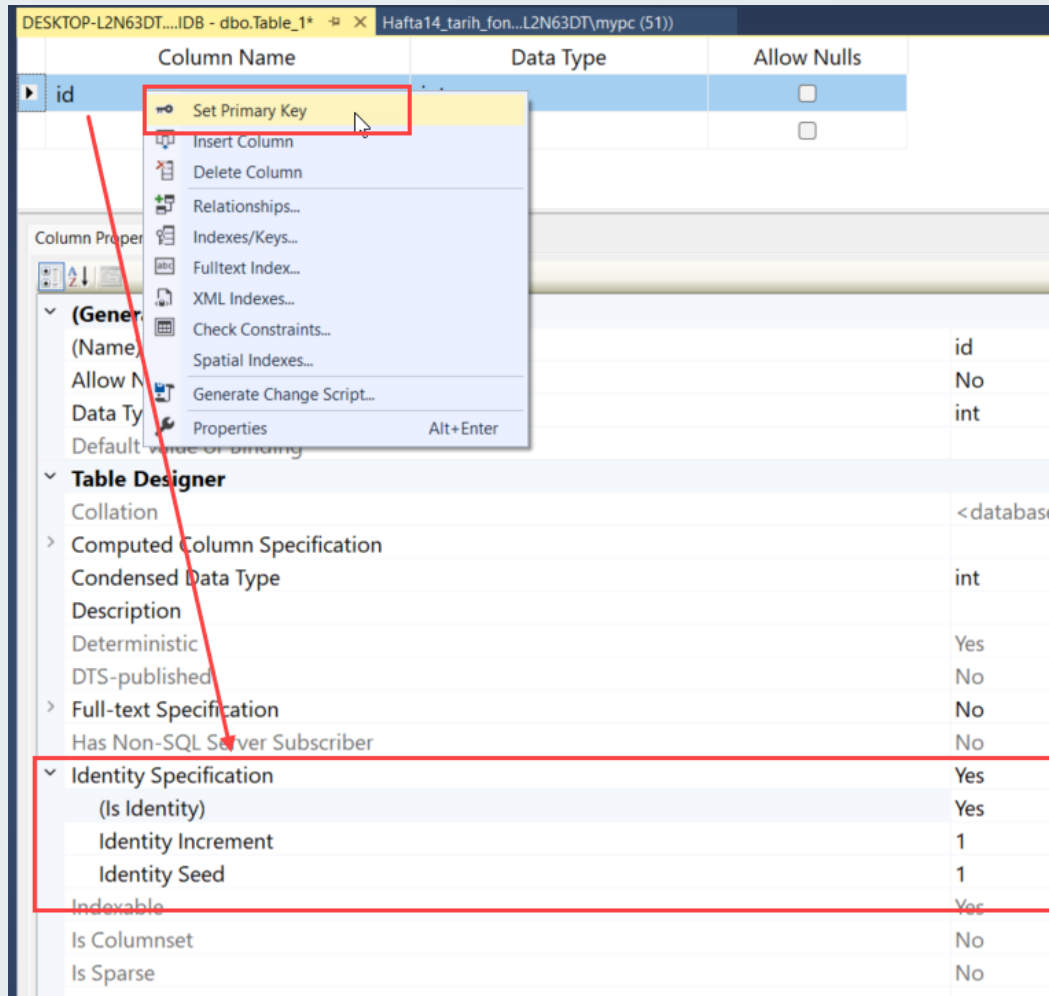
Tarih ve Saat Fonksiyonları -2

Results Messages	
	SİSTEM SAAT
1	2024-12-15 12:30:13.703
	YIL
1	2024
	AY
1	12
	GÜN
1	15
	SAAT
1	12
	DAKİKA
1	30
	SANİYE
1	13
	MİLİ SANİYE
1	703

Tarih ve Saat Fonksiyonları -2



Tarih ve Saat Fonksiyonları -2



Tarih ve Saat Fonksiyonları -2

DESKTOP-L2N63DT....IDB - dbo.Table_1* X Hafta14_tarih_fon...L2N63DT\mypc (51))		
Column Name	Data Type	Allow Nulls
id	int	<input type="checkbox"/>
sistem_saat	datetime	<input type="checkbox"/>
yil	int	<input type="checkbox"/>
ay	int	<input type="checkbox"/>
gun	int	<input type="checkbox"/>
saat	int	<input type="checkbox"/>
dakika	int	<input type="checkbox"/>
saniye	int	<input type="checkbox"/>
milisaniye	int	<input type="checkbox"/>

Choose Name ? X

Enter a name for the table:

Kayit

OK Cancel

Tarih ve Saat Fonksiyonları -2

```
3 INSERT INTO Kayit(  
4   sistem_saat,  
5   yil,ay,  
6   gun,  
7   saat,  
8   dakika,  
9   saniye,  
10  milisaniye)  
11 VALUES(  
12  GETDATE(),  
13  DATEPART(YEAR,GETDATE()),  
14  DATEPART(MONTH,GETDATE()),  
15  DATEPART(DAY,GETDATE()),  
16  DATEPART(HOUR,GETDATE()),  
17  DATEPART(MINUTE,GETDATE()),  
18  DATEPART(SECOND,GETDATE()),  
19  DATEPART(MILLISECOND,GETDATE())  
20 )
```

	id	sistem_saat	yil	ay	gun	saat	dakika	saniye	milisaniye
1	1	2024-12-15 12:40:15.237	2024	12	15	12	40	15	236
2	2	2024-12-15 12:40:27.373	2024	12	15	12	40	27	373
3	3	2024-12-15 12:40:39.623	2024	12	15	12	40	39	623
4	4	2024-12-15 12:40:42.273	2024	12	15	12	40	42	273
5	5	2024-12-15 12:40:44.250	2024	12	15	12	40	44	250

Tarih ve Saat Fonksiyonları -3

- **Verilen Tarihten Tarih Tipi Adını Döndürme– DATENAME()**
 - DATENAME() fonksiyonu verilen tarihten almak istediğimiz kısmın ismini döndürür.
 - Kullanımı; DATENAME(alınacak_bilgi,tarih_saat_bilgisi)

Kısım Adı	Açıklama	Örnek	Dönen Değer
		SELECT GETDATE()	02.04.2012 13:45:20
MONTH	Ay ismini döndürür	SELECT DATENAME(MONTH, GETDATE())	Nisan
WEEKDAY	Gün bilgisini döndürür	SELECT DATENAME(WEEKDAY, GETDATE())	Pazartesi

Tarih ve Saat Fonksiyonları -3

```
SELECT DATENAME(MONTH, GETDATE())  
SELECT DATENAME(MONTH, '2024-02-23')  
--SELECT DATENAME(MONTH, '21-09-2024') --yanlış format  
SELECT DATENAME(WEEK, GETDATE())  
SELECT DATENAME(WEEKDAY, GETDATE())  
SELECT DATENAME(DAYOFYEAR, GETDATE())  
SELECT DATENAME(DAY, GETDATE())  
SELECT DATENAME(YYYY, GETDATE())  
SELECT DATENAME(WW, GETDATE())
```

Results		Messages
	(No column name)	
1	December	
	(No column name)	
1	February	
	(No column name)	
1	51	
	(No column name)	
1	Sunday	
	(No column name)	
1	350	
	(No column name)	
1	15	
	(No column name)	
1	2024	
	(No column name)	
1	51	

Tarih ve Saat Fonksiyonları -4

- İki tarih arasındaki farkı karşılaştırmak– DATEDIFF()
 - DATEDIFF() fonksiyonu verilen iki tarih arasındaki almak istediğimiz bilginin farkını hesaplar.
 - Kullanımı; DATEDIFF(alınacak_bilgi,ilk_tarih,son_tarih)
- SELECT DATEDIFF(YEAR,'01.02.2000', '01.01.2012') →12
- SELECT DATEDIFF(MONTH,'01.01.2012', '02.04.2012') →3
- SELECT DATEDIFF(DAY,'01.01.2012', '02.04.2012') →92

Tarih ve Saat Fonksiyonları -5

➤ Tarihe Belli Değerler Ekleme – DATEADD()

- DATEADD() fonksiyonu belirtilen tarihe belirli gün, ay, yıl vb bilgi eklendiğinde oluşan tarihi hesaplamak için kullanılır.
- Kullanımı; DATEADD(eklenecek_bilgi,eklenme_miktari,tarih)
- `SELECT DATEADD(YEAR,2, '01.01.2012') →01.01.2014`
- `SELECT DATEADD(MONTH,3, '02.04.2012') →02.07.2012`
- `SELECT DATEADD(DAY,29, '02.04.2012') →01.05.2012`

Dönüşüm Fonksiyonları- CAST() ve CONVERT()

Bu işlevler, bir veri türünün ifadesini diğerine dönüştürür.

➤ CAST()

CAST (ifade AS veri_tipi[(uzunluk)])

SELECT CAST('Ali Coşkun' AS NVARCHAR(5)) → Ali C

➤ CONVERT()

CONVERT(veri_tipi [(uzunluk)], ifade [,style])

SELECT CONVERT(INT ,5.45) → 5

SELECT CONVERT(NVARCHAR(5) ,2010) → 2010

Dönüşüm Fonksiyonları- CAST() ve CONVERT()

--değişken tanımlama

```
DECLARE @degisken DECIMAL(5, 2);
```

--değişken değer atama

```
SET @degisken = 193.57;
```

```
SELECT CAST(CAST(@degisken AS VARBINARY(20)) AS DECIMAL(10, 5));
```

--ya da , CONVERT kullanarak

```
SELECT CONVERT(DECIMAL(10, 5), CONVERT(VARBINARY(20), @degisken));
```

```
11 DECLARE @A varchar(2)
```

```
12 DECLARE @B varchar(2)
```

```
13 DECLARE @C varchar(2)
```

```
14 set @A=25
```

```
15 set @B=15
```

```
16 set @C=33
```

```
17 Select CAST(@A as int) + CAST(@B as int) +CAST (@C as int) as Result
```

121 %

Results Messages

	Result
1	73

Dönüşüm Fonksiyonları- CAST() ve CONVERT()

```
19 SELECT CONVERT(VARCHAR(15),GETDATE(),6) AS [Format 1]
20 GO
21 SELECT CONVERT(VARCHAR(16),GETDATE(),106) AS [Format 2]
22 GO
23 SELECT CONVERT(VARCHAR(24),GETDATE(),113) AS [Format 3]
```

121 %

Results

Messages

Format 1

1

01 Jun 23

Format 2

1

01 Jun 2023

Format 3

1

01 Jun 2023 12:42:02:297