

VERİLERİ SORGULAMAK

Tarih / Zaman Veri Tipleri

DATE

- *Tarih tipinden veri saklamaya yarar.*
- *01-01-0001 ile 31-12-9999 arasında tarih değeri alabilir*
- *Formatı; YIL-AY-GÜN*

Time

- *Saat tipinden veri saklamaya yarar. 5 baytlık depolama alanına sahiptir.*
- *Format :*

Saat:Dakika:Saniye:[.Nano saniye]

DateTime

- *Tarih ve saat tipinden veri saklamaya yarar.*
- *8 baytlık veri tipidir. 01-01-1753 ile 31-12-9999 arasında bir tarih değeri alabilir.*
- *Aynı zamanda nanosaniye olarak 3 digit'i desteklemektedir.*
- *Format:*

YYY-MM-DD hh:mm:ss[.nnn]

SmallDateTime-Kısa Tarih Zaman Formatı

- *DateTime veri tipinin daha az kapsamlı halidir.*
- *Nano saniye değerini içermez ve 01-01-1900 ile 06-06-2079 tarihleri arasında değer alabilir. 4 baytlık veri tipidir.*
- *Format:*

YYYY-MM-DD hh:mm:ss

YIL-AY-GÜN SAAT:DAKİKA:SANİYE

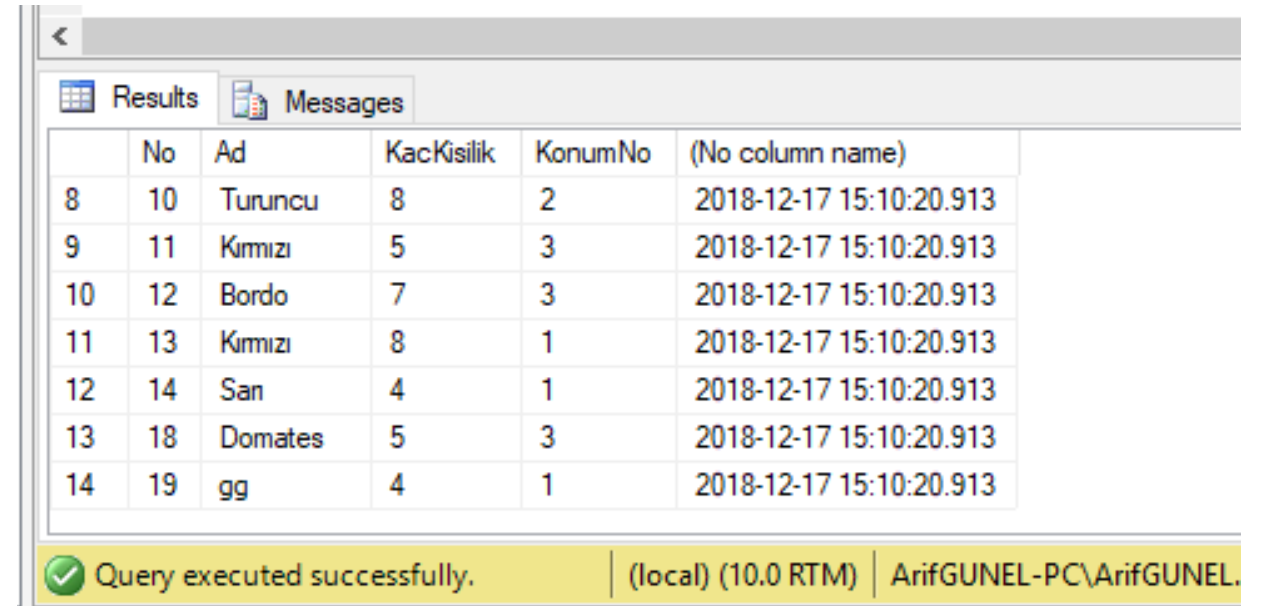
Tarih/Saat Fonksiyonları

- *Veri tabanı işlemlerinde tarih ve saat gibi zaman bilgilerinin önemi yüksektir.*
- *Bu nedenle, SQL Server gibi büyük veri tabanlarında tarih/saat işlemlerini gerçekleştirecek çok sayıda veri tipi ve fonksiyon vardır*

GETDATE-ANLIK TARİH-ZAMAN

- Sistemin anlık tarihini gösterir.
- Sorgunu yapıldığı zamanı veren fonksiyon, her bir kayıt için ayrı ayrı zamanı verir.

```
SELECT *, GETDATE() FROM MASA
```



The screenshot shows a SQL Server query window with the 'Results' tab selected. The query executed is 'SELECT *, GETDATE() FROM MASA'. The results are displayed in a table with 6 columns: 'No', 'Ad', 'KacKisilik', 'KonumNo', and '(No column name)'. The data rows show records 8 through 14 of the MASA table, with the 'GETDATE()' function returning the same timestamp for all rows: '2018-12-17 15:10:20.913'.

	No	Ad	KacKisilik	KonumNo	(No column name)
8	10	Turuncu	8	2	2018-12-17 15:10:20.913
9	11	Kırmızı	5	3	2018-12-17 15:10:20.913
10	12	Bordo	7	3	2018-12-17 15:10:20.913
11	13	Kırmızı	8	1	2018-12-17 15:10:20.913
12	14	San	4	1	2018-12-17 15:10:20.913
13	18	Domates	5	3	2018-12-17 15:10:20.913
14	19	gg	4	1	2018-12-17 15:10:20.913

Query executed successfully. | (local) (10.0 RTM) | ArifGUNEL-PC\ArifGUNEL.

Anlık zaman bilgisinden bilgi çekmek

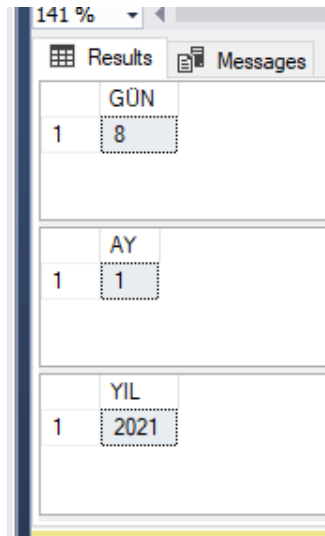
- Mesela anlık zaman bilgisinden gün verisini çekmek istersek;

```
SELECT DAY(GETDATE());
```

```
SELECT MONTH(GETDATE());
```

```
SELECT YEAR(GETDATE());
```

```
SELECT DAY(GETDATE()) GÜN;  
SELECT MONTH(GETDATE()) AY;  
SELECT YEAR(GETDATE()) YIL;
```

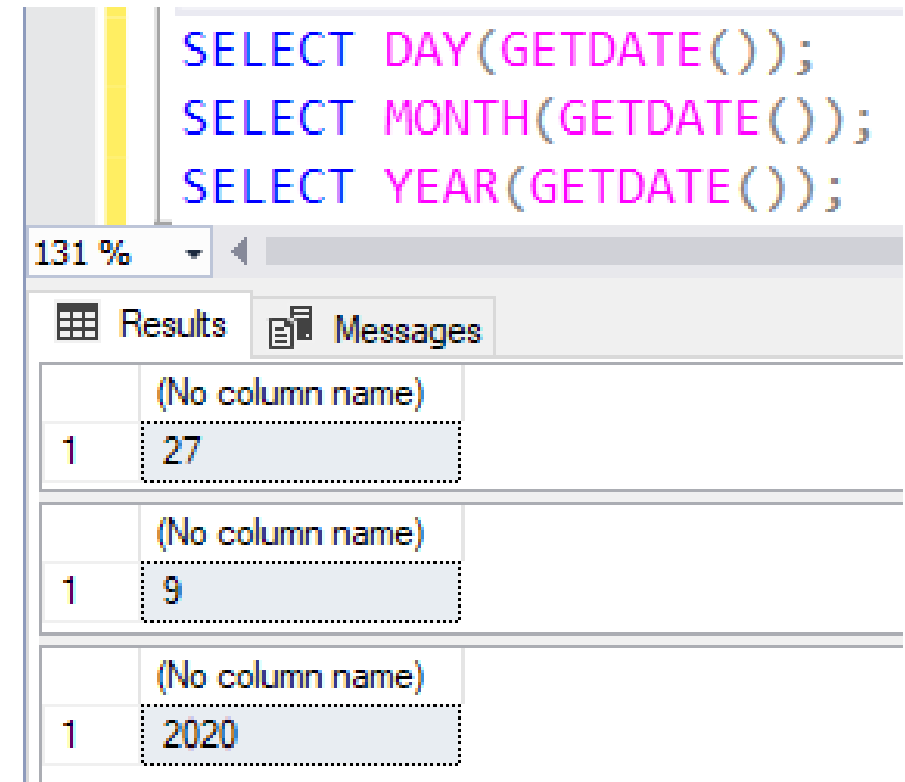


141 %

	GÜN
1	8

	AY
1	1

	YIL
1	2021



131 %

	(No column name)
1	27

	(No column name)
1	9

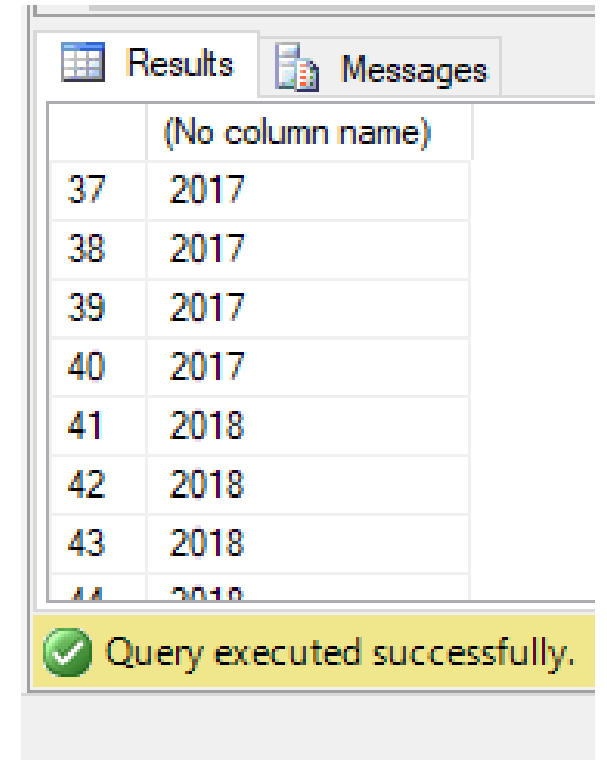
	(No column name)
1	2020

YEAR-YIL

- *Sadece yılı almak için kullanılan fonksiyon, mesela kayıt yılını almak istediğimizde.*

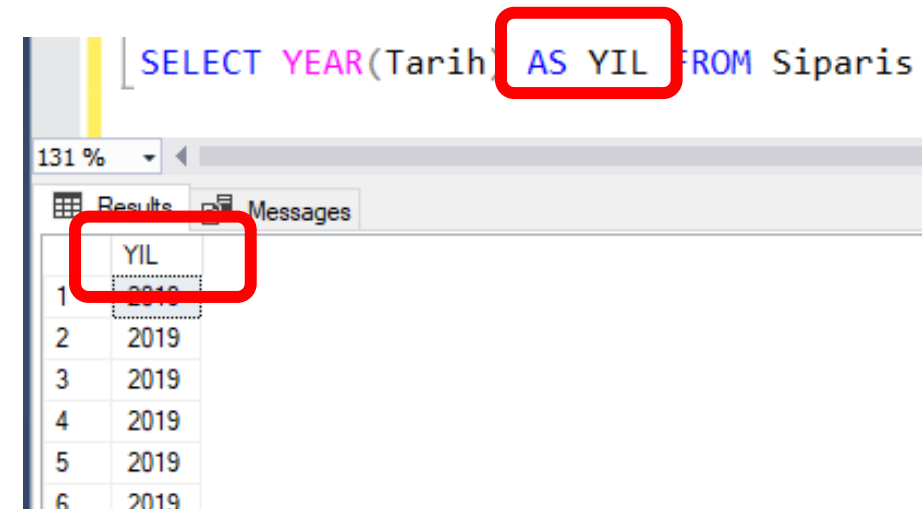
- `SELECT YEAR(Tarih) FROM Siparis`
 - Siparis tablonun Tarih alanının sadece yılını getirir.
- As ile isim verelim gelen sonuç alanına

`SELECT YEAR(Tarih) AS YIL FROM Siparis`



	(No column name)
37	2017
38	2017
39	2017
40	2017
41	2018
42	2018
43	2018
44	2018

Query executed successfully.

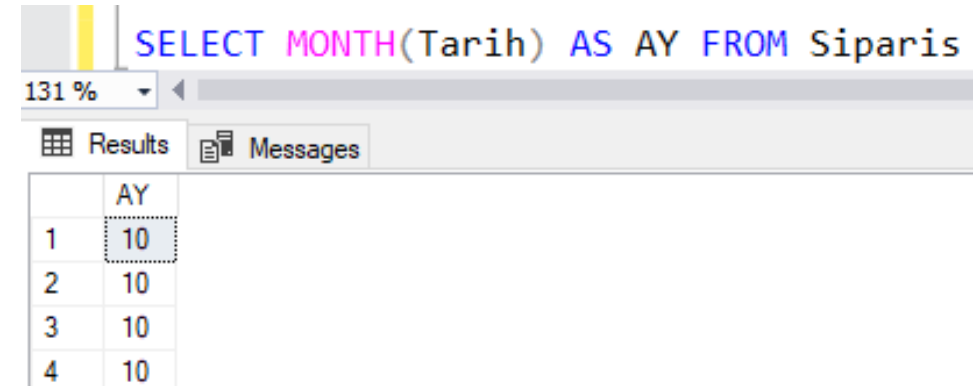


```
SELECT YEAR(Tarih) AS YIL FROM Siparis
```

	YIL
1	2019
2	2019
3	2019
4	2019
5	2019
6	2019

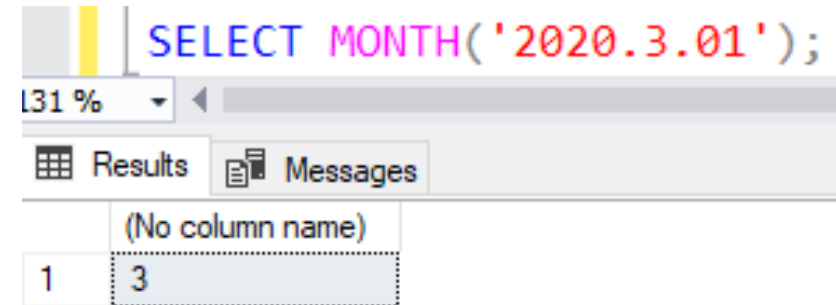
MONTH-AY

- `SELECT MONTH(Tarih) AS AY FROM Siparis`
- `SELECT MONTH('2020.3.01');`
- `SELECT MONTH('2020-3-01');`
- `SELECT MONTH('2020/3/01');`
- *GETDATE ile beraber kullanıp anlık zaman değerinin ay kısmını alır*
- `SELECT MONTH(GETDATE());`



The screenshot shows a SQL Server Enterprise Manager window with the query `SELECT MONTH(Tarih) AS AY FROM Siparis` entered in the query editor. The 'Results' tab is active, displaying a table with two columns: an implicit column and 'AY'. The table contains four rows, all with the value '10' in the 'AY' column.

	AY
1	10
2	10
3	10
4	10

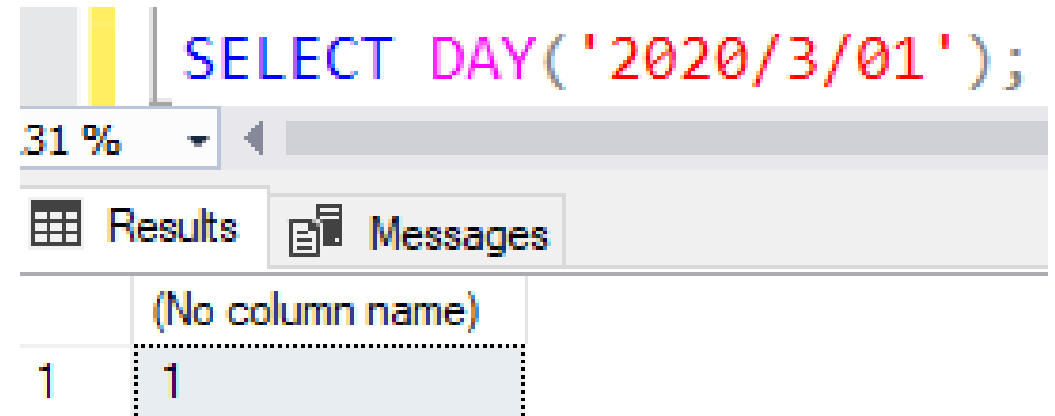


The screenshot shows a SQL Server Enterprise Manager window with the query `SELECT MONTH('2020.3.01');` entered in the query editor. The 'Results' tab is active, displaying a table with two columns: '(No column name)' and a value '3'.

	(No column name)
1	3

DAY-GÜN

- `SELECT DAY('2020/3/01');`



```
SELECT DAY('2020/3/01');
```

31 %

Results Messages

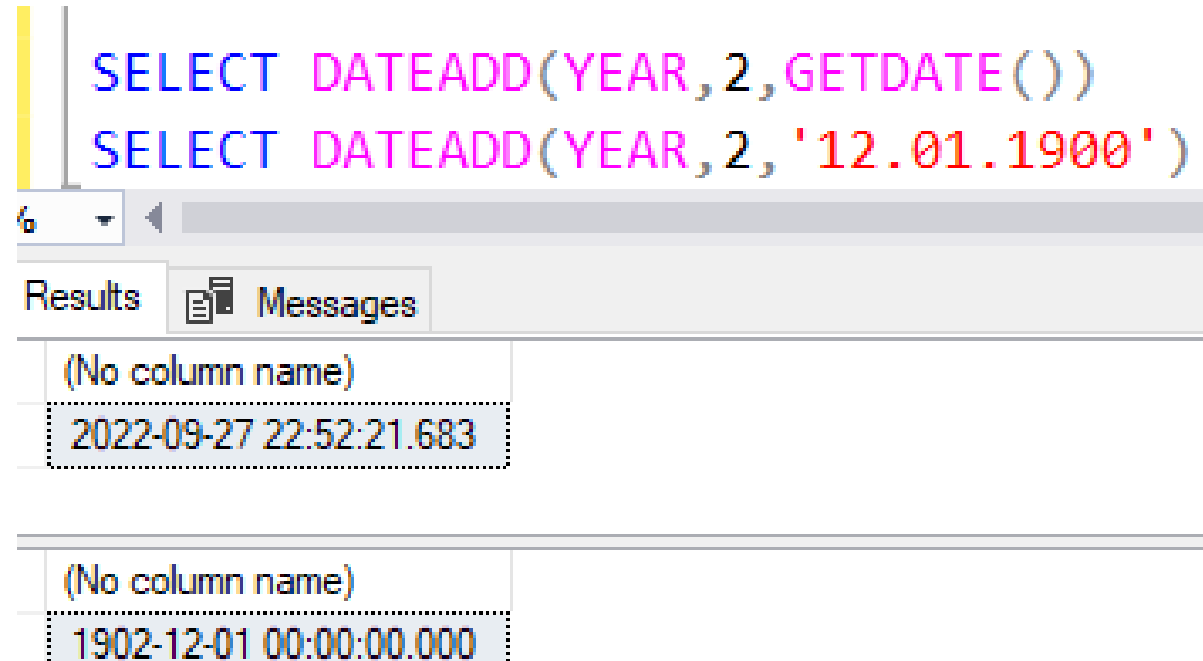
	(No column name)
1	1

DATEADD

- Tarih bilgisin ekleme ya da çıkartma yapmak için kullanılır
- 2 YIL ekle gibi

```
SELECT DATEADD(YEAR,2,GETDATE())
```

```
SELECT DATEADD(YEAR,2,'12.01.1900')
```



The screenshot shows a SQL query execution window with two queries and their results. The first query is `SELECT DATEADD(YEAR,2,GETDATE())` and the second is `SELECT DATEADD(YEAR,2,'12.01.1900')`. The results are displayed in a table with two columns: '(No column name)' and the date/time value.

(No column name)
2022-09-27 22:52:21.683

(No column name)
1902-12-01 00:00:00.000

DATEDIFF

- İki tarih arasındaki yıl/ay/hafta/gün farkını veren fonksiyondur.

```
SELECT DATEDIFF(YEAR, '2000-12-23', '2020-9-18');
```

```
SELECT DATEDIFF(MONTH, '2000-12-23', '2020-9-18');
```

```
SELECT DATEDIFF(WEEK, '2000-12-23', '2020-9-18');
```

```
SELECT DATEDIFF(DAY, '2000-12-23', '2020-9-18');
```

```
SELECT DATEDIFF(YEAR, '2000-12-23', '2020-9-18');
SELECT DATEDIFF(MONTH, '2000-12-23', '2020-9-18');
SELECT DATEDIFF(WEEK, '2000-12-23', '2020-9-18');
SELECT DATEDIFF(DAY, '2000-12-23', '2020-9-18');
```

31 %	
Results Messages	
(No column name)	
1	20
(No column name)	
1	237
(No column name)	
1	1030
(No column name)	
1	7209

SUBSTRING(ALT DİZE) FONKSİYONU

- *String türünden işlem yapan bir fonksiyondur. Belli alandaki metin ifadesinden istenilen kadar karakter alınmasını sağlar.*
- **SELECT SUBSTRING** (Tablo Adı,Alan Adı, Kaçınıcı Karakterden başlayacak, Kaç Karakter) **FROM** TabloAdı

- `SELECT SUBSTRING(Yemek.Ad,1,3) FROM Yemek`
 - Yemek tablosunun Ad alanından baştan ilk harfini alan sorgu.

	(No column name)
1	Taz
2	Ada
3	Urf
4	İsk
5	Bey
6	Pat
7	Köf
8	Kar
9	Bil
10	Kün
11	Kur
12	İma

SELECT

SUBSTRING(Musteri.Ad,1,3),

SUBSTRING(Musteri.Soyad,1,3)

FROM Musteri

Müşteri tablosunu Ad ve Soyad alanlarının ilk karakterini alan sorgu.

Results			Messages		
	(No column name)		(No column name)		
1	Ari		Gün		
2	Ahm		Şer		
3	Ayş		Dağ		

- `SELECT SUBSTRING(Musteri.Ad,2,4),SUBSTRING(Musteri.Soyad,2,4) FROM Musteri`

Müşteri tablosunu Ad ve Soyad alanlarının ikinci karakterinden başlayıp 4 karakter alan sorgu.

	Results	Messages
	(No column name)	(No column name)
1	rif	ünel
2	hmet	erif
3	yşe	ağlı

SELECT

SUBSTRING(Musteri.Ad,2,4)AS AD, SUBSTRING(Musteri.Soyad,2,4)AS
'SOYAD

FROM Musteri

- *Müşteri tablosunu Ad ve Soyad alanlarının ikinci karakterinden başlayıp 4 karakter alan VE gelen sorgu sonucunu AD VE SOYAD yazan sorgu*

- *Not:Yeni adlandırmalar " işaretleri içerisine yazılırsa rengi kırmızı olup anlaşılması kolay olur.*

SELECT

SUBSTRING(Musteri.Ad,2,4)AS 'AD' ,

SUBSTRING(Musteri.Soyad,2,4)AS 'SOYAD'

FROM Musteri

Results			Messages		
	AD	SOYAD			
1	rif	ünel			
2	hmet	erif			
3	yşe	ağlı			

LOWER-Küçük Harfe Dönüştürme

- *Seçtiğimiz metnin tümünü küçük harfe dönüştürür.*
- `SELECT LOWER(Musteri.Ad) AS KüçükHarfliAd FROM Musteri`

Results		Message	
	KüçükHarfliAd		
1	arif		
2	ahmet		
3	ayşe		

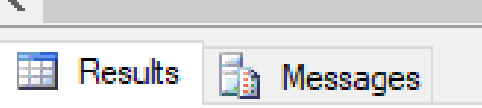
UPPER-Büyük Harfe Dönüştürme

- *Seçtiğimiz metnin tümünü büyük harfe dönüştürür.*
- `SELECT UPPER(Musteri.Ad)AS BüyükHarfliAd FROM Musteri`
- `SELECT UPPER(Ad)AS BüyükHarfliAd FROM Musteri`

Results		Messages	
	BüyükHarfliAd		
1	ARİF		
2	AHMET		
3	AYŞE		

LEN-Karakter Sayısı

- *Seçili alandaki kayıtların kaç adet karakterden oluştuğunu verir.*
- `SELECT LEN(Musteri.Ad) AS BüyükHarfliAd FROM Musteri`



	Büyük Harfli Ad
1	4
2	5
3	4

REPLACE-Değiştirme

- Bir ifade yerine başka bir ifade ile değiştirme.
- Kalıcı olarak değiştirmez listelerken değiştirerek verir.

- `SELECT REPLACE(Musteri.Ad, 'A', 'BEE') FROM Musteri`

A karakterini BEE ile değiştirdi.

Results		Messages	
	(No column name)		
1	BEErif		
2	BEEhmet		
3	BEEyşe		

Hesaplanan sonucu gelen kolona isim yazdırma 3 şekilde yapılabilir(AS , Boşluk = ile atama)

Yeni isim vermek için ya boşluk bırakılır ya da AS kelimesi yazılır.

- `select` KonumNo, KonumNo*KacKisilik **AS** KdvOrani **FROM** Masa
- `select` KonumNo, KonumNo*KacKisilik KdvOrani **FROM** Masa
- `select` KonumNo, KdvOranı=KonumNo*KacKisilik **FROM** Masa

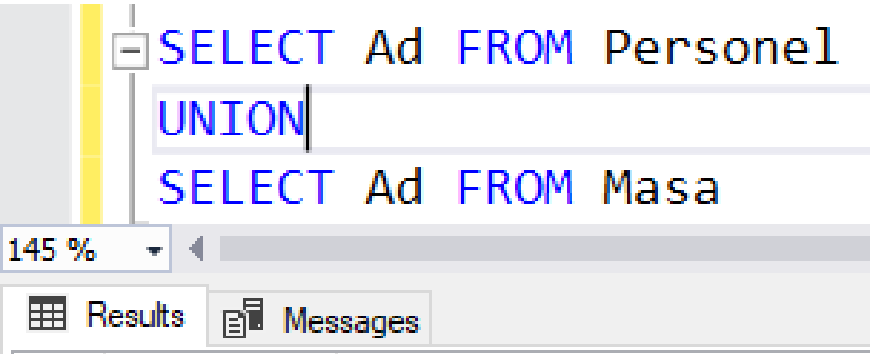
Results		Messages
	KonumNo	KdvOranı
1	1	10
2	2	60
3	2	4
4	2	8

UNION Sorgu Sonuçlarını Birleştirmek

- *Birden fazla tablodan alınan aynı VERİTİP'indeki verileri birleştirmek için kullanılır.*
- *Gelecek olan kolonların aynı veri tipinde olması gerekir.*

- *Personel tablosunun Ad alanı ile Masa tablosunun ad alanlarını birleştirdik*

```
SELECT Ad FROM Personel  
UNION  
SELECT Ad FROM Masa
```



The screenshot shows a SQL query editor with a query window at the top and a results pane at the bottom. The query window contains the following SQL statement:

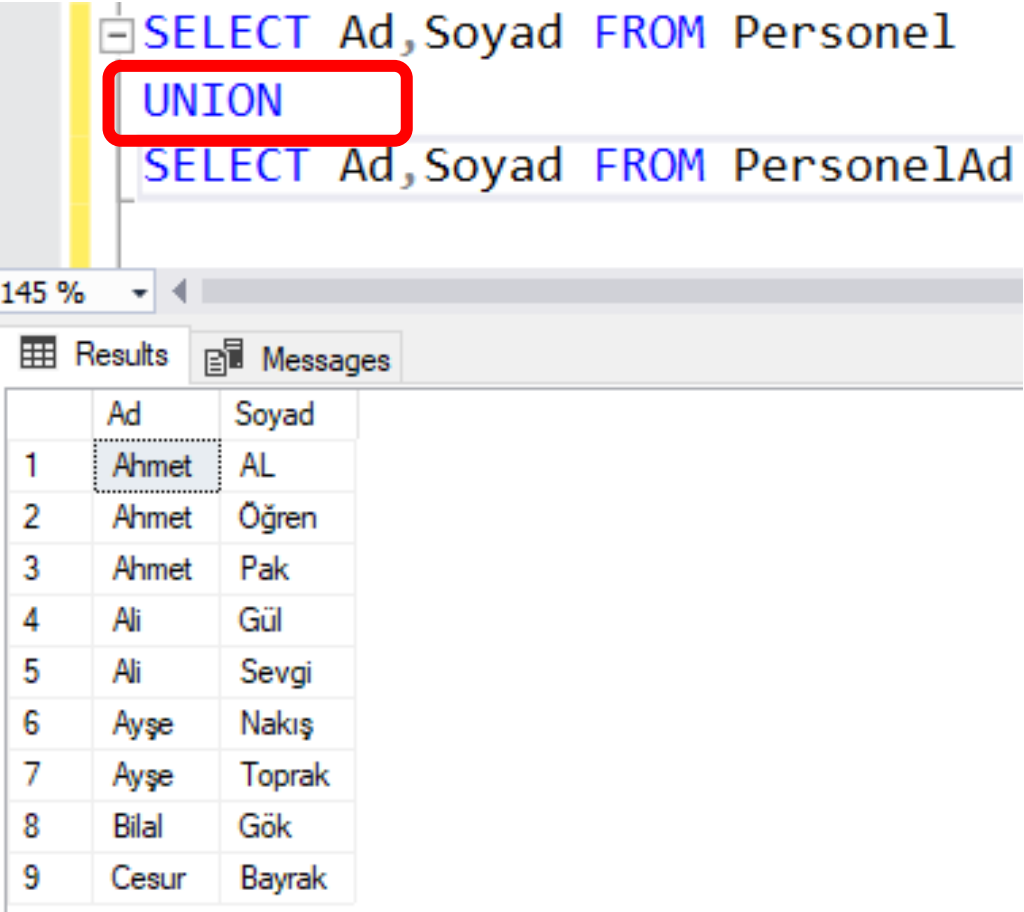
```
SELECT Ad FROM Personel  
UNION  
SELECT Ad FROM Masa
```

The results pane shows a table with two columns: an index and 'Ad'. The table contains 17 rows of data, which are the names from the 'Personel' and 'Masa' tables combined. The first row, 'Ahmet', is highlighted with a dotted border.

	Ad
1	Ahmet
2	Ali
3	ANKARA
4	Ayşe
5	BALIKESİR
6	Bilal
7	Bilecik
8	Cesur
9	ÇANAKKALE
10	Emirgan
11	Eskişehir
12	GÜVERCİN
13	İSTANBUL
14	İstanbul
15	KÜTAHYA
16	Lale
17	Pazaryeri

- İki tablomuz var Personel ve PersonelAd isminde.
- UNION ifadesi ile iki tablonun da Ad ve Soyad alanlarını birleştirip tek bir listede topluyoruz
- İki tabloyu tek bir listede topladık, iki tabloda da aynı olan kayıtlardan sadece bir tanesini alarak liste oluşturdu.

```
SELECT Ad,Soyad FROM Personel  
  
UNION  
  
SELECT Ad,Soyad FROM PersonelAd
```



The screenshot shows a SQL query editor with a query window and a results window. The query window contains the following SQL statement:

```
SELECT Ad,Soyad FROM Personel  
UNION  
SELECT Ad,Soyad FROM PersonelAd
```

The word "UNION" is highlighted with a red rectangle. The results window shows the output of the query, which is a table with 9 rows and 2 columns: Ad and Soyad. The first row is highlighted with a dashed border.

	Ad	Soyad
1	Ahmet	AL
2	Ahmet	Öğren
3	Ahmet	Pak
4	Ali	Gül
5	Ali	Sevgi
6	Ayşe	Nakış
7	Ayşe	Toprak
8	Bilal	Gök
9	Cesur	Bayrak

- Burada UNION ifadesinin yanına **ALL** da ekleyerek iki tablonun ne kadar kaydı varsa hepsini alıyor. Aynı olanlarda dahil.

```
SELECT Ad,Soyad FROM Personel
UNION ALL
SELECT Ad,Soyad FROM PersonelAd
```

145 %

Results Messages

	Ad	Soyad
1	Ahmet	Öğren
2	Ali	Gül
3	Ali	Sevgi
4	Ayşe	Toprak
5	Bilal	Gök
6	Cesur	Bayrak
7	Ahmet	AL
8	Ahmet	Pak
9	Ayşe	Nakış
10	Ahmet	Öğren
11	Ali	Gül
12	Ali	Sevgi
13	Ayşe	Toprak
14	Bilal	Gök
15	Cesur	Bayrak

- *İlk sorguda sadece ikinci tabloya Where şartı yazdık*
- *İkinci sorguda ise iki sorguya da Where şartı ekledik.*

```

SELECT Ad,Soyad FROM Personel
UNION
SELECT Ad,Soyad FROM PersonelAd
WHERE Ad='Ali';

SELECT Ad,Soyad FROM Personel
WHERE Ad='Ali'
UNION
SELECT Ad,Soyad FROM PersonelAd
WHERE Ad='Ali';

```

109 %

Results Messages

	Ad	Soyad
1	Ahmet	AL
2	Ahmet	Öğren
3	Ahmet	Pak
4	Ali	Gül
5	Ali	Sevgi
6	Ayşe	Nakış
7	Ayşe	Toprak
8	Bilal	Gök
9	Cesur	Bayrak

	Ad	Soyad
1	Ali	Gül
2	Ali	Sevgi

Mantıksal Operatörler: AND – OR - NOT

- *Mantıksal operatörler bir ya da birden fazla tablodan farklı durumların birleştirilerek seçilmesi için kullanılır.*

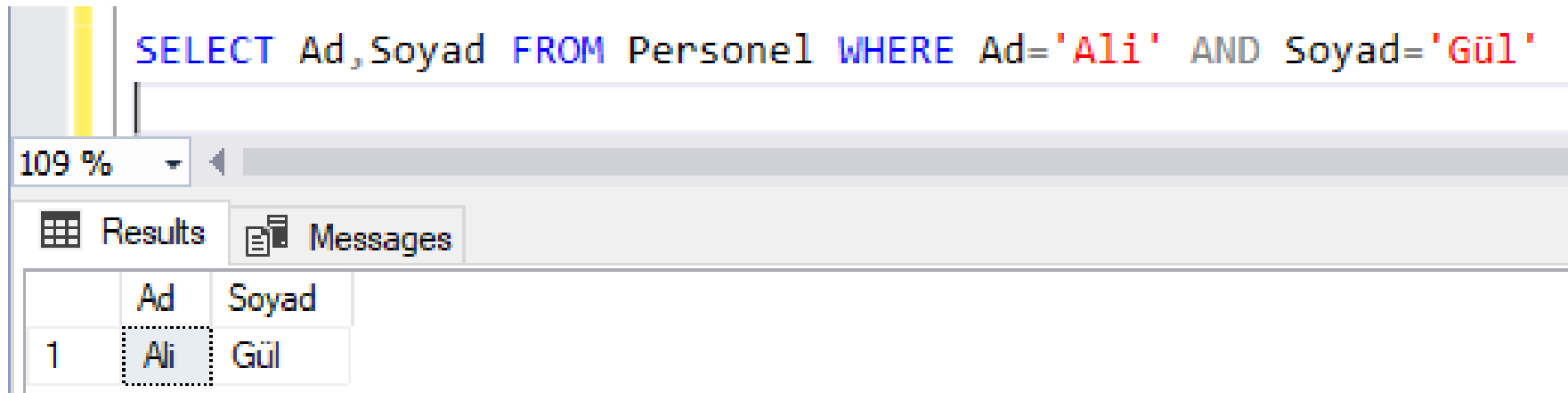
3 temel ifadesi vardır

- *And(Ve)*
- *OR(Veya/yada)*
- *NOT(Değil)*

AND(VE)

- Tüm şartların gerçekleştiği kayıtları getirir.
- Şartlardan biri bile tutmuyorsa o kaydı getirmez.
- Adı Ali ve Soyadı Gül olanlar(iki şartıda sağlamak zorunda)

```
SELECT Ad,Soyad FROM Personel WHERE Ad='Ali' AND Soyad='Gül'
```



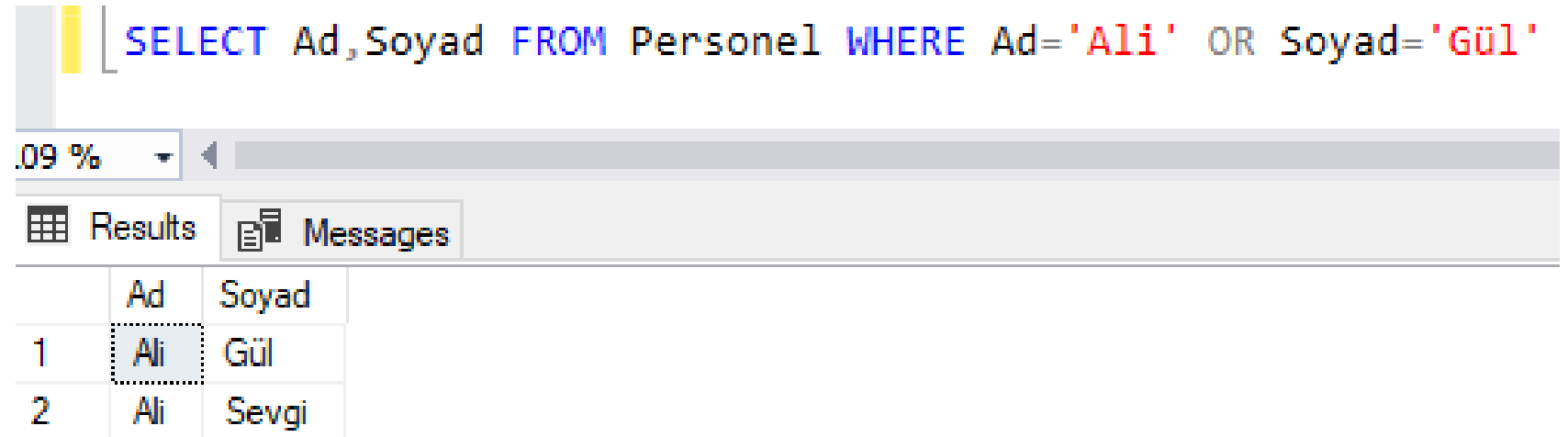
The screenshot shows a SQL query execution window. The query is: `SELECT Ad,Soyad FROM Personel WHERE Ad='Ali' AND Soyad='Gül'`. Below the query, there is a zoom level of 109% and two tabs: "Results" and "Messages". The "Results" tab is active, displaying a table with two columns: "Ad" and "Soyad". The table contains one row with the values "Ali" and "Gül".

	Ad	Soyad
1	Ali	Gül

OR(Veya)

- Belirtilen şartlardan sadece birinin bile uyduğu durumları getiren sorgularda kullanılır.
- Adı Ali Soyadı Gül olanların her birini getir, ikisinden birinin tutması yeterli

```
SELECT Ad,Soyad FROM Personel WHERE Ad='Ali' OR Soyad='GÜL'
```



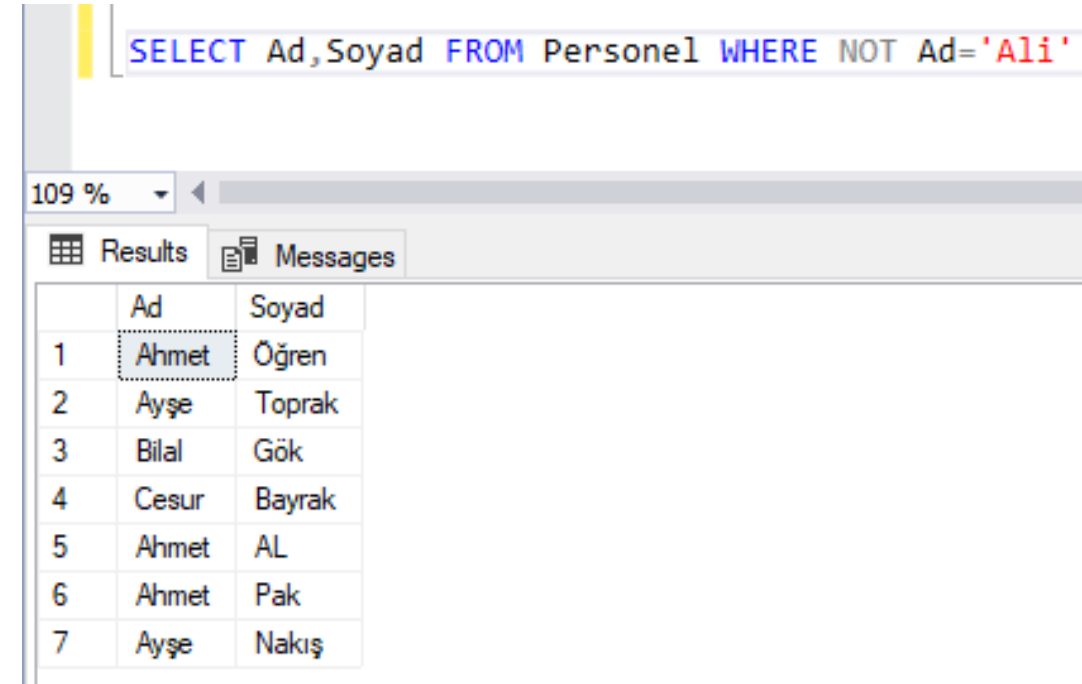
The screenshot shows a SQL query execution interface. At the top, the query is entered in a text box: `SELECT Ad,Soyad FROM Personel WHERE Ad='Ali' OR Soyad='Gül'`. Below the text box, there is a progress bar showing ".09 %". Underneath the progress bar, there are two tabs: "Results" and "Messages". The "Results" tab is active, displaying a table with two columns: "Ad" and "Soyad". The table contains two rows of data: Row 1 with "Ali" and "Gül", and Row 2 with "Ali" and "Sevgi". The cell containing "Ali" in the first row is highlighted with a dashed border.

	Ad	Soyad
1	Ali	Gül
2	Ali	Sevgi

NOT(Değil-Bunun dışında hepsi)

- *Not operatörü yazılan şartın haricindeki tüm durumları getirir.*
- *Adı Ali olmayanları getir*

```
SELECT Ad,Soyad FROM Personel WHERE NOT Ad='Ali'
```



The screenshot shows a SQL query execution window. The query entered is `SELECT Ad,Soyad FROM Personel WHERE NOT Ad='Ali'`. The results are displayed in a table with two columns: Ad and Soyad. The table contains 7 rows of data, all of which are names other than 'Ali'.

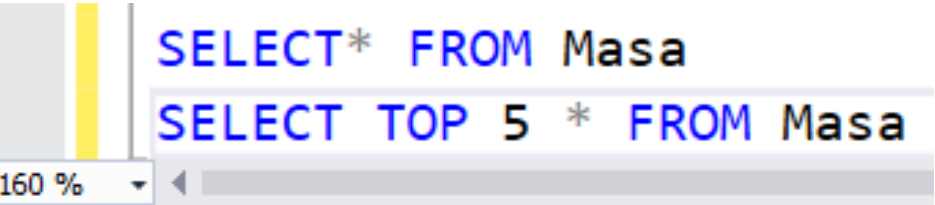
	Ad	Soyad
1	Ahmet	Öğren
2	Ayşe	Toprak
3	Bilal	Gök
4	Cesur	Bayrak
5	Ahmet	AL
6	Ahmet	Pak
7	Ayşe	Nakış

TOP

- *Tablodaki kayıtların sıralamasına göre ilk istenilen sayı kadar kayıt getiren sorgu*
- *İlk 5 kayıt gibi*

```
SELECT* FROM Masa
```

```
SELECT TOP 5 * FROM Masa
```



160 %

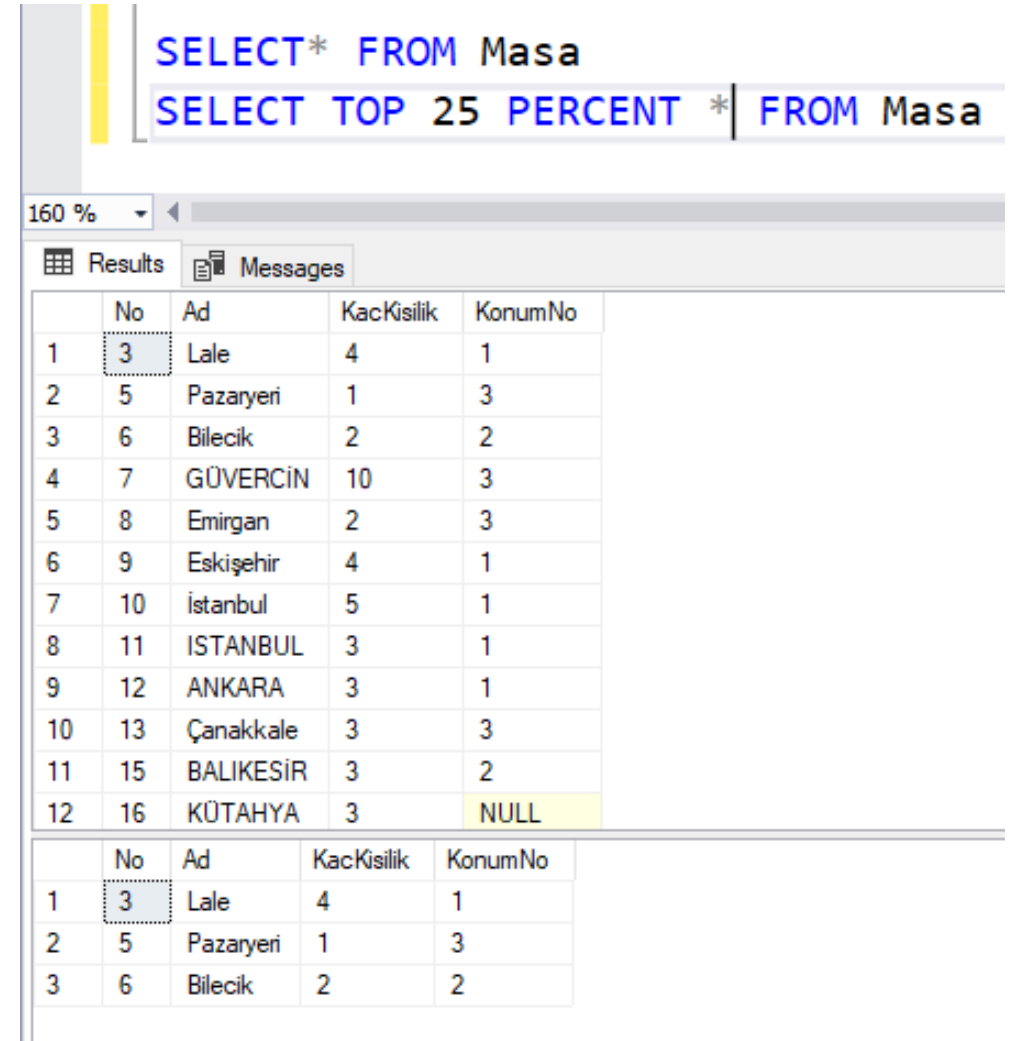
Results Messages

	No	Ad	KacKisilik	KonumNo
1	3	Lale	4	1
2	5	Pazaryeri	1	3
3	6	Bilecik	2	2
4	7	GÜVERCİN	10	3
5	8	Emirgan	2	3
6	9	Eskişehir	4	1
7	10	İstanbul	5	1
8	11	İSTANBUL	3	1
9	12	ANKARA	3	1
10	13	Çanakkale	3	3
11	15	BALIKESİR	3	2

	No	Ad	KacKisilik	KonumNo
1	3	Lale	4	1
2	5	Pazaryeri	1	3
3	6	Bilecik	2	2
4	7	GÜVERCİN	10	3
5	8	Emirgan	2	3

Var olan kayıtların belli oranda getirilmesi

- Mesela 100 kaydımız var biz bunun ilk %25 luk kısmını istediğimizde
- `SELECT* FROM Masa`
- `SELECT TOP 25 PERCENT * FROM Masa`



```
SELECT* FROM Masa
SELECT TOP 25 PERCENT * FROM Masa
```

160 %

Results Messages

	No	Ad	KacKisilik	KonumNo
1	3	Lale	4	1
2	5	Pazaryeri	1	3
3	6	Bilecik	2	2
4	7	GÜVERCİN	10	3
5	8	Emirgan	2	3
6	9	Eskişehir	4	1
7	10	İstanbul	5	1
8	11	İSTANBUL	3	1
9	12	ANKARA	3	1
10	13	Çanakkale	3	3
11	15	BALIKESİR	3	2
12	16	KÜTAHYA	3	NULL

	No	Ad	KacKisilik	KonumNo
1	3	Lale	4	1
2	5	Pazaryeri	1	3
3	6	Bilecik	2	2

BİRDEN FAZLA SÜTUNU BİRLEŞTİRME

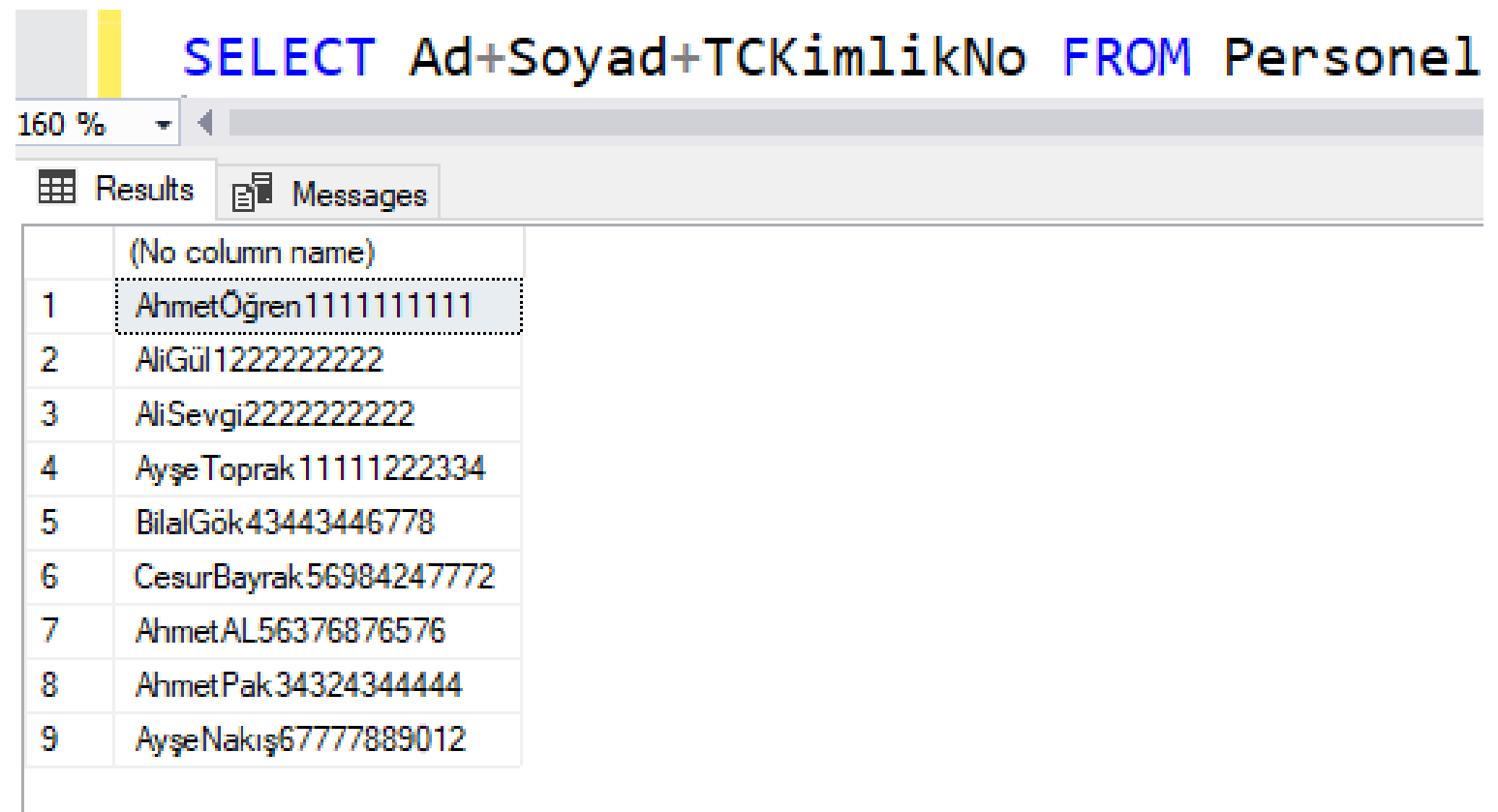
- Sütunları birleştirirken dikkat etmemiz gereken nokta

Metinleri Birleştirme(+Concat)

- Birden fazla sütunda olan verileri tek bir sütunda gösterme işlemi.
- + operatörü kullanılarak yapılır.
- `SELECT Musteri.Ad+Musteri.Soyad FROM Musteri`
- `SELECT Ad+Soyad FROM Musteri`

Results		Messages	
	(No column name)		
1	ArifGünel		
2	Ahmet Şerif		
3	Ayşe Dağlı		

- `SELECT Ad+Soyad+TCKimlikNo FROM Personel`



The screenshot shows a SQL query execution window. At the top, the query `SELECT Ad+Soyad+TCKimlikNo FROM Personel` is entered. Below the query, there is a zoom level of 160% and two tabs: 'Results' and 'Messages'. The 'Results' tab is active, displaying a table with 9 rows. The first row is highlighted with a dotted border. The table has a single column labeled '(No column name)'.

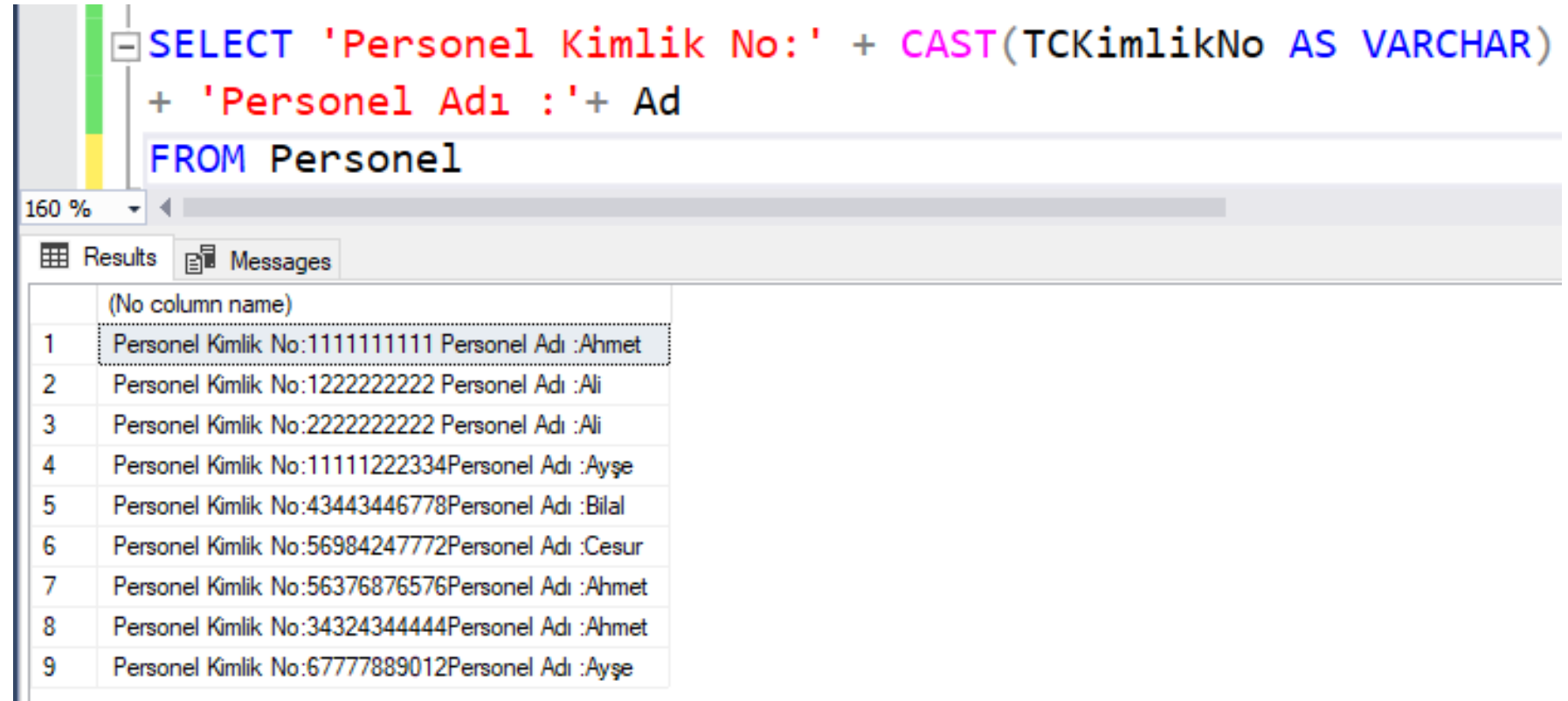
	(No column name)
1	AhmetÖğren1111111111
2	AliGül1222222222
3	AliSevgi2222222222
4	AyşeToprak11111222334
5	BilalGök43443446778
6	CesurBayrak56984247772
7	AhmetAL56376876576
8	AhmetPak34324344444
9	AyşeNakış67777889012

Veri Türleri(Tipleri) Dönüşümleri

- *Veri tiplerinin farklı olması birleştirme gibi bir takım işlemlerde sıkıntı oluşturmakta.*
- *Özellikle sayısal veri türleri ile metinsel veri türlerinin birleştirilmesinde sorun oluşur.B u durumlarda tür dönüşümü uygulanır.*
- *İki adet tür dönüşüm yöntemi bulunmaktadır*
- *CAST VE CONVERT fonksiyonları*

CAST

- *CAST' in Kullanımı*



```
SELECT 'Personel Kimlik No:' + CAST(TCKimlikNo AS VARCHAR)
+ 'Personel Adı :'+ Ad
FROM Personel
```

160 %

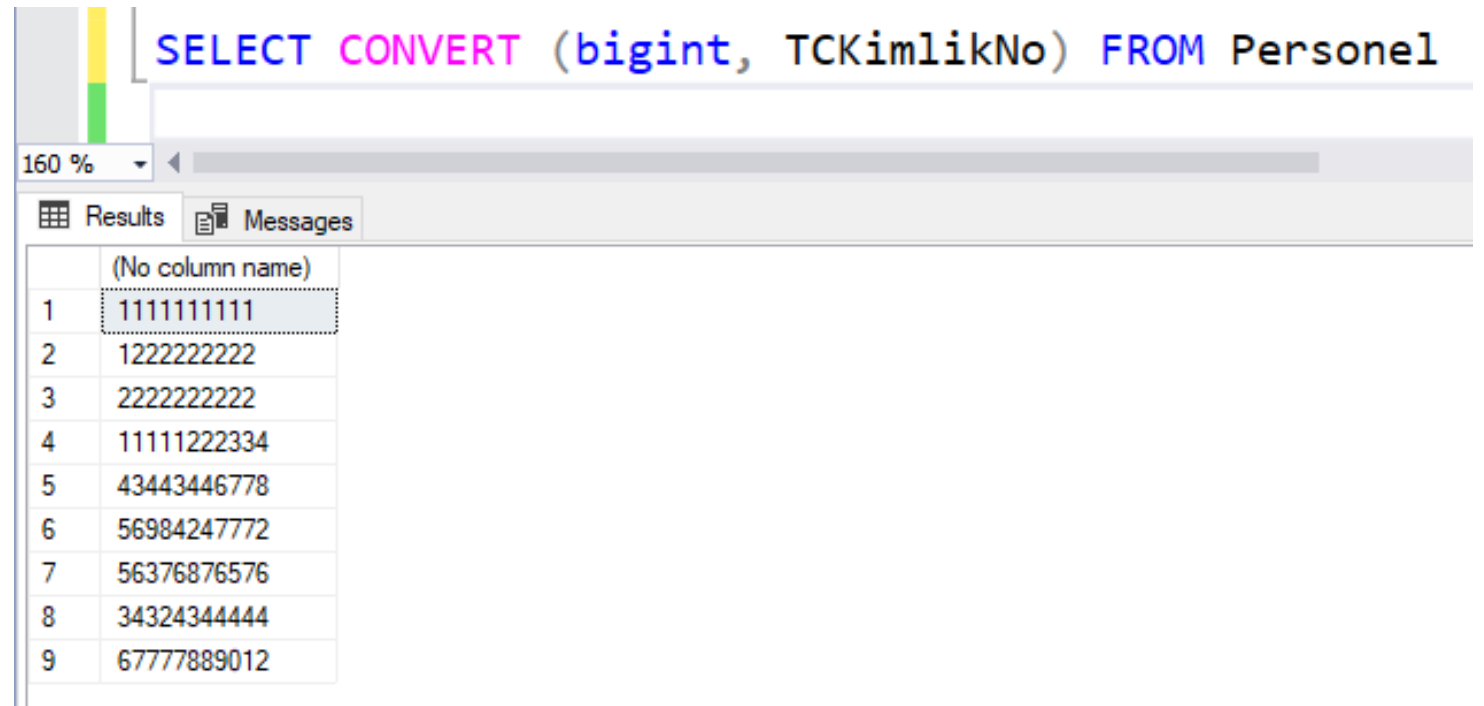
Results Messages

	(No column name)
1	Personel Kimlik No:1111111111 Personel Adı :Ahmet
2	Personel Kimlik No:1222222222 Personel Adı :Ali
3	Personel Kimlik No:2222222222 Personel Adı :Ali
4	Personel Kimlik No:11111222334Personel Adı :Ayşe
5	Personel Kimlik No:43443446778Personel Adı :Bilal
6	Personel Kimlik No:56984247772Personel Adı :Cesur
7	Personel Kimlik No:56376876576Personel Adı :Ahmet
8	Personel Kimlik No:34324344444Personel Adı :Ahmet
9	Personel Kimlik No:67777889012Personel Adı :Ayşe

```
SELECT 'Personel Kimlik No:' + CAST(TCKimlikNo AS VARCHAR)
+ 'Personel Adı :'+ Ad
FROM Personel
```


CONVERT

- `SELECT CONVERT (bigint, TCKimlikNo) FROM Personel`

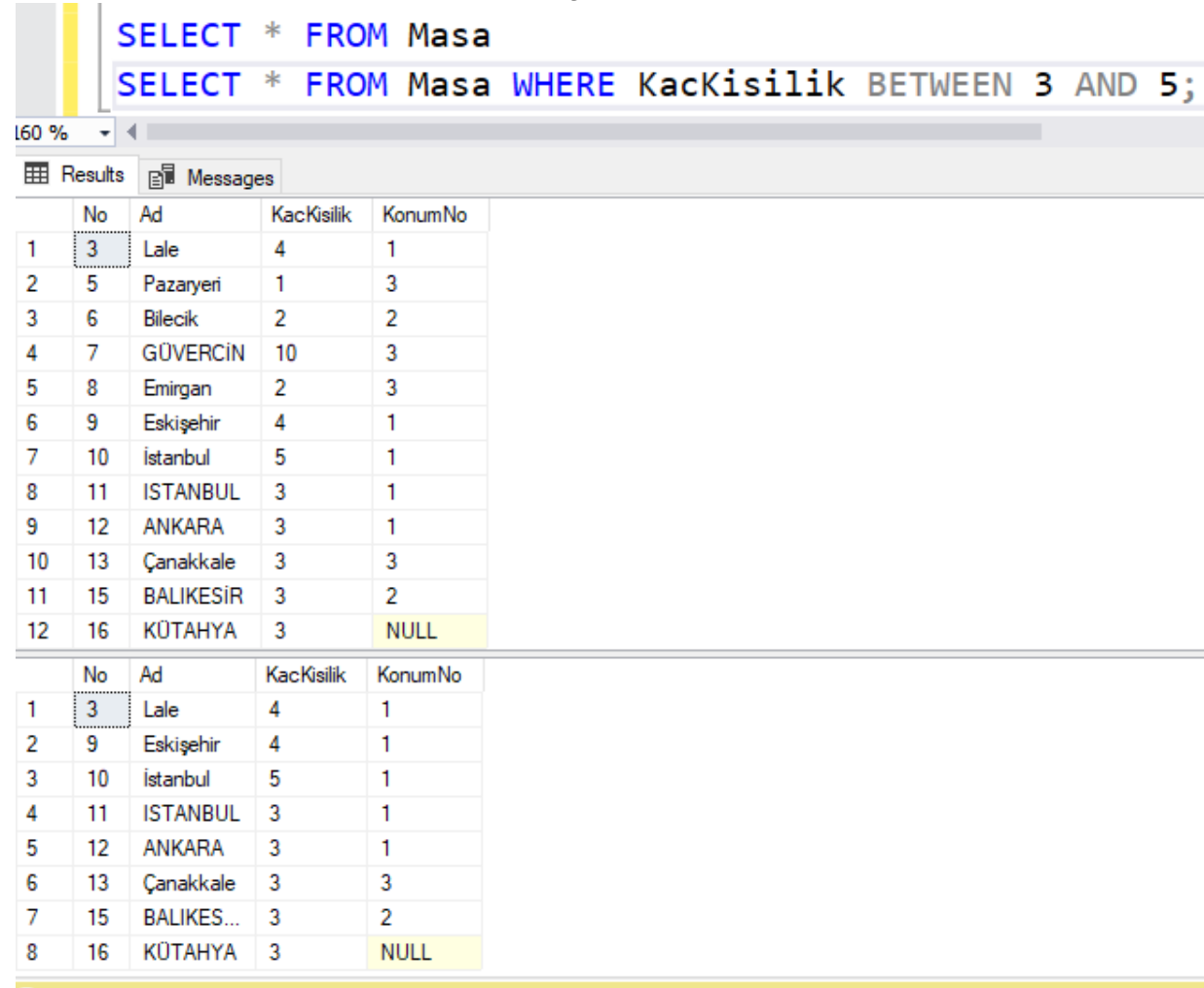


The screenshot shows a SQL query execution window. The query entered is `SELECT CONVERT (bigint, TCKimlikNo) FROM Personel`. The results are displayed in a table with 9 rows. The first row is highlighted. The zoom level is set to 160%.

	(No column name)
1	1111111111
2	1222222222
3	2222222222
4	11111222334
5	43443446778
6	56984247772
7	56376876576
8	34324344444
9	67777889012

BETWEEN AND (ARALIK SORGULAMA)

- *Between komutu belli aralık içerisinde kalan değerlerin sorgulanması için kullanılır*
- **SELECT * FROM Masa WHERE KacKisilik BETWEEN 3 AND 5;**
 - *Masa tablosunda KaçKişilik alanında 3 ile 5 de dağıl olmak üzere arasında olan masaları getiren sorgu.*
- *Metinsel olarak alfabetik sıralam arsındaki değerlerde de kullanılır.*



The screenshot shows a SQL query execution window with the following SQL code:

```
SELECT * FROM Masa  
SELECT * FROM Masa WHERE KacKisilik BETWEEN 3 AND 5;
```

The window displays two result sets. The first result set shows all records from the 'Masa' table, and the second result set shows only the records where 'KacKisilik' is between 3 and 5.

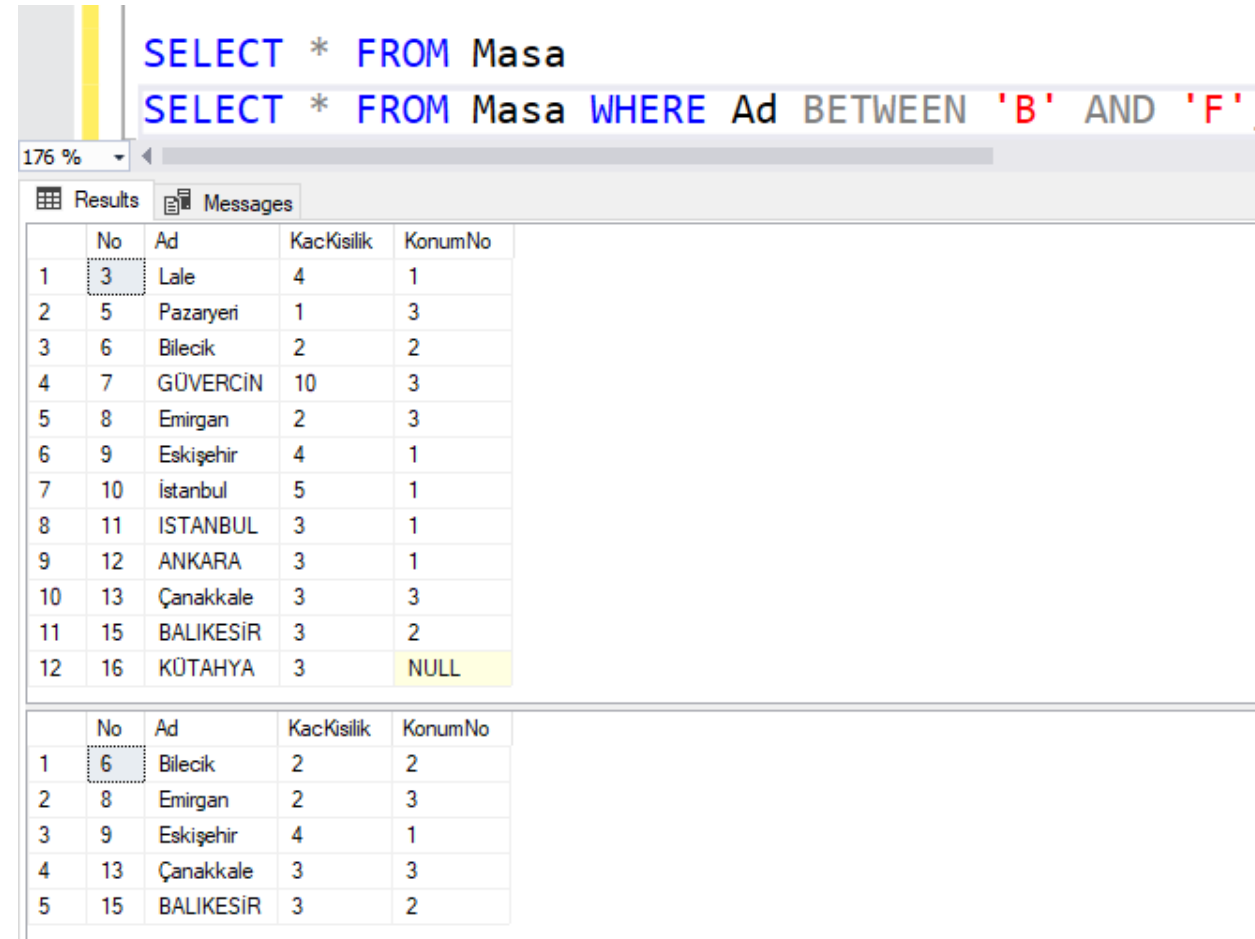
	No	Ad	KacKisilik	KonumNo
1	3	Lale	4	1
2	5	Pazaryeri	1	3
3	6	Bilecik	2	2
4	7	GÜVERCİN	10	3
5	8	Emirgan	2	3
6	9	Eskişehir	4	1
7	10	İstanbul	5	1
8	11	İSTANBUL	3	1
9	12	ANKARA	3	1
10	13	Çanakkale	3	3
11	15	BALIKESİR	3	2
12	16	KÜTAHYA	3	NULL

	No	Ad	KacKisilik	KonumNo
1	3	Lale	4	1
2	9	Eskişehir	4	1
3	10	İstanbul	5	1
4	11	İSTANBUL	3	1
5	12	ANKARA	3	1
6	13	Çanakkale	3	3
7	15	BALIKES...	3	2
8	16	KÜTAHYA	3	NULL

BETWEEN – Metinde Kullanımı

- Alfabetik olarak sıralamada belirlenen karakterler arasında olan harfle başlayan kayıtları getirir.

```
SELECT * FROM Masa  
WHERE Ad BETWEEN 'B' AND 'F';
```



The screenshot shows a SQL query execution window. The query entered is: `SELECT * FROM Masa WHERE Ad BETWEEN 'B' AND 'F';`. The results are displayed in a table with columns: No, Ad, KacKisilik, and KonumNo. The results are filtered to show only records where the 'Ad' is between 'B' and 'F'.

	No	Ad	KacKisilik	KonumNo
1	3	Lale	4	1
2	5	Pazaryeri	1	3
3	6	Bilecik	2	2
4	7	GÜVERCİN	10	3
5	8	Emirgan	2	3
6	9	Eskişehir	4	1
7	10	İstanbul	5	1
8	11	İSTANBUL	3	1
9	12	ANKARA	3	1
10	13	Çanakkale	3	3
11	15	BALIKESİR	3	2
12	16	KÜTAHYA	3	NULL

	No	Ad	KacKisilik	KonumNo
1	6	Bilecik	2	2
2	8	Emirgan	2	3
3	9	Eskişehir	4	1
4	13	Çanakkale	3	3
5	15	BALIKESİR	3	2

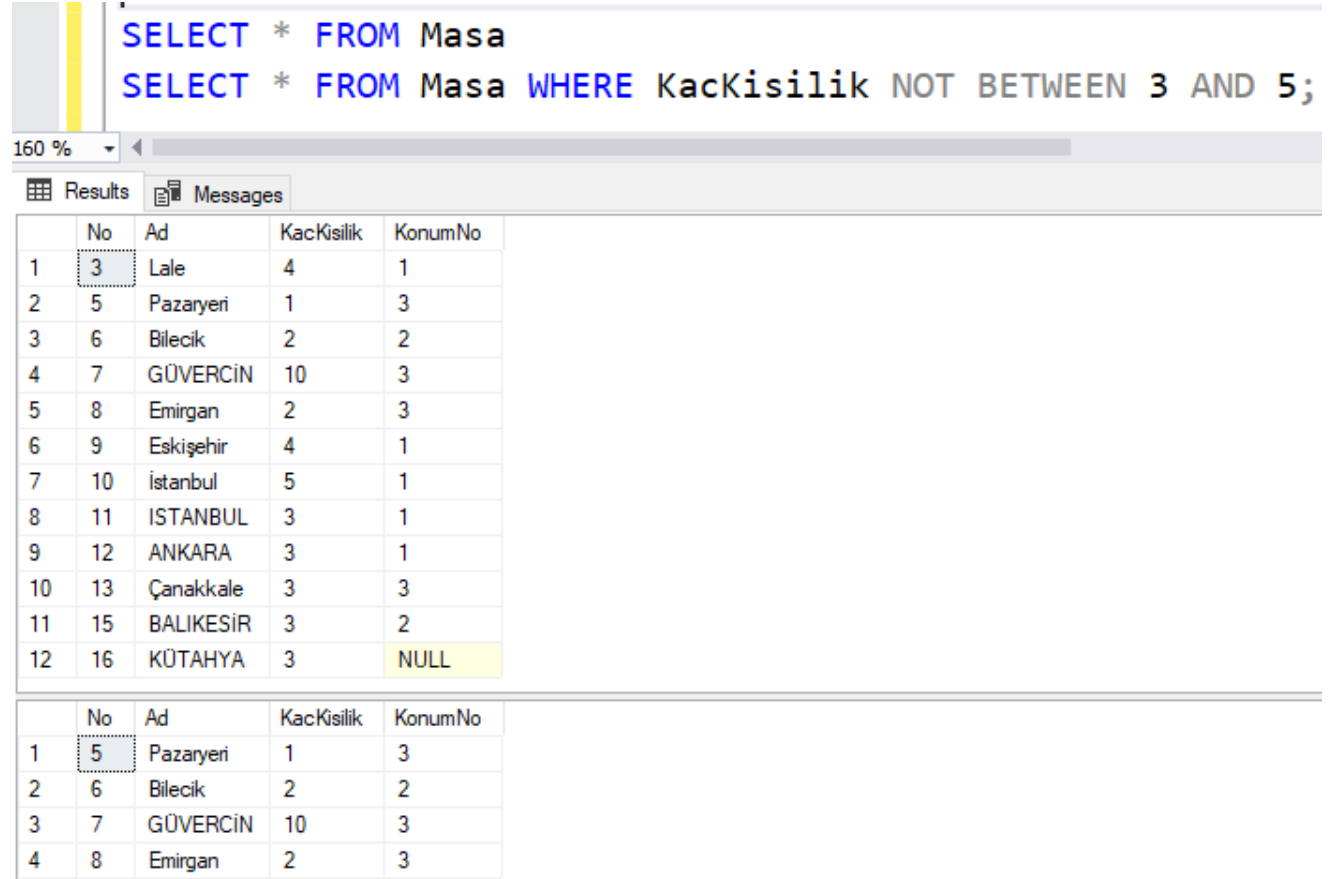
NOT BETWEEN AND (ARALIK HARİCİ SORGULAMA)

- Between operatörünün başına koyduğumuz Not ifadesi 'değil' anlamında düşünülebilir.
- Yani bu aralıkta olanların haricindeki kayıtlar.

```
SELECT * FROM Masa
```

```
SELECT * FROM Masa
```

```
WHERE KacKisilik NOT BETWEEN 3 AND 5;
```



The screenshot shows a SQL query execution window. The query is: `SELECT * FROM Masa WHERE KacKisilik NOT BETWEEN 3 AND 5;`. The results are displayed in a table with columns: No, Ad, KacKisilik, and KonumNo. The table contains 12 rows of data. The first row (No: 3, Ad: Lale, KacKisilik: 4, KonumNo: 1) is highlighted, indicating it is the first result returned by the query. The second row (No: 5, Ad: Pazaryeri, KacKisilik: 1, KonumNo: 3) is also highlighted, indicating it is the second result returned by the query. The third row (No: 6, Ad: Bilecik, KacKisilik: 2, KonumNo: 2) is also highlighted, indicating it is the third result returned by the query. The fourth row (No: 7, Ad: GÜVERCİN, KacKisilik: 10, KonumNo: 3) is also highlighted, indicating it is the fourth result returned by the query. The fifth row (No: 8, Ad: Emirgan, KacKisilik: 2, KonumNo: 3) is also highlighted, indicating it is the fifth result returned by the query. The sixth row (No: 9, Ad: Eskişehir, KacKisilik: 4, KonumNo: 1) is also highlighted, indicating it is the sixth result returned by the query. The seventh row (No: 10, Ad: İstanbul, KacKisilik: 5, KonumNo: 1) is also highlighted, indicating it is the seventh result returned by the query. The eighth row (No: 11, Ad: İSTANBUL, KacKisilik: 3, KonumNo: 1) is also highlighted, indicating it is the eighth result returned by the query. The ninth row (No: 12, Ad: ANKARA, KacKisilik: 3, KonumNo: 1) is also highlighted, indicating it is the ninth result returned by the query. The tenth row (No: 13, Ad: Çanakkale, KacKisilik: 3, KonumNo: 3) is also highlighted, indicating it is the tenth result returned by the query. The eleventh row (No: 15, Ad: BALIKESİR, KacKisilik: 3, KonumNo: 2) is also highlighted, indicating it is the eleventh result returned by the query. The twelfth row (No: 16, Ad: KÜTAHYA, KacKisilik: 3, KonumNo: NULL) is also highlighted, indicating it is the twelfth result returned by the query.

No	Ad	KacKisilik	KonumNo
1	Lale	4	1
2	Pazaryeri	1	3
3	Bilecik	2	2
4	GÜVERCİN	10	3
5	Emirgan	2	3
6	Eskişehir	4	1
7	İstanbul	5	1
8	İSTANBUL	3	1
9	ANKARA	3	1
10	Çanakkale	3	3
11	BALIKESİR	3	2
12	KÜTAHYA	3	NULL

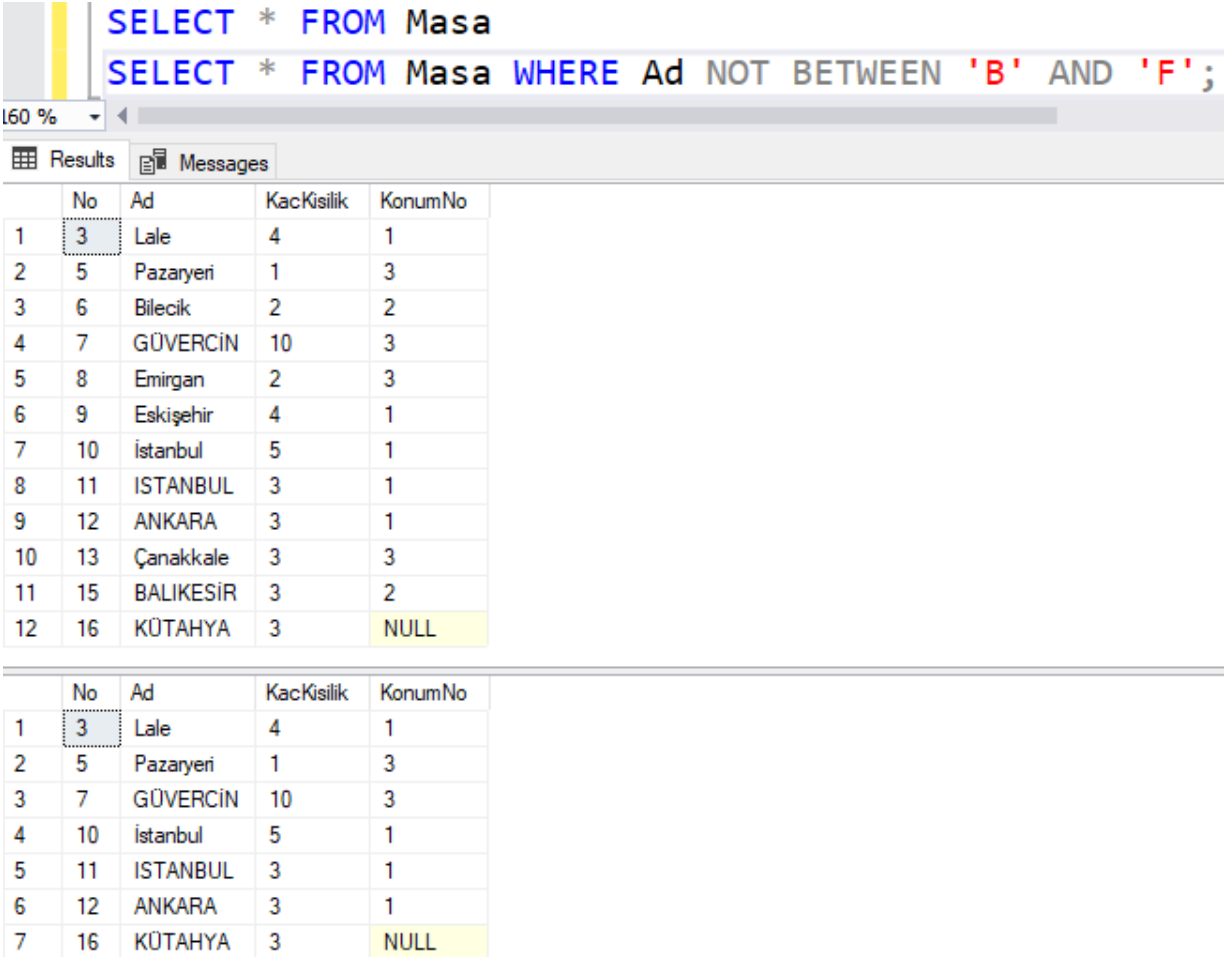
NOT BETWEEN – Metinde Kullanımı

- Sayısal ifadelerde olduğu gibi metin ifadelerinde de aralık dışında kalan tüm kayıtları getirir.*

```
SELECT * FROM Masa
```

```
SELECT * FROM Masa
```

```
WHERE Ad NOT BETWEEN 'B' AND 'F';
```



The screenshot shows a SQL query editor with the following query:

```
SELECT * FROM Masa
SELECT * FROM Masa WHERE Ad NOT BETWEEN 'B' AND 'F';
```

The results are displayed in a table with columns: No, Ad, KacKisilik, and KonumNo. The table shows 12 rows of data, with the first row highlighted in yellow.

	No	Ad	KacKisilik	KonumNo
1	3	Lale	4	1
2	5	Pazaryeri	1	3
3	6	Bilecik	2	2
4	7	GÜVERCİN	10	3
5	8	Emirgan	2	3
6	9	Eskişehir	4	1
7	10	İstanbul	5	1
8	11	İSTANBUL	3	1
9	12	ANKARA	3	1
10	13	Çanakkale	3	3
11	15	BALIKESİR	3	2
12	16	KÜTAHYA	3	NULL

The second table shown in the screenshot is a filtered view of the data, showing only the rows where the 'Ad' is not between 'B' and 'F'.

	No	Ad	KacKisilik	KonumNo
1	3	Lale	4	1
2	5	Pazaryeri	1	3
3	7	GÜVERCİN	10	3
4	10	İstanbul	5	1
5	11	İSTANBUL	3	1
6	12	ANKARA	3	1
7	16	KÜTAHYA	3	NULL

IN OPERATÖRÜ

- *IN Operaörü, between'in aksine belli aralıkta olanları değil aralardan sadece belli değerleri çekmek için kullanılır.*
- *Mantık olarak biraz OR(Veya) yapısına benzer. İstenen değerlerden hangilerine karşılık değer varsa onları getirir diğerlerini getirmez*
- **SELECT * FROM** Masa **WHERE** KacKisilik=4 OR KacKisilik=10 OR KacKisilik=30;
 - Masa tablosunda KaçKişilik alanında 4 ,10 ve 30 a eşit olanları getirir.
- **SELECT * FROM** Masa **WHERE** KacKisilik in (4,10,30);
 - Masa tablosunda KaçKişilik alanında 4 ,10 ve 30 a eşit olanları getirir.

NOT IN OPERATÖRÜ

- *Yazılan değerlerin dışında kalanları getirir.*
- `SELECT * FROM Masa WHERE KacKisilik NOT in (4,10,30);`
 - *Masa tablosunda KaçKişilik alanında 4 ,10 ve 30 a eşit olanlar HARİC diğer tüm kayıtları getirir.*

İngilizce Kelimeler

- *Ascending* : *Artan, yükselen*
- *Descending* : *Azalan, düşen*
- *Select* : *Seçmek*
- *Order* : *Sıra, düzen*
- *Like* : *Eş, benzer*
- *Top* : *Tepe, baş, üst*
- *Percent* : *Yüzde, hisse*

Kaynaklar