İçindekiler

[Uygulamalarla SQL Öğreniyorum 3](#_Toc46883761)

[Giriş 3](#_Toc46883762)

[SQL Ne Demektir? 3](#_Toc46883763)

[İlişkisel Veritabanı Sistemleri (RDMS) 3](#_Toc46883764)

[Temel SQL Komutları 4](#_Toc46883765)

[Data Manipülasyon Komutları 4](#_Toc46883766)

[Database Manipülasyon Komutları 4](#_Toc46883767)

[Select Komutu 4](#_Toc46883768)

[Insert Komutu 5](#_Toc46883769)

[Update Komutu 5](#_Toc46883770)

[Delete Komutu 6](#_Toc46883771)

[Where Şartı Kullanımı 7](#_Toc46883772)

[Distinct Komutu 13](#_Toc46883773)

[Order By Komutu 14](#_Toc46883774)

[Top Komutu 16](#_Toc46883775)

[Örnek Satış Datası 17](#_Toc46883776)

[Aggregate Fonksiyonlar 17](#_Toc46883777)

[Group By Kullanımı 18](#_Toc46883778)

[SQL Server Veri Tipleri 25](#_Toc46883779)

[İlişkisel Veritabanı Sistemleri (RDMS) 28](#_Toc46883780)

[SQL Server Tablo Oluşturma 28](#_Toc46883781)

[Veri Oluşturma 32](#_Toc46883782)

[Join İşlemleri 33](#_Toc46883783)

[INNER JOIN 33](#_Toc46883784)

[LEFT JOIN 33](#_Toc46883785)

[RIGHT JOIN 34](#_Toc46883786)

[FULL JOIN 34](#_Toc46883787)

[ALIAS KULLANIMI 34](#_Toc46883788)

[GROUP BY KULLANIMI 35](#_Toc46883789)

[E-Ticaret Datası Sorgulama 36](#_Toc46883790)

[Örnek-1 36](#_Toc46883791)

[Örnek-2 37](#_Toc46883792)

[Örnek-3 38](#_Toc46883793)

[Örnek-4 38](#_Toc46883794)

[Örnek-5 39](#_Toc46883795)

[Örnek-6 39](#_Toc46883796)

[Sub Query 40](#_Toc46883797)

[SubQuery Giriş 40](#_Toc46883798)

[Örnek 41](#_Toc46883799)

[Örnek: Müşterinin Sepete Eklediği Son Ürün 42](#_Toc46883800)

[String İşlemleri 43](#_Toc46883801)

[ASCII VE CHAR 43](#_Toc46883802)

[SUBSTRING 43](#_Toc46883803)

[CHAR INDEX 44](#_Toc46883804)

[CONCAT, CONCAT\_WS 44](#_Toc46883805)

[FORMAT 45](#_Toc46883806)

[LEFT, RIGHT, LEN 45](#_Toc46883807)

[TRIM, LTRIM, RTRIM 46](#_Toc46883808)

[LOWER, UPPER, REVERSE, REPLICATE 47](#_Toc46883809)

[REPLACE 48](#_Toc46883810)

[Uçtan Uca SQL Server 49](#_Toc46883811)

[Sistem Database’leri 49](#_Toc46883812)

[Master DB 50](#_Toc46883813)

[Model DB 50](#_Toc46883814)

[MSDB 51](#_Toc46883815)

[Temp DB 51](#_Toc46883816)

[T-SQL Kodları 52](#_Toc46883817)

[Data Definition Language (DDL) Kodları 52](#_Toc46883818)

[Değişken Kullanımı 52](#_Toc46883819)

[Date Time fonksiyonları 53](#_Toc46883820)

[T-SQL’de Döngüler 55](#_Toc46883821)

[Index 57](#_Toc46883822)

[Kaynakça 58](#_Toc46883823)

# Uygulamalarla SQL Öğreniyorum

## Giriş

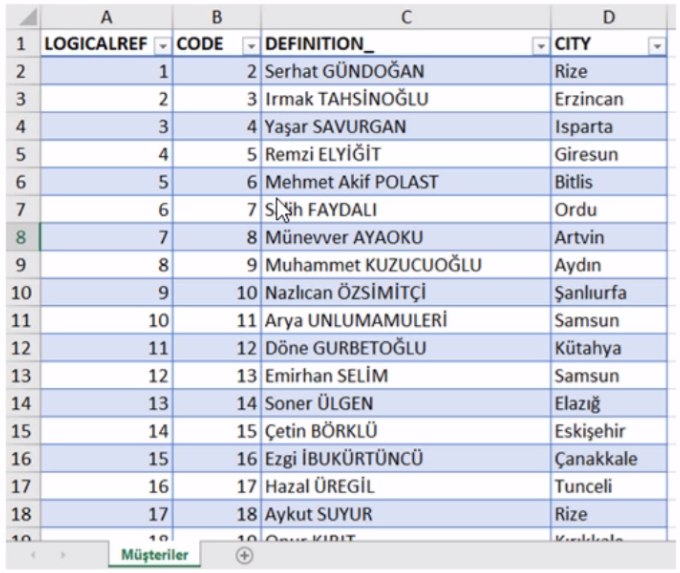
### SQL Ne Demektir?

**SQL = Structured Query Language (**Yapısal Sorgulama Dili**)**

Veritabanlarındaki verileri yönettiğimiz bir çeşit kodlama dilidir.

### İlişkisel Veritabanı Sistemleri (RDMS)

Tekrar eden verileri tekilleştirmek amacı ile yapılandırılan veritabanı sistemleridir.



## Temel SQL Komutları

### Data Manipülasyon Komutları

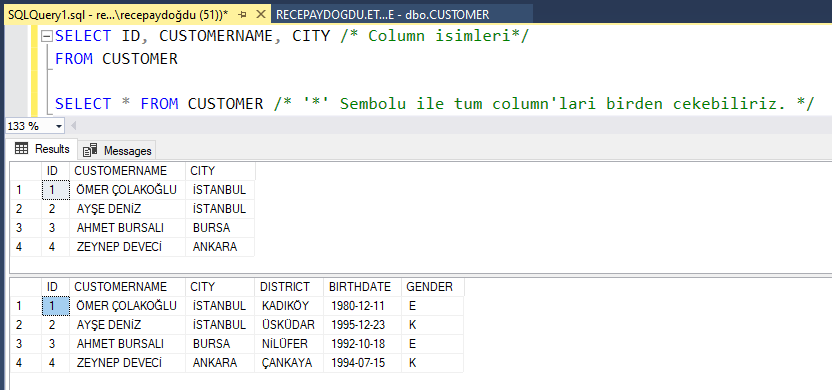
* **SELECT :** Veritabanından kayıtları çeker.
* **UPDATE :** Bir tablodaki kaydın bir ya da daha fazla alanını günceller, değiştirir.
* **DELETE :** Bir tablodan kayıt siler.
* **INSERT INTO :** Tabloya yeni kayıt ekler.
* **TRUNCATE TABLE :** Tablonun içini boşaltır.

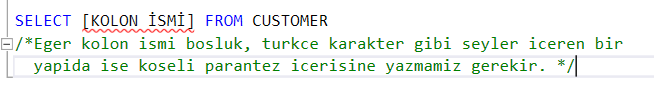
### Database Manipülasyon Komutları

* **CREATE DATABASE :** Yeni veritabanı oluşturur.
* **ALTER DATABASE :** Bir veritabanının özelliklerini değiştirir.
* **CREATE TABLE :** Yeni bir tablo oluşturur.
* **ALTER TABLE :** Bir tablonun özelliklerini değiştirir.
* **DROP TABLE :** Bir tabloyu tamamen siler.
* **CREATE INDEX :** Index oluşturur.
* **DROP INDEX :** Index’i siler.

### Select Komutu

**Select** komutu veritabanından kayıtları çekmemizi sağlar.



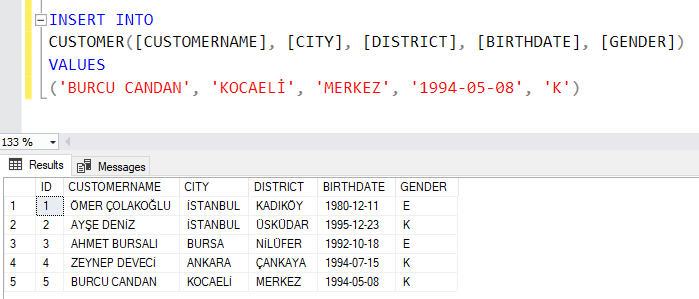


### Insert Komutu

Bir tabloya kayıt eklediğimiz komut.

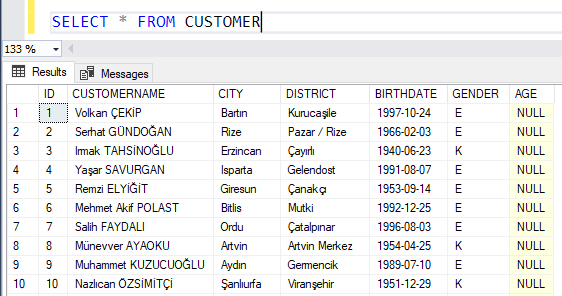
CUSTOMER isimli tablomuzda şuan 4 kaydımız var. 

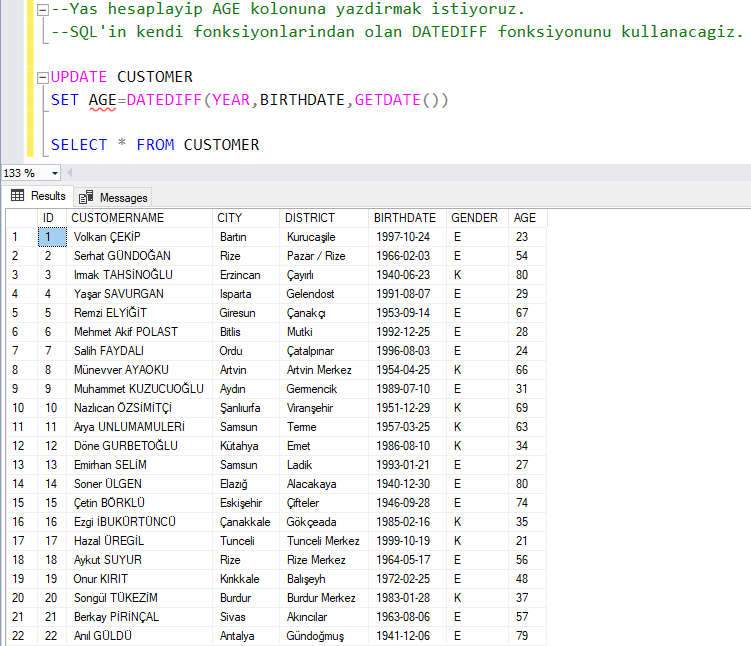
**INSERT INTO** komutu ile yeni bir kayıt ekleyelim.



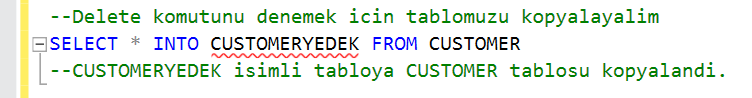
### Update Komutu

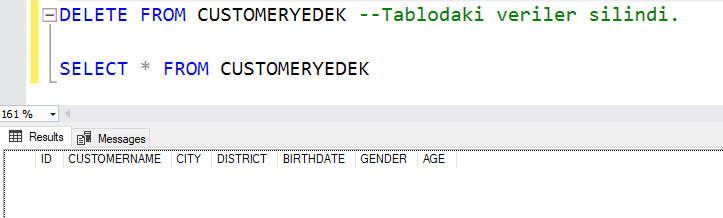
Veritabanındaki tablolarda herhangi bir ya da birden fazla alanı değiştirmek istediğimizde **UPDATE** komutunu kullanırız.

AGE adında yeni bir sütun ekledik. 



### Delete Komutu

Tablonun içindeki verileri siler. 



CUSTOMERYEDEK isimli tabloya yeni bir kayıt eklediğimizde içi boş olduğundan, ID değerinin 1’den başlamasını bekleriz. Fakat öyle olmaz, önceden 1000 adet kaydımız vardı. Tabloyu DELETE ile boşalttıktan sonra yeni bir kayıt eklediğimizde ID 1001’den saymaya başlayacaktır.

Bu durumun önüne geçmek için **TRUNCATE TABLE** komutunu kullanmamız gerekir.

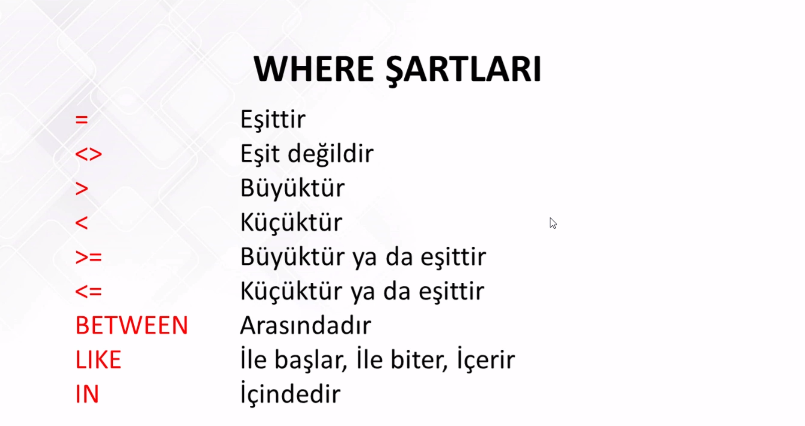


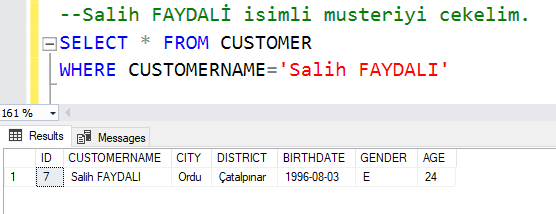
### Where Şartı Kullanımı

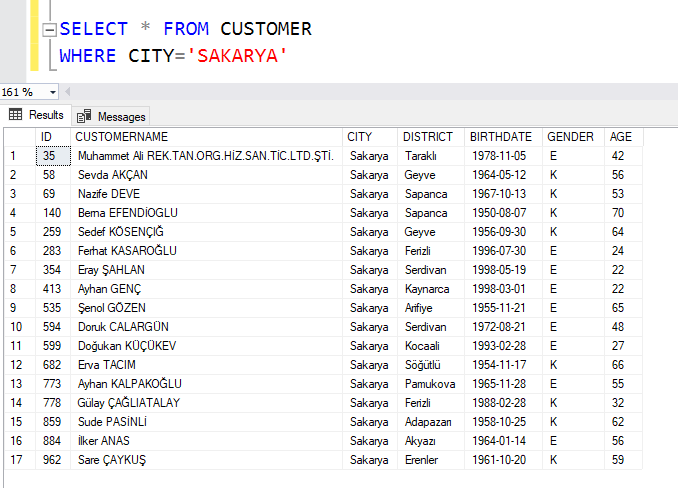
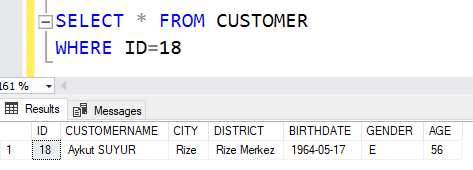
Normalde **SELECT \* FROM CUSTOMER** derken tüm müşterileri çekeriz.   
İsmi *Ömer Çolakoğlu* olan müşteriyi çekmek istediğimizde **WHERE** şartı bize yardımcı olacaktır.

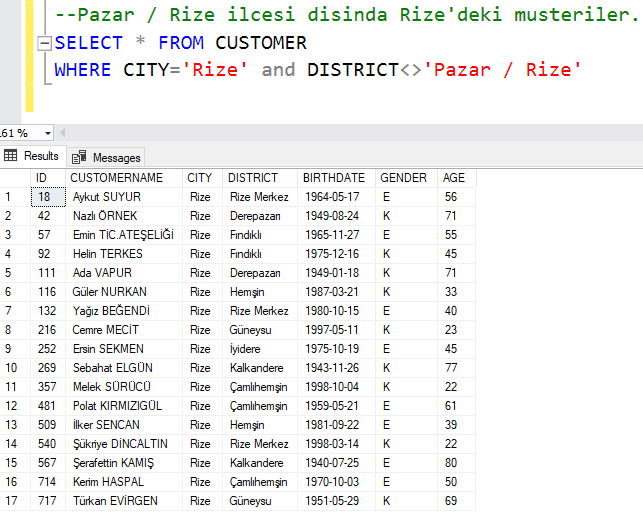
Ya da yaşı 20’den büyük olan müşterileri çekmek istediğimiz zaman yine **WHERE** şartı kullanırız.

**WHERE** şartı sadece SELECT’de değil UPDATE ve DELETE komutlarında da geçerlidir.

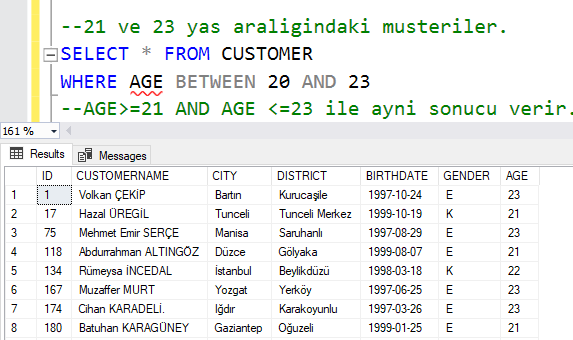


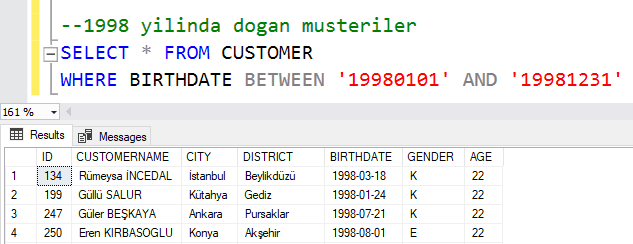




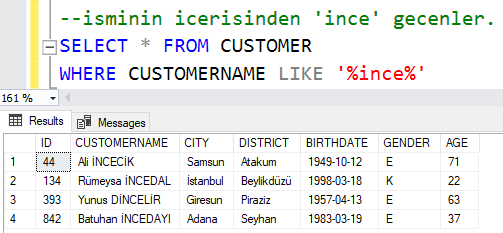
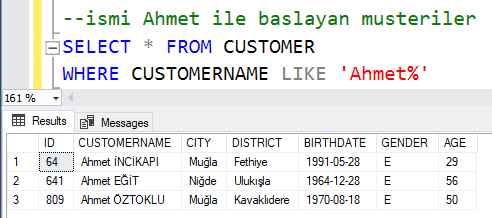
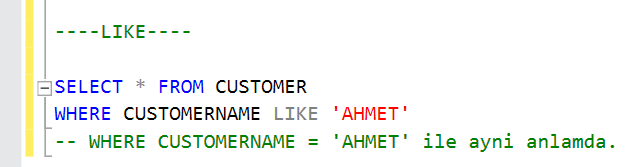


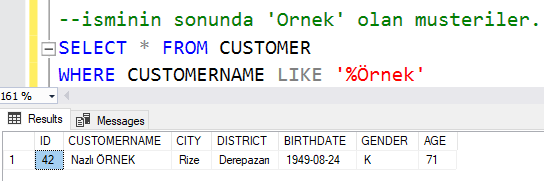
#### BETWEEN



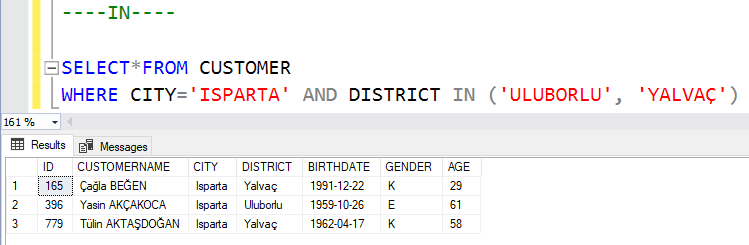


#### LIKE

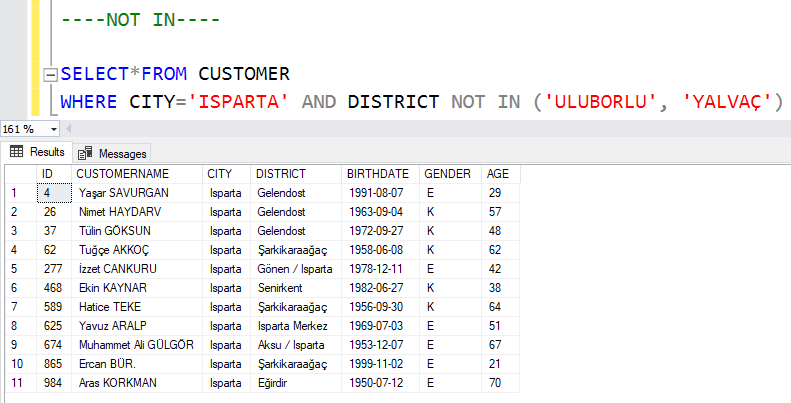




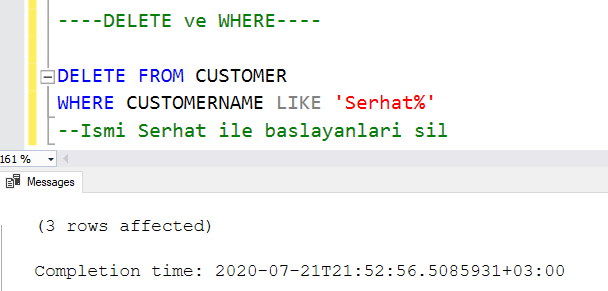
#### IN



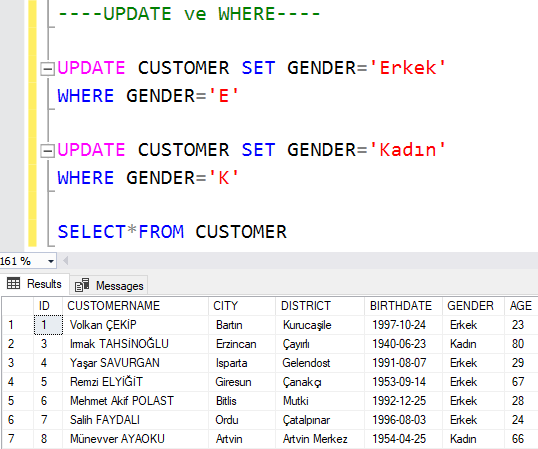
#### NOT IN



#### DELETE Komutu ile WHERE Şartı Kullanımı

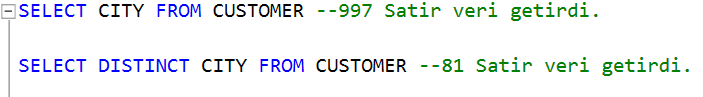


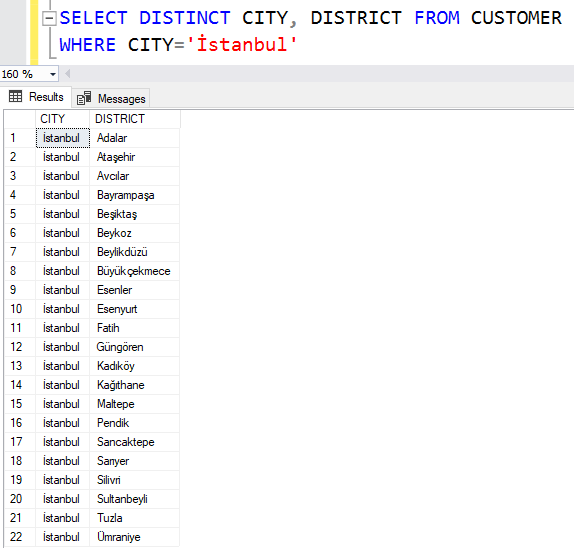
#### UPDATE Komutu ile WHERE Şartı Kullanımı



### Distinct Komutu

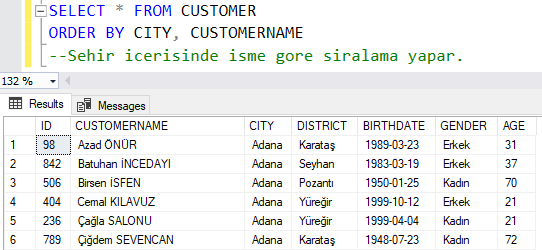
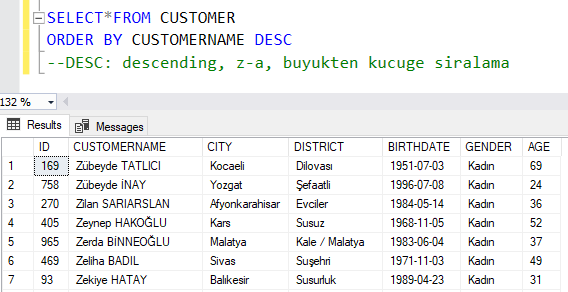
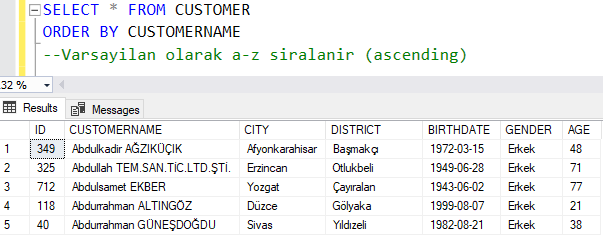
**DISTINCT** komutunu SELECT içerisinde kullanırız, tekrar eden satırlar için tek bir değer döndürür.



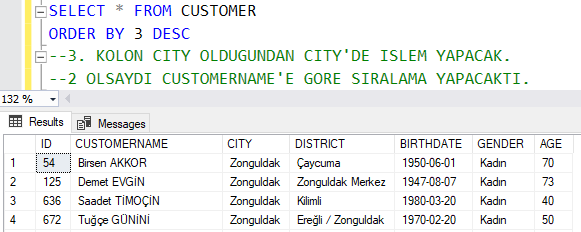


### Order By Komutu

**ORDER BY** bir sıralama komutudur.

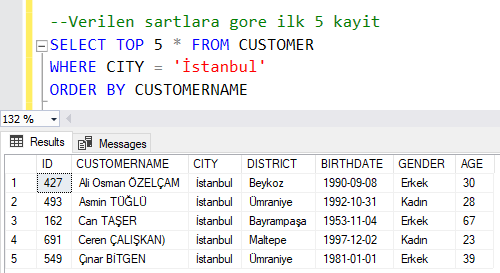


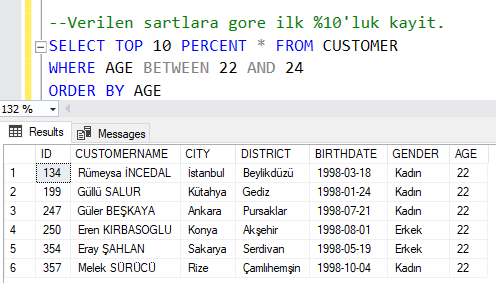




### Top Komutu

**TOP** Komutu veri setinde belirli bir sayı kadar ya da belirli bir yüzde kadar satır döndürmemizi sağlar.

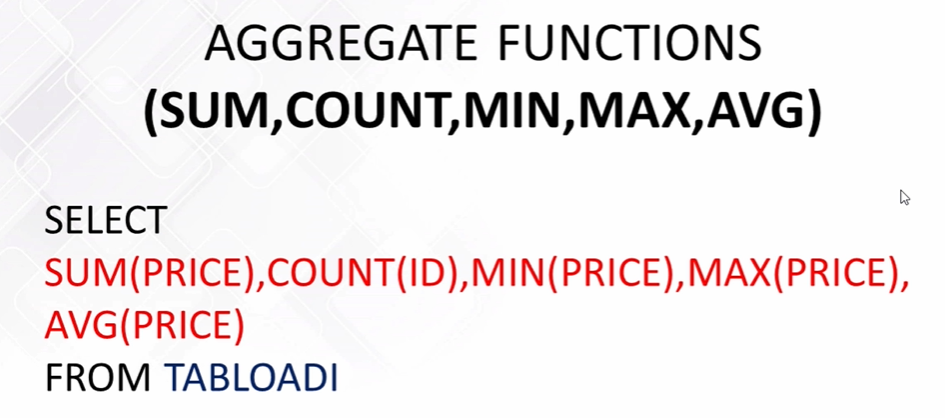


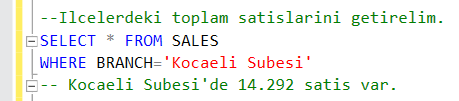


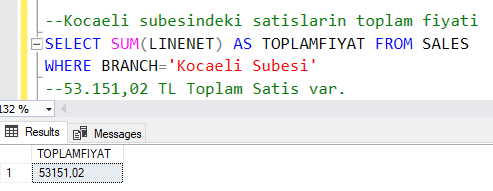
## Örnek Satış Datası

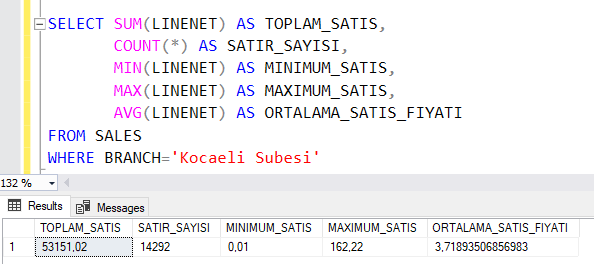
Gerçek bir veri seti üzerinde işlemler yapmak için örnek satış veritabanımızı SQL Server’a ekledik.

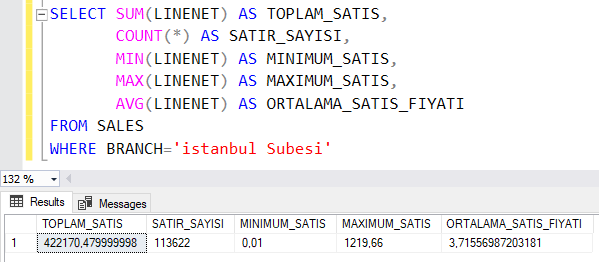
### Aggregate Fonksiyonlar



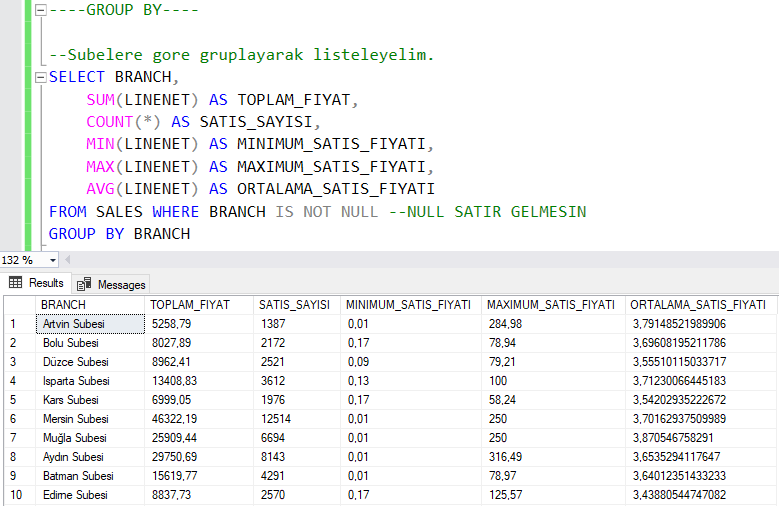








### Group By Kullanımı

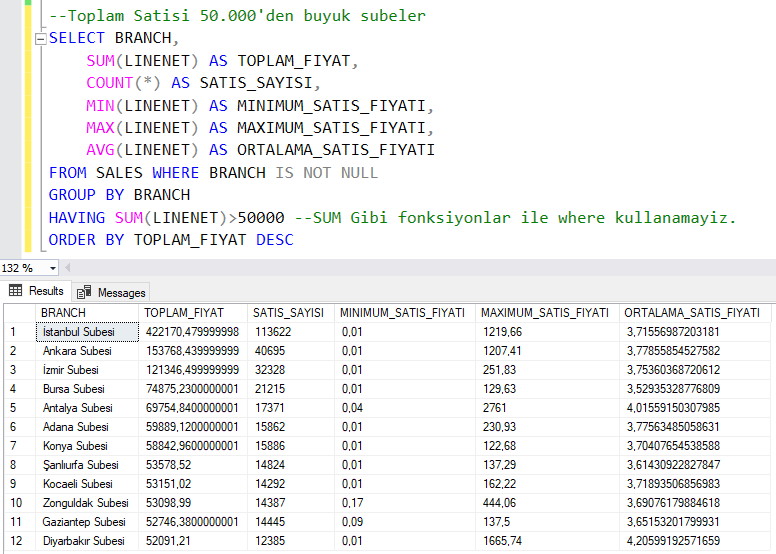


81 adet şube getirdi.

#### En Çok Satış Yapan Şubeler

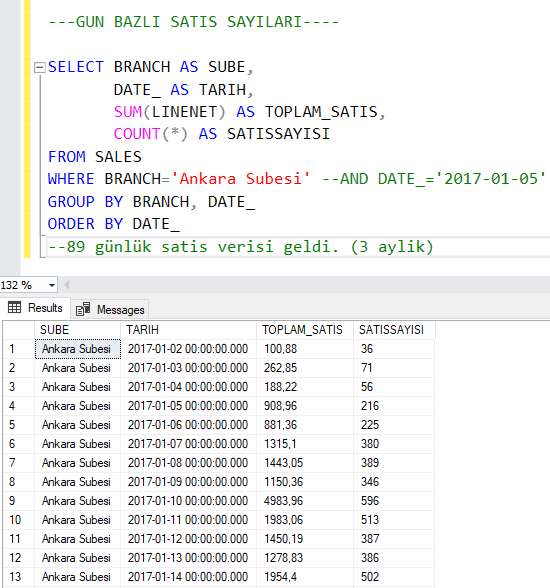


#### Toplam Satış Fiyatı 50.000’den Büyük Olan Şubeler



**SUM** Gibi fonksiyonlar ile koşul sağlamak istediğimizde **WHERE** yerine **HAVING** kullanmalıyız.

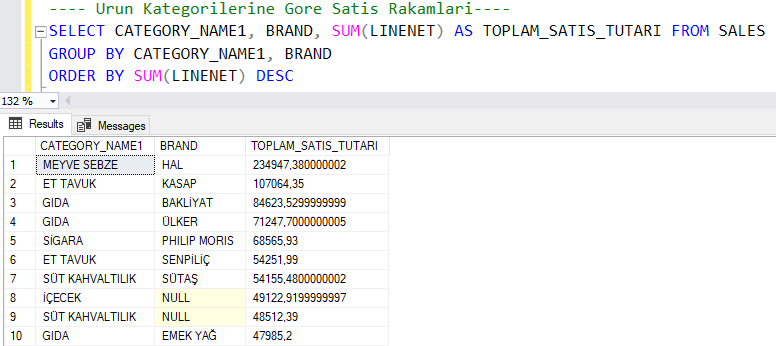
#### Bir Mağazanın Gün Bazlı Satışları

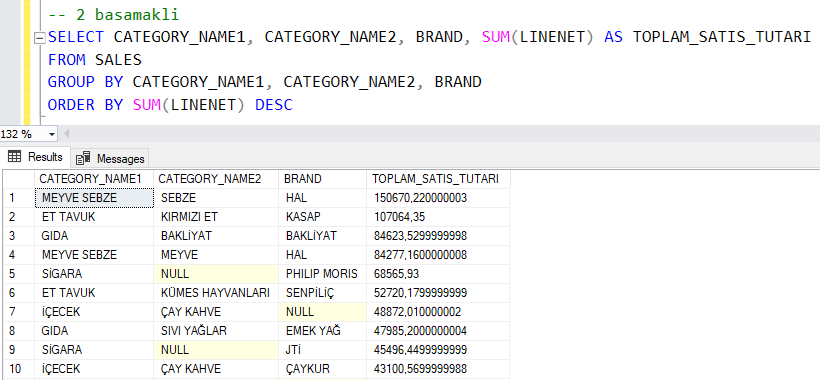


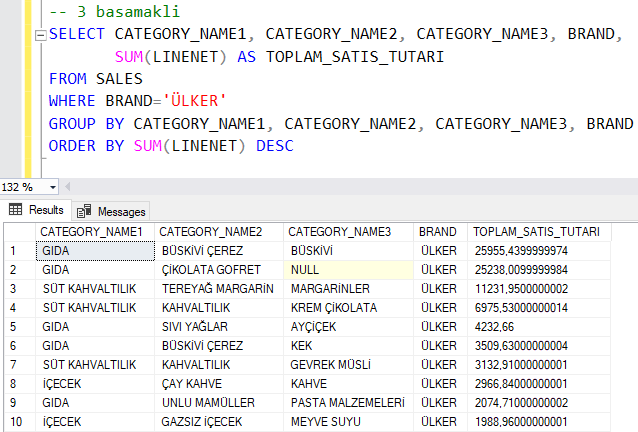
#### Bir Gündeki Mağaza Bazlı Satışlar



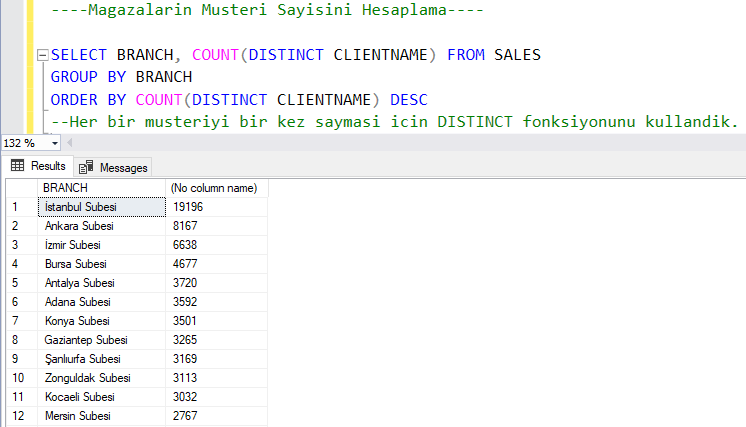
#### Ürün Kategorilerine Göre Toplam Satış Tutarları

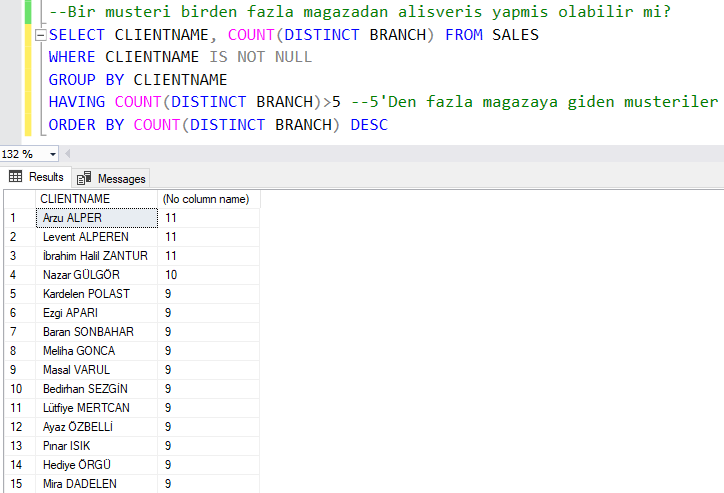


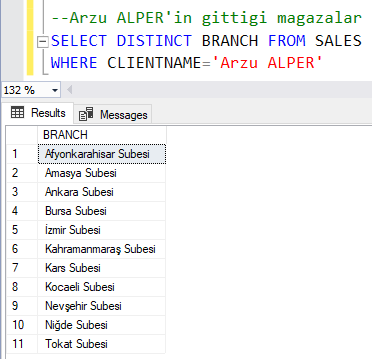




#### Mağazaların Müşteri Sayısını Hesaplama

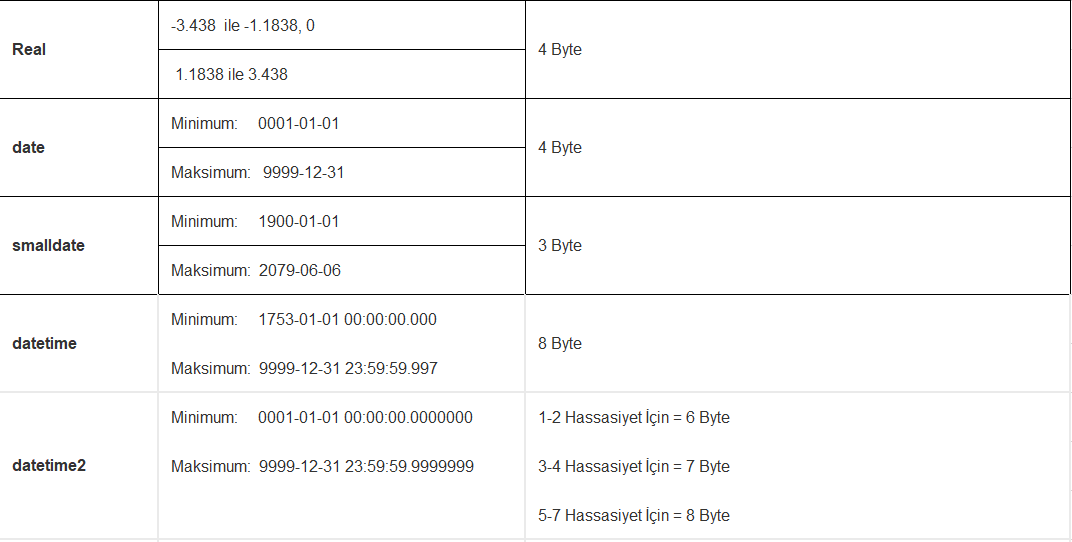
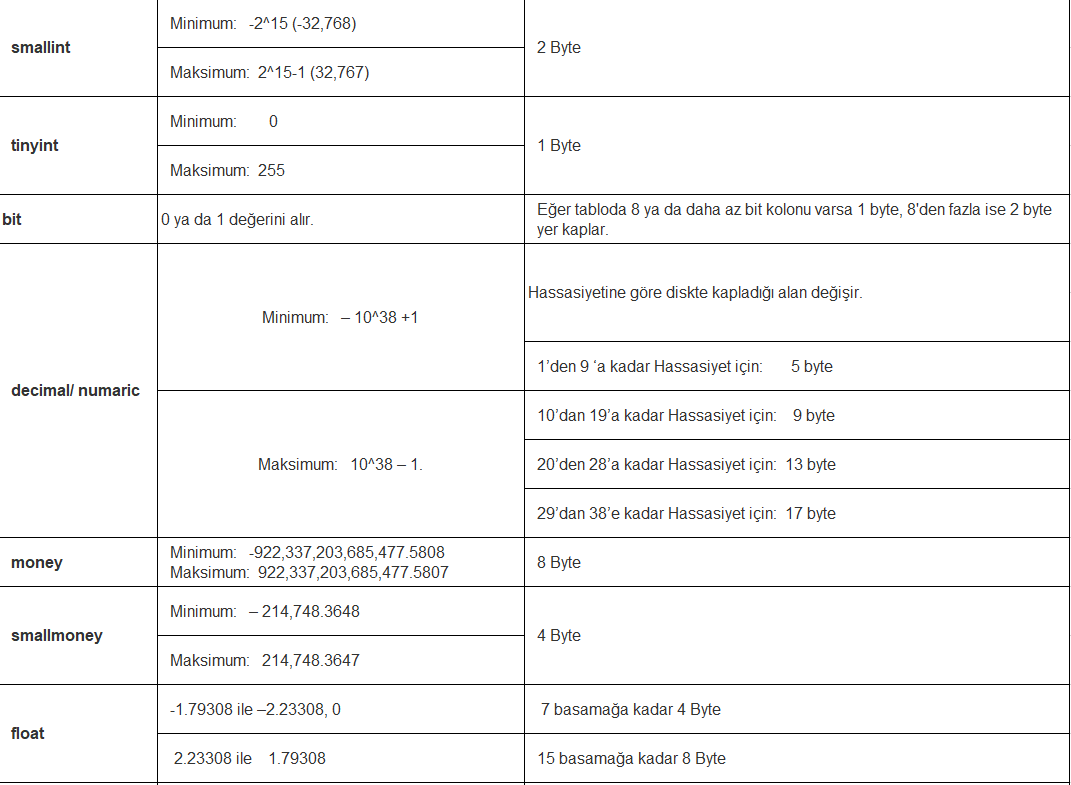


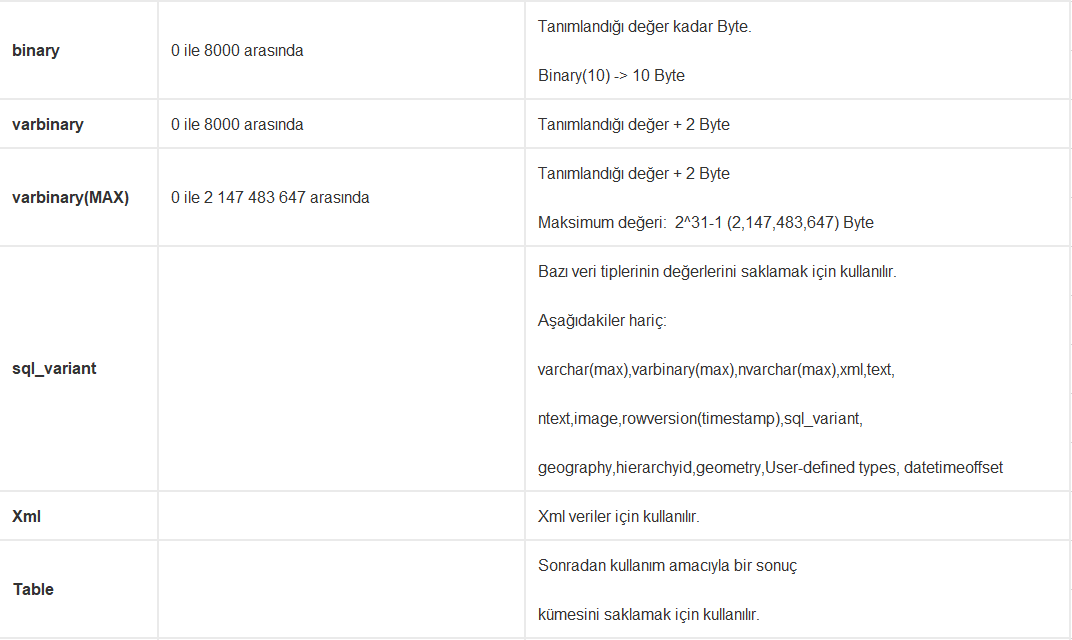
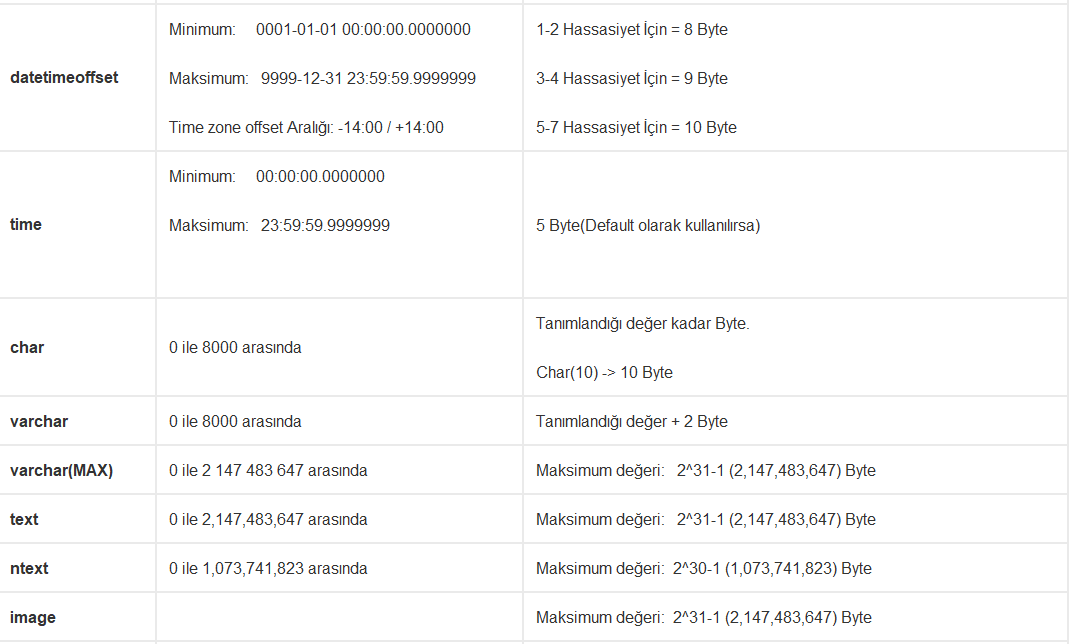


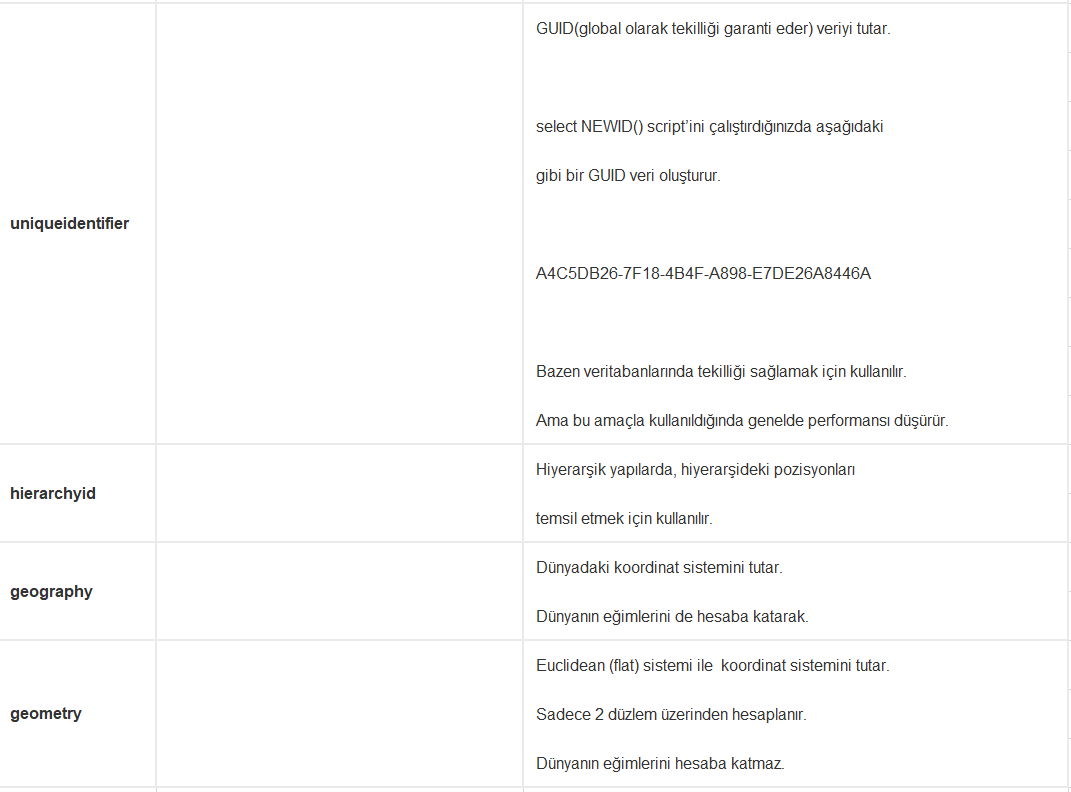


## SQL Server Veri Tipleri

[Veri Tipleri Excel dosyası](veri_tipleri.xlsx)ndan detayları görebiliriz.

[](veri_tipleri.xlsx)

[](veri_tipleri.xlsx)

[](veri_tipleri.xlsx)

## İlişkisel Veritabanı Sistemleri (RDMS)

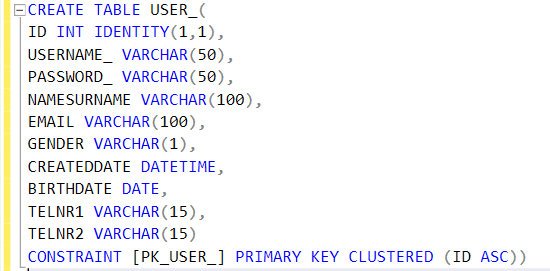


Bu E-Ticaret sisteminin veritabanını nasıl oluşturabileceğimize [bu dosyadan](../RDMS/vtOlusturma.xlsx) göz atalım.

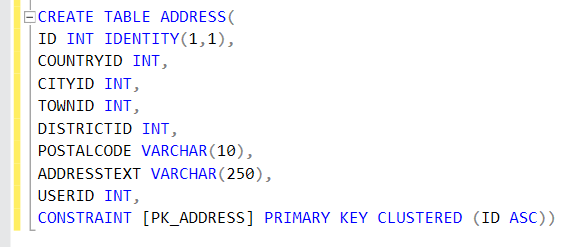
### SQL Server Tablo Oluşturma

Excel üzerinde oluşturulan tablo ve alanları SQL Server üzerinde oluşturalım.

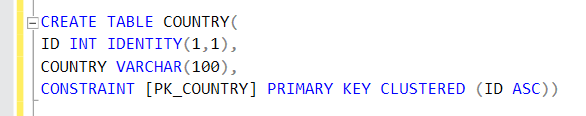
**USER TABLOSU**



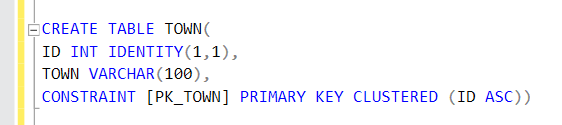
**ADDRESS TABLOSU**



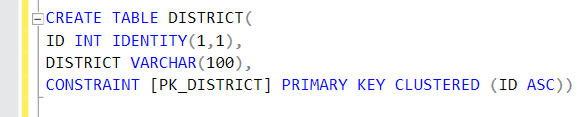
**COUNTRY TABLOSU**

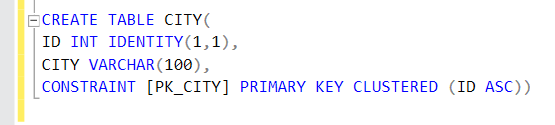


**TOWN TABLOSU**

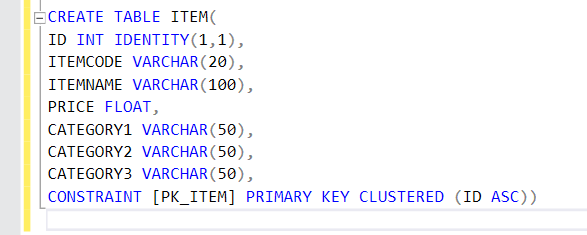


**DISTRICT TABLOSU**

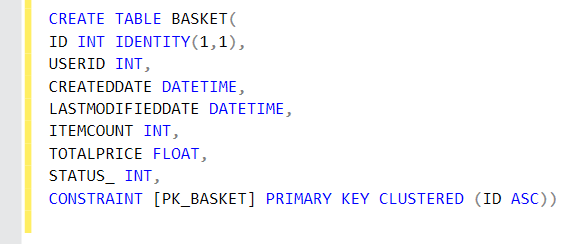
**CITY TABLOSU**



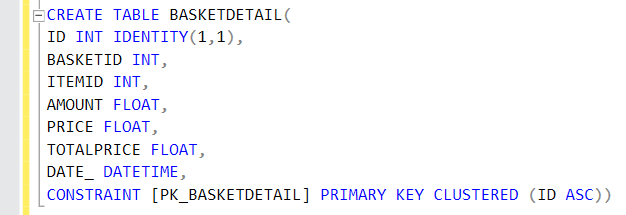
**ITEM TABLOSU**



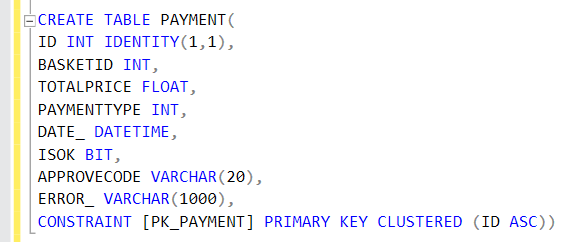
**BASKET TABLOSU**



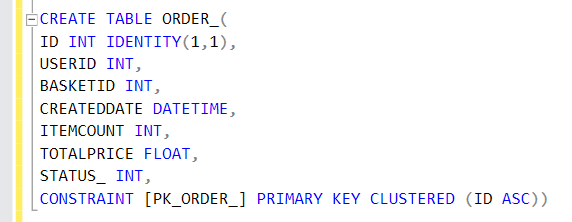
**BASKETDETAIL TABLOSU**



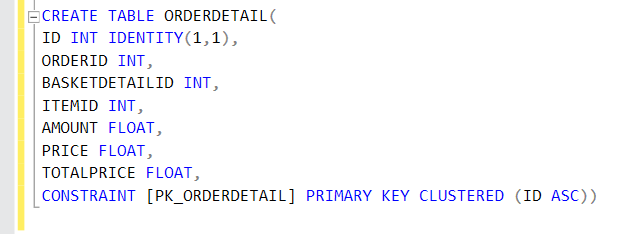
**PAYMENT TABLOSU**



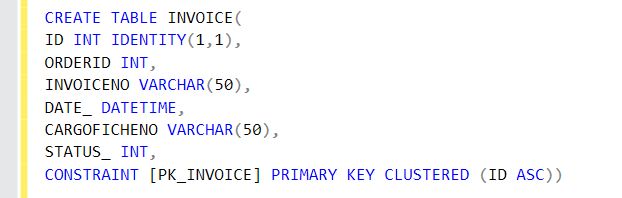
**ORDER TABLOSU**



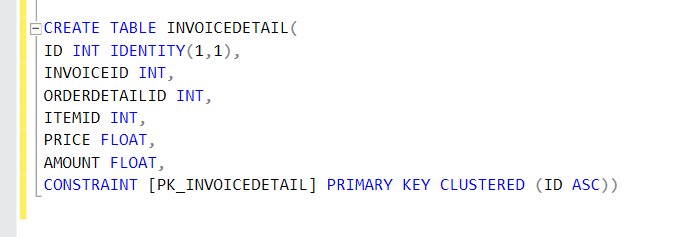
**ORDERDETAIL TABLOSU**



**INVOICE TABLOSU**



**INVOICEDETAIL TABLOSU**

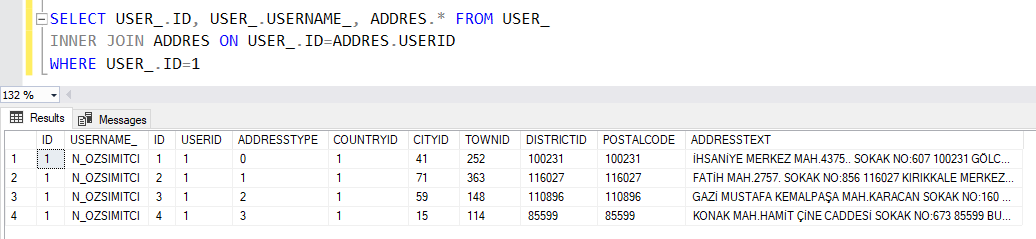
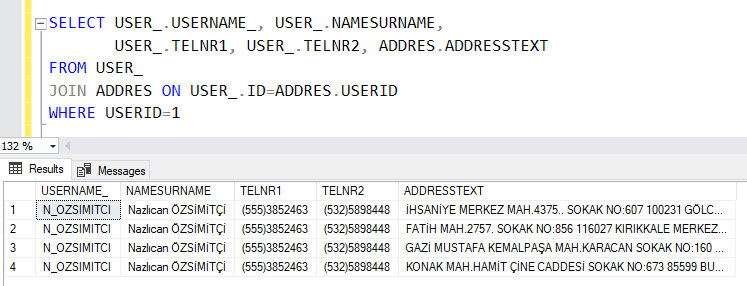


### Veri Oluşturma

RDMS isimli klasördeki ETRADE2.BAK dosyasını restore database işlemi ile SQL Server’a ekledik.

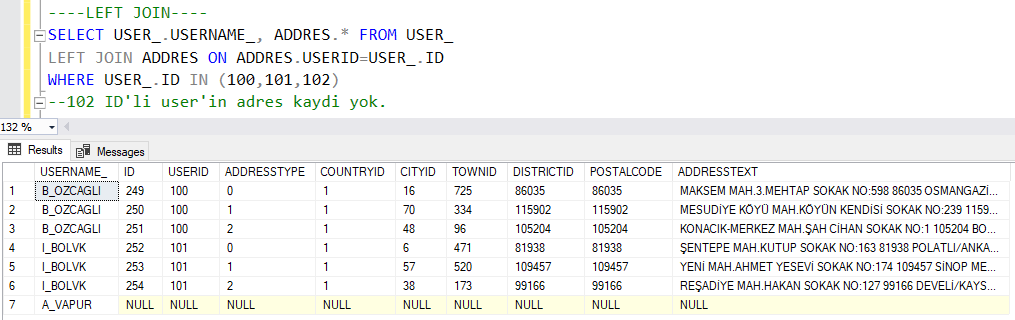
## Join İşlemleri

### INNER JOIN



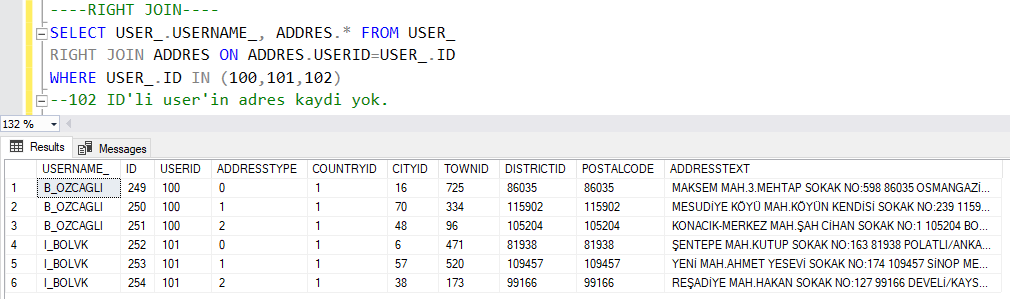
### LEFT JOIN

Sol taraftaki tablonun tüm elemanlarını getirir. Sol taraftaki satırın karşılığı sağ taraftaki tabloda yoksa yerine NULL yazar.



### RIGHT JOIN

Sağ taraftaki tablonun tüm elemanlarını getirir. Sağ taraftaki satırın karşılığı sol taraftaki tabloda yoksa yerine NULL yazar.

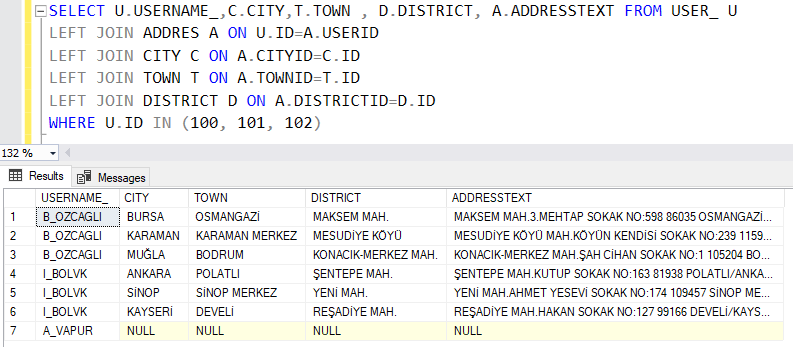


### FULL JOIN

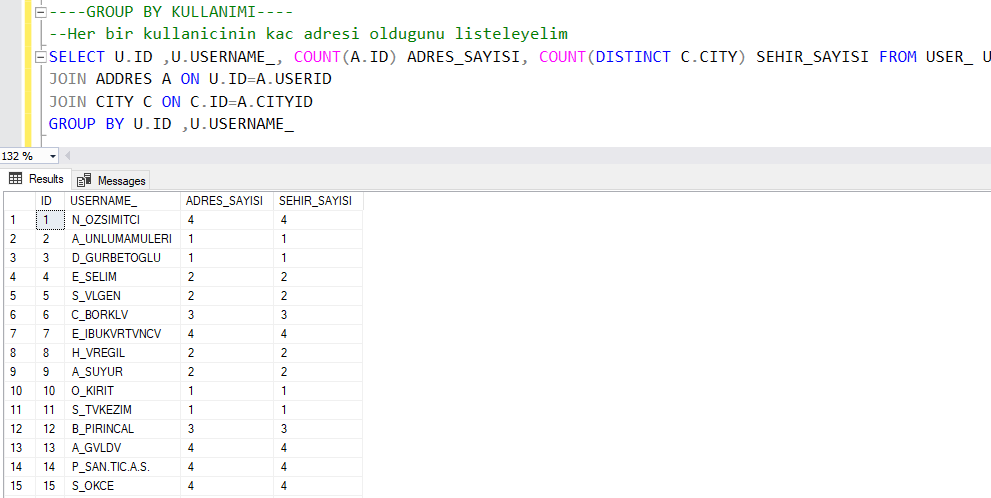
İki tablonun birleşimi.

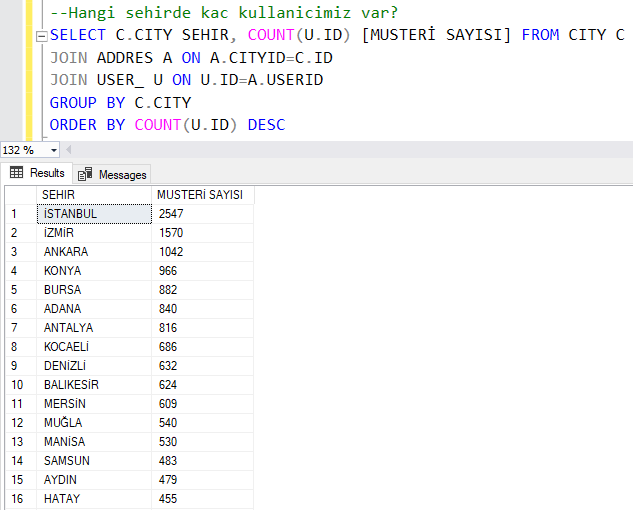


### ALIAS KULLANIMI



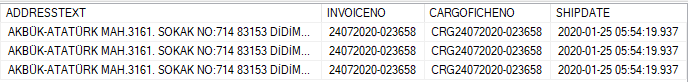
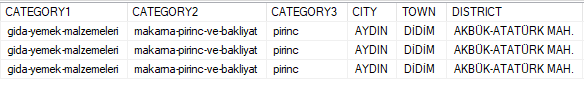
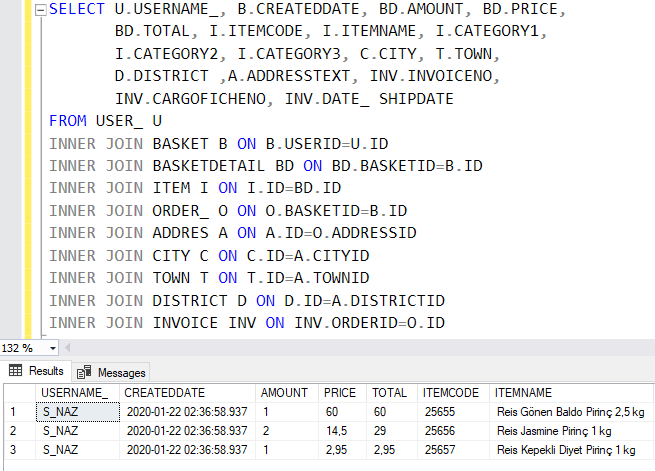
### GROUP BY KULLANIMI



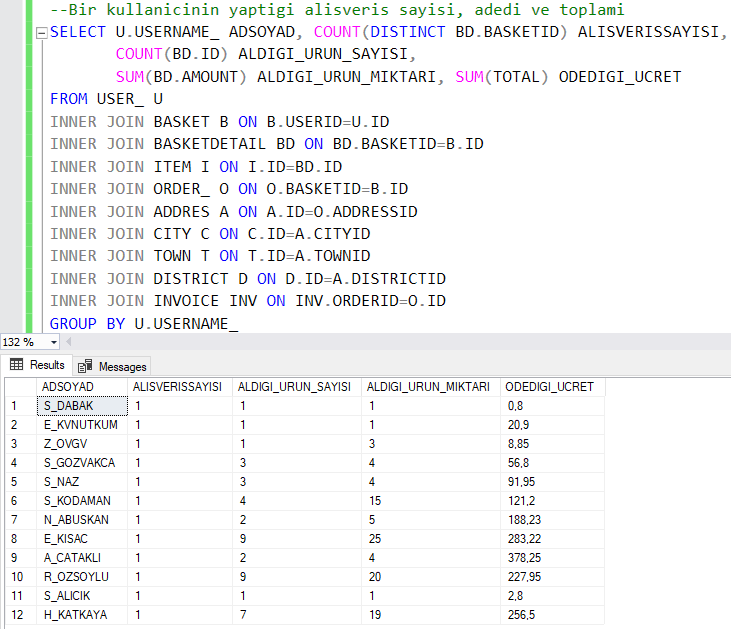


## E-Ticaret Datası Sorgulama

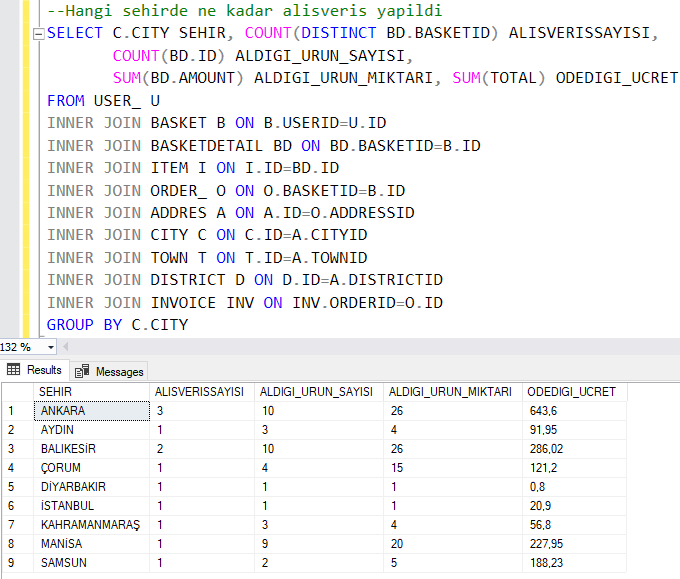
### Örnek-1



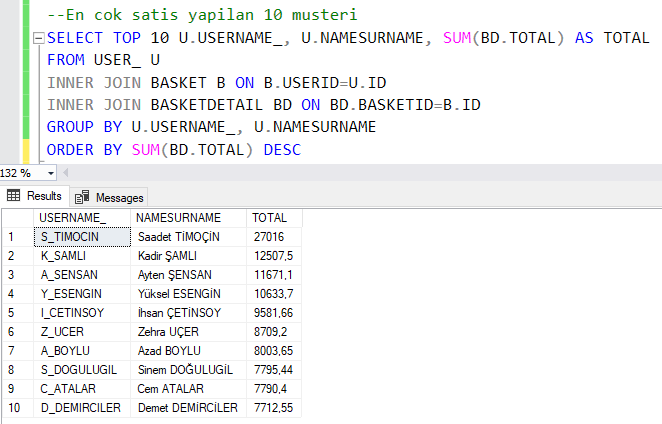
### Örnek-2



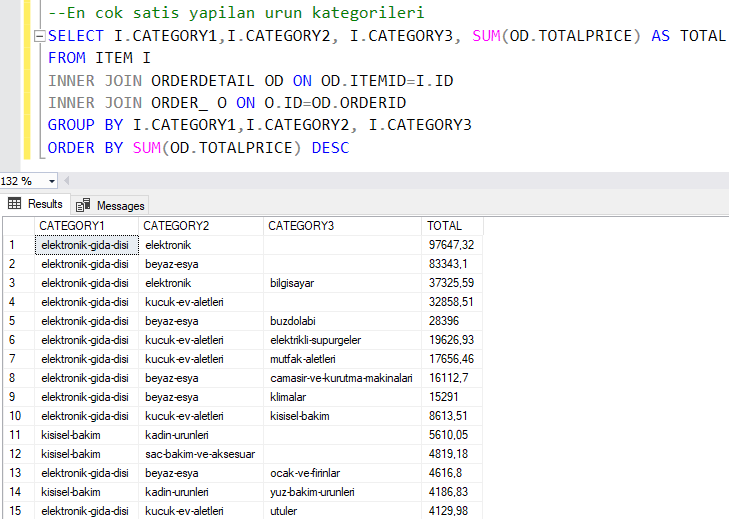
### Örnek-3



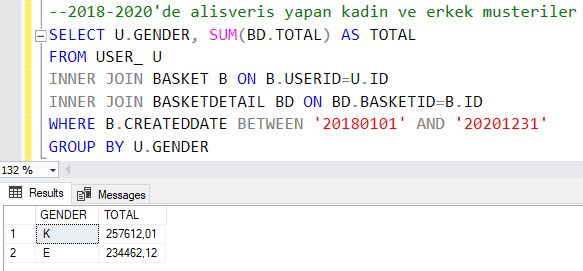
### Örnek-4



### Örnek-5

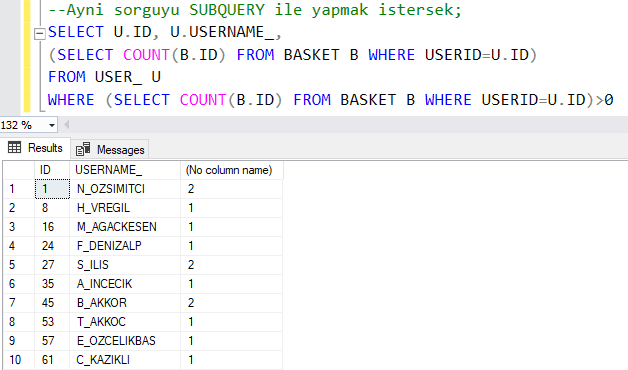
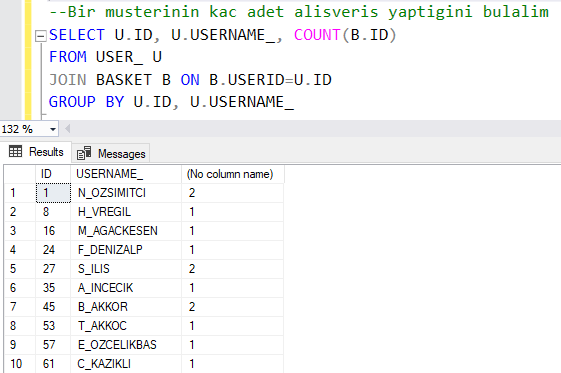


### Örnek-6



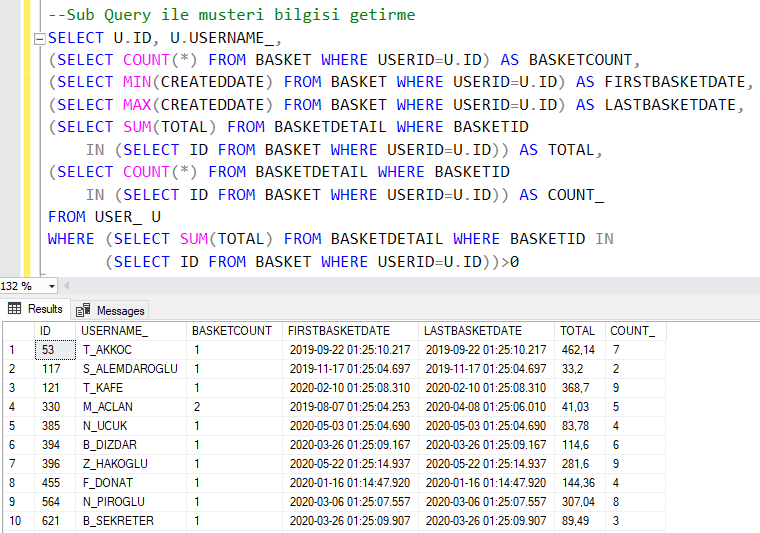
## Sub Query

### SubQuery Giriş

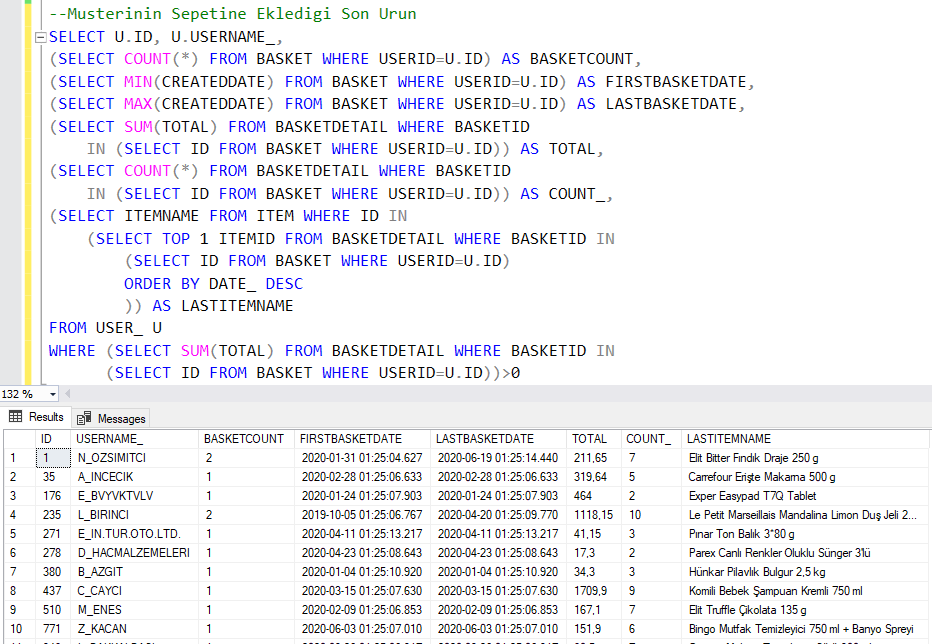


Hız olarak Join işlemi ve SubQuery işlemi neredeyse aynıdır. Kod yazımı olarak ise join biraz daha kolay diyebiliriz. Ancak bazı örnekler var ki sadece Sub Query kullanarak yapabiliriz.

### Örnek



### Örnek: Müşterinin Sepete Eklediği Son Ürün

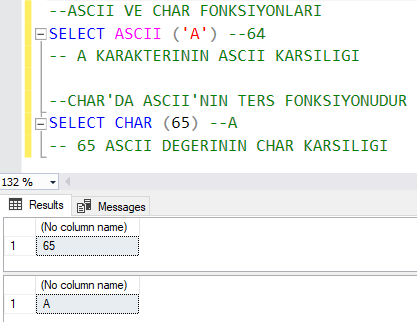


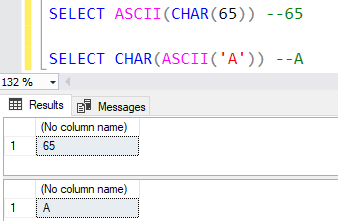
## String İşlemleri

### ASCII VE CHAR

ASCII fonksiyonu içerisine aldığı karakterin ASCII türünden değerini verir. (Byte olarak karşılığını verir.)

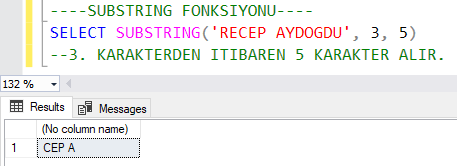
CHAR fonksiyonu ise ASCII fonksiyonunun tersidir.





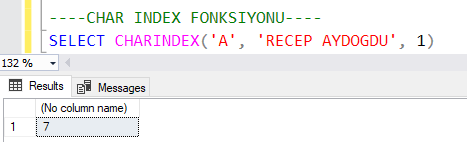
### SUBSTRING

SUBSTRING fonksiyonu bir string içerisinden belirli bir yerden bir yere kadar olan bölümü almamızı sağlar.



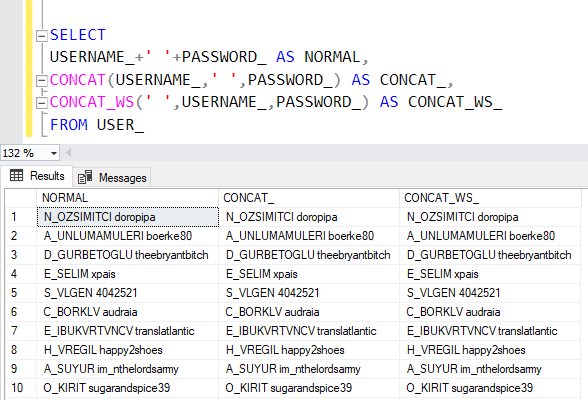
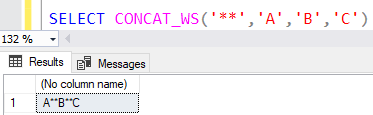
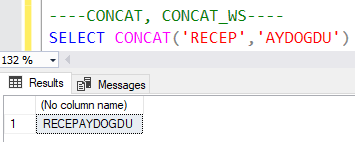
### CHAR INDEX

Bir stringin içerisinde başka bir stringi buldurup onun pozisyonunu öğrenmemizi sağlar.



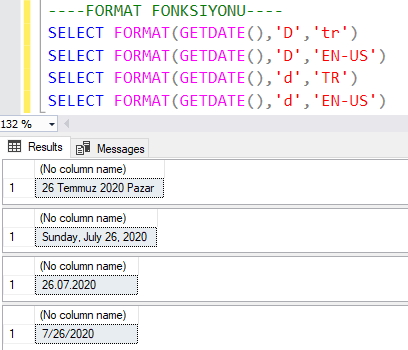
### CONCAT, CONCAT\_WS

CONCAT, iki veya daha fazla stringin yan yana birleştirilmesini sağlar.

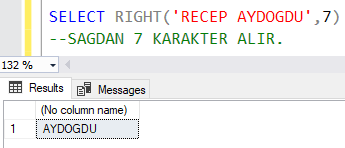


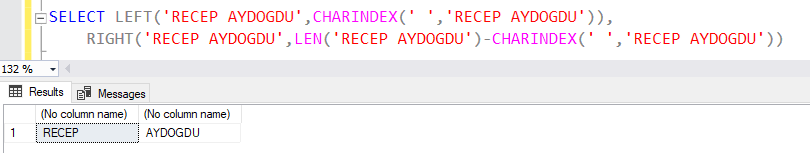
### FORMAT

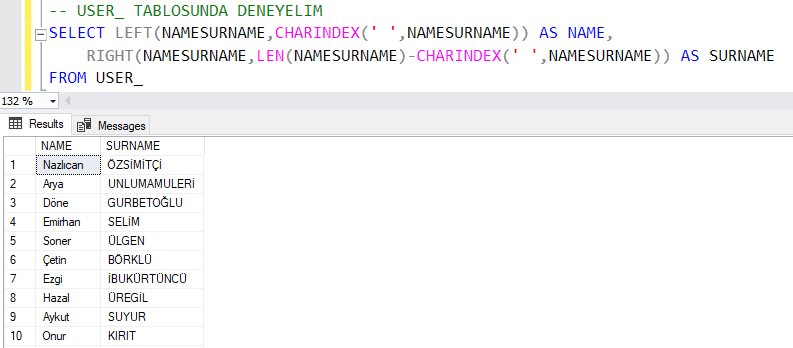
Sayı yada tarih türündeki değerleri local olarak istediğimiz formatta, istediğimiz dile uygun şekilde yazdırmamızı sağlar.



### LEFT, RIGHT, LEN

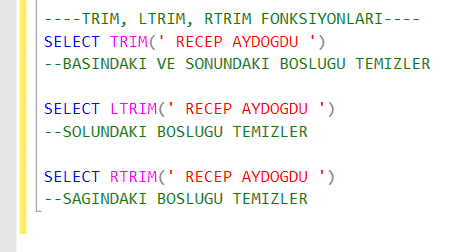






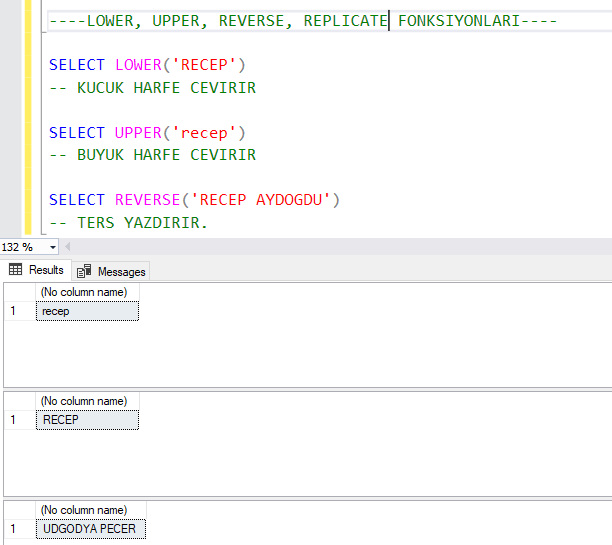
### TRIM, LTRIM, RTRIM

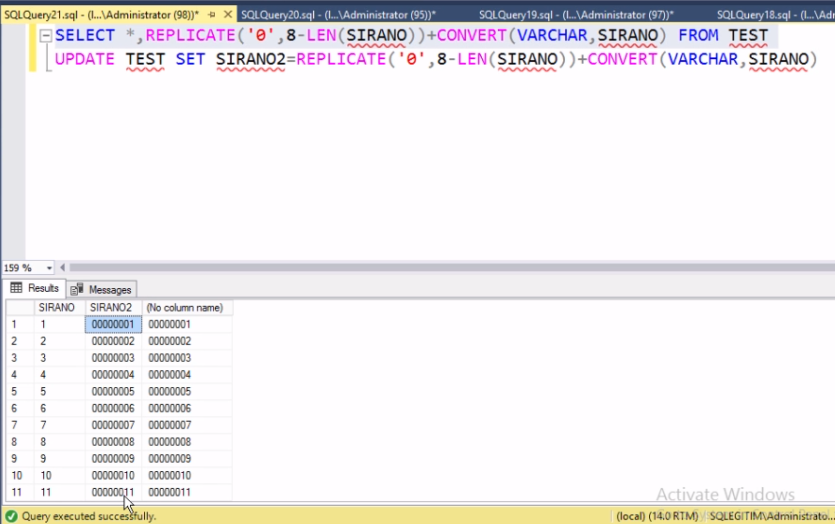
Boşlukları temizlemek için kullanılır.





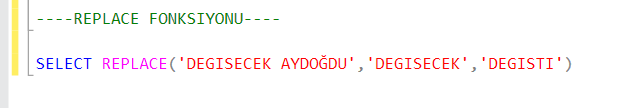
### LOWER, UPPER, REVERSE, REPLICATE





### REPLACE

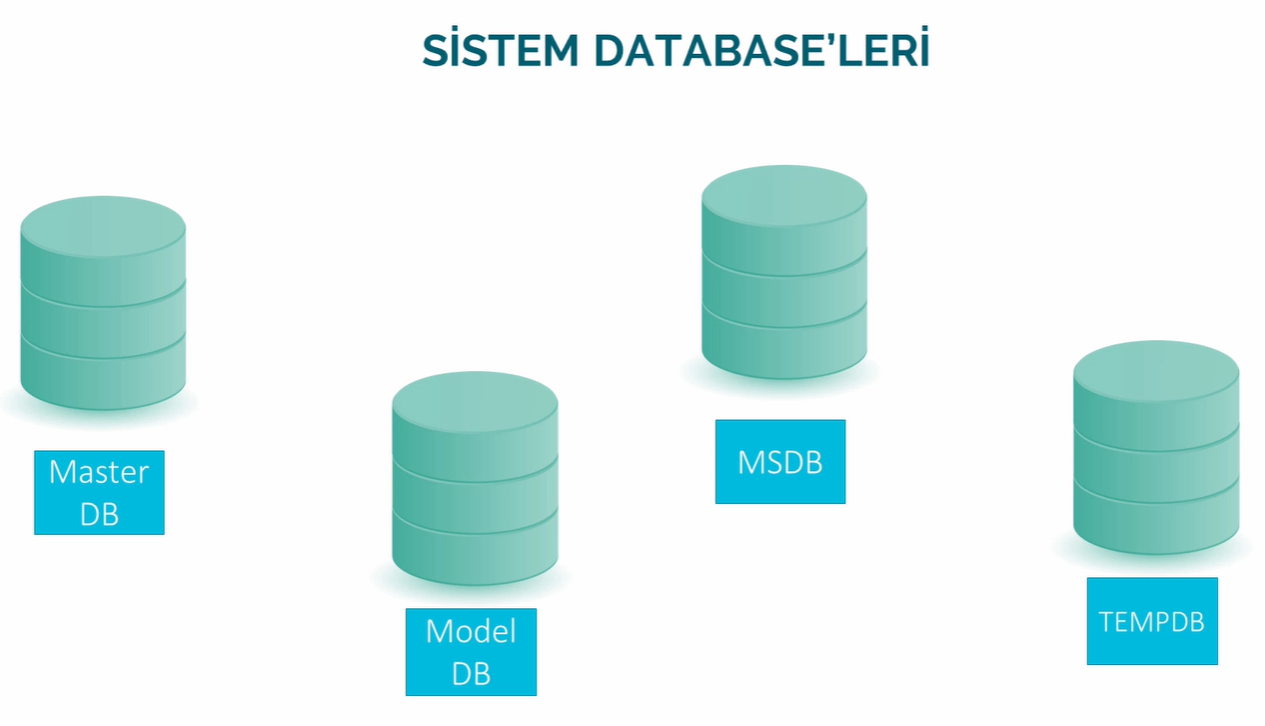
Bir şey ile başka bir şeyi yer değiştiren fonksiyondur.

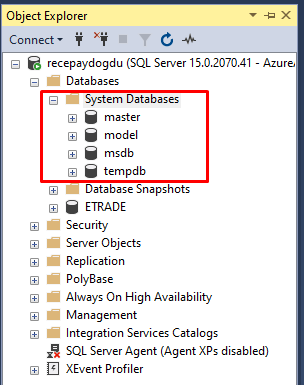


# Uçtan Uca SQL Server

## Sistem Database’leri

SQL Server kendisini yönetirken bazı databaseler ile yönetir, biz bunlara Sistem Database’leri deriz.

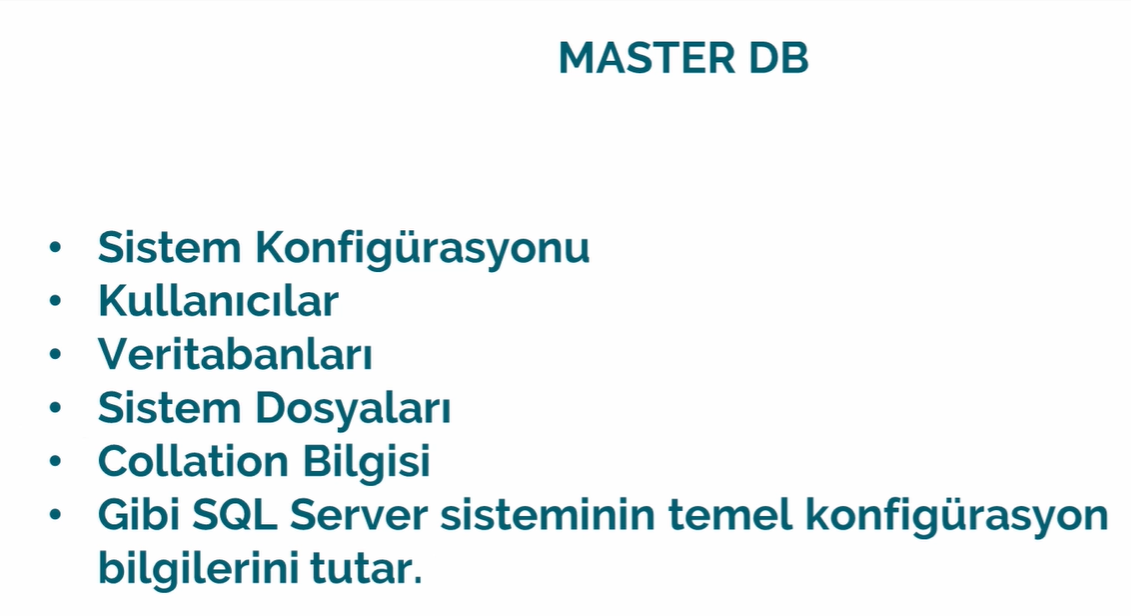




### Master DB

**MASTER** database’inin dosyası bizim için en önemlisidir.

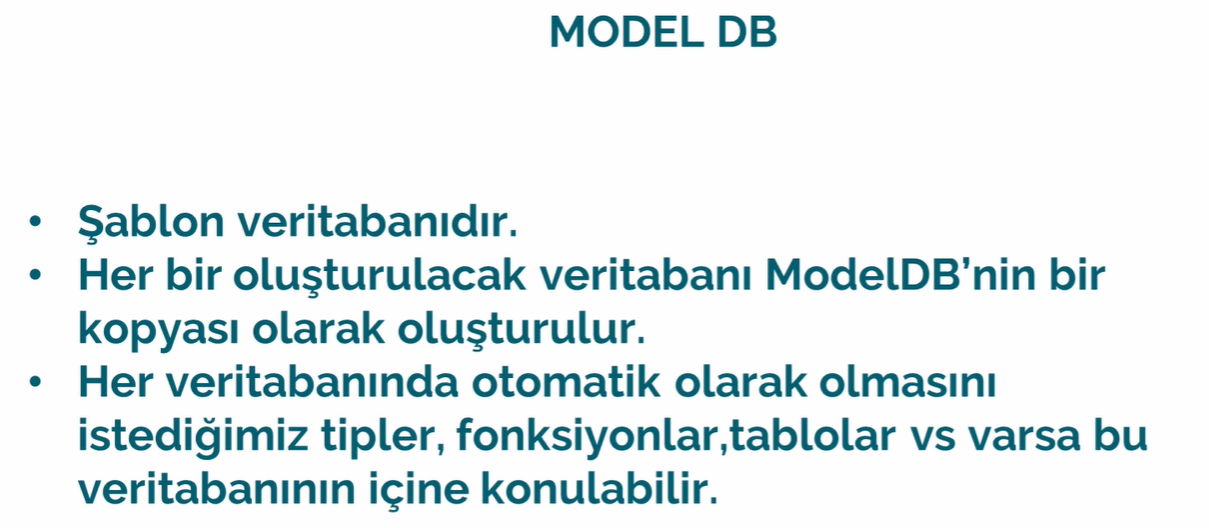
Master DB olmazsa SQL Server çalışmaz.



### Model DB

**MODEL** database’i ise yeni bir database oluştururken baz alınan database’dir.

Model DB olmazsa SQL Server çalışmaz.



### MSDB



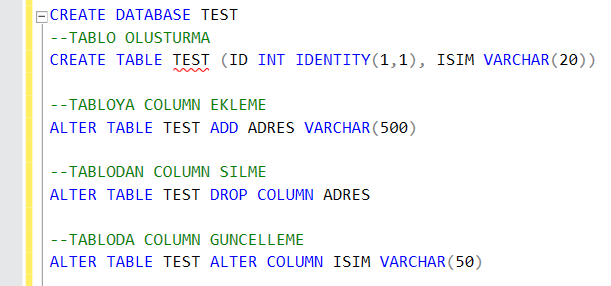
### Temp DB

**TEMPDB**’de ise geçici tablolar tutulur. Geçici tablo oluşturmak için tablo oluşturulurken başına # işareti konulur. Tek # işareti kullanarak oluşturulan tablolara sadece oluşturulduğu sessiondan ulaşılabilir. Çift # yani ## işareti kullanarak oluşturulan tablolara ise farklı sessionlardan da ulaşılabilir.

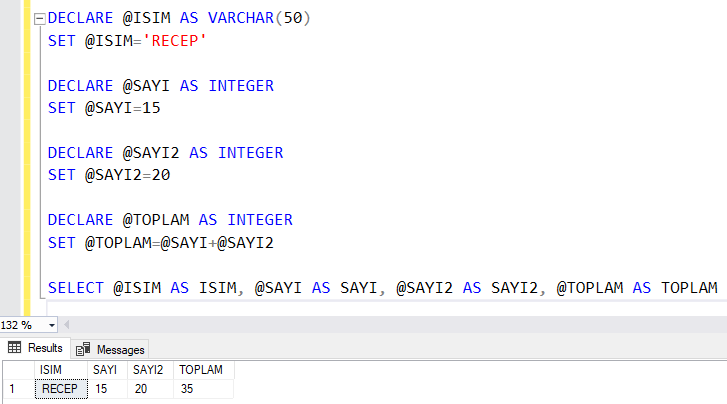


## T-SQL Kodları

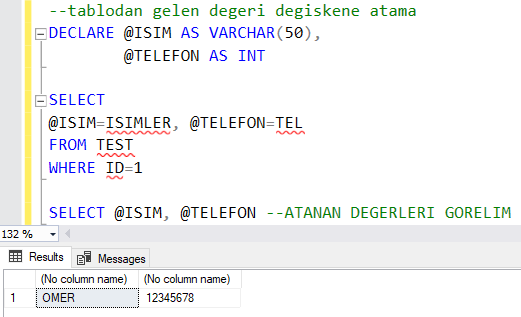
### Data Definition Language (DDL) Kodları



### Değişken Kullanımı



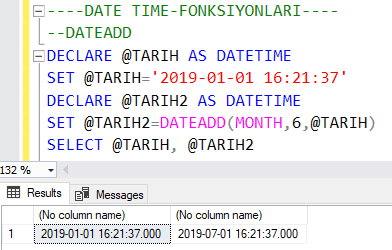
#### Tablodan Gelen Değeri Değişkene Atama



### Date Time fonksiyonları

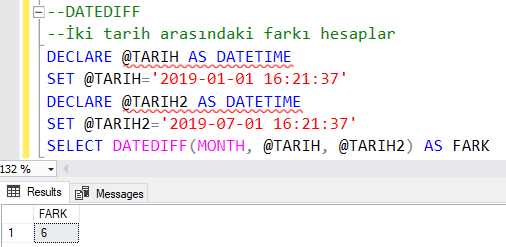
#### DATEADD

Belirlediğimiz tarihe ekleme yapmamızı sağlar.



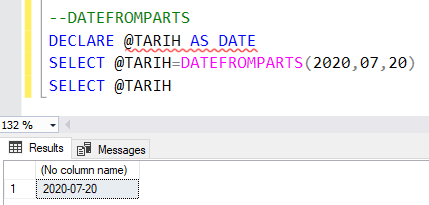
#### DATEDIFF

İki tarih arasındaki farkı hesaplar.



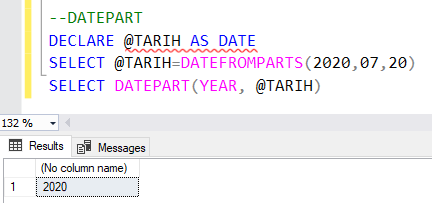
#### DATEFROMPARTS

İçine girdiğimiz değerler ile tarih oluşturur.

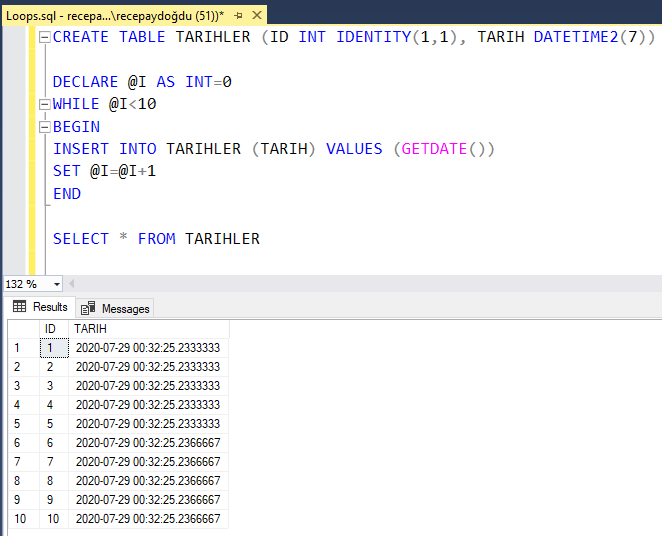


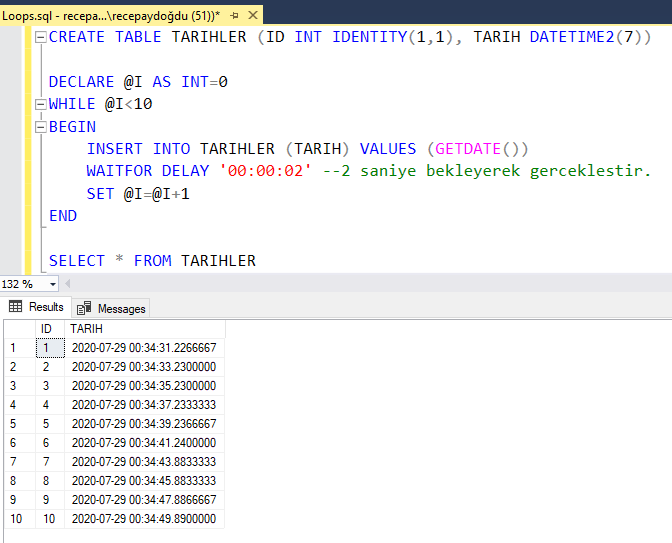
#### DATEPART

İçine girdiğimiz tarihin istediğimiz bölümünü verir.

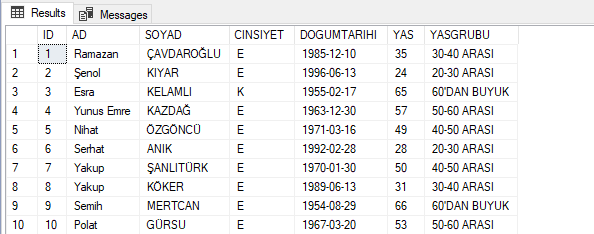
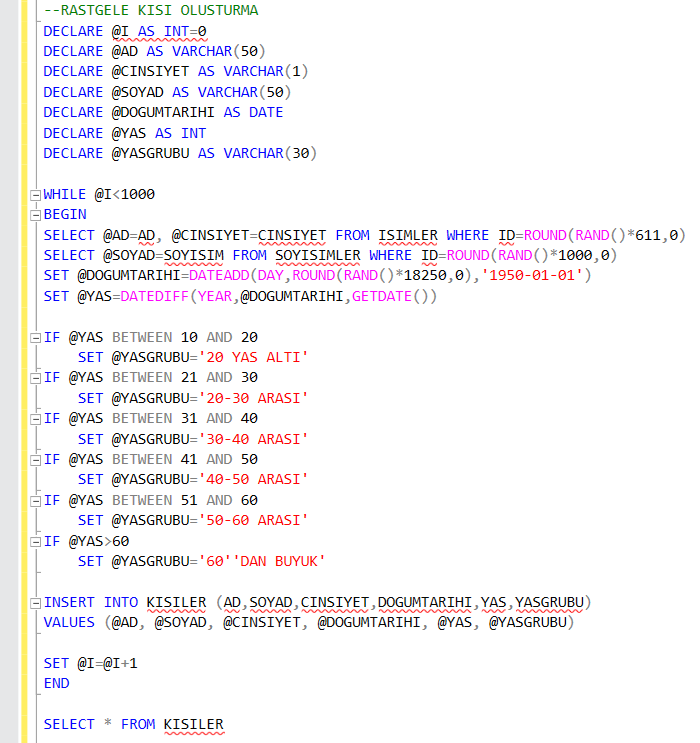


### T-SQL’de Döngüler





#### Rastgele Kişi Oluşturma



## Index

# Kaynakça

* **Uygulamalarla SQL Öğreniyorum – Ömer Çolakoğlu**

[**https://www.udemy.com/course/sql-ogreniyorum/**](https://www.udemy.com/course/sql-ogreniyorum/)

* **Uçtan Uca SQL Server – Ömer Çolakoğlu**

[**https://www.udemy.com/course/uctan-uca-sql-server-egitimi/**](https://www.udemy.com/course/uctan-uca-sql-server-egitimi/)