İçindekiler

[Uygulamalarla SQL Öğreniyorum 2](#_Toc46349214)

[Giriş 2](#_Toc46349215)

[SQL Ne Demektir? 2](#_Toc46349216)

[İlişkisel Veritabanı Sistemleri (RDMS) 2](#_Toc46349217)

[Temel SQL Komutları 3](#_Toc46349218)

[Data Manipülasyon Komutları 3](#_Toc46349219)

[Database Manipülasyon Komutları 3](#_Toc46349220)

[Select Komutu 3](#_Toc46349221)

[Insert Komutu 4](#_Toc46349222)

[Update Komutu 4](#_Toc46349223)

[Delete Komutu 5](#_Toc46349224)

[Where Şartı Kullanımı 6](#_Toc46349225)

[Distinct Komutu 12](#_Toc46349226)

[Order By Komutu 13](#_Toc46349227)

[Top Komutu 15](#_Toc46349228)

[Örnek Satış Datası 16](#_Toc46349229)

[Aggregate Fonksiyonlar 16](#_Toc46349230)

[Group By Kullanımı 17](#_Toc46349231)

[SQL Server Veri Tipleri 24](#_Toc46349232)

[İlişkisel Veritabanı Sistemleri (RDMS) 27](#_Toc46349233)

# Uygulamalarla SQL Öğreniyorum

## Giriş

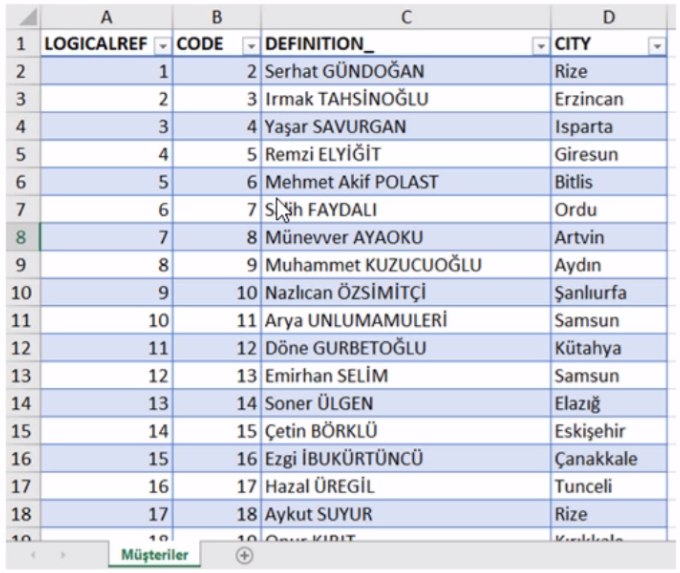
### SQL Ne Demektir?

**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

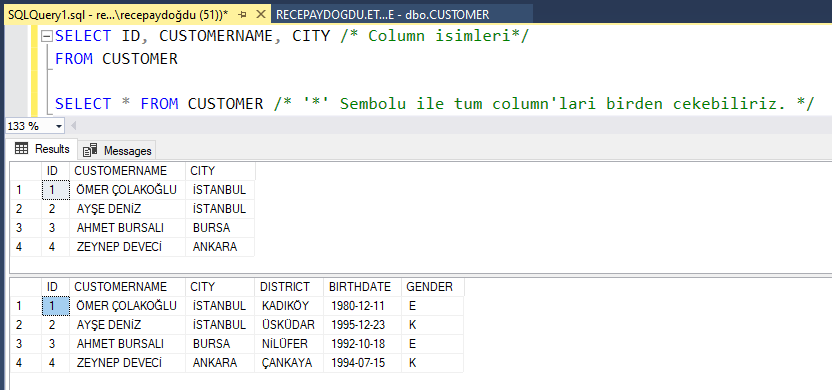
* **SELECT :** Veritabanından kayıtları çeker.
* **UPDATE :** Bir tablodaki kaydın bir ya da daha fazla alanını günceller, değiştirir.
* **DELETE :** Bir tablodan kayıt siler.
* **INSERT INTO :** Tabloya yeni kayıt ekler.
* **TRUNCATE TABLE :** Tablonun içini boşaltır.

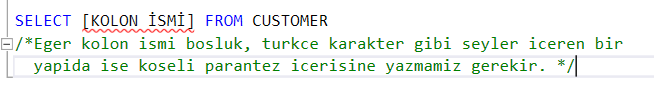
### Database Manipülasyon Komutları

* **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

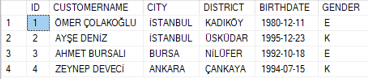
**Select** komutu veritabanından kayıtları çekmemizi sağlar.



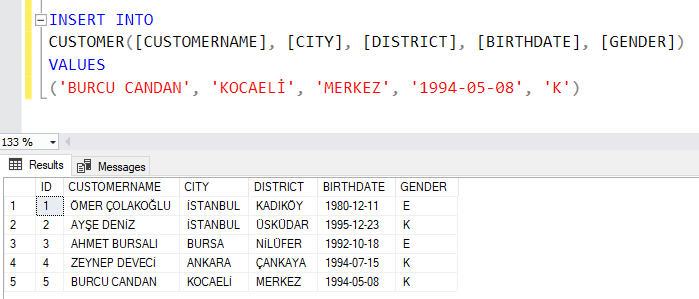


### Insert Komutu

Bir tabloya kayıt eklediğimiz komut.

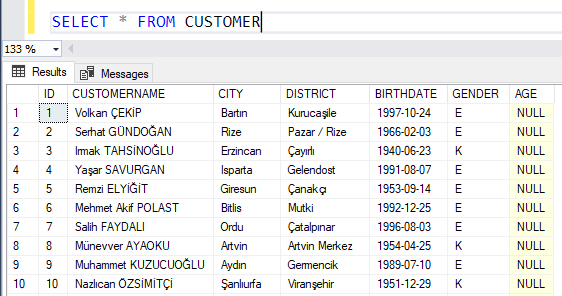
CUSTOMER isimli tablomuzda şuan 4 kaydımız var. 

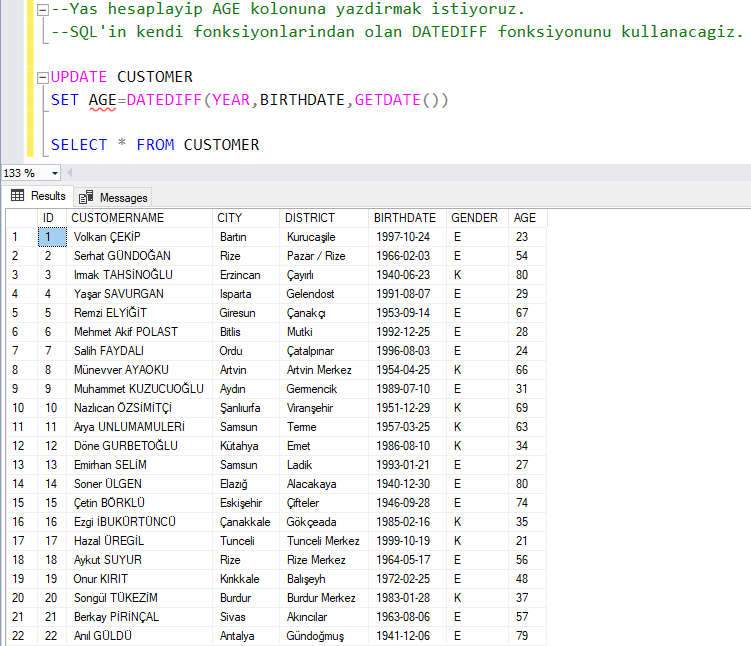
**INSERT INTO** komutu ile yeni bir kayıt ekleyelim.



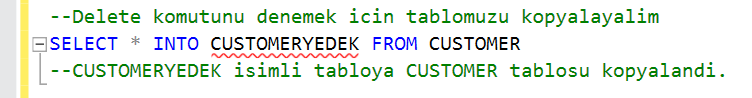
### Update Komutu

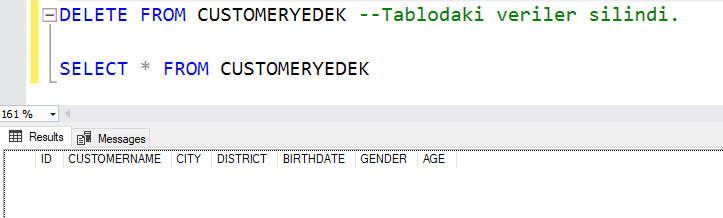
Veritabanındaki tablolarda herhangi bir ya da birden fazla alanı değiştirmek istediğimizde **UPDATE** komutunu kullanırız.

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.

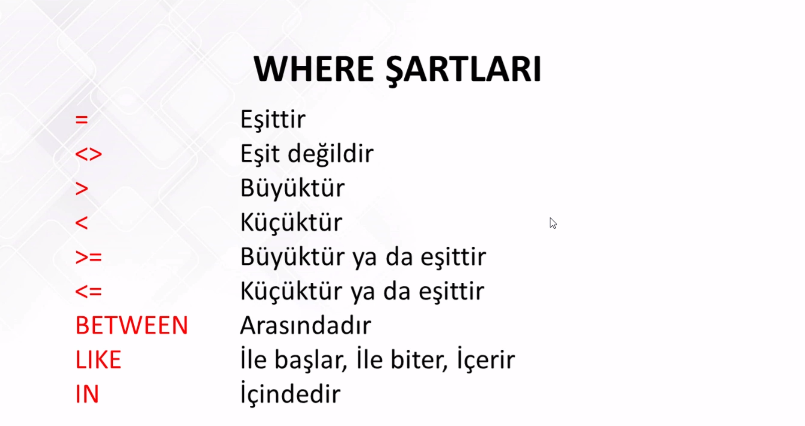


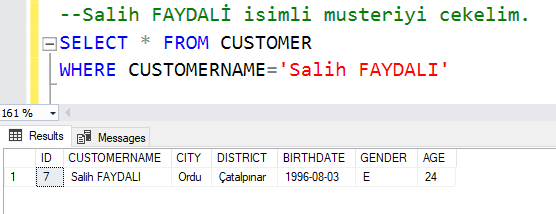
### Where Şartı Kullanımı

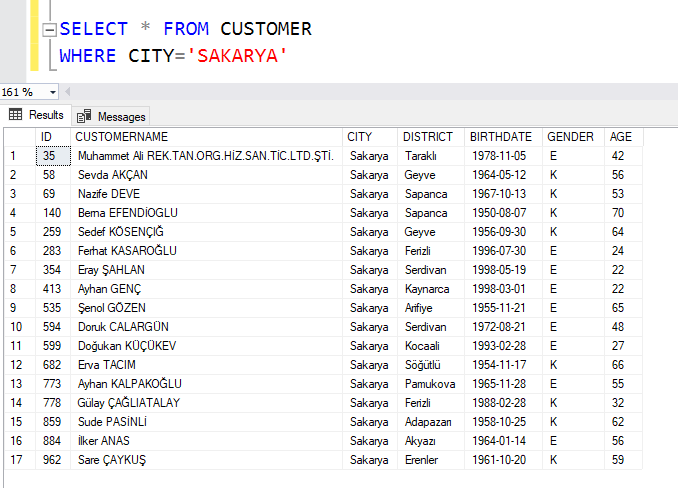
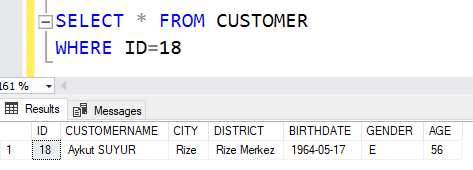
Normalde **SELECT \* FROM CUSTOMER** derken tüm müşterileri çekeriz.   
İsmi *Ömer Çolakoğlu* olan müşteriyi çekmek istediğimizde **WHERE** şartı bize yardımcı olacaktır.

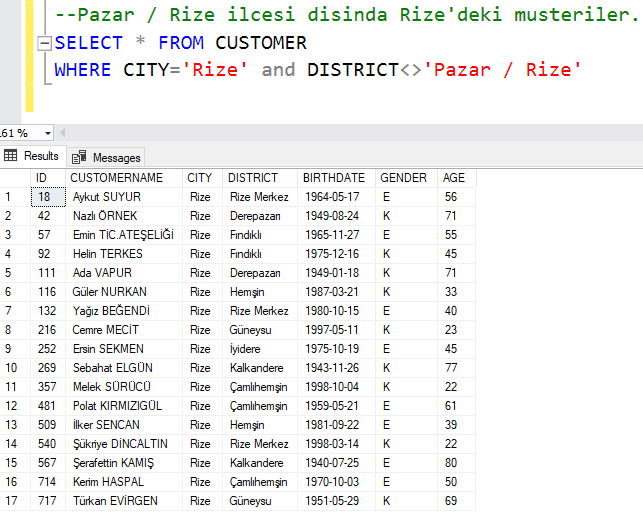
Ya da yaşı 20’den büyük olan müşterileri çekmek istediğimiz zaman yine **WHERE** şartı kullanırız.

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

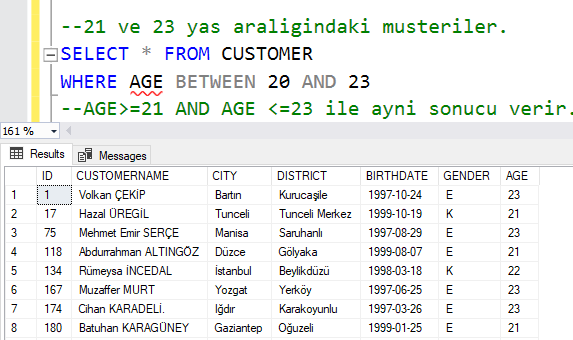


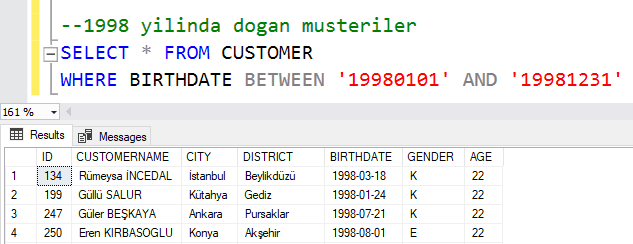




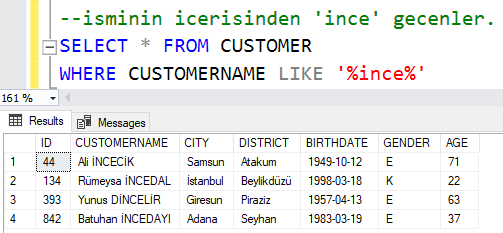
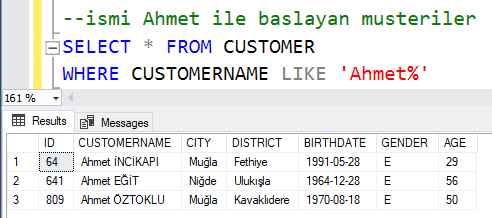
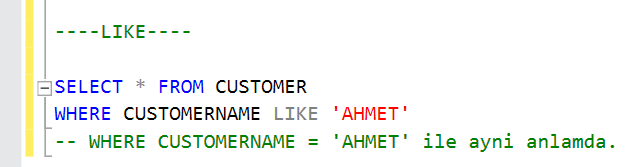


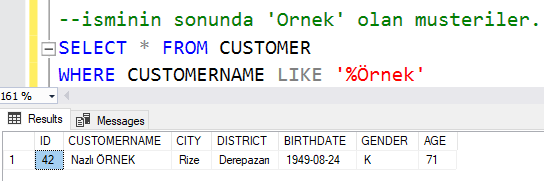
#### BETWEEN



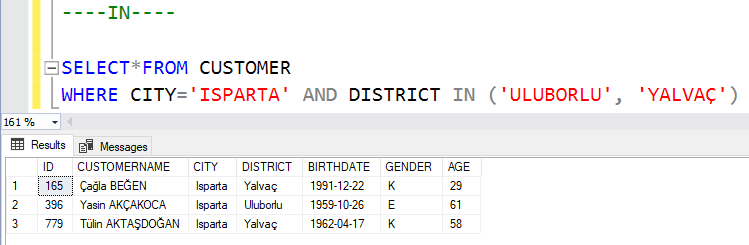


#### LIKE

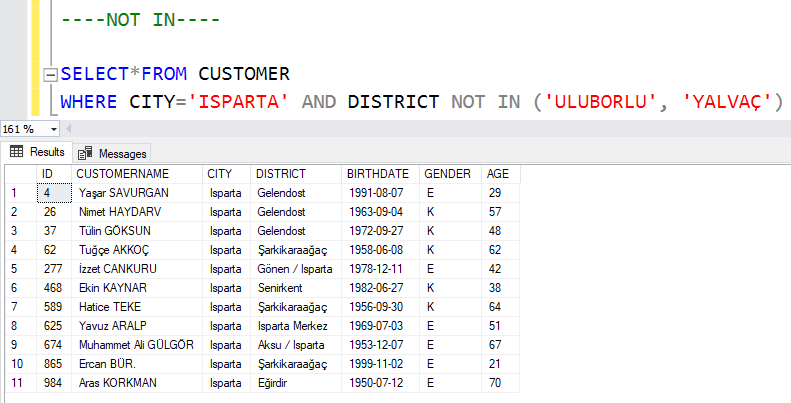




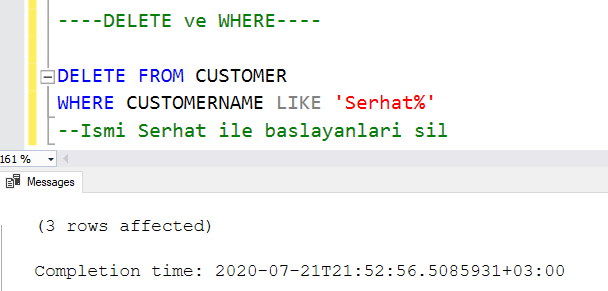
#### IN



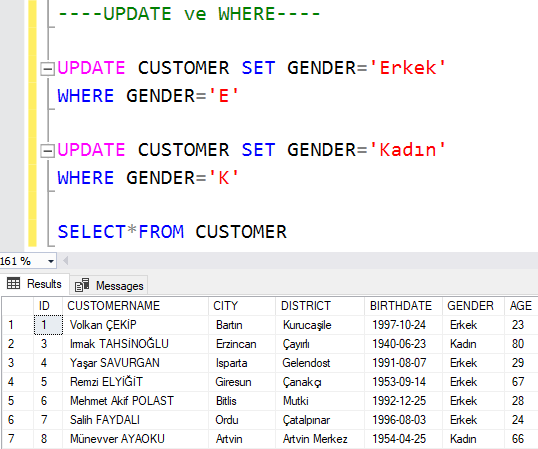
#### NOT IN



#### DELETE Komutu ile WHERE Şartı Kullanımı

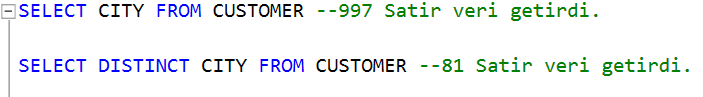


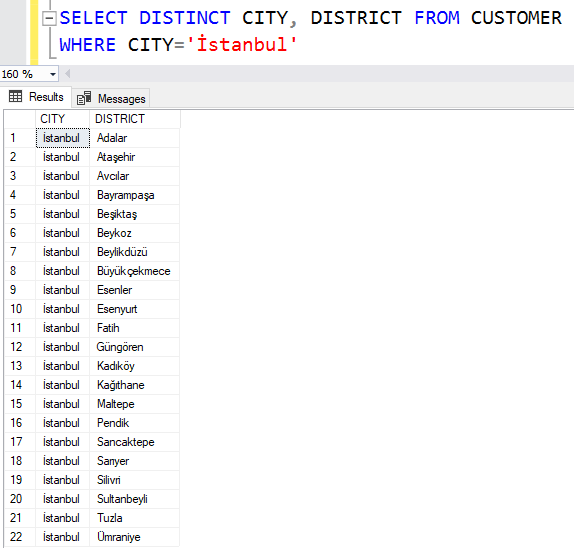
#### UPDATE Komutu ile WHERE Şartı Kullanımı



### Distinct Komutu

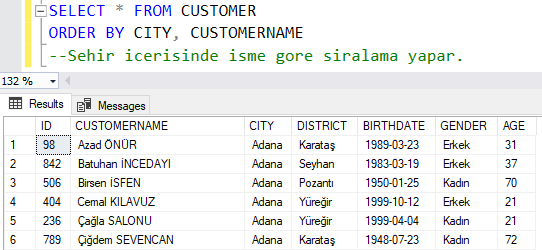
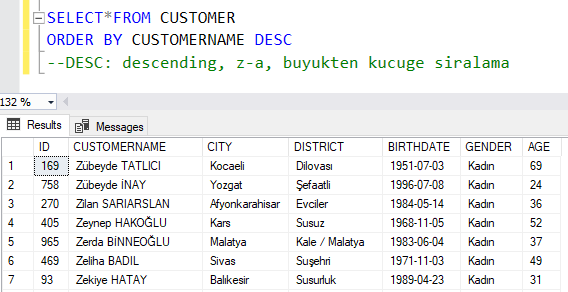
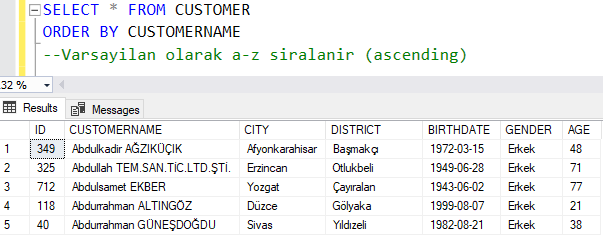
**DISTINCT** komutunu SELECT içerisinde kullanırız, tekrar eden satırlar için tek bir değer döndürür.

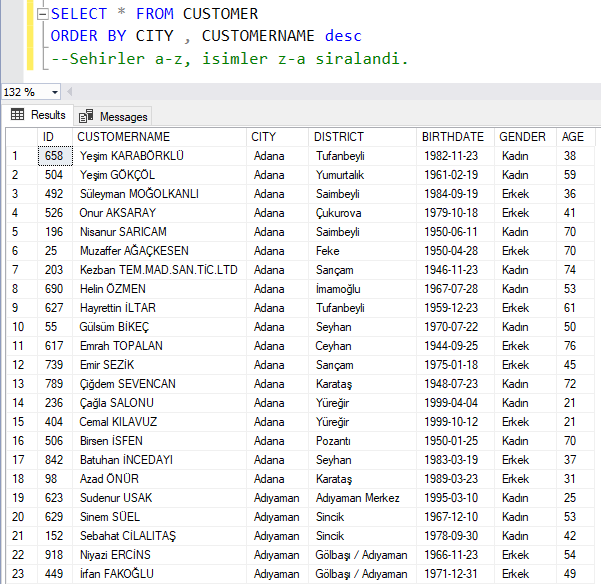


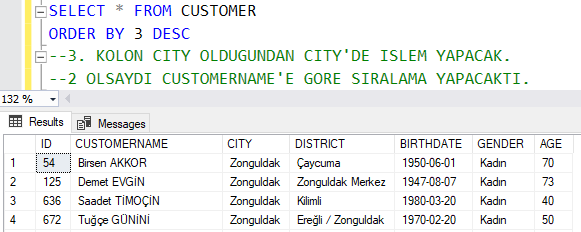


### Order By Komutu

**ORDER BY** bir sıralama komutudur.

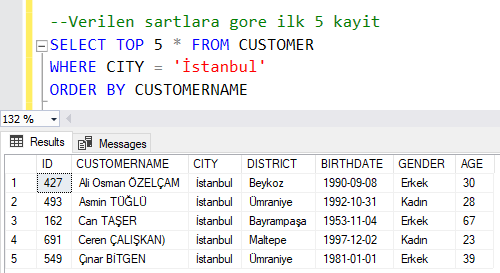


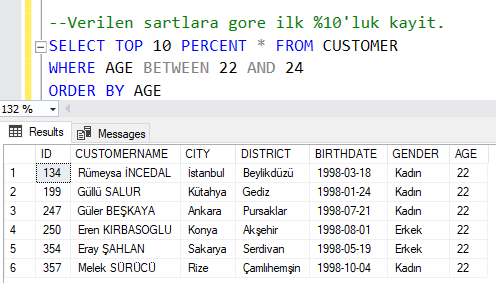




### Top Komutu

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

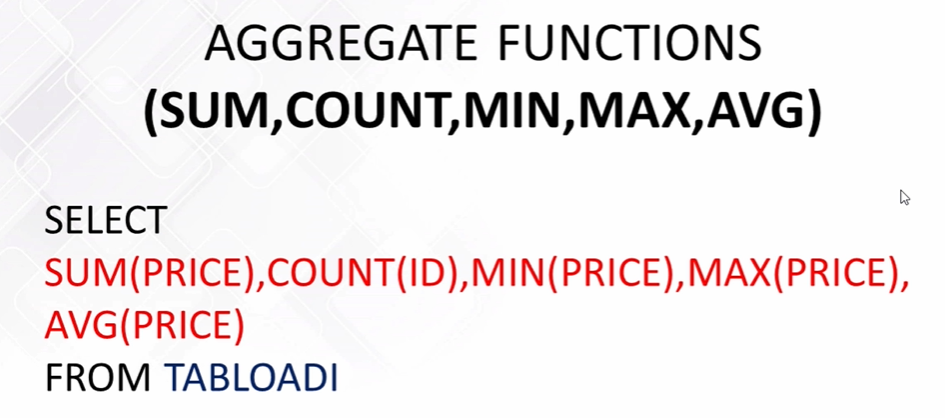


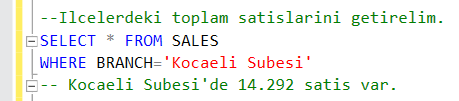


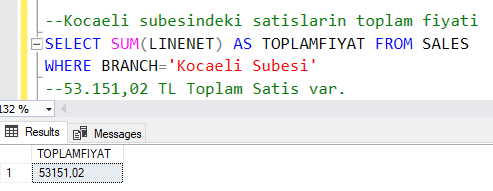
## Örnek Satış Datası

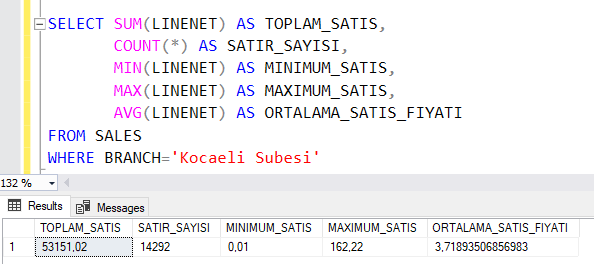
Gerçek bir veri seti üzerinde işlemler yapmak için örnek satış veritabanımızı SQL Server’a ekledik.

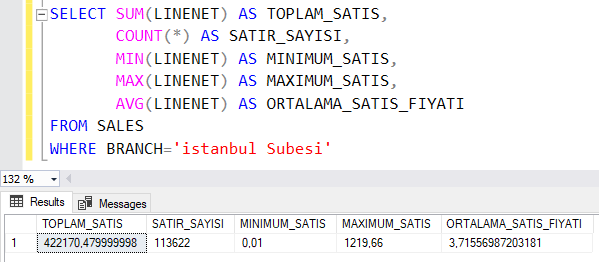
### Aggregate Fonksiyonlar



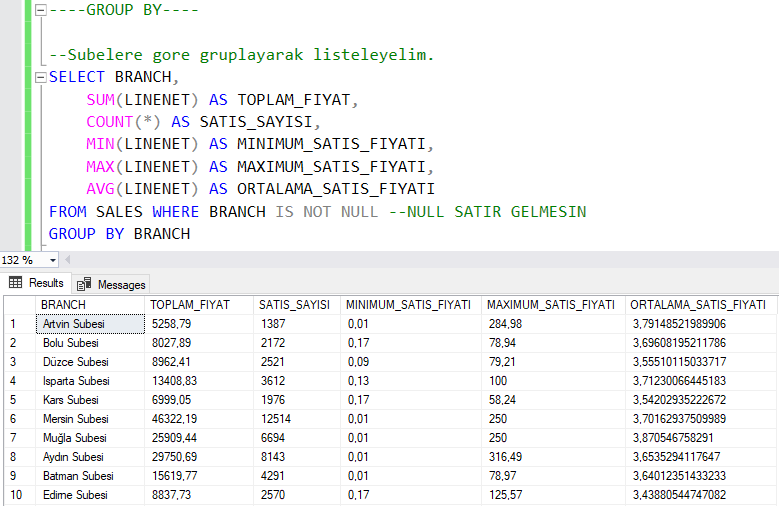






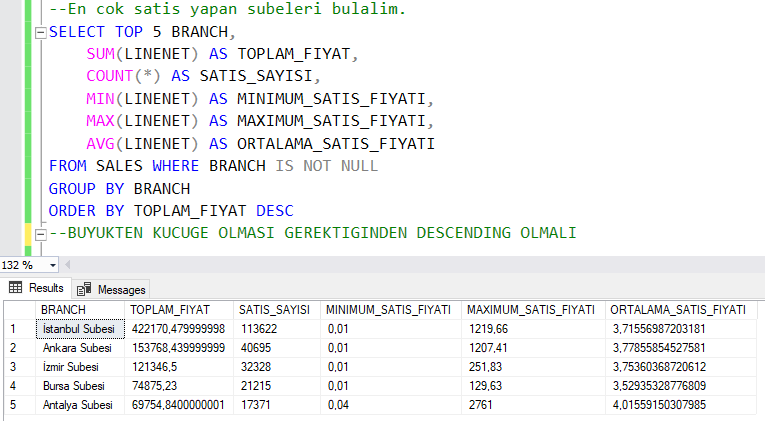


### Group By Kullanımı

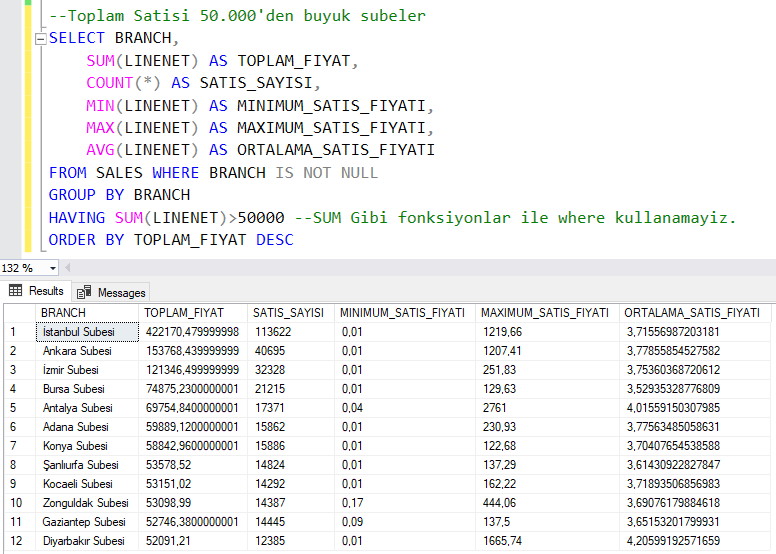


81 adet şube getirdi.

#### En Çok Satış Yapan Şubeler

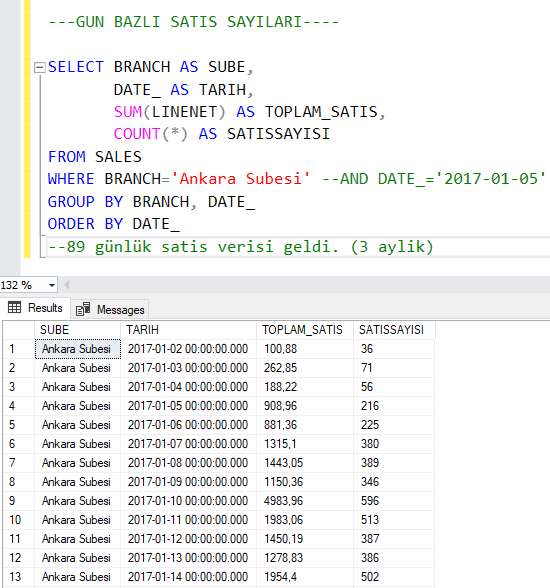


#### Toplam Satış Fiyatı 50.000’den Büyük Olan Şubeler

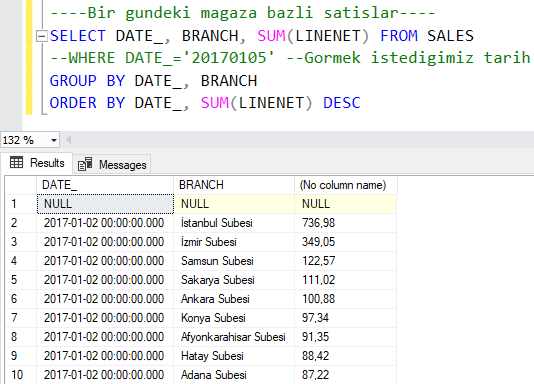
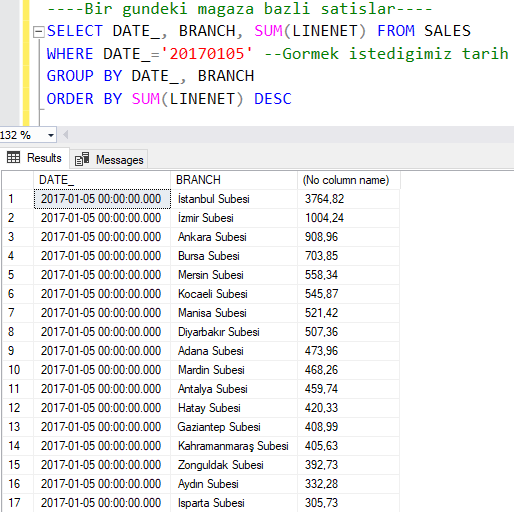


**SUM** Gibi fonksiyonlar ile koşul sağlamak istediğimizde **WHERE** yerine **HAVING** kullanmalıyız.

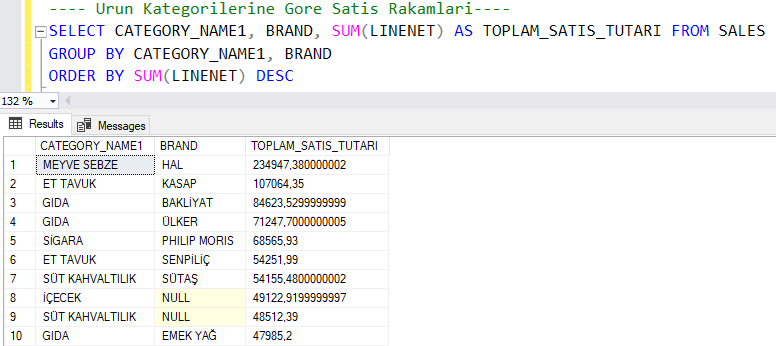
#### Bir Mağazanın Gün Bazlı Satışları

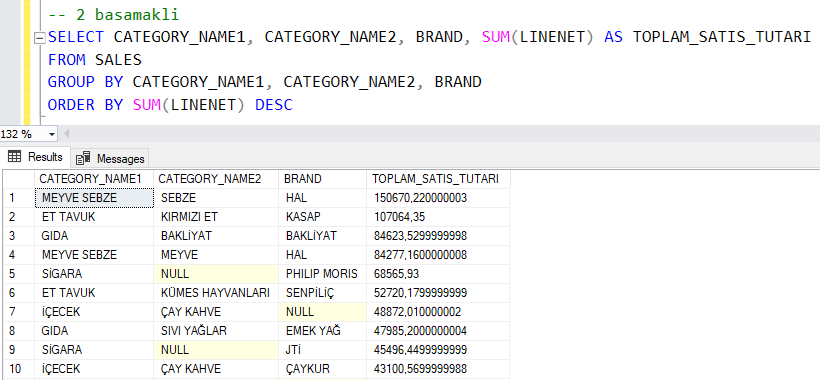


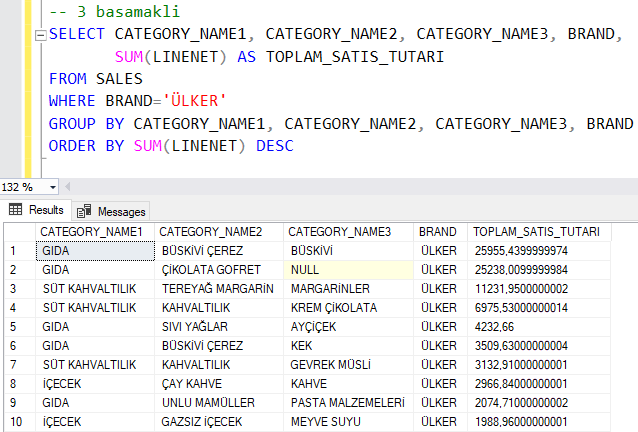
#### Bir Gündeki Mağaza Bazlı Satışlar



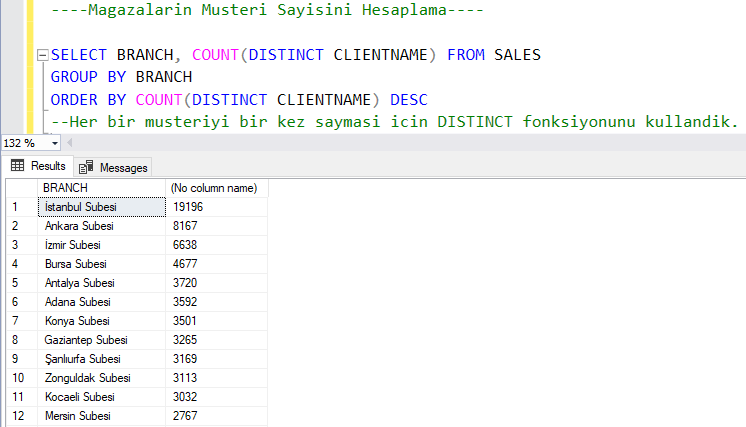
#### Ürün Kategorilerine Göre Toplam Satış Tutarları

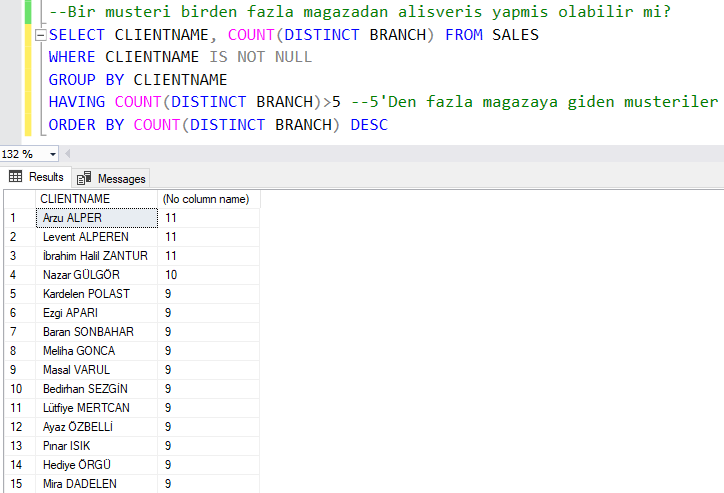


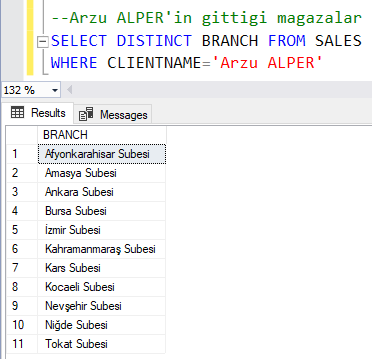




#### Mağazaların Müşteri Sayısını Hesaplama

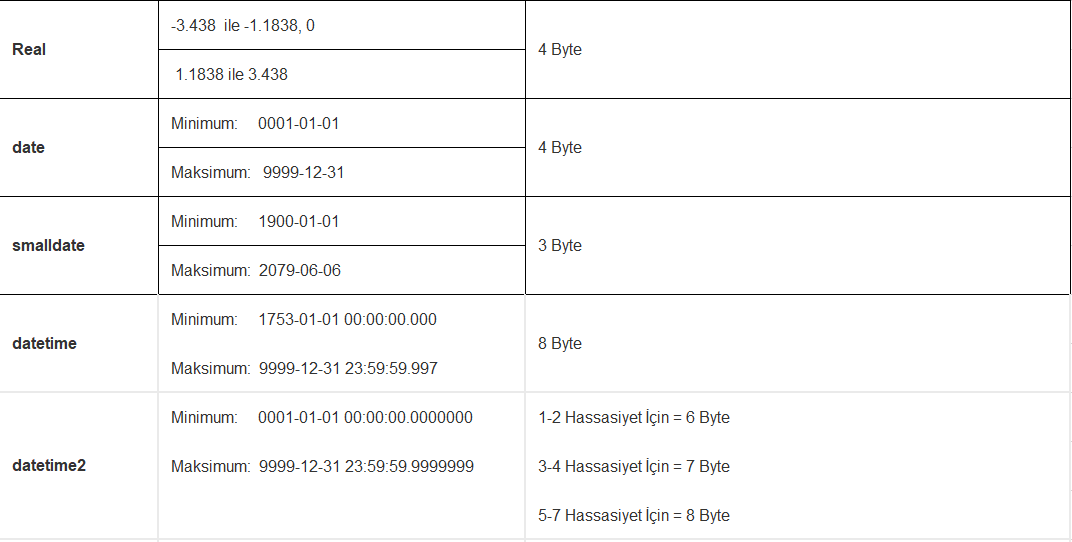
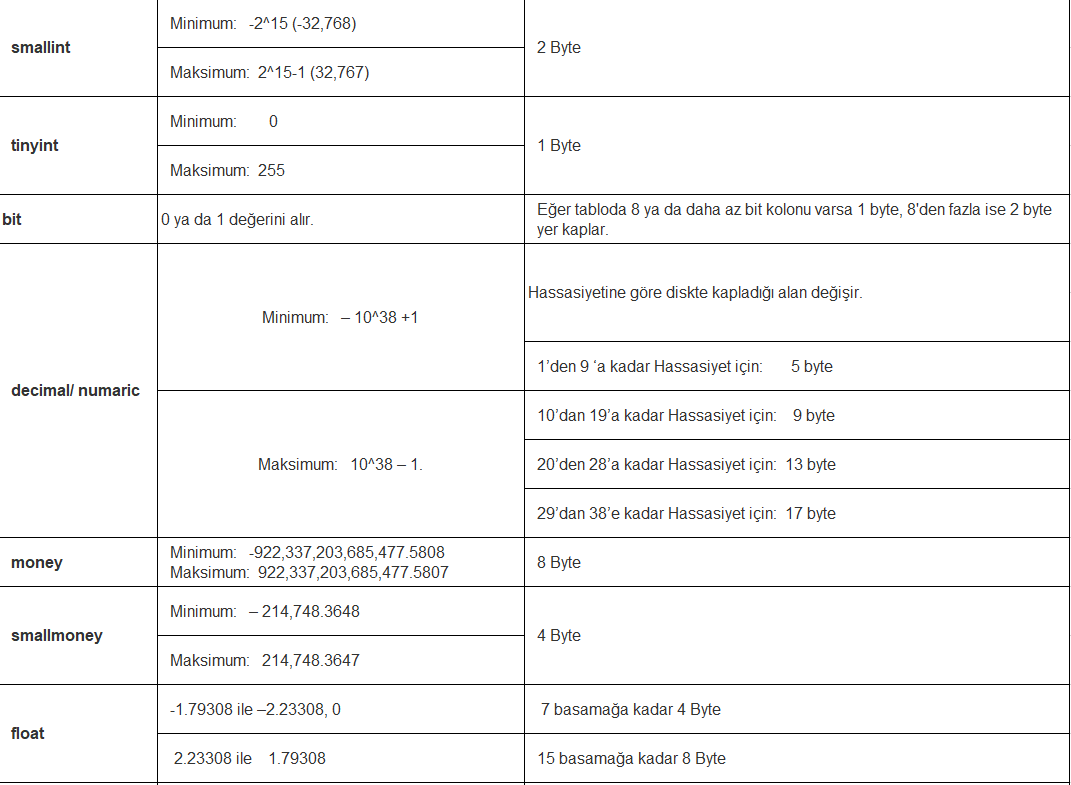


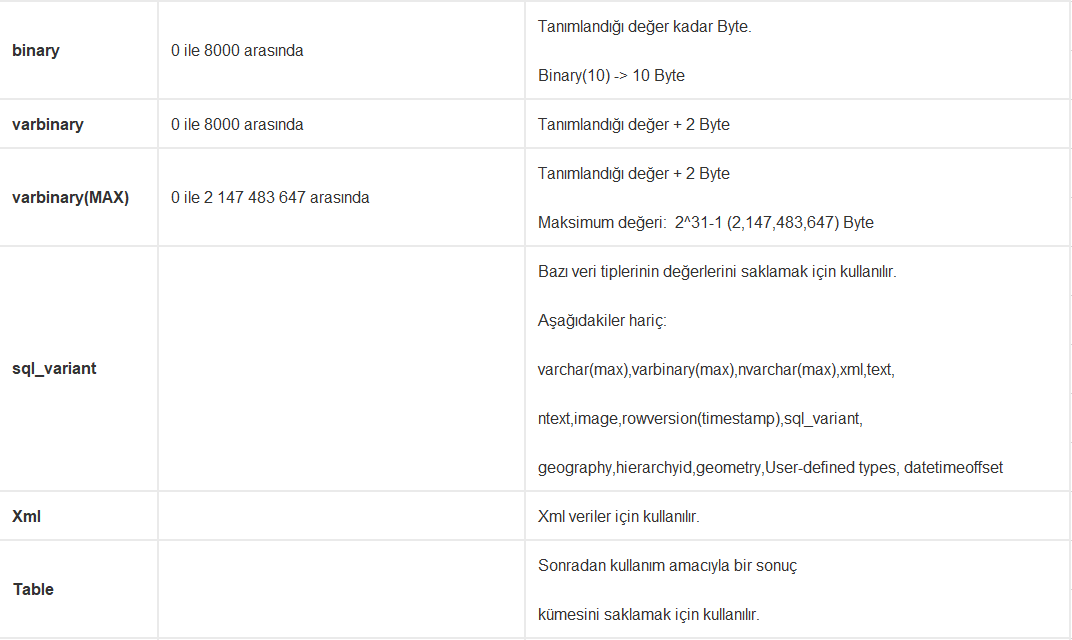
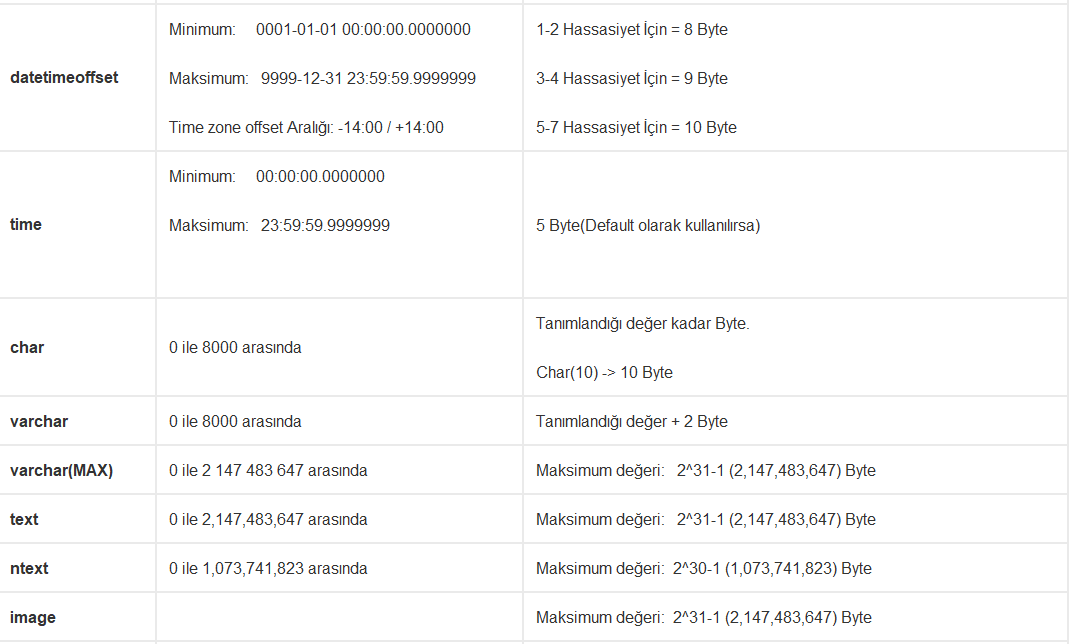


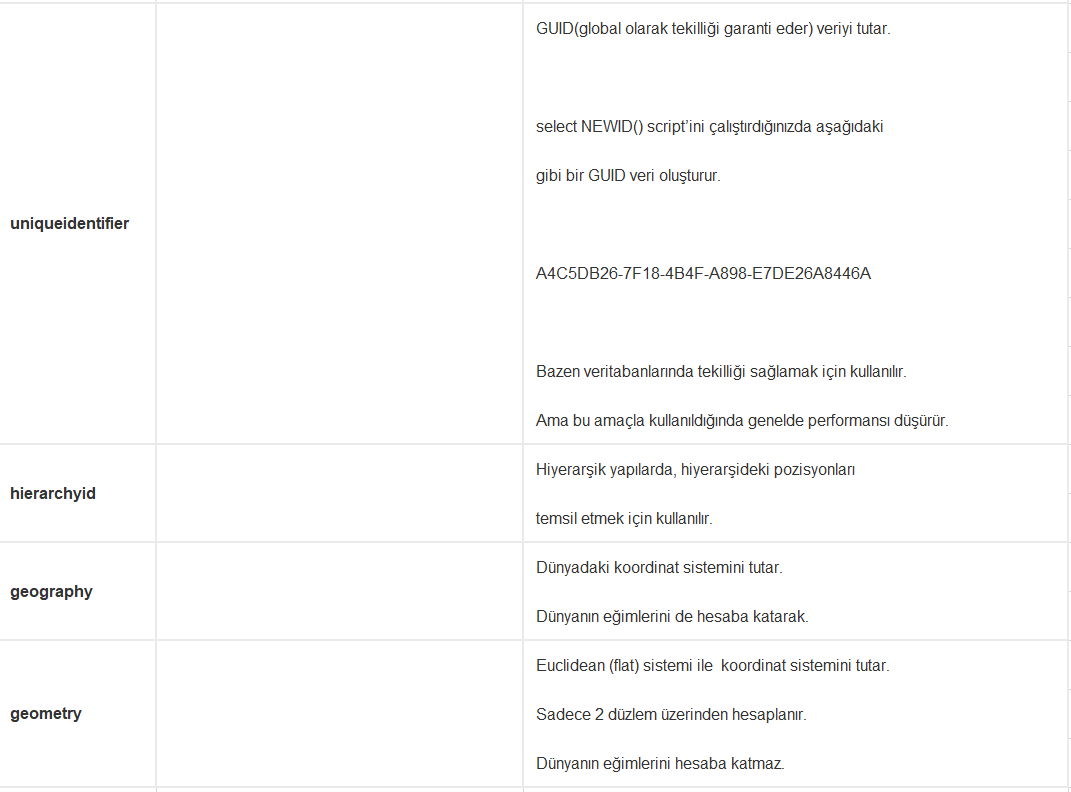


## SQL Server Veri Tipleri

[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 veritabanı nasıl oluşturabileceğimize göz atalım.