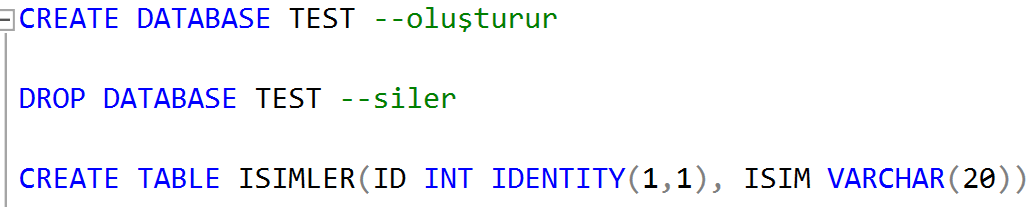
**TSQL Komutları**

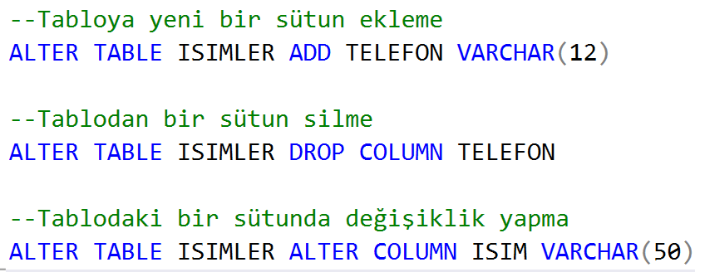
SQL haricinde herhangi bir programlama dilinde yapabileceğin hemen hemen her şeyi yaptığın dil **TSQL**’dir.

1. **DDL (Data Definition Language) Komutları**

* Sadece TSQL’e özel değildir. SQL dilinin de komutlarıdır. Ancak temel için önemlidir.
* Her SQL Server objesi (database, tablo, kullanıcı, view, procedure) **CREATE** komutu ile oluşturulur, **ALTER** komutuyla değiştirilir ve **DROP** komutuyla silinir. Bu komutlara **DDL komutları** denir.



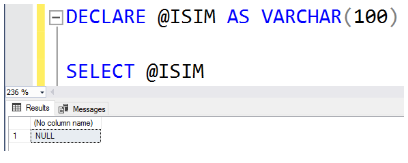
* DDL komutları **KOMUT NESNETÜRÜ NESNEADI** şeklinde tanımlanır.
* Eğer beraberinde parametre tanımlayacaksan parantez içine yazarsın.
  + Tanımlanan nesne tablo ise parantez içine **SÜTUNADI VERİTÜRÜ** şeklinde tablonun kolonları yazılır.
  + Eğer otomatik artan bir kolon olacaksa **IDENTITY (1, 1)** şeklinde tanımlarsın.
* Tablo silineceği zaman da **DROP TABLE ISIMLER** şeklinde silinir.
* Sorgu yapmadan önce ilgili veri tabanını seçmek için **USE VERİTABANI** komutunu kullanırsın.



1. **TSQL’de Değişken Kullanımı**

**Tanımlama**

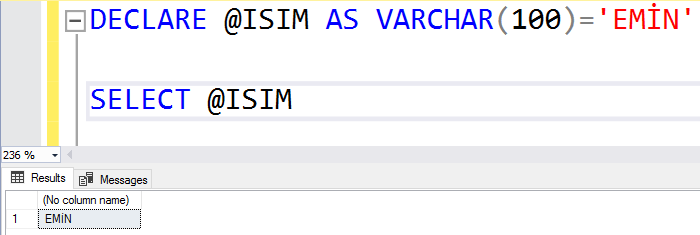
* Neredeyse her programlama dilinde ihtiyaca göre tanımlanabilen değişkenler TSQL’de de tanımlanabilir.



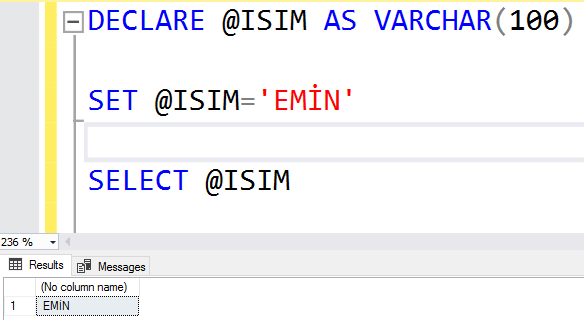
* Değişken tanımlanırken **DECLARE** komutu kullanılır ve değişken olduğunu belirtmek için isminin başına **@** getirilir. Ardından veri türü tanımlanarak değişken oluşturulur.
* SQL’de **SELECT** her türlü değeri çekmek için kullanılır. Burada henüz değişken boş olduğu için *NULL* getirir.

**Değer Atama**

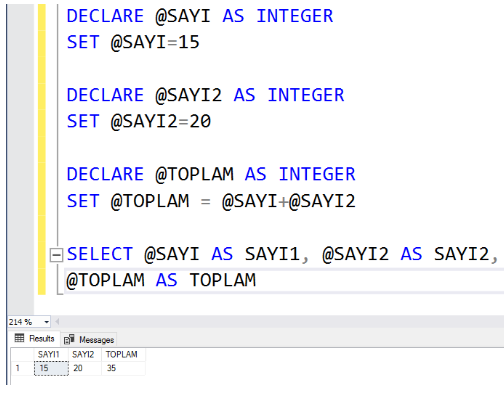
* Tanımlama yaparken değer atama yapılabilir.



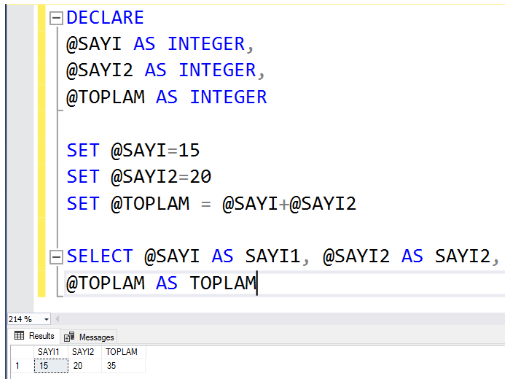
* Değer atama sonradan da yapılabilir.



* Tamsayı değişkenleri **INTEGER** ile tanımlanır.
* İki farklı değişken kullanarak yeni bir değişken üretilebilir.
* Tablodan kolonları çekermiş gibi değişkenler virgül ile ayrılabilir ve alias verilebilir.

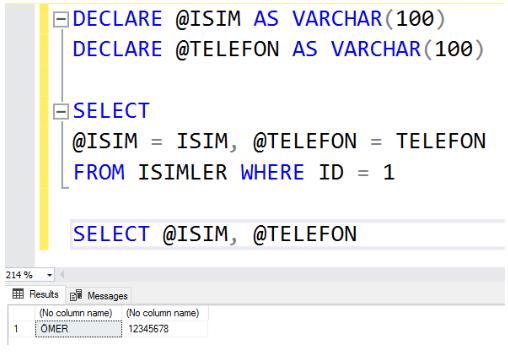


* Değişkenler ayrı ayrı tanımlanabileceği gibi, aralarına virgül konularak aynı **DECLARE** içinde de tanımlanabilir.

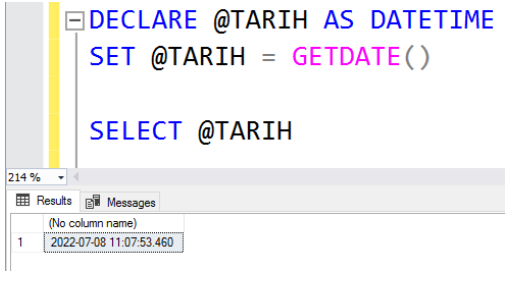


**Tablo Satırını Değişken Olarak Atama**

* Önce tablodaki bir satırı seçersin. Daha sonra tanımladığın değişkenleri o satırın kolonları ile eşlersin.



* Tarih değişkeni oluşturma



**Özet**

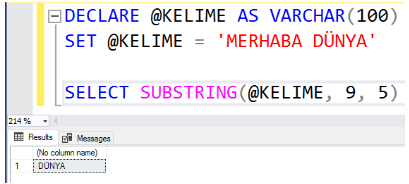
* SQL Server’da değişken **DECLARE** komutu ile tanımlanır.
* Değişkene bir isim verilir ve başına **@** getirilir.
* Daha sonra **AS** diyerek değişkenin türü tanımlanır.
* **SET** ile değişkenin ismi kullanılarak değişkene değer atama yapılır.
* Tanımlanan bu değişken üzerinden matematik, tarih ve string işlemleri türüne göre yapılabilir.
* Veri tabanından dönen değeri değişkene atarken **SELECT** ile değişken adı ve kolon adı eşlenir. Ancak bunun için tablodan tek satır dönmesi gerekir. Birden fazla değer dönerse son değeri alır.

1. **String Functions**

* TSQL’de string işlemlerini gerçekleştiren fonksiyonlar
* [String Functions (Transact-SQL)](https://learn.microsoft.com/en-us/sql/t-sql/functions/string-functions-transact-sql?view=sql-server-ver16)

**SUBSTRING**

* Bir string ifadenin bir kısmını alan fonksiyon
* ***SUBSTRING (expression, start, length)***
  + Expression: string ifade
  + Start: substring başlangıcı
  + Length: uzunluğu



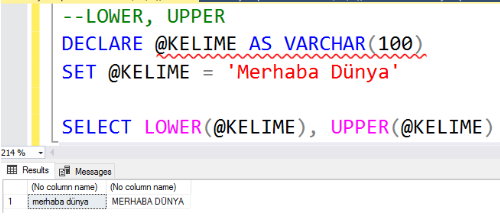
**LOWER, UPPER**

**LOWER**

* String ifadeyi küçük harfli yapan fonksiyon

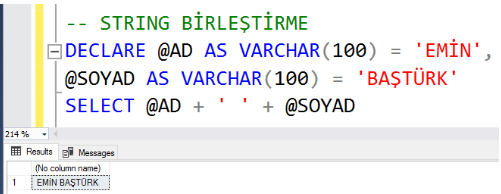
**UPPER**

* String ifadeyi büyük harfli yapan fonksiyon



**String Birleştirme**

* İki string **+** ile yan yana getirilerek birleştirilebilir.



**RIGHT, LEFT**

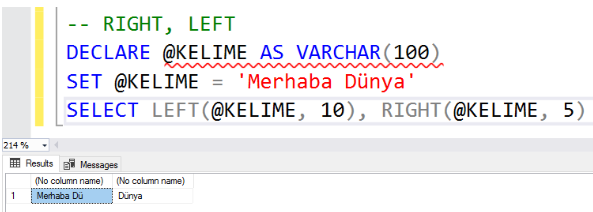
* SUBSTRING’e benzer çalışan fonksiyonlardır.

**RIGHT**

* String’i sağdan karakter sayarak alır.

**LEFT**

* String’i soldan karakter sayarak alır.



**LTRIM, RTRIM, TRIM**

* Boşlukları temizleyen fonksiyonlardır.
* Kirli verileri temizlemek için işe yararlar.

**LTRIM**

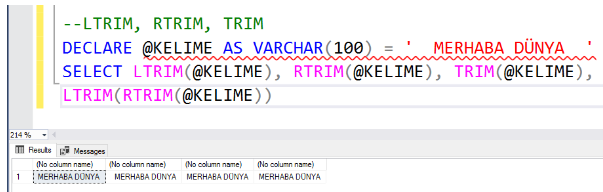
* Soldan boşluk temizler.

**RTRIM**

* Sağdan boşluk temizler.

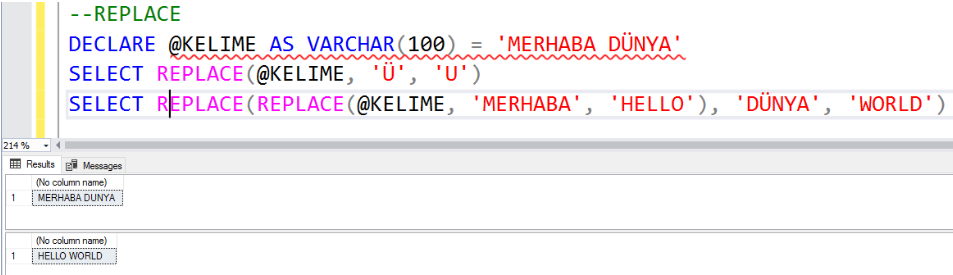
**TRIM**

* ­Soldan ve sağdan boşluk temizler.



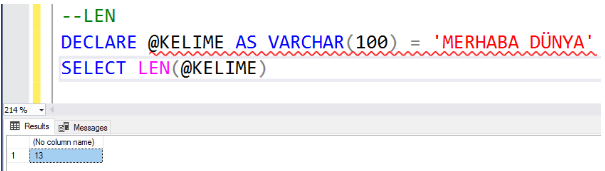
**REPLACE**

* String içindeki bazı karakterleri başka karakterlerle değiştirmek için kullanılan fonksiyon.
* İç içe de kullanılabilir.



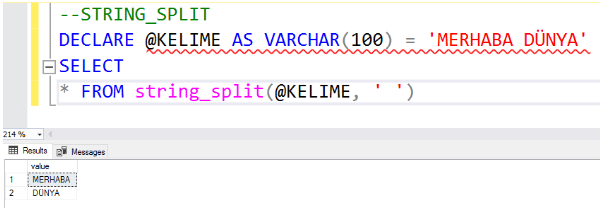
**LEN**

* String ifadenin uzunluğunu veren fonksiyon.



**STRING\_SPLIT**

* String’i belirli bir ayraca göre bölerek ayırdığı stringleri satır satır tablo halinde getirir.
* Diğerlerinden farklı olarak **FROM**’dan sonra kullanılır.

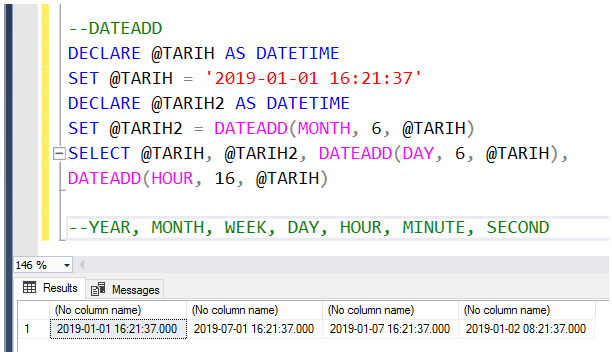


1. **Datetime Functions**

* [Datetime Functions (Transact-SQL)](https://learn.microsoft.com/en-us/sql/t-sql/functions/date-and-time-data-types-and-functions-transact-sql?view=sql-server-ver16)

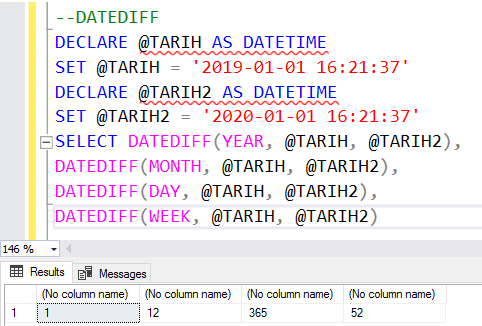
**DATEADD**

* Bir tarihe bir sayı ekleyip başka bir tarih elde etmeye yarayan fonksiyon.
* Fonksiyon içine eklenmek istenen tarih periyodu, değeri ve tarih girilir.



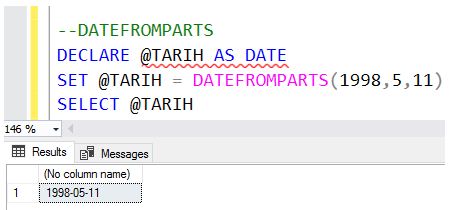
**DATEDIFF**

* İki tarih arasından fark alan fonksiyon.
* Fonksiyon içine alınmak istenen tarih periyodu, başlangıç ve bitiş tarihleri girilir.



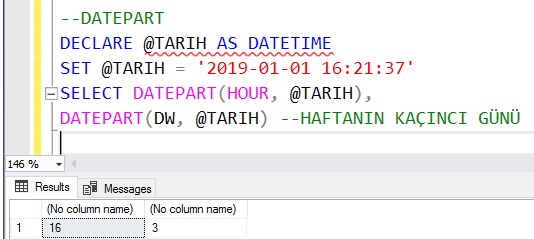
**DATEFROMPARTS**

* Fonksiyon içine yıl, ay ve gün girerek date nesnesi oluşturan fonksiyon.

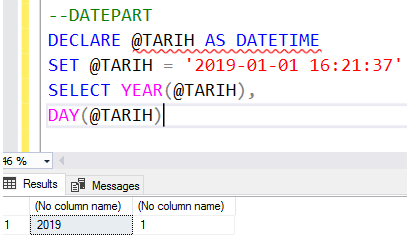


**DATEPART**

* Tarihin istediğimiz kısmını getiren fonksiyon.

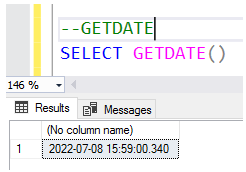


* Sık kullanılan tarih formatları için ayrı ayrı fonksiyonlar da bulunur.

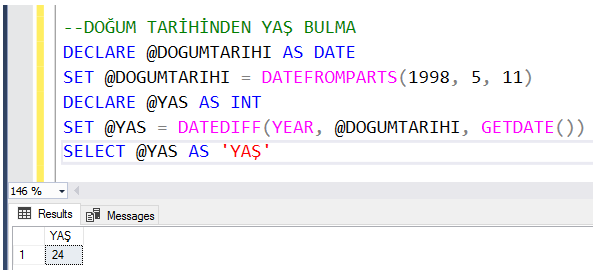


**GETDATE**

* Anın tarih ve zaman bilgisini getiren fonksiyon

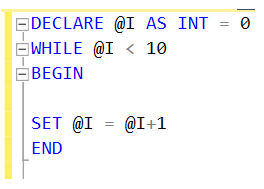


**Doğum tarihinden yaş bulma**

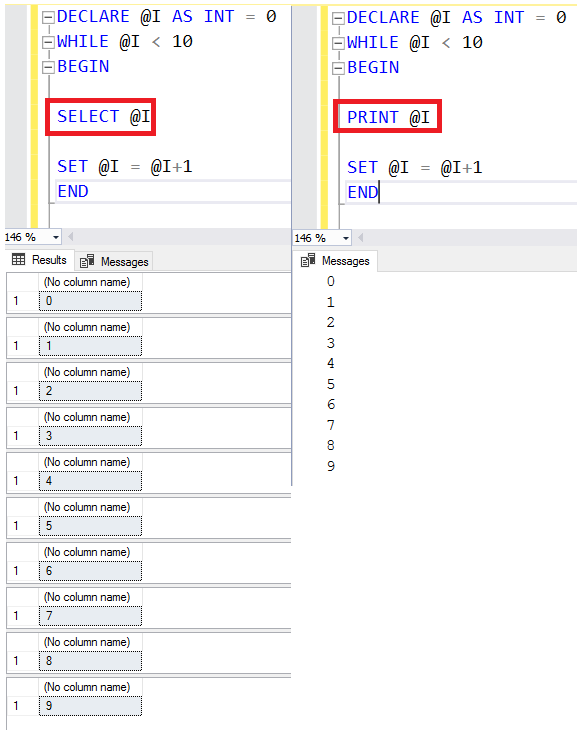
****

1. **Döngüler**

* TSQL’de for döngüsü yoktur ancak **while** döngüleri vardır.
* While döngüsü içerisinde kullanmak üzere değişken tanımlamak gerekir. (**@I**)
* Temel yapı aşağıdaki gibidir;



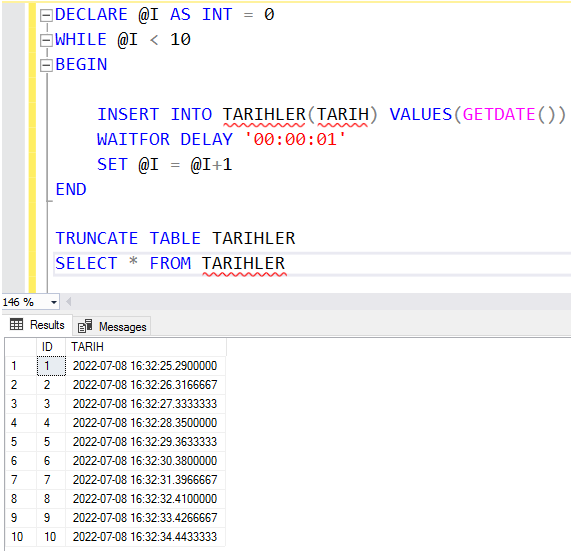
* Döngü değişkeni tanımlanır.
* Koşula bağlı olarak döngü başlar.
* Döngü değişkeni döngünün son satırında güncellenir.
* Koşul devam ettiği sürece işlemler sürer.



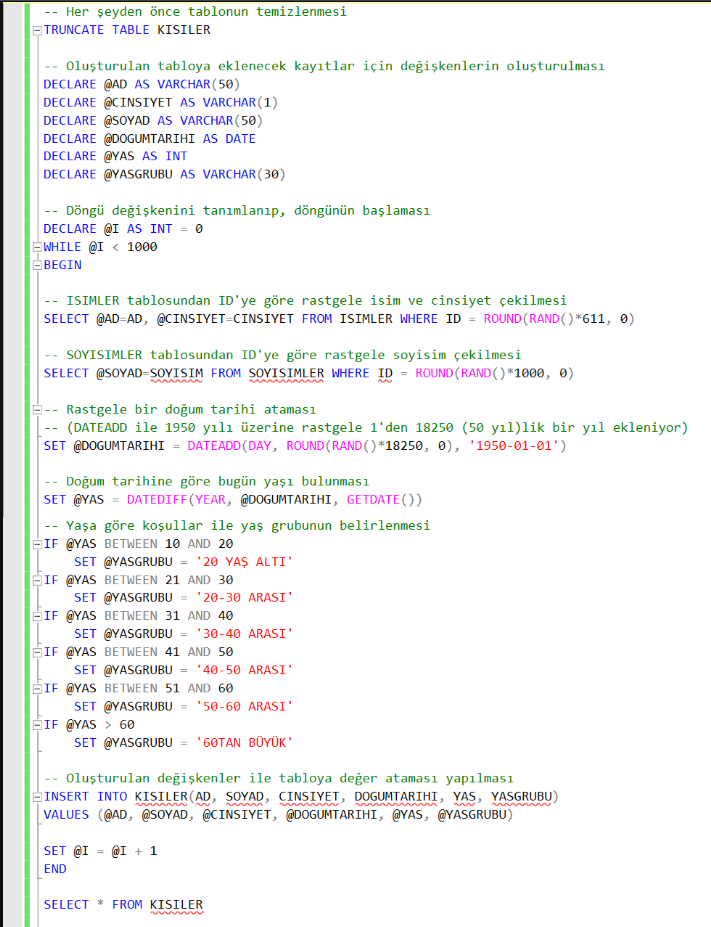
* Döngü değişkenini yazdırmak için **SELECT** ya da **PRINT** komutları kullanılır.

**Döngü ile kayıt girmek**

* 1 saniye gecikme ile bir tabloya anlık tarih bilgisi döngü ile aşağıdaki şekilde girilebilir.



**İsim ve soy isim dataları kullanarak rastgele kayıtlar oluşturma**

****