GO Komutunun Kullanımı

Go komutu bir T-SQL ifadesi değildir, fakat TSQL kodlarında sıkça kullanılır. GO komutu öncesinde yazılmış olan tüm sql ifadelerini veya bir önceki GO ifadesine kadar olan tüm sql ifadelerini tek bir execution plan dahilinde SQL Servera gönderir. SQL Server GO komutunu scripti servera göndermek için bir işaret olarak algılar.

Go komutu T-SQL ifadesi olmadığı için yeni bir satırda tek başına yazılır , aynı satırda sadece açıklama satırı yer alabilir.

GO komutunun kullanımı ile ilgili basit bir örnek

[?](http://www.yazilimmutfagi.com/10166/veritabani/sql-server/go-komutunun-kullanimi-go-komutunu-kullanarak-bir-sorgu-birden-fazla-nasil-calistirilir.aspx)

|  |  |
| --- | --- |
| 1  2  3 | DECLARE @Mesaj VARCHAR(30)  SELECT @Mesaj = 'Yazılım Mutfağı.'  GO -- @Mesaj değişkeni GO komutundan sonra geçerliliğini kaybeder. |

Dikkat edilmesi gereken noktalar:

* GO komutu diğer T-SQL komutları ile aynı asatırda kullanılamaz
* GO komutundan önce tanımlanan bir değişken GO komutundan sonra kullanılamaz. Çünkü daha önceki kod yığınının sona erdiğini belirtmektedir.
* Kodun sonunda GO kullanılmadığı veya tüm sorgu bitmediği sürece komutlar sunucuya gönderilmez.

GO Komutunun Çok Bilinmeyen Kullanım Yöntemi

GO komutu kendisinden önce yazılan sql sorgusunu sonlandırıp servera gönderir dedik. Peki bunu sadece bir kezmi yapabilir. GO komutundan sonra bir sayı belirterek birden fazla aynı sorguyu çalıştırabiliriz.

Aşağıdaki örnekte görebileceğiniz gibi GO komutu ile önceki sorguyu 10 kez çalıştırarak tablomuza 10 adet insert işlemi yapmış olduk.

[?](http://www.yazilimmutfagi.com/10166/veritabani/sql-server/go-komutunun-kullanimi-go-komutunu-kullanarak-bir-sorgu-birden-fazla-nasil-calistirilir.aspx)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | CREATE TABLE tablo (deger BIGINT)  INSERT INTO tablo(deger) SELECT 1  GO    INSERT INTO tablo(deger) SELECT SUM(deger) FROM tablo  GO 10    SELECT \* FROM tablo |

Yukarıdaki sorgu sonucu tablomuzdaki veriler aşağıdaki gibi oldu.

|  |
| --- |
| **deger** |
| 1 |
| 1 |
| 2 |
| 4 |
| 8 |
| 16 |
| 32 |
| 64 |
| 128 |
| 256 |
| 512 |