

EXECUTE READER / EXECUTE NONQUERY / EXECUTE SCALAR

- **Connection:** Çalıştırılacak olan sorgunun hangi bağlantı üzerinde işlem yapacağı belirtilmektedir.
 - **CommandType:** Çalıştırılacak olan SQL cümlesinin tipini belirlemek için kullanılmaktadır.
 - **CommandText:** Çalıştırılacak olan sorgu cümlesi yazılmaktadır.
 - **Parameters:** T-SQL cümlesi ya da Stored Procedure içerisinde kullanılan parametreleri setlemek için kullanılmaktadır.
-
- Eğer sorgunuzdan tek bir değer dönecekse ExecuteScalar ,
 - Eğer SQL sorgusu sonucunda birden fazla satır dönüyorsa, o zaman ExecuteReader,
 - SQL sorgusu sonucunda bir dönüş beklenmiyorsa o zaman ExecuteNonQuery kullanılır.

ExecuteReader

Bir ya da birden fazla satırların sonuç olarak döneceği sorgularda SqlCommand' ın ExecuteReader özelliği kullanılmaktadır. ExecuteReader geriye SqlDataReader tipinde veri döndürmektedir.

SqlDataReader, sadece okunabilir olarak kullanılmaktadır. Satır satır okuma işlemi yapılmaktadır. SqlDataReader kullanımı boyunca veritabanı bağlantısı açık olacaktır. Çünkü SqlDataReader veritabanı ile bağlantılı olarak çalışmaktadır. Okuma işlemi sona erdiğinde SqlDataReader bağlantısının kapatılması gerekmektedir.

```
1 reference
private void button1_Click(object sender, EventArgs e)
{
    SqlConnection conn = new SqlConnection("Server = SEVDE; Database = Northwind; Trusted_Connection = true;");
    try
    {
        if (conn.State == ConnectionState.Closed)
        {
            conn.Open();
            SqlCommand cmd = new SqlCommand("select * from Products", conn);
            SqlDataReader reader = cmd.ExecuteReader();
            while (reader.Read())
            {
                MessageBox.Show(reader.GetValue(0) + " - " + reader.GetValue(1) + " - " + reader.GetValue(2));
            }
            reader.Close();
            cmd.Dispose();
            conn.Close();
        }
        else
        {
            conn.Close();
        }
    }
    catch (Exception)
    {
        MessageBox.Show("Beklenmedik bir hata alındı. B001");
    }
}
```

ExecuteNonQuery

UPDATE, INSERT ve DELETE işlemlerinde kullanılmaktadır. İşlem sonucuna göre geriye int tipinde değer döndürmektedir.

UPDATE, INSERT ve DELETE deyimleri için döndürülen değer, komutun etkilediği satır sayısıdır. Diğer tüm deyim türleri için dönüş değeri -1'dir.

Eklene veya güncelleştirilen bir tabloda tetikleyici olduğunda, dönüş değeri hem ekleme veya güncelleştirme işleminden etkilenen satır sayısını hem de tetikleyiciden veya tetikleyicilerden etkilenen satır sayısını içerir.

```
1 reference
private void button2_Click(object sender, EventArgs e)
{
    SqlConnection conn = new SqlConnection("Server = SEVDE; Database = Northwind; Trusted_Connection = true;");

    using (SqlCommand cmd = new SqlCommand("INSERT Categories(OrderDate, Freight) values(@date, @freight)", conn))
    {
        cmd.Parameters.AddWithValue("@date", textBox1.Text);
        cmd.Parameters.AddWithValue("@freight", textBox2.Text);

        try
        {
            if (conn.State == ConnectionState.Closed)
            {
                conn.Open();
                int etkilenen = cmd.ExecuteNonQuery();

                MessageBox.Show(etkilenen > 0 ? "Kayıt Eklendi" : "Başarısız Kayıt ekleme işlemi!!!!");
            }
            else
            {
                conn.Close();
            }
        }
        catch (Exception)
        {
            MessageBox.Show("Beklenmedik bir hata alındı. B001");
        }
    }
}
```

ExecuteScalar

Tek alanlık bir değeri geri döndürmek için kullanılmaktadır. Object tipinde veri döndürmektedir.

Veritabanından veri çekerken geri dönüş olarak tek satır bekliyorsak, bu metodu kullanıyoruz. Gönderdiğimiz komut ile birden fazla satır gelebilecek olduğunda bile ExecuteScalar kullandığımızda her zaman sonucun ilk satırı elimize ulaşacaktır.

Sorguyu yürütür ve sorgu tarafından döndürülen sonuç kümesindeki ilk satırın ilk sütununu döndürür. Ek sütunlar veya satırlar yok sayılır.

```
1 reference
private void button3_Click(object sender, EventArgs e)
{
    SqlConnection conn = new SqlConnection("Server = SEVDE; Database = Northwind; Trusted_Connection = true;");

    try
    {
        if (conn.State == ConnectionState.Closed)
        {
            conn.Open();
            SqlCommand cmd = new SqlCommand("select Count(*) from Shippers", conn);
            int count = Convert.ToInt32(cmd.ExecuteScalar());
            MessageBox.Show("Satır sayısı : " + count);
        }
        else
        {
            conn.Close();
        }
    }
    catch (Exception)
    {
        MessageBox.Show("Beklenmedik bir hata alındı. B001");
    }
}
```