

# Nikom HMI: SQL DataBase\_Class Eğitim Dokümanı

Bu doküman, C# tabanlı Nikom HMI projelerinde MSSQL veri tabanı işlemlerini kolaylaştıran DatabaseClass kütüphanesinin kullanımını kapsar.

## 1. Giriş ve Kurulum

DatabaseClass, SQL komutlarını bilen geliştiriciler için MSSQL erişimini basitleştiren bir kütüphanedir. İşlemlere başlamadan önce sınıfın örneğini (instance) oluşturmalı ve bağlantı dizesini (connection string) tanımlamalısınız.

**Önemli İpucu:** Bellek yönetimi için nesneyi buton olayları içinde değil, global alanda tanımlamanız önerilir.

```
C#
// Global Tanımlama
NikomHMI.DatabaseClass DB = new
NikomHMI.DatabaseClass ("Server=SERVER_ADRESI;Database=DB_ADI;User
Id=KULLANICI;Password=SIFRE;");
```

---

## 2. Veri Listeleme (SELECT İşlemi)

Veri tabanından veri çekmek ve bir DataGridView üzerinde göstermek için GetTable metodu kullanılır.

### Temel Kullanım:

```
C#
DataTable dt = DB.GetTable("SELECT * FROM tblReport");
dgvReport.DataSource = dt;
```

### Parametrelili ve Filtreli Sorgu (Tarih Aralığı):

SQL'e tarih gönderirken formatın yyyy-MM-dd olması kritiktir.

```
C#
string start = dtpStart.Value.ToString("yyyy-MM-dd 00:00:00");
string end = dtpEnd.Value.ToString("yyyy-MM-dd 23:59:59");

string sorgu = $"SELECT * FROM tblReport WHERE RecordDate >= '{start}' AND
RecordDate <= '{end}'";
dgvReport.DataSource = DB.GetTable(sorgu);
```

---

### 3. Veri Ekleme (INSERT İşlemi)

PLC'den gelen verileri SQL'e kaydetmek için `SendQuery` metodu kullanılır. `string.Format` kullanımı kodun okunabilirliğini artırır.

#### PLC'den Veri Çekme ve Kaydetme:

```
C#
// PLC Bağlantısını al
var connect = hmiButtonInsert.Connect;
object val1 = connect.TagList.GetValue(86); // Örn: Sıcaklık

// Sorguyu Hazırla
string query = string.Format("INSERT INTO tblReport (Value1, Operator)
VALUES ('{0}', '{1}')" , val1, "Murat");

// Sorguyu Gönder
DB.SendQuery(query);
```

---

### 4. Veri Silme (DELETE İşlemi)

Tablodan belirli bir satırı silmek için genellikle `DataGridView` üzerinde seçili olan satırın ID'si kullanılır.

#### Seçili Satırı Silme:

```
C#
if (dgvReport.SelectedRows.Count > 0)
{
    int selectedId =
Convert.ToInt32(dgvReport.SelectedRows[0].Cells[0].Value);
    string deleteQuery = $"DELETE FROM tblReport WHERE ID = {selectedId}";

    DB.SendQuery(deleteQuery);
    // Listeyi yenilemek için tekrar Select çağrılabilir.
}
```

---

## 5. Stored Procedure Çalıştırma (Run Procedure)

Endüstriyel sistemlerde güvenlik veya standart gereği direkt sorgu yerine prosedürler tercih edilebilir.

### Parametrelili Prosedür Kullanımı:

C#

```
// Önceki parametreleri temizle
DB.ClearParameter();

// Parametreleri ekle (@isaretini kullanmanıza gerek yoktur, kütüphane
halleder)
DB.AddParameter("RecordDate", DateTime.Now);
DB.AddParameter("Value1", 25.5);
DB.AddParameter("OperatorName", "Remzi");

// Prosedürü çalıştır
DB.RunProcedure("ST_InsertReport");
```

---

### 💡 Genel İpuçları

- **Hata Denetimi:** Sorgularınızı göndermeden önce `MessageBox.Show(sorgu);` ile kontrol ederek tırnak işaretlerinin ve formatın doğruluğundan emin olun.
  - **Görsel Düzenleme:** `GetTable` işleminden sonra `dgvReport.Columns[0].Visible = false;` komutuyla ID gibi sütunları gizleyebilirsiniz.
  - **PLC Veri Tipleri:** PLC'den `GetValue` ile alınan veriler `object` tipindedir; matematiksel işlem yapacaksanız uygun tipe (`int`, `double` vb.) cast etmeyi unutmayın.
-