

HMI_SQL : Geniş bir SQL Erişim Sisteminin parçasıdır , Olay Komut ları ile geniş bir kullanım alanı sunar. Tek başına kullanılabacağı gibi , HMI_Button , HMI_Triger , DataGridView ile birlikte kullanılabilir.

Tek başına kullanıldığında ; PLC den gelen Yaz komutu ile , tanımlı tabloya (Tablo Adı) Satır ekler. Tablo Sütunları ise **VT Tablo Yönetimi** Özelliğinde Tanımlanır. SQL Parametreleri (Mssql server adresi , hangi Database) Connection String ile tanımlanır. Tablo DataBase içerisinde yoksa bile o tabloyu oluşturur , tablo varsa tabloya , PLC den aldığı verileri satır olarak ekler.

***MSSQL içerisinde mutlaka DataBase oluşturulmalıdır. Tablo oluşturmanıza gerek yok , sistem tabloyu otomatik oluşturur**

Mutlaka MasterPage Formuna Eklenebilir. Diğer Formlarda Kullanılmaz. MasterPage e eklendikten sonra , Diğer formlardan da erişilebilir.

Connect Seçimi : HMI_SQL mutlaka bir HMI_Connect nesnesine bağlanmalıdır.

Yaz : PLC den gelen veri yaz komutunun tag adresi buradan seçilir.

Yazıldı : PLC e yazıldı geri bildirimini yapılır.

Tablo Adı : Arada boşluk bırakmadan tablo adı yazınız. Eğer tablo DataBase içerisinde yoksa otomatik oluşturur.

VT Tablo Yönetimi : Tablo sütunları ile Bu Sütunların PLC de bağlandığı Tag adresleri burada tanımlanır. Bu tanımlamaları yaptıktan sonra yeni sütun tanımlarsanız , o sütun yeni sütun olarak tabloya eklenir.

Connection String : Database bağlantı parametreleri ile DataBase adı burada belirlenir.

Alternate String : İlk Connection String ile bağlantı gerçekleşmez ise bu parametre ile bağlantı sağlanır. Projenin yazıldığı bilgisayar ile çalıştığı bilgisayar farklı ise kullanışlı bir özelliktir. İlkinde çalıştığı bilgisayarın ikincisinde ise yazıldığı bilgisayarın MSSQL Connection Stringi yazılır.

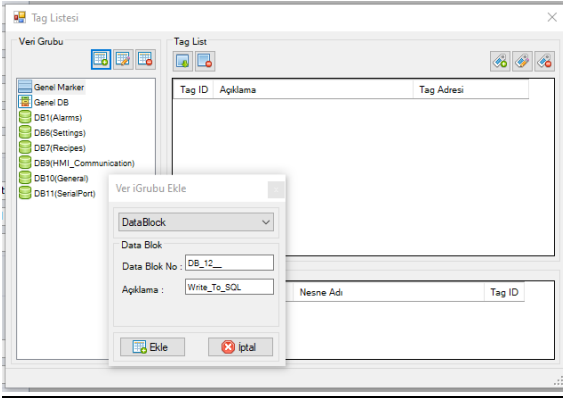
Connection String e kısaca değinilecektir.

DB12(Write_To_SQL)			
	Name	Data type	Offset
1	Static		
2	Write	Bool	0.0
3	Written	Bool	0.1
4	Record_Date	DTL	2.0
5	Flow_Value	Real	14.0
6	Pressure_Value	Real	18.0
7	Operator_ID	DInt	22.0

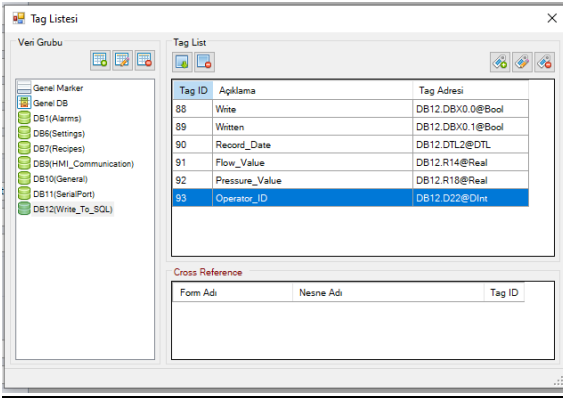
Öncelikle yandaki gibi PLC içerisinde DataBlock tanımlayınız. **DataBlock un Optimized Block Access özelliğini kaldırmayı unutmayınız.**

Bu Örneğimizde DB12(Write_To_SQL) Data Blocku tanımlanmıştır. Write ve Written dışındakiler , Rapor da kayıt altına alınacak Adreslerdir.

Bu adreslerin değerleri , her Write komutu geldiğinde ilgili tabloya satır olarak eklenecektir.



MasterPage üzerindeki HMI_Conect nesnesine DB12(Write_To_SQL) Data Blocku nu Eklıyoruz

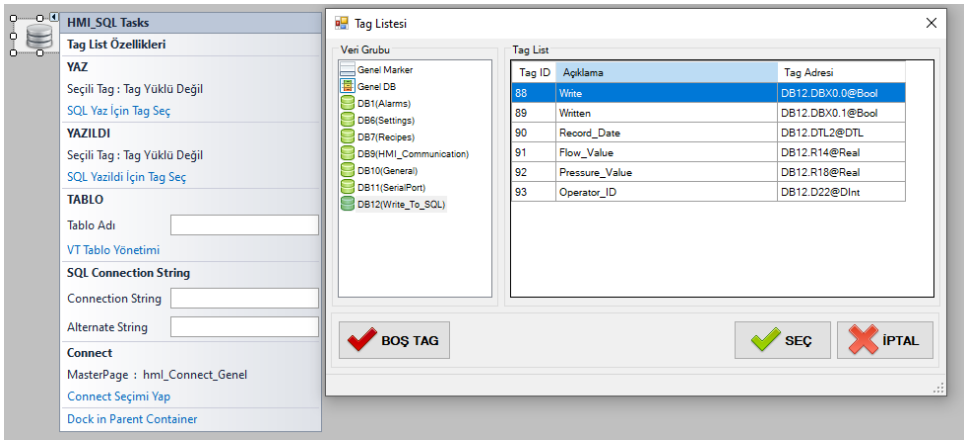


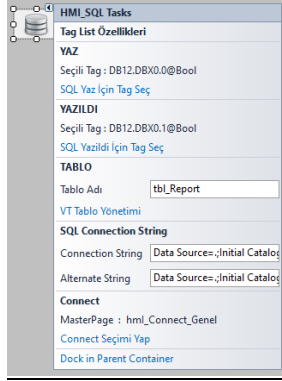
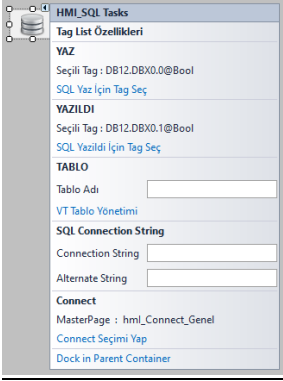
DB12(Write_To_SQL) Data Blockunun Taglarını TagList e ekliyoruz.

İsterseniz tek tek , isterseniz HMI_Connect konusunda anlattığımız gibi topluca ekleyebilirsiniz.

Topluca ekledikten sonra mutlaka kontrol ediniz.

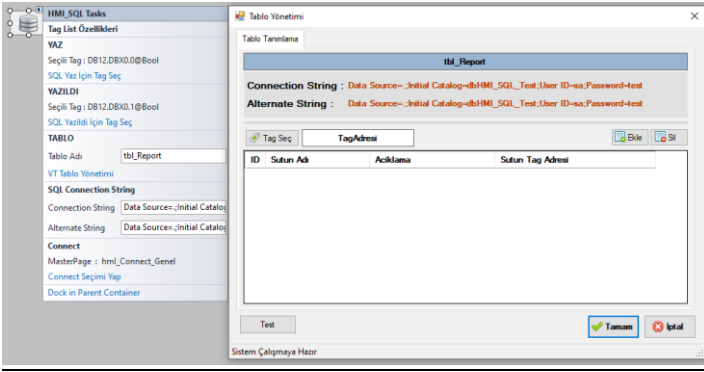
HMI_SQL in komut adreslerini giriyoruz. Yaz komutu için seçtiğimiz tag adresinin değeri true olduğunda (yüksele kenarında) VT Tablo Yönetiminde tanımladığımız adreslerin değerleri , ilgili tabloya (Tablo Adı ile tanımladığımız) satır olarak eklenecektir.





Yaz ve Yazıldı için Tag adresi seçimi yapılmıştır.

Tablo adı ve Connection String parametreleri girilmiştir.



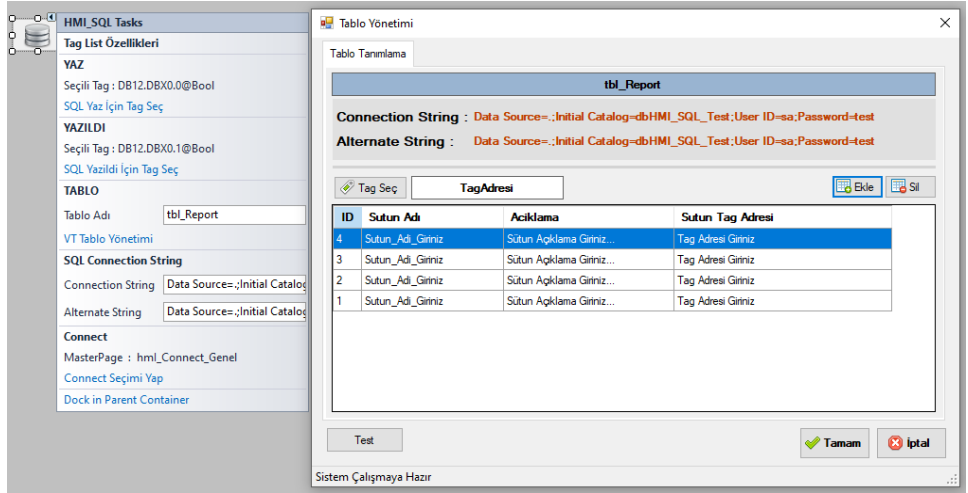
VT Tablo Yönetimi Özeliği İle

Tablonun Sütunları tanımlanır ,

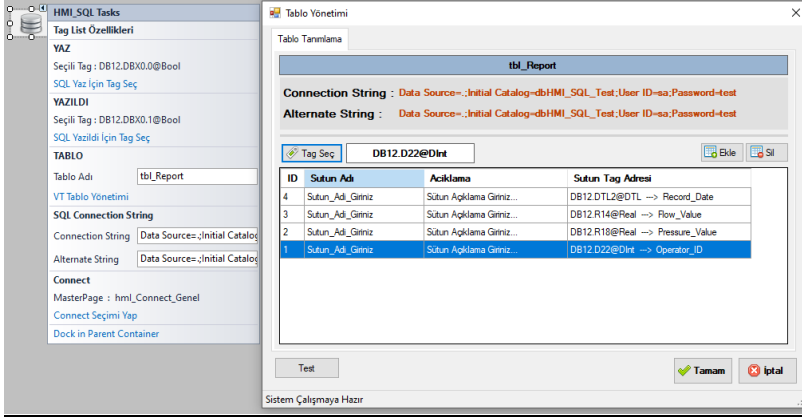
Bu Sütunların değerini alacağı Tag Adresleri Bağlanır

Eğer bu tablonun değerleri bir DataGridView kontrolünde gösterilecekse , Sütun Başlıklarında gösterilmesi için Açıklama Yazılır yani açıklama da yazdığımız DataGridView Kontrolündeki Sütun Başlığıdır.

Aşağıda 4 adet sütun tanımlanmıştır.



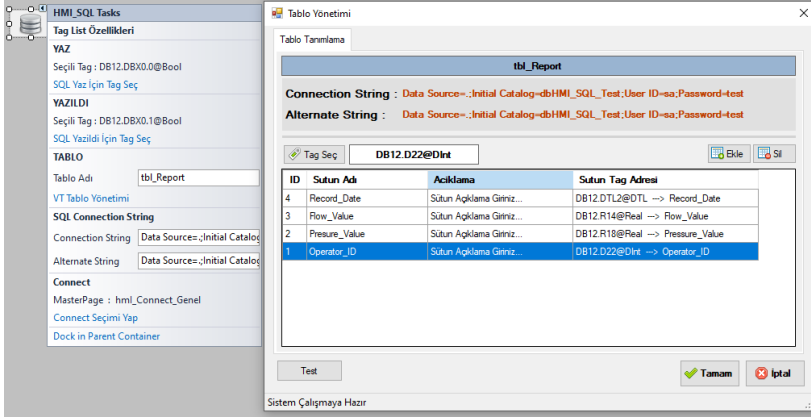
İlk Olarak bu sütunlar için tag adresi ataması yaptık , Sütun Adlarını bu tag adreslerinin adlarına göre yapacağız.



The screenshot shows the HMI_SQL Tasks window on the left and the Tablo Yönetimi window on the right. The HMI_SQL Tasks window has the 'TABLO' section selected, showing the table name 'tbl_Report'. The Tablo Yönetimi window shows the table configuration for 'tbl_Report'. The table has the following columns:

ID	Sütun Adı	Açıklama	Sütun Tag Adresi
4	Sütun_Ad_Giriniz	Sütun Açıklama Giriniz...	DB12.DTL2@DTL -> Record_Date
3	Sütun_Ad_Giriniz	Sütun Açıklama Giriniz...	DB12.R14@Real -> Flow_Value
2	Sütun_Ad_Giriniz	Sütun Açıklama Giriniz...	DB12.R18@Real -> Pressure_Value
1	Sütun_Ad_Giriniz	Sütun Açıklama Giriniz...	DB12.D22@Dirn -> Operator_ID

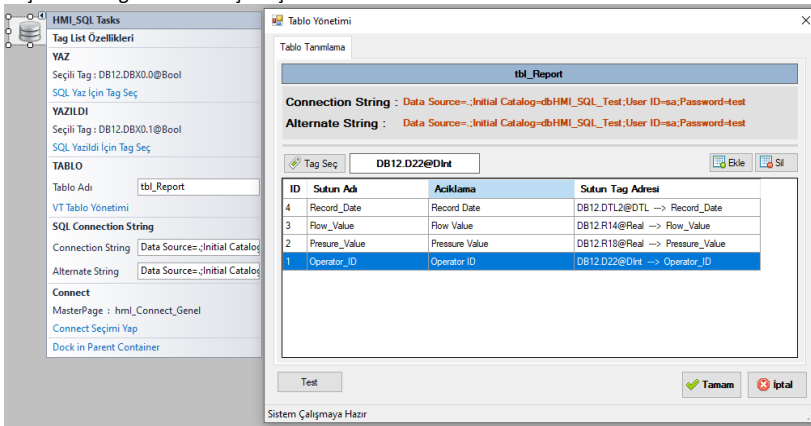
Daha Sonra aşağıdaki gibi sütun adlarını giriyoruz , metin girişi için ilgili hücreye çift tıklayız.



The screenshot shows the HMI_SQL Tasks window on the left and the Tablo Yönetimi window on the right. The HMI_SQL Tasks window has the 'TABLO' section selected, showing the table name 'tbl_Report'. The Tablo Yönetimi window shows the table configuration for 'tbl_Report'. The table has the following columns:

ID	Sütun Adı	Açıklama	Sütun Tag Adresi
4	Record_Date	Sütun Açıklama Giriniz...	DB12.DTL2@DTL -> Record_Date
3	Flow_Value	Sütun Açıklama Giriniz...	DB12.R14@Real -> Flow_Value
2	Pressure_Value	Sütun Açıklama Giriniz...	DB12.R18@Real -> Pressure_Value
1	Operator_ID	Sütun Açıklama Giriniz...	DB12.D22@Dirn -> Operator_ID

Aşağıda ise açıklama girişi yapılmaktadır. Eğer bu tablonun değerleri bir DataGridView kontrolünde gösterilecekse , Sütun Başlıklarında gösterilmesi için Açıklama Yazılır.



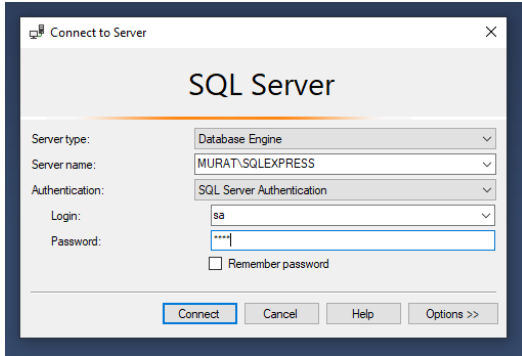
The screenshot shows the HMI_SQL Tasks window on the left and the Tablo Yönetimi window on the right. The HMI_SQL Tasks window has the 'TABLO' section selected, showing the table name 'tbl_Report'. The Tablo Yönetimi window shows the table configuration for 'tbl_Report'. The table has the following columns:

ID	Sütun Adı	Açıklama	Sütun Tag Adresi
4	Record_Date	Record Date	DB12.DTL2@DTL -> Record_Date
3	Flow_Value	Flow Value	DB12.R14@Real -> Flow_Value
2	Pressure_Value	Pressure Value	DB12.R18@Real -> Pressure_Value
1	Operator_ID	Operator ID	DB12.D22@Dirn -> Operator_ID

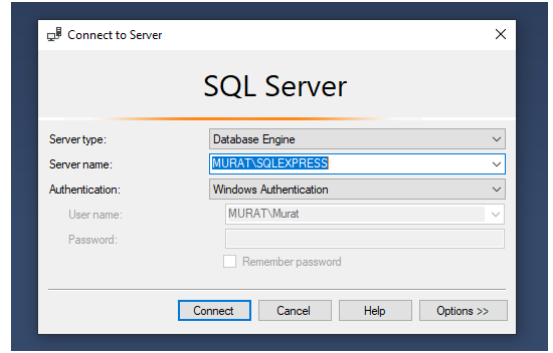
Connection String : Database bağlantı parametreleri ile DataBase adı burada belirlenir. Kullanıcı adı ve şifre ile veri tabanına giriş sağlayabileceğiniz gibi , Windows kullanıcısı olarak da bağlanabilirsiniz.

Data Source veya Server Name i öğrenmek için Management Studio yu açınız oradan Server Name yi öğrenebilirsiniz.

Parametreleri aşağıdaki gibi test edebilirsiniz
sa kulanıcısı ile giriş



Windows kullanıcısı ile kullanıcı adı ve şifresi olmadan giriş



Örnek 1:

Data Source=.;Initial Catalog=db_HMI_SQL;User ID=sa;Password=test;

Data Source=.; Programın çalıştığı Bilgisayarda yüklü MSSQL e bağlanır.

Initial Catalog=db_HMI_SQL; MSSQL içindeki db_HMI_SQL database sine bağlanır

User ID=sa; kullanıcı adı : sa

Password=test şifre ise test dir

Örnek 2 :

Data Source= MURAT\SQLEXPRESS;Initial Catalog=db_HMI_SQL;User ID=sa;Password=test;

Data Source= MURAT\SQLEXPRESS; MURAT Bilgisayarındaki Adı SQLEXPRESS olan MSSQL e bağlanır

Initial Catalog=db_HMI_SQL; MSSQL içindeki db_HMI_SQL database sine bağlanır

User ID=sa; kullanıcı adı : sa

Password=test şifre ise test dir

Örnek 3 :

Data Source= 192.168.0.5\SQLEXPRESS;Initial Catalog=db_HMI_SQL;User ID=sa;Password=test;

Data Source= 192.168.0.5\SQLEXPRESS; 192.168.0.5 Bilgisayarındaki Adı SQLEXPRESS olan MSSQL e bağlanır , IP ile bağlanmada gerekli port ayarlarında yapılmalıdır. Daha fazla bilgi için MSSQL e Uzak Bağlantı İle Bağlanmayı araştırınız.

Initial Catalog=db_HMI_SQL; MSSQL içindeki db_HMI_SQL database sine bağlanır

User ID=sa; kullanıcı adı : sa

Password=test şifre ise test dir

Örnek 4:

Server=localhost;Database=db_HMI_SQL;Integrated Security=True;

Server=localhost;Programın çalıştığı Bilgisayarda yüklü MSSQL e bağlanır.

Database=db_HMI_SQL; MSSQL içindeki db_HMI_SQL database sine bağlanır

Integrated Security=True; Windows Kullanıcısı ile MSSQL e giriş yapılıyor , Bu girişte kullanıcı adı ve şifreye gerek duyulmaz. Bilgisayara Admin yetkisine sahip kullanıcı ile giriş yapılmalıdır.

Aşağıda Connection String değiştirilmiştir ve Windows Kullanıcısı ile MSSQL e giriş yapılacak şekilde ayarlanmıştır.

The screenshot shows two windows. The left window, titled 'HMI_SQL Tasks', displays configuration options for 'tbl_Report' including 'YAZ', 'YAZILDI', and 'TABLO' sections. The 'TABLO' section shows the table name 'tbl_Report' and the 'SQL Connection String' set to 'Server=localhost;Database=...'. The right window, titled 'Tablo Yönetimi', shows the 'tbl_Report' table structure with columns: Record_Date, Flow_Value, Pressure_Value, and Operator_ID. The 'Connection String' and 'Alternate String' are both set to 'Server=localhost;Database=db_HMI_SQL;Integrated Security=True;'. A 'Test' button is visible at the bottom of the right window.

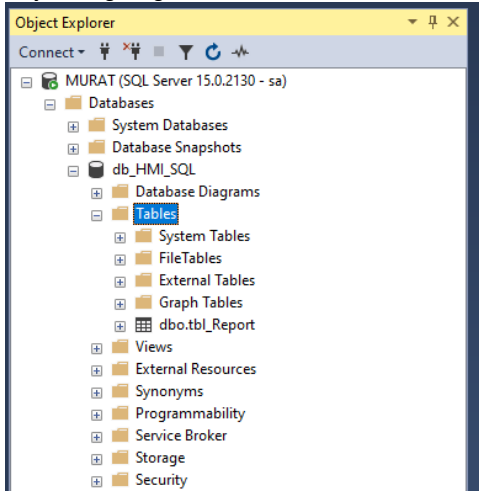
The screenshot shows the 'Object Explorer' window in SQL Server. The 'Connect' dropdown is set to 'MURAT (SQL Server 15.0.2130 - sa)'. The tree view shows the following structure:

- Object Explorer
- Connect - MURAT (SQL Server 15.0.2130 - sa)
- Databases
 - System Databases
 - Database Snapshots
 - db_HMI_SQL
 - Database Diagrams
 - Tables
 - Views
 - External Resources
 - Synonyms
 - Programmability
 - Service Broker
 - Storage
 - Security

Yandaki gibi , mutlaka MSSQL içerisinde db_HMI_SQL adında bir DataBase tanımlayınız.

Programı çalıştırdığınızda , parametreler doğru girilmişse tablo ve sütunları , Tag Adreslerinin tiplerine göre tanımlanır.

Programı çalıştırdığınızda tablonun otomatik olarak oluşturulduğunu göreceksiniz.



Tablo sütunları da Tag Adreslerinin tipine göre otomatik olarak oluşturulur.

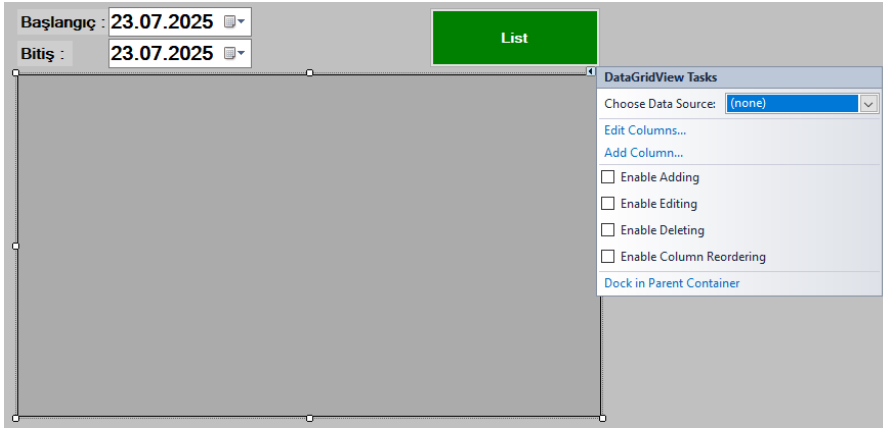
Column Name	Data Type	Allow Nulls
ID	bigint	<input type="checkbox"/>
Operator_ID	bigint	<input type="checkbox"/>
Pressure_Value	float	<input type="checkbox"/>
Flow_Value	float	<input type="checkbox"/>
Record_Date	datetime	<input type="checkbox"/>
		<input type="checkbox"/>

Event Olay Yönetimi İle SQL İşlemleri : HMI_Button ve HMI_Trigger daki Event Olay Yönetimi ile , kod gerektirmeden SQL sorgu yollayabilir , DataGridView de görüntülenmesini sağlayabilirsiniz. NikCom_HMI bunun için size geniş bir kütüphane sunmaktadır.

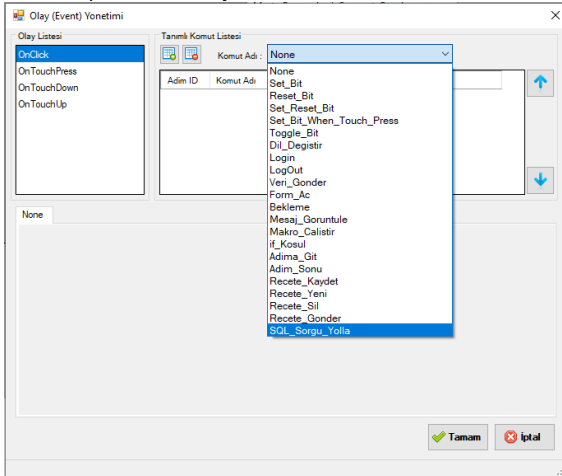
Bu işlemler için MasterPage da tanımlanmış bir HMI_SQL kontrolüne ihtiyacınız vardır. Bu nesne üzerinden Veri Tabanına bağlanılacaktır.

Aşağıdaki gibi Form üzerine kontrolleri yerleştiriniz.

- 2 Adet HMI_Label
- 2 Adet DateTimePicker , dateTimePicker_Baslangic ve dateTimePicker_Bitis
- 1 Adet HMI_Button , Text Düzenle Özelliğine “List” yazınız.
- 1 Adet DataGridView , ekranda görüldüğü gibi Enable Adding , Enable Editing , Enable Deleting özelliklerindeki seçimi kaldırınız



HMI_Buttonu seçiniz ve üst sağdaki kulakçığı tıklayınız , açılan pencereden Event Olay Yönetimini Seçiniz.



Gelen ekrandan (yandaki ekran) komut listesinden SQL_Sorgu_Yolla komutunu çalıştırınız.

Aşağıdaki Ekranı açıklayalım.

Sorgu Tipi ile hangi sorguyu yollayacağınızı belirleyebilirsiniz.

- Select Sorgusu
- Ekleme Sorgusu
- Silme Sorgusu
- Genel Sorgu : Bu sorgu tipi ile kendi kompleks sorgularınızı rahatlıkla Mssql e yollayabilirsiniz.

DataGrid Seç : Sorgunuz eğer bir Select sorgusu ise bu sorgu sonucunun görüntüleneceği bir DataGridView kontrolü mutlaka olmalıdır.

SQL Seçimi : Bu komut mutlaka bir HMI_SQL kontrolüne ihtiyaç duyar. Bağlantı parametrelerini bu kontrolden alır , Genel Sorgu dışındaki sorgularda işlem yapacağı tablo ve sütun bilgilerini bu kontrolden alır.

Genel Sorgu HMI_SQL Kontrolünün bağlantı parametrelerini kullanır ama HMI_SQL in tablosuna erişmek zorunda değildir.

Editör Ekranı : Kullandığınız sorgunu ilgili editörleri bu ekranda görüntülenir.

Parametre Bilgileri : Sorgularda kullanmak için 8 adet parametre kullanabilirsiniz , bu parametreler 0 ile 7 arasında numarandırılır. Parametre veri kaynağı Tag adresi olabileceği gibi Form üzerindeki tanımlı bir kontrolde olabilir (DateTimePicker , TextBox gibi)

Olay (Event) Yonetimi

Olay Listesi

- OnClick
- OnTouchPress
- OnTouchDown
- OnTouchUp

Tanımlı Komut Listesi

Komut Adı : SQL_Sorgu_Yolla

Adım ID	Komut Adı	Açıklama
1	SQL Sorgu Yolla	SQL_Sorgu : Select_Sorgusu

SQL Sorgu Yolla

Sorgu Tipi Seçiniz: Select_Sorgusu

DataGrid Seç

DatGrid Seçiniz...

SQL Seçiniz : Seç SQL Seçiniz

Select Editör

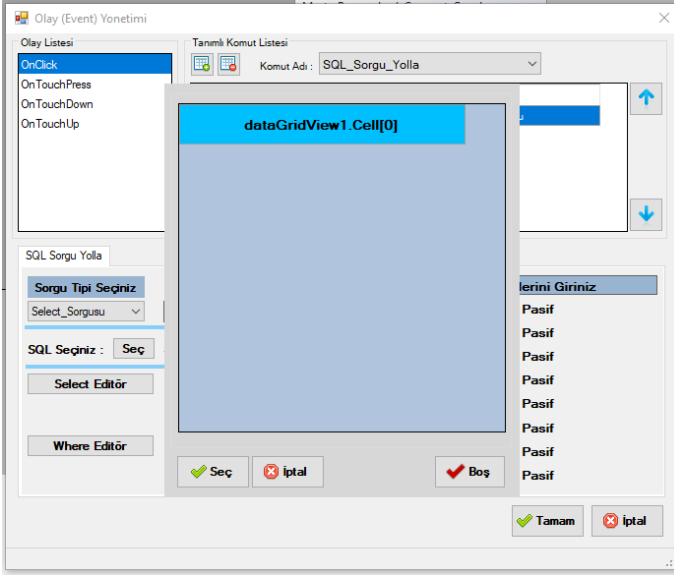
Where Editör

Parametre Bilgilerini Giriniz

Parametre	Durum	Seçim	Durum
{0}	Pasif	Seç	Pasif
{1}	Pasif	Seç	Pasif
{2}	Pasif	Seç	Pasif
{3}	Pasif	Seç	Pasif
{4}	Pasif	Seç	Pasif
{5}	Pasif	Seç	Pasif
{6}	Pasif	Seç	Pasif
{7}	Pasif	Seç	Pasif

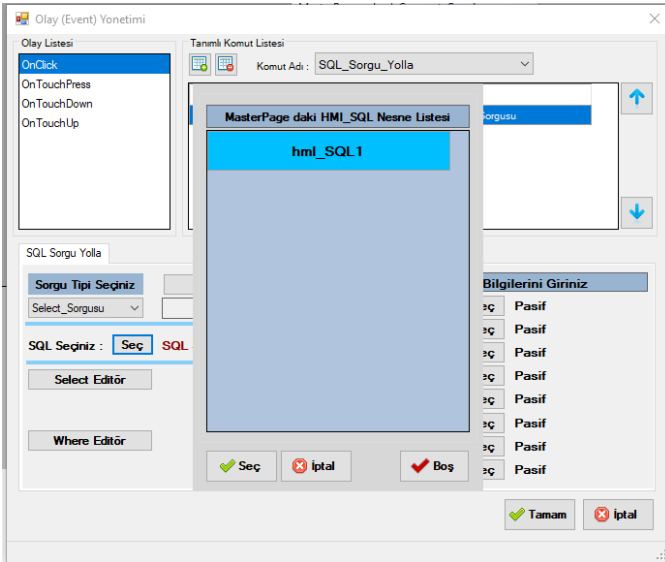
Tamam İptal

Select Sorgusu ile Başlayalım :



Öncelikle Select sorgumuz için bir DataGridView kontrolü tanımlayalım.

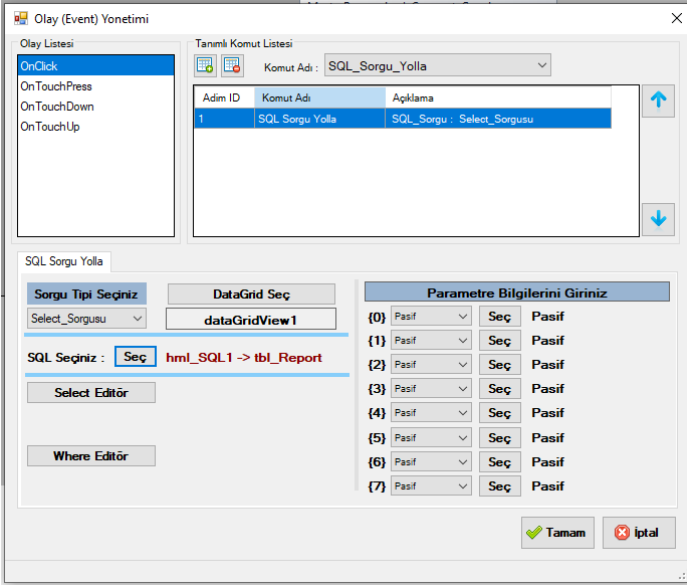
Form üzerine yerleştirdiğimiz DataGridView kontrolünü Seçiyoruz.



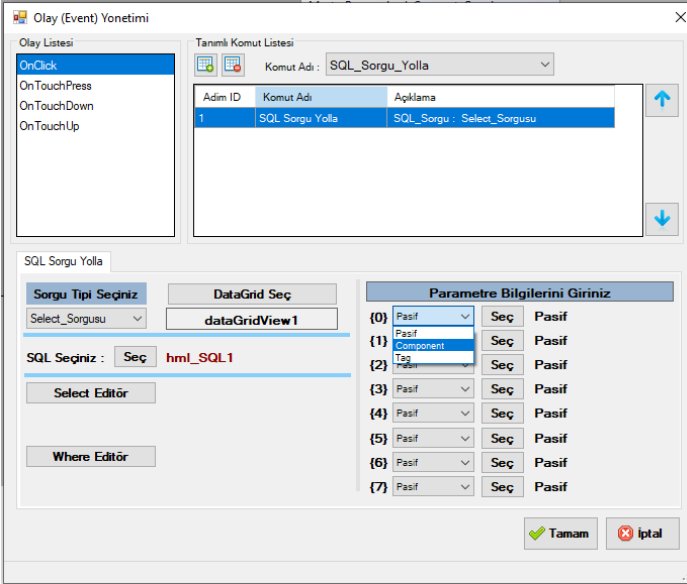
Daha sonra ise MSSQL ile bağlantımızı sağlayacak HMI_SQL nesnesini seçiyoruz.

Select sorgusunda sorgulanacak tablo , HMI_SQL nesnesinde tanımladığımız tablodur.

Başka tablonun sorgulanmasını istiyorsanız Genel_Sorgu yu seçiniz.



Yandaki ekranda gördüğünüz gibi tbl_Report tablosunda Select sorgusu yapacağız.



Artık sorgumuzda kullanacağımız parametreleri seçebiliriz.

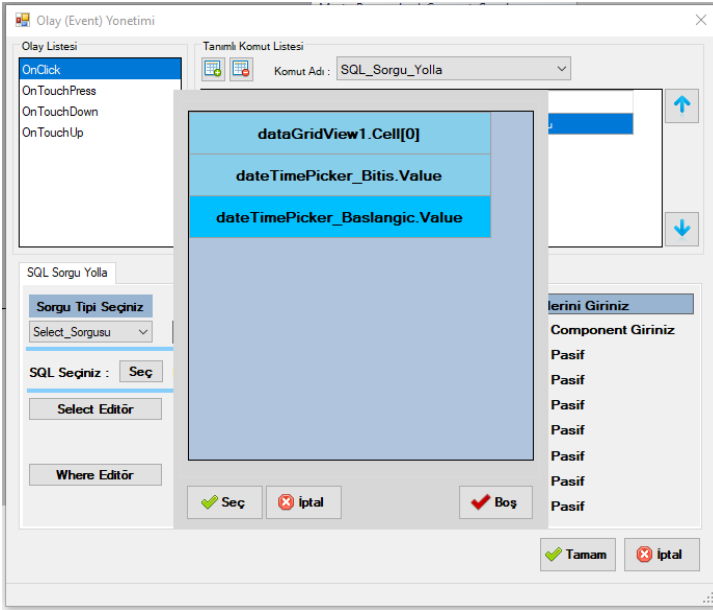
Component veya Tag olarak parametre tanımlaması yapabiliriz.

Sorgumuzu

- dateTimePicker_Baslangic
- dateTimePicker_Bitis

Kontrolerine göre yapacağız.

Bu kontrollerden tarih seçeceğiz , iki tarih aralığındaki satırları getireceğiz

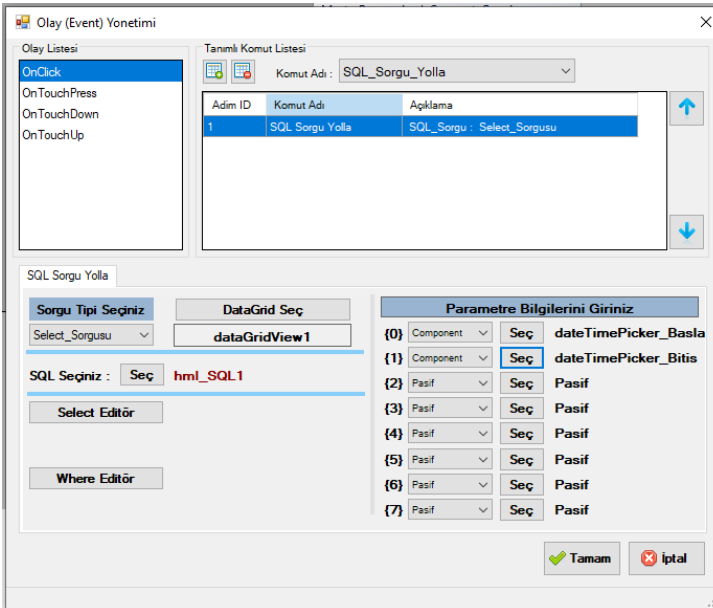


Component seçeneğini seçiniz ve Seç buttonuna tıklayınız.

Ekrandaki gibi seçebileceğiniz kontroller ekrana gelir.

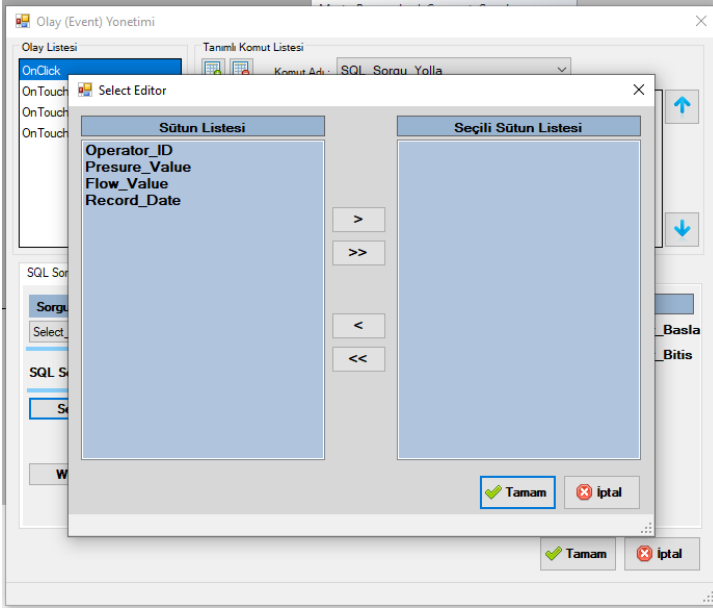
Aşağıdaki gibi 2 adet parametre tanımlayınız.

{0} : dateTimePicker_Baslangic
{1} : dateTimePicker_Bitis

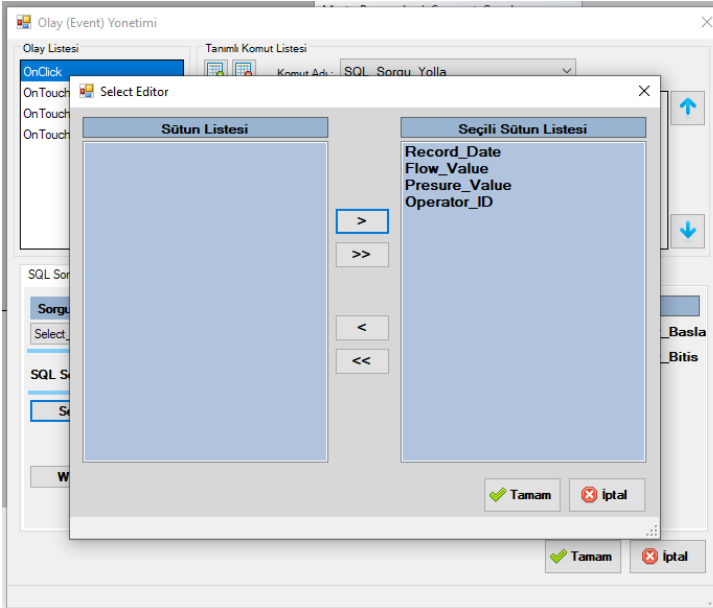


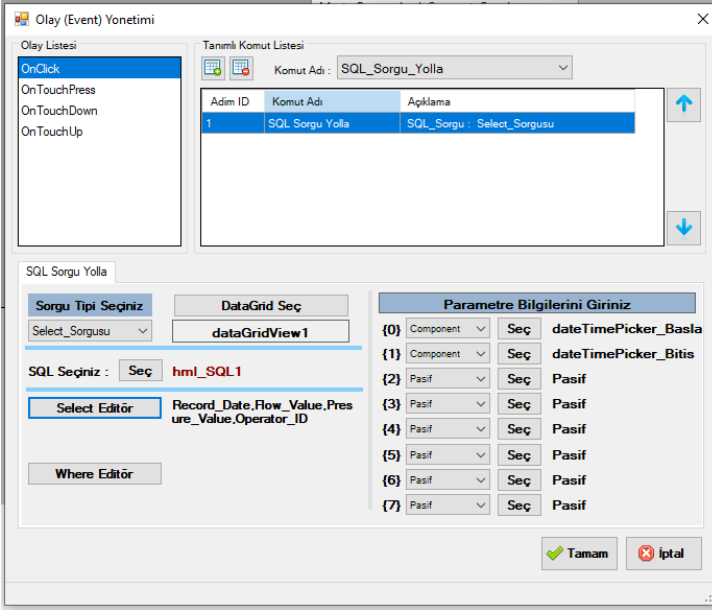
Yanda 2 adet parametre tanımlanmıştır.

{0} : dateTimePicker_Baslangic
{1} : dateTimePicker_Bitis



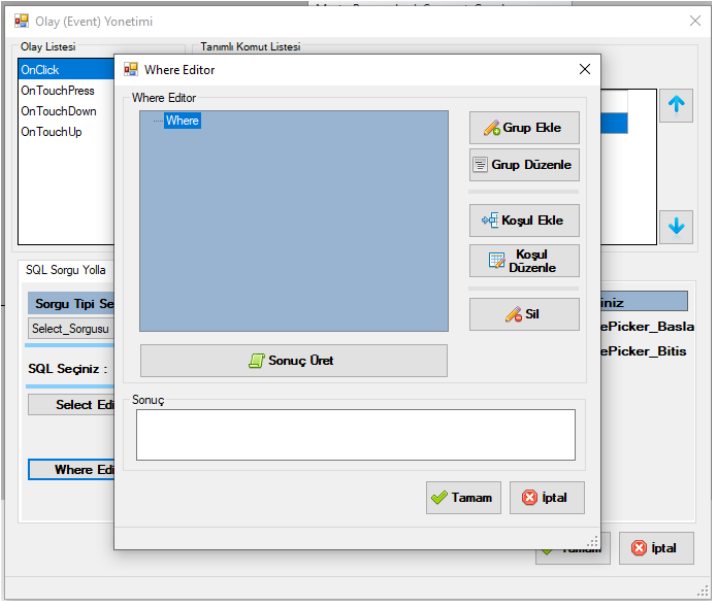
Select Editör Buttonuna tıklayınız ,
DataGridView de görüntülenmesini
istediğiniz sütunları sağ tarafa
taşıyınız.





Seçilen 4 sütunu dataGridView1 kontrolünde görüntüleyeceğiz

Bu şekilde bırakırsak tüm satırlar listelenir ama biz Where ile tarih aralığı belirlemek istiyoruz.



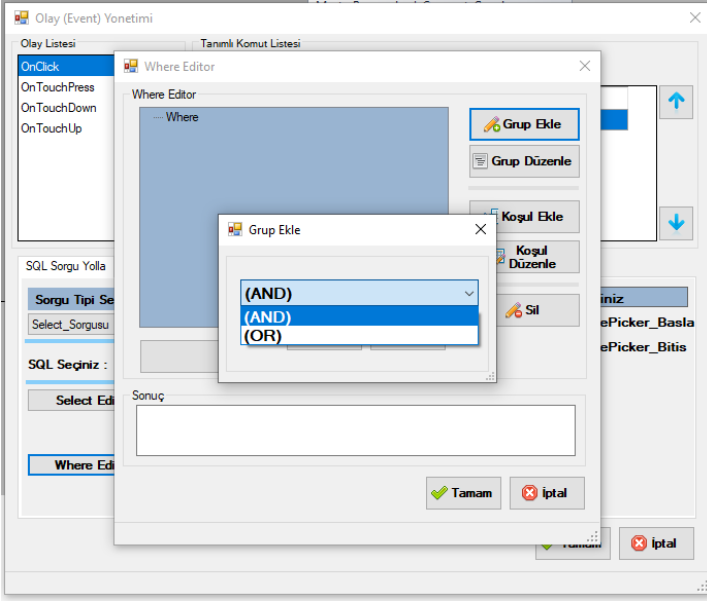
Where Editöre tıklayınız.

Bu editör kompleks Where sorguları oluşturmamızı sağlar.

Grup Ekle ile AND , OR işlemleri Yapabiliriz

Koşul Ekle ile karşılaştırma komutları yazabiliriz.

Sonuç Üret e tıkladığımızda ise sorgumuzun son hali aşağıya yazılır.

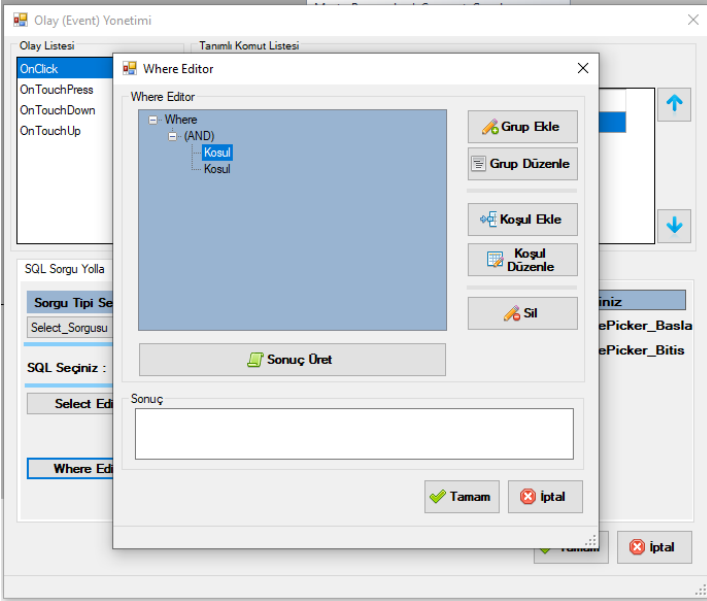


Sorgumuz bir aralık sorgusudur. Başlangıçtan büyük ve Bitişten küçük satırlar listelenecektir.

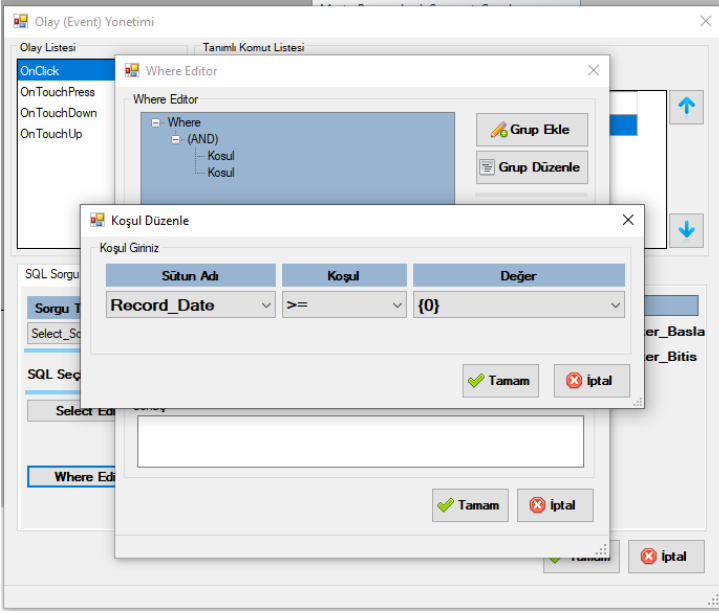
Yani iki koşul ve bu iki koşul da AND ile bağlanacaktır.

Başlangıçtan büyük AND Bitişten küçük

Grup Ekle ile AND operatörünü ekliyoruz ve bu operatöre 2 adet koşul tanımlıyoruz



AND Operatörünü eklediğimizde 2 adet koşul Default olarak gelir. İsterseniz daha fazla koşulu bu operatöre bağlayabilirsiniz.



Koşul seçeneğine **çift tıkladığımızda** , yada Koşul seçeneğini seçip **Koşul Düzenle** butonuna tıkladığımızda ekrana koşul tanımlaması yapacağımız Form gelir.

Yandaki gibi Koşulumuzu yazabiliriz , burada

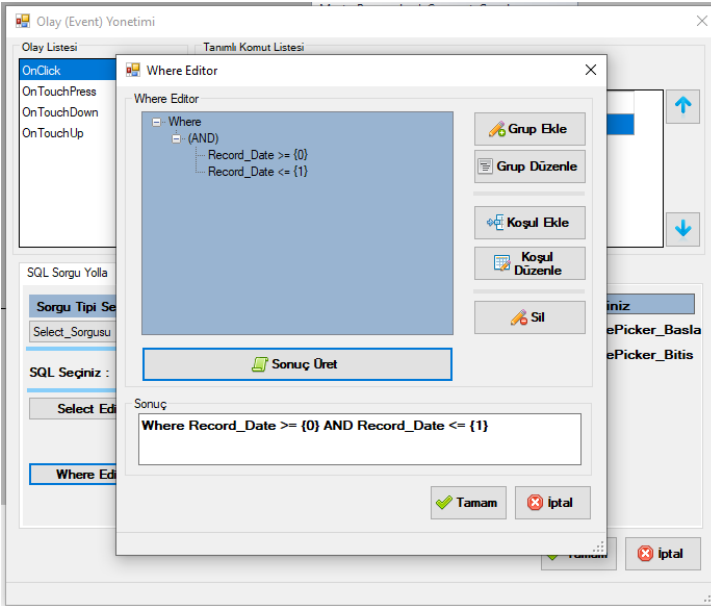
{0} : dateTimePicker_Baslangic

Kontrolüdür.

Record_Date büyük eşit dateTimePicker_Baslangic daki seçili tarihten

İsterseniz Değer bölümünden **Dinamik** Seçimi yapıp kendiniz de yazı ile parametre girişi yapabilirsiniz.

Dinamik girişi kompleks bir yapıdır ve daha sonra anlatılacaktır.



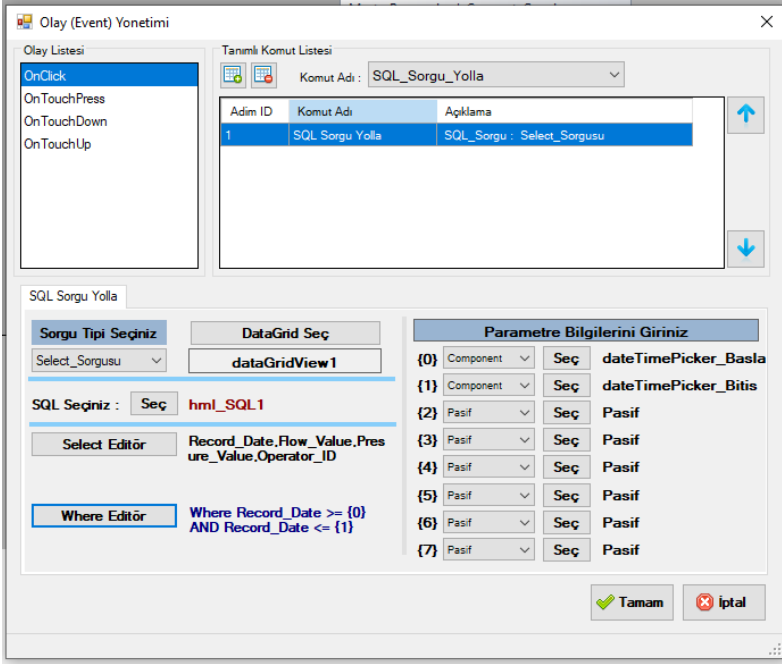
Yandaki ekranda İki Koşul tanımlanmıştır.

{0} : dateTimePicker_Baslangic
{1} : dateTimePicker_Bitis

Sonuç Üret Butonuna tıklayınız ve Where sorgusunu kontrol ediniz.

Where Editör ile çok kompleks sorgular yazabilirsiniz.

Tamam Butonuna tıklayınız ve Event Olay Yönetimi Ekranına geçiniz.



Sorgumuzun tüm girdileri yandaki gibidir.

Tamam Buttonuna tıktlatınız.

Artık sorgumuzu kontrol edebiliriz.

ID	Operator_ID	Pressure_Value	Flow_Value	Record_Date
1	1	5	10	2025-07-23 00:0...
2	1	6	15	2025-07-22 00:0...

Tablomuza Managment Studio u kullanarak yandaki gibi 2 satır ekleyiniz. Tarihlere dikat ediniz.

Başlangıç : 21.07.2025

Bitiş : 25.07.2025

List

Record Date	Flow Value	Pressure Value	Operator ID
23.07.2025	10	5	1
22.07.2025	15	6	1

Projemizi çalıştıralım ve yandaki gibi Başlangıç ve Bitiş tarih aralığı verip List HMI_Buttonuna tıktlayınız

İlgili Sonuçlar dataGridView1 kontrolünde listelenmiştir.

```
2 references
public partial class Rapor : Form
{
    0 references
    public Rapor()
    {
        InitializeComponent();
    }

    1 reference
    private void Rapor_Load(object sender, EventArgs e)
    {
        dateTimePicker_Baslangic.Value = DateTime.Now;
        dateTimePicker_Bitis.Value = DateTime.Now;
    }
}
```

Eğer DateTimePicker kontrollerinin , Form her yüklendiğinde O günün tarihini almasını istiyorsanız

- Formda boş bir yeri çift tıklayınız
- Gelen Formun Load olayı , yani Rapor_Load olayına yandaki kodu yazınız.
- Büyük küçük harf ayırımına dikkat ediniz.

Ekleme Sorgusu :

SQL Sorgu Yolla

Sorgu Tipi Seçiniz
Ekleme_Sorgusu

SQL Seçiniz : Seç **hml_SQL1**

Ekleme Editör

Tanımlı Komut Listesi

Adım ID	Komut Adı	Açıklama
1	SQL Sorgu Yolla	SQL_Sorgu : Ekleme_Sorgusu

Parametre Bilgilerini Giriniz

{0}	Tag	Seç	DB12.DTL2@DTL
{1}	Tag	Seç	DB12.R14@Real
{2}	Tag	Seç	DB12.R18@Real
{3}	Tag	Seç	DB12.D22@DInt
{4}	Pasif	Seç	Pasif
{5}	Pasif	Seç	Pasif
{6}	Pasif	Seç	Pasif
{7}	Pasif	Seç	Pasif

Tamam İptal

Projenize yeni bir HMI_Button kontrolü ekleyiniz.

Connect Seçimi ve Text Düzenle yaptıktan sonra Event Olay Yönetimi Özelliğini Tıklayınız

SQL_Sorgu_Yola Komutunu ekleyiniz

Sorgu Tipi : Ekleme_Sorgusu yapınız.

SQL Seçiniz.

DB12(Write_To_SQL) datablock unda tanımladığımız 4 adet Tag adresini parametre olarak tanımlayınız.

Eklenecek satır , değerlerini bu Tag Adreslerinden alacaktır.

Yani HMI_Buttona tıklandığında , eklenen satırın , alan değerleri , Bu tag adreslerinin değeridir.

Insert Editor

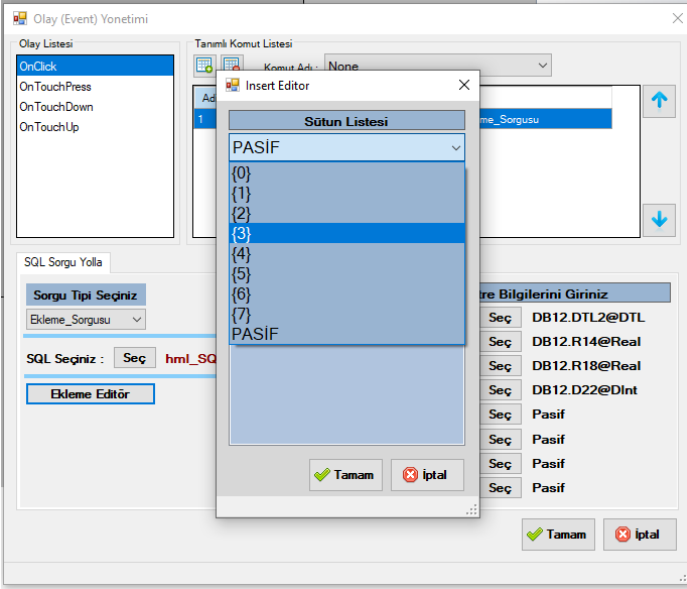
Sütun Listesi

PASIF

Operator_ID
Pressure_Value
Flow_Value
Record_Date

Tamam İptal

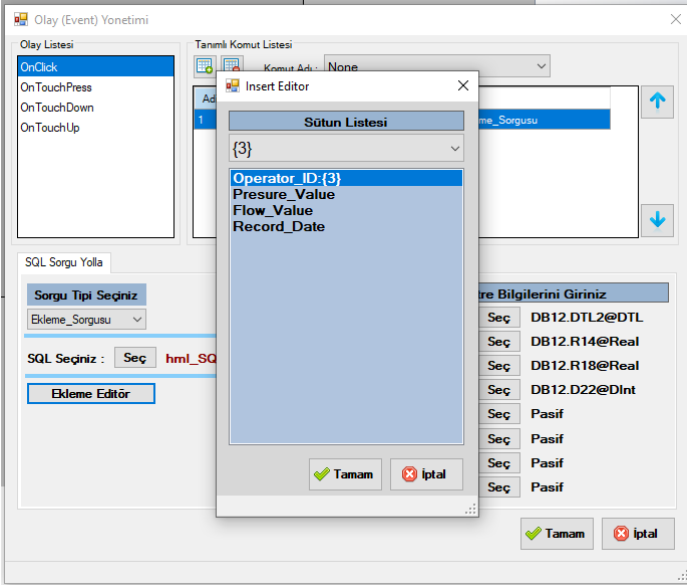
Ekleme işleminde hangi sütunun hangi parametre ile eşleşeceğini buradan belirliyoruz.



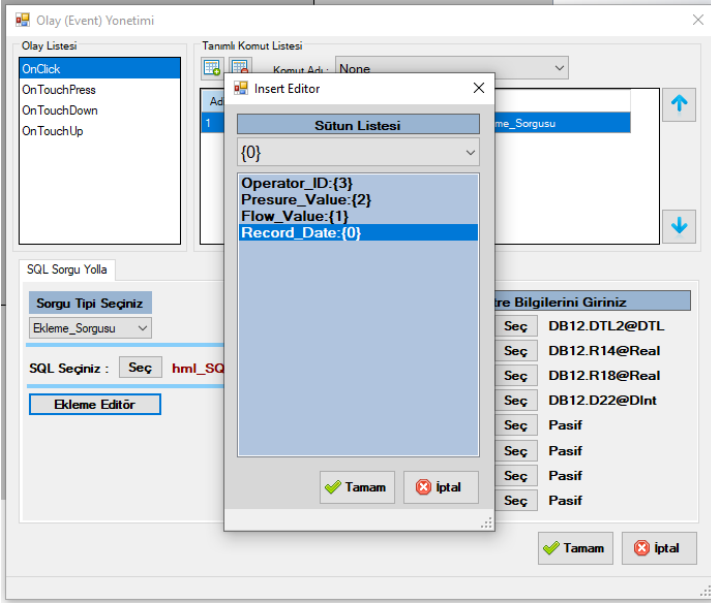
Yandaki ekranda Operator_ID sütunu ile {3} parametresini eşleştirdik.

Yani Her satır eklendiğinde , satırın Operator_ID değeri , {3} nolu parametrenin değeridir.

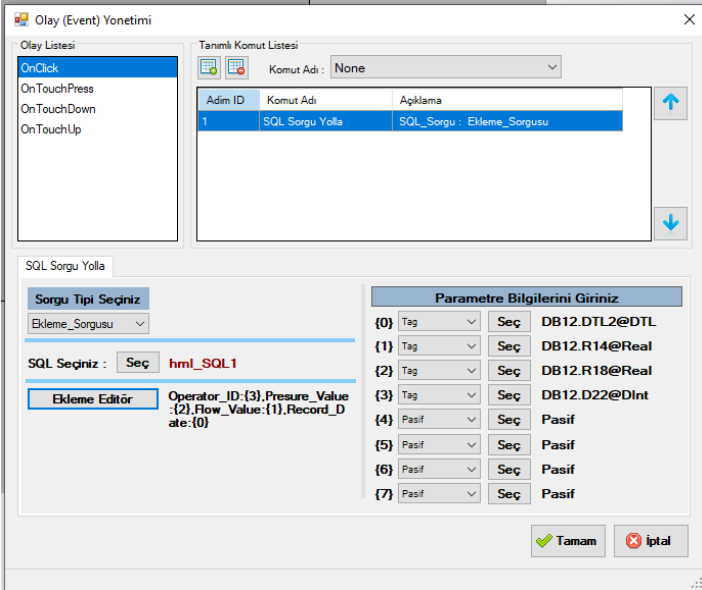
Operator_ID değerini {3} nolu parametreden alır.



Operator_ID değerini {3} nolu parametreden alır.



Diğer sütunlar içinde , yandaki gibi parametre seçimi yapılmıştır.



Ekleme Sorgumuzun son hali yandaki gibidir.

Projemizi çalıştırıp HMI_Button a tıkladığımızda; Sorgumuz çalışır , Parametrede belirlediğimiz tag adreslerinin değerlerini alır ve ekleme işlemini yapar.

Silme Sorgusu :

Adim ID	Komut Adı	Açıklama
1	SQL Sorgu Yolla	SQL_Sorgu : Silme_Sorgusu

(0)	Tag	Seç	DB12.DTL2@DTL
(1)	Tag	Seç	DB12.R14@Real
(2)	Tag	Seç	DB12.R18@Real
(3)	Tag	Seç	DB12.D22@DInt
(4)	Component	Seç	dataGridView1
(5)	Pasif	Seç	Pasif
(6)	Pasif	Seç	Pasif
(7)	Pasif	Seç	Pasif

Projenize yeni bir HMI_Button kontrolü ekleyiniz.

Connect Seçimi ve Text Düzenle yaptıktan sonra Event Olay Yönetimi Özelliğini Tıklayınız

SQL_Sorgu_Yola Komutunu ekleyiniz

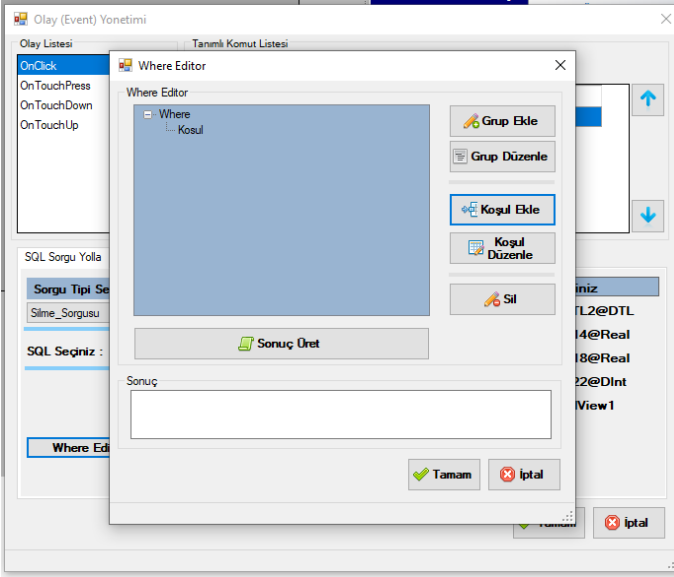
Sorgu Tipi : Silme_Sorgusu yapınız.

{4} nolu parametreye dataGridView1 kontrolü eklenmiştir. dataGridView1 kontrolünün parametre için alınan hücresi Cell[0] dir , bu hücrenin visiblesi false de olabilir , yani visible false olsa bile Cel[0] parametre değeri olarak alınır.

Önemli Uyarı : HMI SQL sisteminde otomatik tanımlanan vede visible false olan ID sütunu vardır. Bu sütun özel tanımlı bir sütundur ve her satır için değeri farklıdır. HMI SQL kontrolünde , otomatik olarak tanımlanır. HMI SQL Select sorgularında bu sütun alınır DataGridView kontrolüne eklenir ama gösterilmez. DataGridView in Cell[0] ı HMI SQL Select sorgusunda işte bu ID sütunudur

Silme Sorgusunda Editör olarak sadece Where editörü kullanılır.

HMI_SQL de tanımlı tabloya Where komutu ile sorgu yolanır. Eğer Where komutunu kullanmazsanız , HMI_Button nesnesine tıkladığınızda , tüm satırlar silinir. Where komutu ile ; silinecek satır için koşul oluşturuyoruz.

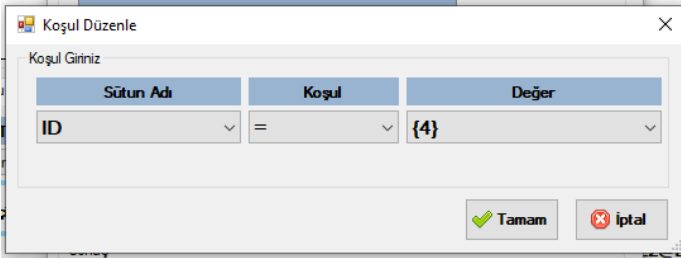


Where Editör Buttonuna tıklayınız.

Yandaki gibi açılan ekranda sadece 1 adet koşul ekleyiniz.

Bu koşul içerisine ID sütunu eşit ise (otomatik tanımlanan) dataGridViev1 in seçili satırının Cell[0] hücresine koşulunu oluşturacağız.

Yani dataGridViev1 de seçili satırı sileceğiz.



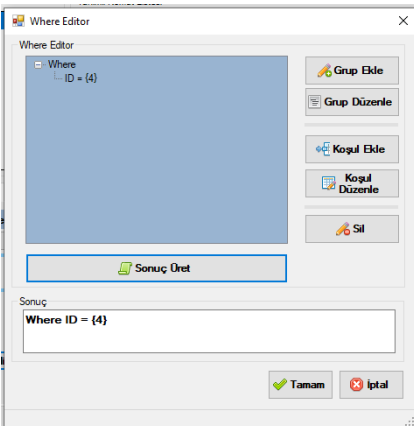
Özel tanımlı ID sütunu Koşul Düzenle ekranında mutlaka bulunur. Bu sütun HMI_SQL kontrolünde kullanıcıdan bağımsız olarak , otomatik tanımlanan sütundur ve değeri her satır için farklıdır.

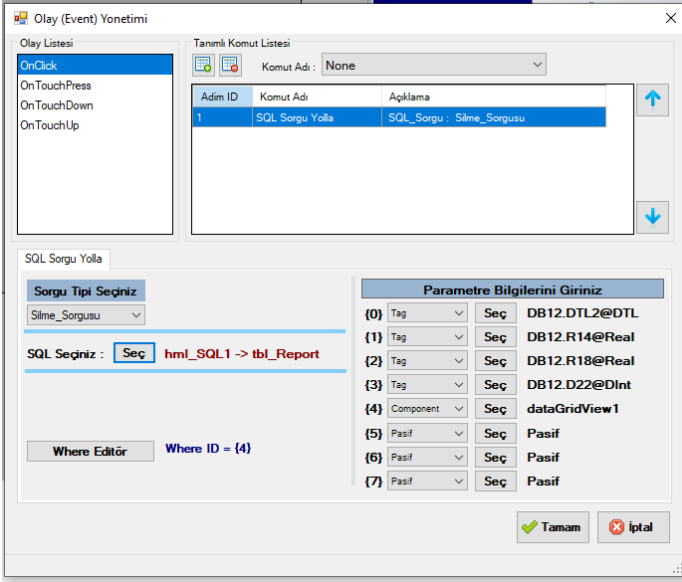
ID = {4}

ID sütunu eşit ise (otomatik tanımlanan) dataGridViev1 in seçili satırının Cell[0] hücresine

{4} : dataGridViev1 SelectedCell[0]

Where Sorgumuzun son hali yandaki gibidir.





Silme_Sorgu muzun son hali yandaki gibidir.

HMI_Button a tıklandığında silme sorgusu çalışır.

Yani dataGridView1 de seçili satırı silinir.

Genel Sorgu : Bu sorgu tipi ile kendi kompleks sorgularınızı rahatlıkla Mssql e yollayabilirsiniz.

Genel_Sorgu HMI_SQL Kontrolünün bağlantı parametrelerini kullanarak ama HMI SQL in tablosuna erişmek zorunda değildir. İçerisinde Kompleks Join işlemleri gibi sorgularıda yapabilirsiniz.

Projenize yeni bir HMI_Button kontrolü ekleyiniz.

Connect Seçimi ve Text Düzenle yaptıktan sonra Event Olay Yönetimi Özelliğini Tıklayınız

SQL_Sorgu_Yola Komutunu ekleyiniz

Sorgu Tipi : Genel_Sorgu yapınız.

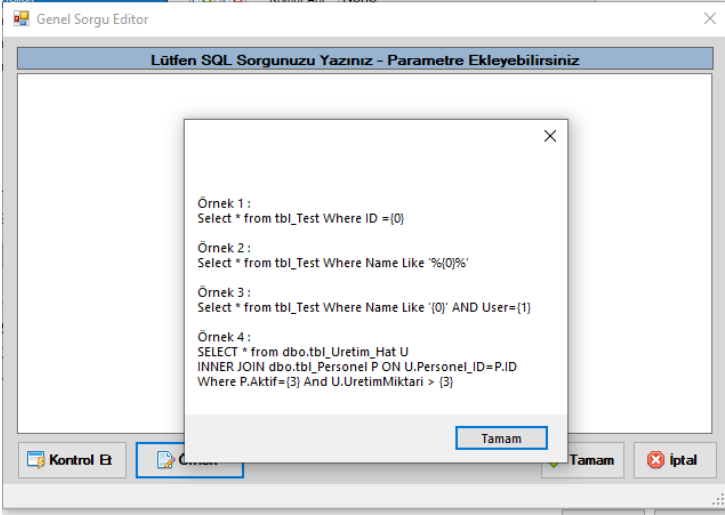
Bu sorguda , Genel Sorguyu Select komutu ile kullanacağız.

{4} : dateTimePicker_Baslangic
{5} : dateTimePicker_Bitis

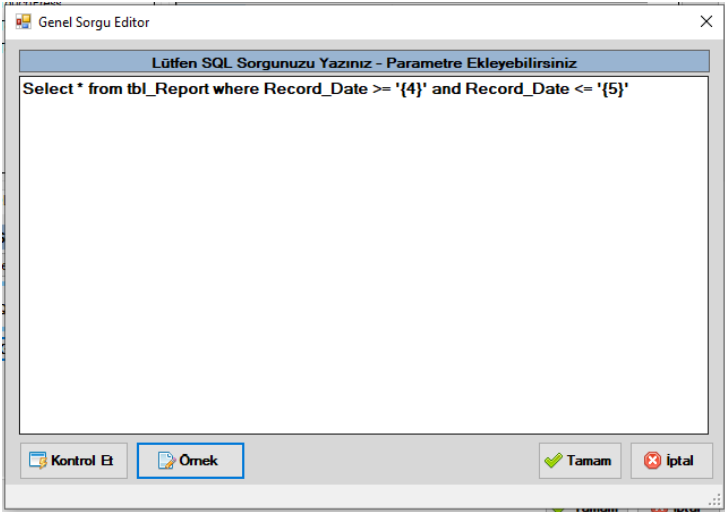
Parametrelerini tanımladık ve bu kontrollerden tarih seçeceğiz , iki tarih aralığındaki satırları getireceğiz.

Genel Sorgu Editorüne tıkladığımızda yandaki sorgu ekranı çıkar.

Burada MSSQL de New Query de olduğu gibi direkt sorgunuzu yazabilirsiniz, parametre işaretini kullanarak { } , parametrelerinizi sorgu içerisine yerleştirebilirsiniz.



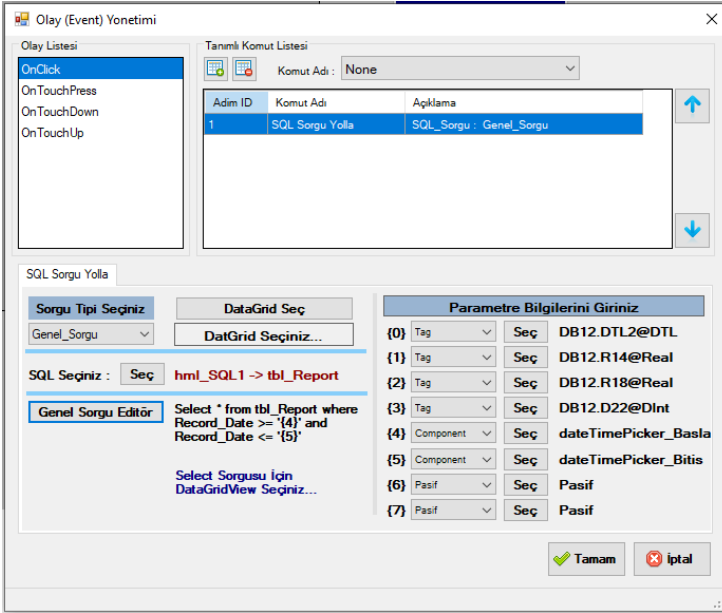
Örnek Butonuna tıkladığınızda bazı sorgu örneklerini görebilirsiniz.



Yanda gördüğünüz gibi sorgunuzu rahatlıkla yazabilirsiniz.

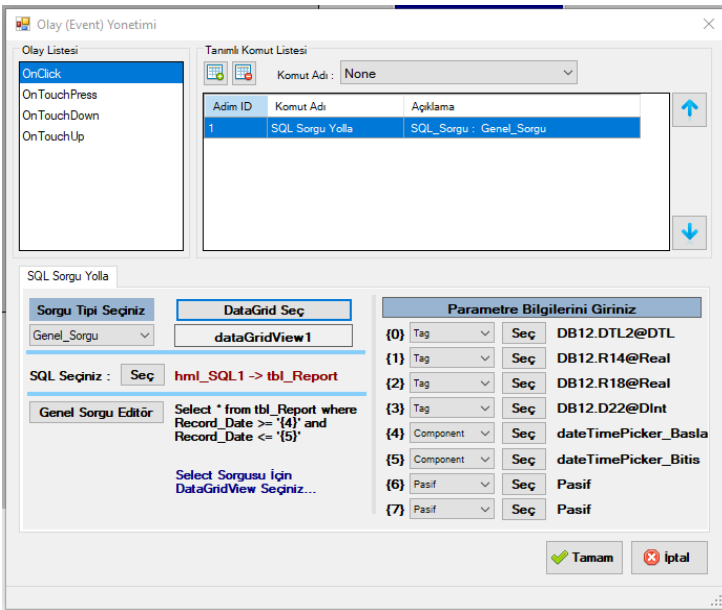
Genel_Sorgu da SQL Sorgu kuralları geçerlidir , bundan dolayı String , Char ve Date tipleri için tırnak("") kullanmayı unutmayınız.

Burada tbl_Report tablosunda sorgu işlemi yapılmıştır. **Genel Sorgu için bu zorunlu değildir.** HMI_SQL kontrolünün kullandığı DataBase in diğer tabloları içinde sorgu yazabilirsiniz.



Sorgumuz bir Select sorgusu olduđu için , sonucun yazılacağı bir DataGridView e de ihtiyacımız var.

DataGrid Se buttonuna tıklayıp , DataGridView Seiniz.





Sorgumuzun son hali yandaki gibidir.

Genel_Sorgu ile

HMI_SQL kontrolünün kullandığı DataBase in tablolarında ,

Insert , Update , Delete , Select sorgularını parametre kullanarak rahatlıkla yazabilirsiniz.

Öncelikle Tarih Aralığı Seçiniz ve İçerisine Genel_Sorgu ile Select Sorgusu yazılmış HMI_Button kontrolüne tıklayınız.

Başlangıç :	20.07.2025 	Send Query
Bitiş :	25.07.2025 	

	ID	Operator_ID	Pressure_Value	Flow_Value	Record_Date
▶	1	1	5	10	23.07.2025
	2	1	6	15	22.07.2025

Where Editör Koşul İçerisinde {Dinamik} Kullanımı : Koşul değişkeninin daha esnek bir şekilde yazılmasını sağlar.

Örneğin ; Operatör adı ile ilgili bir sorgu yapacağımızı varsayalım. Br HMI_TextBox tanımlayalım , HMI_TextBox a girdiğimiz veriyi Operatör Adlarında aratalım. Bunu yaparkende **içerisinde geçen** sorgusunu yapalım. Yani HMI_TextBox a girdiğimiz metni , Operatörün adının içinde arasın , Operatörün Adı Murat ise girdiğimiz metin “mu” ise eşleşme sağlansın (Murat in içerisinde “mu” geçtiği için) . Bunu yapmak için MSSQL deki % operatörünü kullanacağız.

Aşağıdaki SQL Sorguları bu işlemi yapar.

```
Select * from tbl_Report Where Operator_Name Like '%mu%'
```

yada

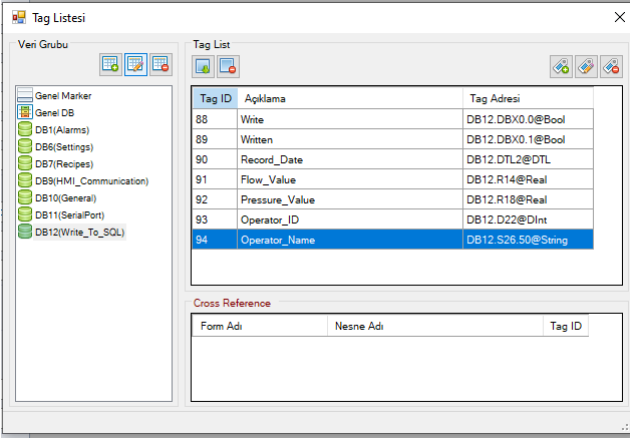
```
Select Record_Date, Flow_Value, Pressure_Value, Operator_ID, Operator_Name  
from tbl_Report Where Operator_Name Like '%mu%'
```

İşte biz bu sorguyu sağlayacak formu tasarlayıp , dinamik parametreyi kullanarak , parametremizi '%{2}%' şeklinde yazacağız , {2} verisini HMI_TextBox dan alacak.

Başlayalım :

DB12(Write_To_SQL)			
	Name	Data type	Offset
1	Static		
2	Write	Bool	0.0
3	Written	Bool	0.1
4	Record_Date	DTL	2.0
5	Flow_Value	Real	14.0
6	Pressure_Value	Real	18.0
7	Operator_ID	DInt	22.0
8	Operator_Name	String[50]	26.0

Öncelikle DB12(Write_To_SQL) DataBlock una Operator_Name isminde String[50] tipinde bir değişken tanımlayalım.



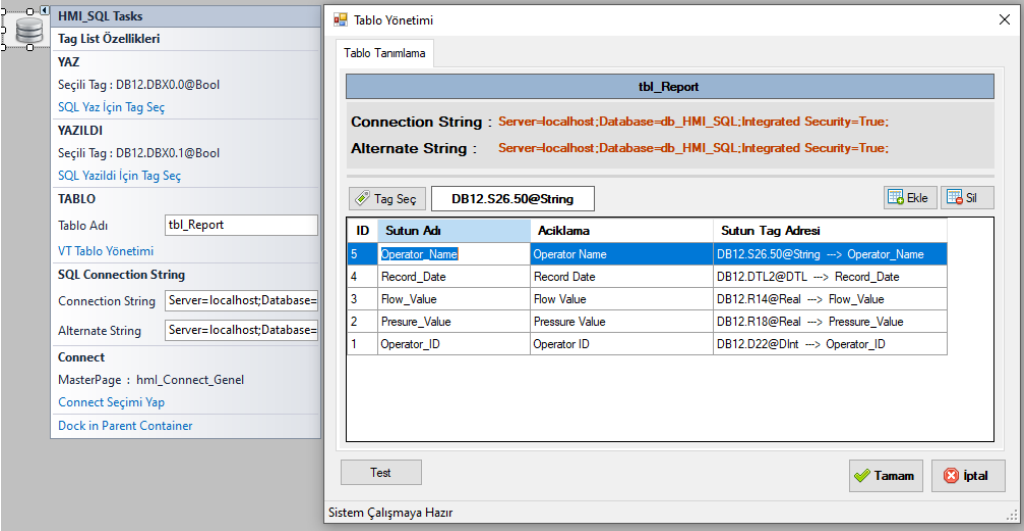
MasterPage üzerindeki HMI_Connect kontrolüne bu tagı ekleyelim.

Yanda DB12(Write_To_SQL) DataBlock una Operator_Name isminde String[50] tipinde bir değişkeni yazdık.

Bunu ister manuel ekleyebilirsiniz , isterseniz kopyalayıp yapıştırabilirsiniz.

Kopya-Yapıştır işlemi için HMI_Connect kontrolünün kullanımına bakabilirsiniz.

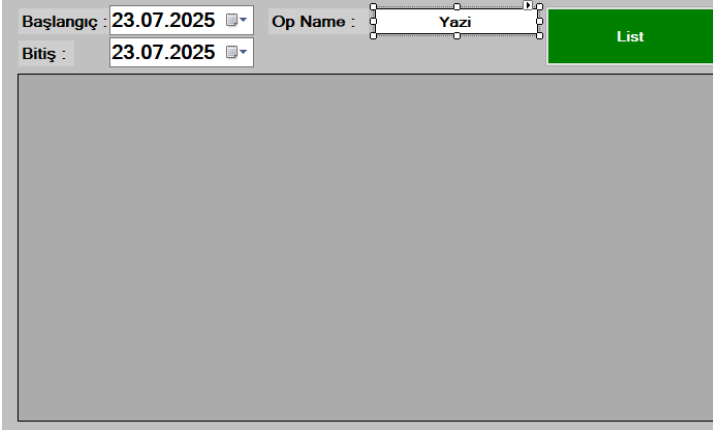
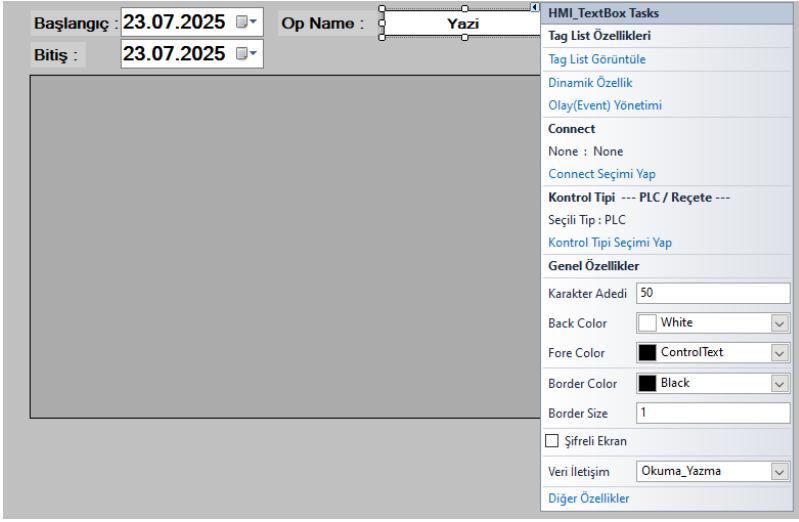
Daha sonra ise mutlaka , MasterPage de tanımladığımız HMI_SQL nesnesine bu yeni sütunu ekliyoruz. Aşağıda Operator_Name nin eklenmiş hali gösterilmiştir.



Artık yeni formumuzu tasarlayabiliriz. Aşağıdaki gibi OP Name için bir HMI_TextBox kontrolünü formumuza ekleyelim.

HMI_TextBox a Connect eklemesi yapmayınız. Eğer Connect ataması yapmazsanız , HMI_TextBox ve NumericBox u gene kullanabilirsiniz. Dokunmatik klavye ile veri girişi yapabilir ve ayrıca kod tarafında kontrolün Text özelliği ile veriyi alabilirsiniz.

Yani HMI_TextBox ve HMI_NumericBox a Connect ataması yapmazsanız , size dokunmatik ekran klavyesini kullandırtan bir kontrol gibi davranır.

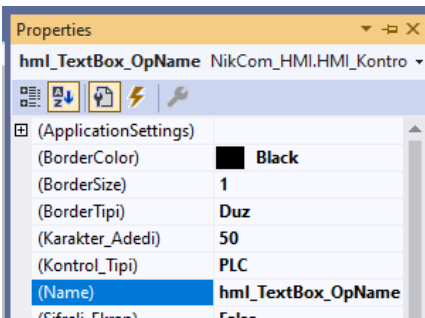


Yandaki ekranda Operator_Name için HMI_TextBox kontrolü eklenmiştir.

Bu kontrolün **Name** özeliği Properties ekranından

hml_TextBox_OpName

yapılmıştır.

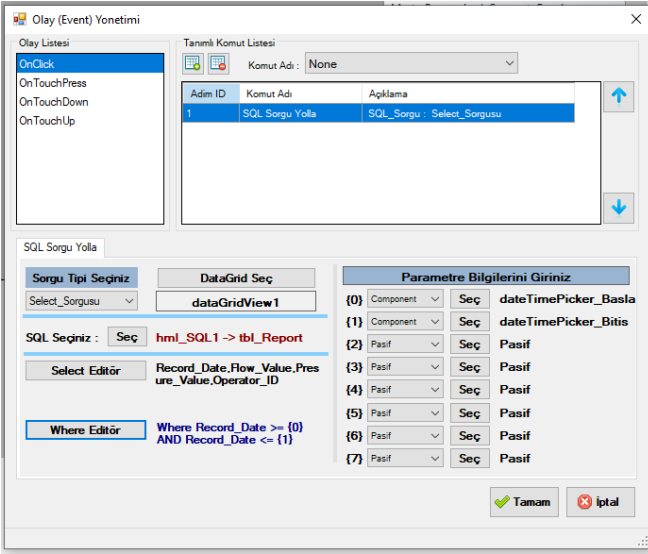


Yanda kontrolün Properties ekranı gösterilmiştir.

Properties ekranına Visual studio da

ile ulaşabilirsiniz.

Kontrolü tıkladığınızda , kontrolün özellikler Properties ekranında gösterilir.



Yanda daha önceden parametrelerini girdiğimiz "List" HMI_Button kontrolünün Event Olay Yönetim Özelliği görüntülenmiştir.

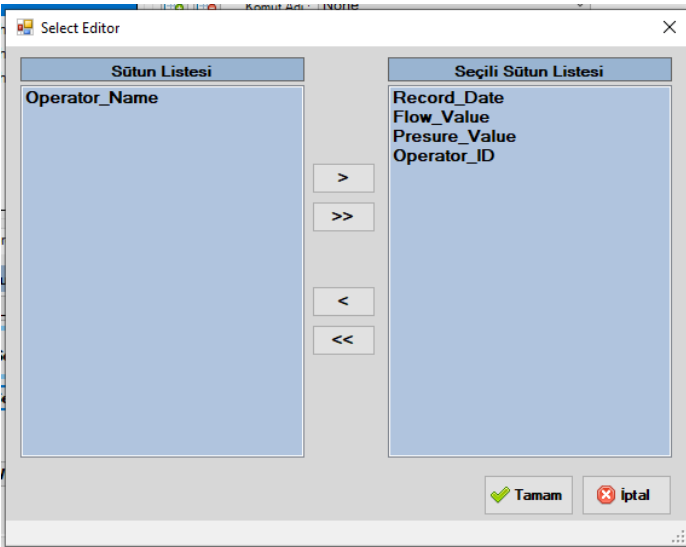
Bu Özelliğe

Select Editor den Operator_Name sütununu Sorgumuza ekleyeceğiz

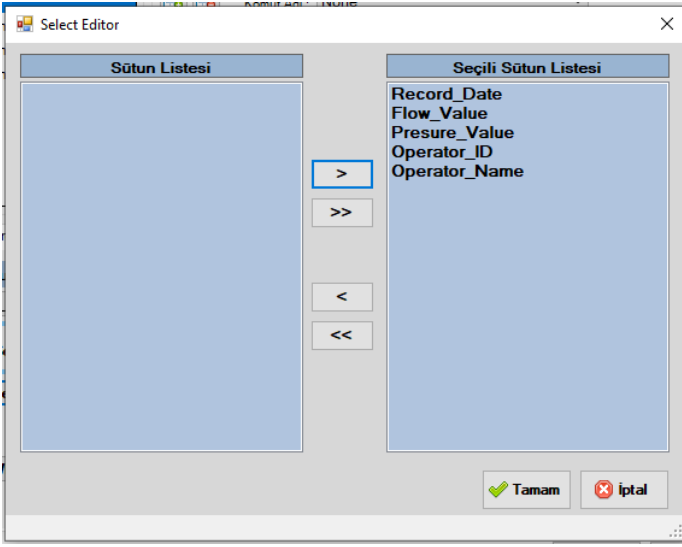
{2}: hml_TextBox_OpName

Parametresini ekleyeceğiz.

Where Editör ile de yeni koşul tanımlayıp, '%{2}%' için Dinamik değer girişi yapacağız.



Öncelikle Select Editör den Operator_Name Sütununu seçelim ve Sağa taşıyalım



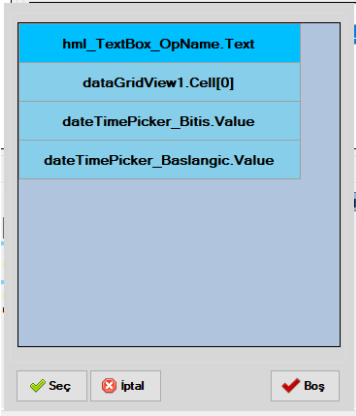
Operator_Name Sütununu Sağa taşıdık.

Parametre Bilgilerini Giriniz			
{0}	Component	Seç	dateTimePicker_Basla
{1}	Component	Seç	dateTimePicker_Bitis
{2}	Pasif	Seç	Pasif
{3}	Pasif	Seç	Pasif
{4}	Pasif	Seç	Pasif
{5}	Pasif	Seç	Pasif
{6}	Pasif	Seç	Pasif
{7}	Pasif	Seç	Pasif

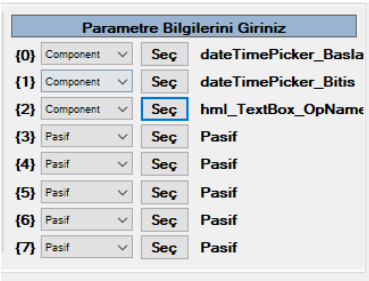
HMI_TextBox u Parametre bölümünde {2} nolu parametre olarak ekliyoruz.

Parametre Bilgilerini Giriniz			
{0}	Component	Seç	dateTimePicker_Basla
{1}	Component	Seç	dateTimePicker_Bitis
{2}	Component	Seç	Component Giriniz
{3}	Pasif	Seç	Pasif
{4}	Pasif	Seç	Pasif
{5}	Pasif	Seç	Pasif
{6}	Pasif	Seç	Pasif
{7}	Pasif	Seç	Pasif

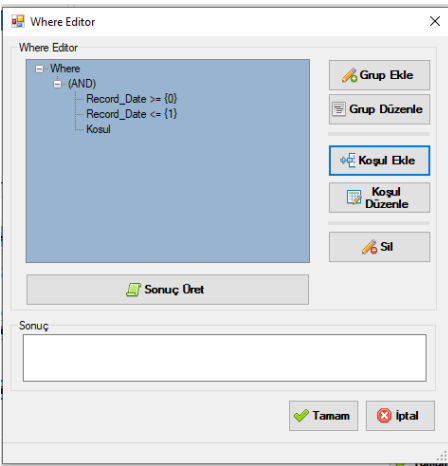
{2} için Component i seçip , Seç buttonuna tıklarız.



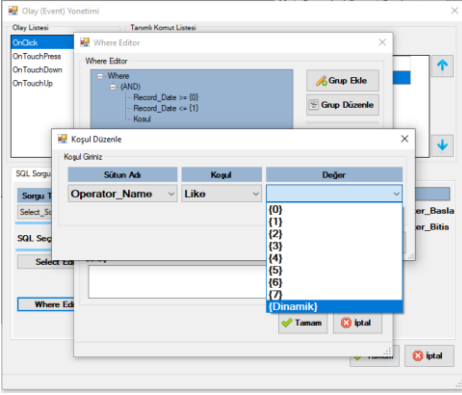
Gelen ekrandan , yeni eklediğimiz HMI_TextBox u seçip parametre olarak yüklüyoruz.



{2} parametresine hml_TextBox_OpName i yükledik.



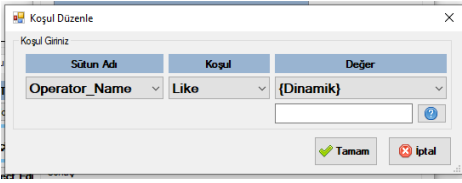
Artık Where editör e Koşul eklemesi yapabiliriz. AND altına yeni koşulumuzu ekleyelim.



Yeni eklediğimiz koşula çift tıklayalım ve açılan Koşul Düzenle ekranından

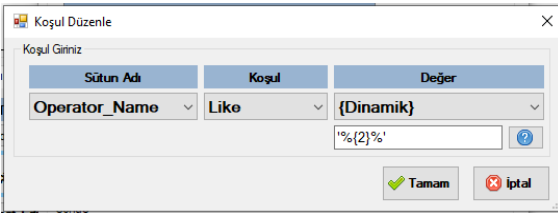
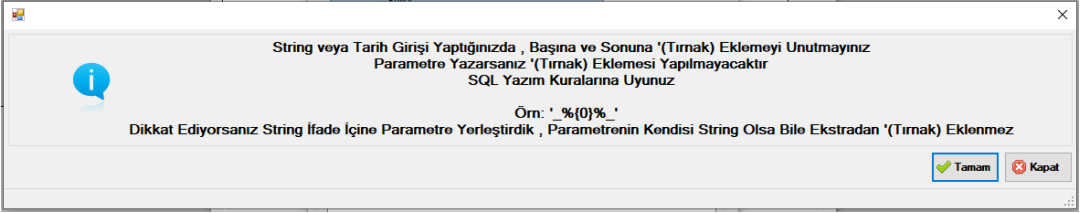
Operator_Name Like {Dinamik}

Seçimi Yapalım



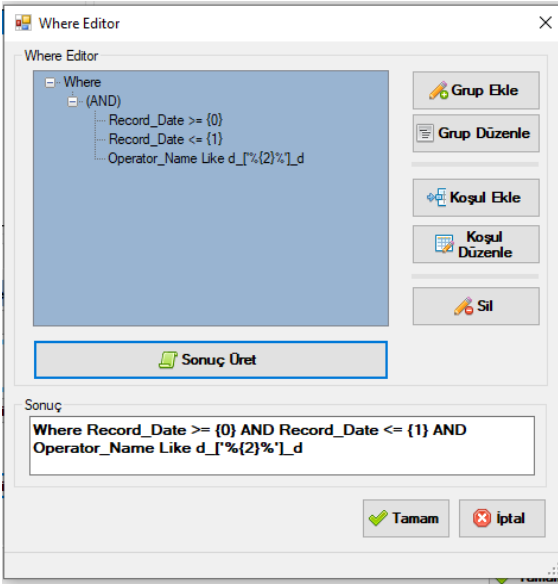
Dinamik seçimi yaptığımızda aşağıda parametre girişi için bir TextBox görüntülenir.

Yanındaki (?) Buttonuna mutlaka tıklayınız. Butona tıkladığınızda çıkan uyarı mesajını dikkatlice okuyunuz.



Parametre girişini yandaki gibi yapalım , burada " (Tırnak) işaretlerine dikkat ediniz.

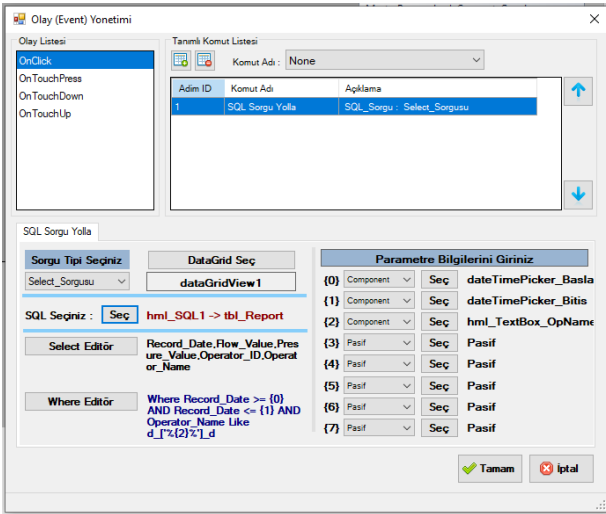
SQL Komutlarında String ve Date tipleri için tırnak kullanılmalıdır.



Dinamik parametre eklendiğinde sorgu

d_! _d içerisine alınır.

Ekrandaki yazıma dikkat ediniz , bu Where Editörüne özgü bir yazım şeklidir.



Sorgumuzun son hali yandaki gibidir.

Column Name	Data Type	Allow Nulls
ID	bigint	<input type="checkbox"/>
Operator_ID	bigint	<input type="checkbox"/>
Pressure_Value	float	<input type="checkbox"/>
Flow_Value	float	<input type="checkbox"/>
Record_Date	datetime	<input type="checkbox"/>
Operator_Name	nvarchar(50)	<input type="checkbox"/>

ID	Operator_ID	Pressure_Value	Flow_Value	Record_Date	Operator_Name
1	1	5	10	2025-07-23 00:0...	Murat
2	1	6	15	2025-07-22 00:0...	Murat
3	2	8	20	2025-07-22 00:0...	Remzi

Başlangıç : 21.07.2025 Op Name :

Bitiş : 25.07.2025

Record Date	Flow Value	Pressure Value	Operator ID	Operator Name
23.07.2025	10	5	1	Murat
22.07.2025	15	6	1	Murat
22.07.2025	20	8	2	Remzi

Başlangıç : 21.07.2025 Op Name :

Bitiş : 25.07.2025

Record Date	Flow Value	Pressure Value	Operator ID	Operator Name
23.07.2025	10	5	1	Murat
22.07.2025	15	6	1	Murat

Projemizi çalıştırdığımızda , MSSQL tbl_Report tablosuna yandaki gibi Operator_Name sütununun eklendiğini görebilirsiniz.

Managment Studio ile tabloya yeni bir satır ekleyiniz ve Operator_Name sütununun satırlarında ekrandaki gibi doldurunuz.

Projemizi çalıştırınız. Tarih aralığı seçip List HMI_Buttonuna tıkladığımızda satırların geldiğini görürsünüz.

Burada hml_TextBox_OpName şimdilik boş

hml_TextBox_OpName içerisine "mu" yazınız ve List buttonuna tıklayınız.

Belirtilen tarih aralığındaki , içerisinde "mu" geçen Operatörler listlenecektir.