

## 05\_Dinamik Rapor

- Widget | Dinamik Rapor
  - Tanım
  - Kullanımı
  - Ekran
    - Görünüm
    - Ayarlar
    - SQL Sorgu
    - Parametreler
    - Kolonlar
  - Veritabanı İşlemleri
  - Bağlı Modüller
  - Örnek 1
    - SQL Sorgu
    - Parametreler
    - Kolonlar
    - View
  - Örnek 2
    - SQL Sorgu
    - Parametreler
    - Kolonlar
    - View
  - Örnek 3
    - SQL Sorgu
    - Parametreler
    - Kolonlar
  - Örnek 4
    - SQL Sorgu
    - Parametreler
    - Kolonlar
  - Örnek 5

## Widget | Dinamik Rapor

### Tanım

Veritabanındaki istenilen bilgiyi SQL tipi ile sorgusu girilerek rapor çekmeye yarayan ve istenen tüm sayfalara eklenebilen widget eklentisidir. Gösterilmek istenen kolonlar "Kolonlar" kısmından ayarlanabilen, kısıt olarak "Parametreler" kısmından filtrelenebilen ve kullanıcıya bu bilgileri ekrana dökme ya da Excel, Word, Pdf olarak alabilme özelliği sunan bir widgettır.

### Kullanımı

Özelleştirilebilir sistem sayfalarına ya da yeni oluşturulan sayfalara widget olarak eklenmektedir. Eklendikten sonra test amaçlı en altta verilen test girdileri ile kullanım sağlanabilmektedir.

"Tasarım Modu" açıldığında "Sayfa Tasarımı Düzenle" özelliği ile sayfa layout üzerinden "Yeni Widget Ekle" ile eklenebilir.

### Ekran

#### Görünüm

Tüm sayfalara eklenebilir widget bileşenidir. Belirlenen ayarlara göre cari rapor bilgisini vermektedir. Örnek raporumuz Stored Procedure ile çalışmaktadır. Parametre olarak Cari Kodu, Başlangıç ve Bitiş Tarihi alıp bu kısıtlara göre arama yapmaktadır. Ayarlar kısmında belirtilen 6 adet kolon bu kısıtlara göre dolmaktadır.



Cari Kodu \*

[12011001] BOZÜYÜK OTOMOTIV



Başlangıç Tarihi



Bitiş Tarihi



Rapor

Excel

Word

Pdf

Yazdır

Cari Kodu	TARİH	BELGE_NO	ACIKLAMA	HKA	BORC	ALACAK
12011001	19.01.2012 00:00:00	000000000258664	FT.MIZ 0E0000000003324	A	762,28	0,00
12011001	12.01.2012 00:00:00	000000000258456	FATURAMIZ	A	161,71	0,00
12011001	10.01.2012 00:00:00	000000000258415	FT.MIZ E00000000032659	A	366,97	0,00
12011001	09.01.2012 00:00:00	000000000258388	FATURAMIZ	A	3.337,98	0,00

## Ayarlar

Yetkiniz dahilinde Tasarım modu ile "Ayarlar" kısmından eklentinin parametre ekranı açılmaktadır.

- **Başlık** Widget başlık bilgisi
- **SQL Sorgu Tipi** SQL'de birden fazla; Store Procedure,View gibi sorgu tipleri vardır.Raporda gösterilmesini istediğiniz tip burada belirtilmektedir.
- **SQL Sorgu** Kullanılacak olan sorgu burada belirtilmektedir. **NOT: Bu kısımda where kısıtı kullanılmalıdır. (yoksa bile where 1=1 şeklinde yazılmalıdır. Aksi halde sorgu varsayılan olarak TBLCASABIT içinden çeker.)**
- **Parametreler** Veri filtresi için burada kısıtlar belirtilmektedir.
- **Kolonlar** Raporda gösterilmek istenilen kolonlar burada belirtilmektedir. **NOT: ColumnType değeri için 1 normal yazı, 2 buton ile yeni sekmede, 3 buton ile pop up olarak açılmayı sağlar.**
- **NDS** Tutar kısmında virgülden sonra gösterilmek istenen rakam sayısı burada belirlenmektedir.

**Ayarlar** ×

Başlık	<input type="text"/>
SQL SorguTipi	StoreProcedure <span>▼</span>
SQL Sorgu	<pre>exec PRTSP_TEST_CARI_HAREKET @BranchId,@CARI_KOD,@FormData</pre>
Parametreler	<pre>[ {   "Name": "CARI_KOD",   "ColumnName": "CARI_KOD",   "Label": "3D957E9",   "DefaultValue": "",   "Required": true,</pre>
Kolonlar	<pre>[{"ColumnName":"CARI_KOD","DisplayName":"3D957E9"}]</pre>
NDS	<input type="text"/> 2

### SQL Sorgu

SQL sorgu kısmında kullanılan stored procedure ya da view ın içeriğinde;

- StoreId,
- BranchId,
- UserId,
- CurrentAccountCode,
- FormData Xml bulunmalıdır.

StoreId kullanıcının o an ki mağaza bilgisini, BranchId şube bilgisini, UserId kullanıcı bilgisini, CurrentAccountCode o an ki carinin kodunu almaktadır. FormData parametresi ise birbiriyle senkron çalışan işlemler için gerekli parametreyi sağlamaktadır.

**CREATE PROCEDURE** [dbo].[procedure\_name]

@StoreId **VARCHAR**(35),

@BranchId **VARCHAR**(35),

@UserId **VARCHAR**(35),

@CurrentAccountCode **VARCHAR**(35),

@FormData **Xml**

Şeklinde yaratılmaktadır.

## Parametreler

- **Name** kısıtı stored proceduredeki parametreyi etkilemektedir.

\*\*Dinamik rapor ve Parametre (Kısıt) widget ayarlarında; verilen tarih kısıtları için "Parametreler" bölümünde "Name" alan isminin içinde büyük ya da küçük harflerle "TARİH" veya "DATE" kelimeleri geçmesi gerekmektedir. Aksi durumda, bu kısıtlar tarih formatında algılanmaz.

\*\*Dinamik rapor ve Parametre (Kısıt) widget ayarlarında; verilen ondalık kısıtları için "Parametreler" bölümünde "Name" alan isminin içinde büyük ya da küçük harflerle "ONDALIK" veya "DOUBLE" kelimeleri geçmesi gerekmektedir. Aksi durumda, bu kısıtlar ondalık formatında algılanmaz.

- **ColumnName** ise veritabanındaki ilgili kolon adıdır. Örnekte geçirilen CARI\_KOD parametresi parametreler kısmında CARI\_KOD olarak verilmiş olup, veritabanındaki ilgili kolon ismi de CARI\_KOD olduğundan bu şekilde kullanılmıştır.
- **Label** parametresi raporda gösterilecek olan kolon başlığıdır. Buraya dil kodu da basılabilir.
- **DefaultValue** alanına verilen değer boş gelen veriyi doldurmaya yönelik kullanılmaktadır.
- **Required** true ise o kolonun boş bırakılamayacağı anlamındadır. False değeri alırsa boş bırakılabilir.
- **DataType** kolon verisinin tipinin belirlendiği alandır. Bu alanlar ;
  - String(Metin) = 1,
  - Int(Tamsayı) = 2,
  - Date(Tarih) = 3,
  - Boolean(Doğru/Yanlış) = 4,
  - CurrentAccount (Cari)= 5,
  - ResellerCode(Plasiyer Kodu) = 6,
  - ResellerId (Plasiyer ID)= 7,
  - Inventory(Stok) = 8,
  - ProjectCode (Proje Kodu)= 9,
  - Salesman (Satıcı)= 10,
  - Invoice (Fatura)= 11,
  - WarehouseCode(Depo) = 12,
  - ReturnInvoices (İade Faturası)= 13,
  - SpecialCode1(Özel Kod1) = 14,
  - SpecialCode2 (Özel Kod2)= 15,
  - ConditionalCode (Koşul Kodu)= 16,
  - AllCurrentAccounts (Tüm Cariler)= 17,
  - Double (Ondalık) = 18

şeklinde tanımlanmıştır.

- **Operator** veritabanında yapılan aramalarda kullanılan büyüklük, küçüklük, ya da eşitlik işlemlerinin belirlendiği alandır. Bunların kullanımı ise;
  - Eşitlik=1,
  - BüyükEşit=2,
  - KüçükEşit=3,
  - İçerir=4

şeklinde tanımlanmıştır. Örnek olarak cari kodu kolonunda "12" numarası geçen carileri aramak istiyorsanız operatör alanında 4'ü kullanırsak bu işlemi gerçekleştirecektir. Örneğimizde bitiş tarih kolonunda operatör=3 kullanılmıştır. Bu tarihe eşit ve küçük olan tarihleri getirmektedir. Başlangıç tarihinde ise büyük eşit görevi olan 2 numaralı alan başlangıç tarihine eşit ve büyük tarihleri getirmektedir.

## Kolonlar

Kolonlar kısmında kolonun veritabanındaki karşılığı ve raporda görülen kolon başlık adı girilmiştir.

- **DisplayName** dil kodu olarak da verilebilmektedir.
- **ColumnType** ;
  - DataType=1,
  - Link=2,
  - ModalLink=3

Şeklinde tanımlanmıştır. Bu şekilde kolondaki veriye link verilmesi ya da modalda açılması sağlanmaktadır. Bunun için ColumnType alanına belirtilen değerlerin girilmesi yeterlidir. Kolonlara buton da eklenebilmektedir.

- **Url** link alan kolonlara ya da eklenen butonlara gideceği adresin belirtildiği alandır. Eğer adres için parametre gerekiyorsa parametre ve değer ColumnData dizisi ile yollanmaktadır.

- **ColumnData** parametre dizisidir. Url için gereken parametre "**ParamName**" ile belirlenir. Değer ise kolonlardaki ya da veritabanında hangi kolonda bulunuyorsa o kolon adını "**ColumnName**" ile belirtmek yeterlidir. Örneğin; "ColumnData": [{ "ParamName": "id", "ColumnName": "PAYMENT\_ORDER\_NO" } ]
- **IconStyle** bu butonlara görünümü değiştirmek için kullanılmaktadır.
- **IsHidden** ise "**true**" iken kolonu gizlemektedir.

## Veritabanı İşlemleri

Uyarlamaya açık Sql sorguları kullanılmaktadır. Herhangi bir veritabanı sorgusu kullanılabilir.

## Bağlı Modüller

- Özelleştirilebilir sistem sayfaları ve yeni oluşturulan tüm sayfalara eklenip çalışabilmektedir.

## Ornek 1

Örnek, cari kodu zorunlu olarak girilmekle birlikte, başlangıç ve bitiş tarihi filtreleri olarak arama yapan dinamik

rapordur. Cari kodu alanı 'DataType' alanı 5 verilmiştir. Bu da bize cari rehberi kullanmamızı sağlamaktadır.



Cari Kodu *	<input type="text" value="[12011001] BOZÜYÜK OTOMOTIV"/>	<input type="button" value="x"/>	<input type="button" value="Q"/>
Başlangıç Tarihi	<input type="text"/>	<input type="button" value="Calendar"/>	
Bitiş Tarihi	<input type="text"/>	<input type="button" value="Calendar"/>	

Rapor

Excel

Word

Pdf

Yazdır

Cari Kodu	TARİH	BELGE_NO	ACIKLAMA	HKA	BORC	ALACAK
12011001	19.01.2012 00:00:00	000000000258664	FT.MIZ 0E0000000003324	A	762,28	0,00
12011001	12.01.2012 00:00:00	000000000258456	FATURAMIZ	A	161,71	0,00
12011001	10.01.2012 00:00:00	000000000258415	FT.MIZ E00000000032659	A	366,97	0,00
12011001	09.01.2012 00:00:00	000000000258388	FATURAMIZ	A	3.337,98	0,00

**Ayarlar**

Başlık

SQL SorguTipi

StoreProcedure

SQL Sorgu

exec PRTSP\_TEST\_CARI\_HAREKET  
@BranchId,@CARI\_KOD,@FormData

Parametreler

[  
{  
"Name": "CARI\_KOD",  
"ColumnName": "CARI\_KOD",  
"Label": "3D957E9",  
"DefaultValue": "",  
"Required": true,  
}

Kolonlar

[{"ColumnName":"CARI\_KOD","DisplayName":"3D957E9"}]

NDS

2

### SQL Sorgu

exec PRTSP\_TEST\_CARI\_HAREKETCARI\_KOD,@FormData

### Parametreler

```
[{  
"Name": "CARI_KOD",  
"ColumnName": "CARI_KOD",  
"Label": "3D957E9",  
"DefaultValue": "",  
"Required": true,  
"DataType": 5,  
"Operator": 1  
},  
  
{  
"Name": "TARIH1",  
"ColumnName": "TARIH",  
"Label": "7D83F15E",
```

```

"DefaultValue": "",
"Required": false,
"DataType": 3,
"Operator": 2
},
{
"Name": "TARIH2",
"ColumnName": "TARIH",
"Label": "4574AD00",
"DefaultValue": "",
"Required": false,
"DataType": 3,
"Operator": 3
}]

```

## Kolonlar

```
[{"ColumnName":"CARI_KOD","DisplayName":"Cari Kodu"}]
```

## View

```

CREATE PROCEDURE PRTSP_TEST_CARI_HAREKET
@BranchId VARCHAR(35),
@CARI_KOD VARCHAR(35),
@FormData Xml
AS
SELECT CARI_KOD,TARIH,BELGE_NO,ACIKLAMA,HKA,BORC,ALACAK FROM DENEME_VIEW_CARIHAREKET
(NOLOCK) WHERE SUBE_KODU = @BranchId and CARI_KOD = @CARI_KOD

```

## Ornek 2

Bu örnek ise müşteri, durum, sipariş no, proje kodu, stok kodu, depo, başlangıç tarihi, bitiş tarihi filtreleri alıp buna göre raporlama yapmaktadır. Müşteri filtresi 'DataType' 5 olarak girildiği için cari rehber açılmaktadır. Aynı şekilde proje kodu, stok kodu, depo için de rehber kullanılmıştır.





Müşteri *	[12011001] BOZÜYÜK OTOMOTIV	x ▼	Q
Durum	Beklemede	▼	
Sipariş No			
Proje Kodu	Lütfen seçiniz	▼	Q
Stok Kodu	Lütfen seçiniz	▼	Q
Depo	[4] ÇİN	x ▼	Q
Başlangıç Tarihi	11.11.2015		📅
Bitiş Tarihi	11.11.2016		📅

Rapor

Excel

Word

Pdf

Yazdır

FATIRS_NO	Carı Kodu	CARI_ISIM	TARİH	GENELTOPLAM	STOK_KODU	STOK_ADI	DEPO_KODU	DURUM	KALEMDURUM
B00000000000309	12011001	BOZÜYÜK OTOMOTIV	14.03.2016 00:00:00	1.104,42	AFE5585072	ALFA ROMEO 146 1,4 16V EKSANTRİK KİT -331	4	Beklemede	1/0
B00000000000303	12011001	BOZÜYÜK OTOMOTIV	07.03.2016 00:00:00	7.092,47	KL164F	BMC TM30 FORD 900 MİN YAPRAK BALATA ARKA	4	Beklemede	301/0
B00000000000303	12011001	BOZÜYÜK OTOMOTIV	07.03.2016 00:00:00	7.092,47	MONMR829	AMORTISOR ON ( BMW : 3 SERİ E21 )	4	Beklemede	2/0
B00000000000437	12011001	BOZÜYÜK OTOMOTIV	29.09.2016 00:00:00	333,16	VAL786011	( CLASSIC ) DEBRİYAJ SETİ ( VOLKSWAGEN: POLO CLASSIC 1.6 96 ) 801439	4	Beklemede	1/0

**Ayarlar** ×

SQL SorguTipi

View

SQL Sorgu

SELECT FATIRS\_NO,CARI\_KODU,CARI\_ISIM, TARIH, GENELTOPLAM, STOK\_KODU, STOK\_ADI, DEPO\_KODU, DURUM, KALEMDURUM FROM DENEME\_DINAMIK\_RAPOR2 (NOLOCK) WHERE SUBE\_KODU = @BranchId

Parametreler


{  
"Name": "CARI\_KODU",  
"ColumnName": "CARI\_KODU",  
"Label": "Müşteri",  
"DefaultValue": "",  
"Required": true,  
"DataType": 5,  
"Operator": 1  
}


Kolonlar

[{"ColumnName":"CARI\_KODU","DisplayName":"Cari Kodu"}]

NDS

2

 Kaydet

 İptal

### SQL Sorgu

```
SELECT FATIRS_NO,CARI_KODU,CARI_ISIM, TARIH, GENELTOPLAM, STOK_KODU, STOK_ADI, DEPO_KODU, DURUM, KALEMDURUM FROM DENEME_DINAMIK_RAPOR2 (NOLOCK) WHERE SUBE_KODU = @BranchId
```

### Parametreler

```
[{  
"Name": "CARI_KODU",  
"ColumnName": "CARI_KODU",  
"Label": "Müşteri",  
"DefaultValue": "",  
"Required": true,  
"DataType": 5,  
"Operator": 1  
},
```

```
{  
"Name": "DURUM",  
"ColumnName": "DURUM",  
"Label": "Durum",  
"DefaultValue": "",
```

```
"Required": false,
"DataType": 1,
"Operator": 1,
"DataSource" : [{"Text": " Hepsi", "Value": ""}, {"Text": "Beklemede", "Value": "Beklemede"}, {"Text": "
Faturalasmis", "Value": "Faturalasmis"}, {"Text": "Kismi Yüklendi.", "Value": "Kismi Yüklendi."}]
},
{
"Name": "FATIRS_NO2",
"ColumnName": "FATIRS_NO",
"Label": "Sipariş No",
"DefaultValue": "",
"Required": false,
"DataType": 1,
"Operator": 1
},
{
"Name": "PROJE_KODU",
"ColumnName": "PROJE_KODU",
"Label": "Proje Kodu",
"DefaultValue": "",
"Required": false,
"DataType": 9,
"Operator": 1
},
{
"Name": "STOK_KODU",
"ColumnName": "STOK_KODU",
"Label": "Stok Kodu",
"DefaultValue": "",
"Required": false,
"DataType": 8,
"Operator": 1
},
{
"Name": "DEPO_KODU",
"ColumnName": "DEPO_KODU",
"Label": "Depo",
"DefaultValue": "",
"Required": false,
"DataType": 12,
"Operator": 1
},
{
"Name": "TARİH1",
"ColumnName": "TARİH",
"Label": "Başlangıç Tarihi",
"DefaultValue": "",
"Required": false,
"DataType": 3,
"Operator": 2
},
{
"Name": "TARİH2",
"ColumnName": "TARİH",
"Label": "Bitiş Tarihi",
"DefaultValue": "",
```

```
"Required": false,  
"DataType": 3,  
"Operator": 3  
}]
```

## Kolonlar

```
[{"ColumnName":"CARI_KOD","DisplayName":"Cari Kodu"}]
```

## View

```
CREATE VIEW DENEME_DINAMIK_RAPOR2  
AS  
SELECT S.SUBE_KODU, S.FATIRS_NO, S.CARI_KODU, S.GENELTOPLAM, S.TARİH TARİH, S.ODEMEGUNU, C.  
CARI_ISIM, C.PLASIYER_KODU, T.STOK_KODU, ST.STOK_ADI, S.PROJE_KODU ,T.DEPO_KODU,  
(CASE WHEN (T.STHAR_GCMİK = T.FİRMA_DOVTUT) THEN 'Faturalaşımiş'  
        WHEN (T.FİRMA_DOVTUT = 0) THEN 'Beklemede'  
        ELSE 'Kısmi Yüklendi.' END) as DURUM, CONCAT(CONVERT(varchar(20),CONVERT(int,T.  
STHAR_GCMİK)),',',CONVERT(varchar(20),CONVERT(int,T.FİRMA_DOVTUT ))) AS KALEMDURUM  
FROM TBLŞIPAMAS S (NOLOCK)  
JOIN TBLŞIPATRA T (NOLOCK) ON S.FATIRS_NO = T.FİŞNO  
JOIN TBLŞASABİT C (NOLOCK) ON S.CARI_KODU = C.CARI_KOD  
JOIN TBLŞTSABİT ST (NOLOCK) ON ST.STOK_KODU = T.STOK_KODU
```

## Örnek 3

3. örneğimiz de yine cari kodu,başlangıç ve bitiş tarihi filtreleri alıp, buna göre raporlama yapmaktadır. Son kolona ise buton eklenmiştir. Bu buton tıklanılan ödemenin detayını açmaktadır. Url kısmında adres belirtilmiştir. Butona tıklandığı zaman bu adrese gidecektir. Aldığı parametreler ise DataColumn'da verilmektedir.



Cari Kodu

[12011001] BOZÜYÜK OTOMOTIV



Başlangıç Tarihi



BitişTarihi



Rapor

Excel

Word

Pdf

Yazdır

ACCOUNT_NAME	CC_NUMBER	CC_INSTALLMENT	CC_AMOUNT	Bilgi
BOZÜYÜK OTOMOTIV	2342*****4232	1	100,00	
BOZÜYÜK OTOMOTIV	5434*****3534	1	10,60	
BOZÜYÜK OTOMOTIV	2323*****9023	1	100,00	
BOZÜYÜK OTOMOTIV	3342*****3432	1	21,20	

**Ayarlar** ×

Başlık

SQL SorguTipi

View

SQL Sorgu

SELECT ACCOUNT\_NAME,CC\_NUMBER,  
CC\_INSTALLMENT, CC\_AMOUNT, PAYMENT\_ORDER\_NO  
FROM PRTV\_PAYMENTREPORT (NOLOCK)

Parametreler


```
"Name": "ACCOUNT_CODE",  
"ColumnName": "ACCOUNT_CODE",  
"Label": "Cari Kodu",  
"DefaultValue": "",  
"Required": false,
```


Kolonlar

```
[{  
"ColumnName": "PAYMENT_ORDER_NO",  
"DisplayName": " Bilgi",  
"ColumnType": 3,
```

NDS

2

 **Kaydet**

 **İptal**

### SQL Sorgu

```
SELECT ACCOUNT_NAME,CC_NUMBER, CC_INSTALLMENT, CC_AMOUNT, PAYMENT_ORDER_NO FROM  
PRTV_PAYMENTREPORT (NOLOCK)  
WHERE REQUEST_TYPE = 'GET'
```

### Parametreler

```
[{  
"Name": "ACCOUNT_CODE",  
"ColumnName": "ACCOUNT_CODE",  
"Label": "Cari Kodu",  
"DefaultValue": "",  
"Required": false,  
"DataType": 5,  
"Operator": 1  
},
```

```
{  
"Name": "TARIH1",  
"ColumnName": "DATE",  
"Label": "Başlangıç Tarihi",  
"DefaultValue": "",
```

```

"Required": false,
"DataType": 3,
"Operator": 2
},
{
"Name": "TARIH2",
"ColumnName": "DATE",
"Label": "BitişTarihi",
"DefaultValue": "",
"Required": false,
"DataType": 3,
"Operator": 3
}]

```

## Kolonlar

```

[{
"ColumnName": "PAYMENT_ORDER_NO",
"DisplayName": " Bilgi",
"ColumnType": 3,
"ColumnData": [{
"ParamName": "id",
"ColumnName": "PAYMENT_ORDER_NO"
} ],
"Url": "../OrderFlow/Payment/PaymentDetail",
"IconStyle": "icon-info",
"LinkText": ""
}]

```

## Ornek 4

Son örneğimizde ise cari kodu girerek tablodan veri çekilmektedir. Aynı sayfaya eklediğimiz grafik rapor da bu cari kodunu prosedure un içinde yazdığımız 'FormData' parametresi ile almaktadır. Bu parametre ile aldığı cari kodla raporu dinamik raporla birlikte senkron şekilde çizdirmektedir.Örnek olarak;

```

set @CurrentAccountCode= @FormData.value('(/WidgetFormParam/DataForm/
FormElement[@name="CARI_KOD"]/@formvalue)[1]','varchar(max)')

```

"Name" alanı için gerekli kısmı "Profiler" yardımıyla doldurabilirsiniz.

Cari Kodu

[12011001] BOZÜYÜK OTOMOTIV

x ▼

Q

Rapor

Excel

Word

Pdf

Yazdır

GRAFİK RAPOR

Kayıt Bulunamadı

Rapora bastıktan sonra grafik rapor da yukarıda girilen kodu alıp buna göre raporlama yapmaktadır. Senkron bir şekilde raporlar çizilmektedirler.

Cari Kodu

[12011001] BOZÜYÜK OTOMOTIV

x ▼

Q

Rapor

Excel

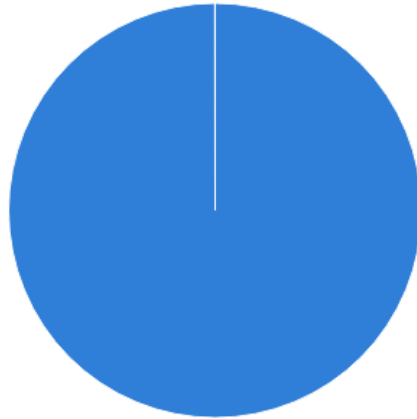
Word

Pdf

Yazdır

SUBE_KODU	ISLETME_KODU	CARI KODU	CARI_TEL	CARI_IL	ULKE_KODU	CARI_ISIM	CARI_TIP
-1	1	—	0228 3140642	BILECIK	TR	BOZÜYÜK OTOMOTIV	A

GRAFİK RAPOR



Sipariş



**Ayarlar** ×

Başlık	<input type="text" value="DİNAMİK RAPOR"/>
SQL SorguTipi	<input type="text" value="View"/>
SQL Sorgu	<input type="text" value="SELECT TOP 10 * FROM TBLCASABIT(NOLOCK) WHERE 1=1"/>
Parametreler	<input type="text" value='[{"Name":"CARI_KOD","ColumnName":"CARI_KOD","Label":"Cari"}]'/>
Kolonlar	<input type="text" value='[{"ColumnName":"CARI_KOD","DisplayName":"CARI KODU"}]'/>
NDS	<input type="text" value="2"/>

**Ayarlar** ×

Başlık	<input type="text" value="GRAFİK RAPOR"/>
Sql Procedure	<input type="text" value="PRTSP_BORA_CHART_SALES_REPORT_DAILY"/>
Grafik Tipi	<input type="text" value="Pie"/>
Başlık	<input type="text"/>
Metin	<input type="text"/>
Açıklama Görüntüle	<input checked="" type="checkbox"/>
Son Ek	<input type="text"/>
Ön Ek	<input type="text"/>
NDS	<input type="text" value="2"/>
JS Fonksiyon Adı	<input type="text"/>
Detay Link	<input type="text"/>

#### SQL Sorgu

Rapor oluşturulduğu zaman otomatik olarak gelen sorgu.

#### Parametreler

Rapor oluşturulduğu zaman otomatik olarak gelen parametreler.

## Kolonlar

Rapor oluşturulduğu zaman otomatik olarak gelen kolonlar.

Grafik raporda ise PRTSP\_BORA\_CHART\_SALES\_REPORT\_DAILY kullanılarak 'Pie' çizdirilmiştir.

## Ornek 5

### NOT:

Dinamik Rapor içerisinde Javascript sahası (JS) desteklenmiştir.

Bu JS 'Rapor' butonuna tıklandığında çalışır.

### ÖR:

```
$(document).ready(function ()
{
var StokDurum = 0.0;
$(".item-row").each(function( index ) {

var newval = parseFloat($( $( this ).find("td")[5]).text().replace('.', '').replace(',', '.'));
if(newval < 0.0){ $( $( this ).find("td")[5]).html("<div style='color:red;'><b>" +$( $( this ).find("td")[5]).text()
+"</b></div>"); }
if(newval == 0.0){ $( $( this ).find("td")[5]).html("<div style='color:#E07F2C;'><b>" +$( $( this ).find("td")
[5]).text()+"</b></div>"); }
if(newval > 0.0){ $( $( this ).find("td")[5]).html("<div style='color:green;'><b>" +$( $( this ).find("td")[5]).
text()+"</b></div>"); }

});

});
```