## POWER BI NOTLARI

1-Hesaplama yapabilmek icin üst kısımdaki modellemeye basıp ondan sonra yeni ölcü diyerek ekran acılır.

Row Context=DAX'ta row context yaratan iki şey var:  iterator fonksiyonlar  (sonu X ile bitenler + FILTER gibi komutlar) ve  hesaplanmış sütunlar .

Row context en basit tanımıyla bulunduğumuz satır demek, hangi satırda olduğumuzun farkında olmak demek.

Filter Context=Dilimleyicide yapılan her seçim, matriste satıra veya sütuna koyduğumuz renk ve yıl, tabular engine'nin gördüğü veri kümesini hep filtreliyor. Bu yüzden de bu konseptin adı: **Filter Context**.

a)Sum()=Toplama işlemi yapar.

Acıklaması: "Satışlar" tablosundaki "Tutar" sütununu bul,  bu sütunda gördüğün  tüm rakamları topla.

b) SUMX ('İterate Edilecek Tablo'; İlgili tablonun her bir satırı için hesaplanacak deyim )=

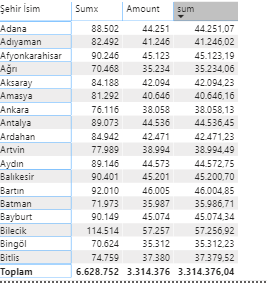
-Tüm Satırda Toplama işlemi yapar.

Örn: Sum = SUM('Order'[Amount])

Örn: Sumx = SUMX('Order';'Order'[Amount]+'Order'[Amount])

Acıklama=İlk kısımda Tablonun ismi,ikinci kısımda ise işlem her satır için tekrarlanır.

Çıktısı=



# Sayma Fonksiyonları

"Satışlar" tablosuna git, bu tabloda  gördüğün tüm satırlardaki  "Fiyat" ve "Miktar" sütunlarını her bir satır için tek tek çarp, sonra hepsini topla.

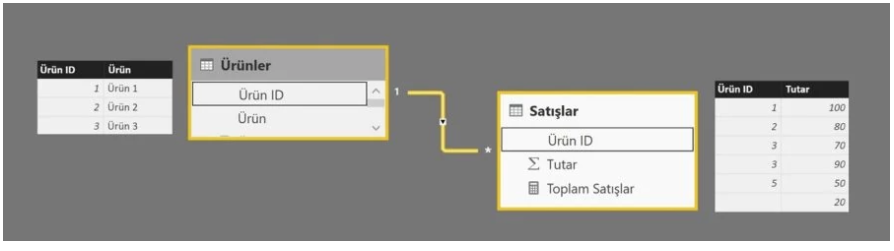
a)COUNTROWS (Tablo) : Verilen tabloda gördüğü satır sayısını döndürür.

b)CountA()=Bir sütundaki boş olmayan hücre sayısını sayar.

c)Count()=COUNT işlevi, bir sütundaki boş olmayan değerleride içeren hücre sayısını sayar.

d)DiscountCount()=Bir sütundaki farklı değerlerin sayısını sayar.

e)COUNTBLANK: Sütunda gördüğü "boş" değerleri sayar.



Örnler:

(1)CountRows ÜRÜNLER := COUNTROWS('Ürünler')

(2) CountRows SATIŞLAR := COUNTROWS('Satışlar')

(3) CountBlank ÜRÜNLER := COUNTBLANK('Ürünler'[Ürün ID])

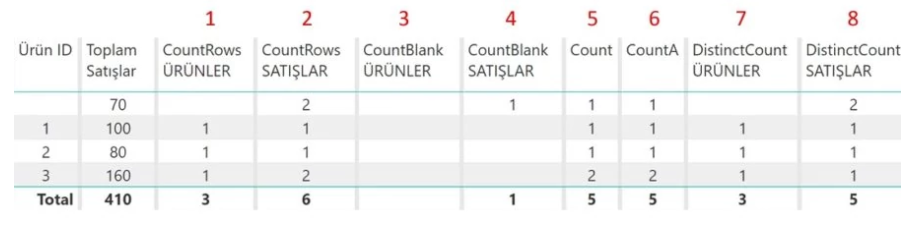
(4) CountBlank SATIŞLAR := COUNTBLANK('Satışlar'[Ürün ID])

(5) Count := COUNT('Satışlar'[Ürün ID])//Yukarıda göründügü gibi satıslar tablosunda

6 satır olmasına ragmen bir hücre boş oldugu için 5 adet yazdı.

(6) CountA := COUNTA('Satışlar'[Ürün ID])

 (7) DistinctCount ÜRÜNLER := DISTINCTCOUNT('Ürünler'[Ürün ID])

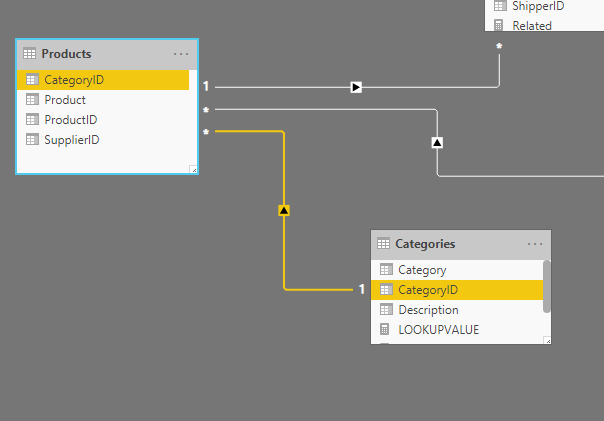
(8) DistinctCount SATIŞLAR := DISTINCTCOUNT('Satışlar'[Ürün ID]) //tekil olan boşla birlikte aldı. 

# RELATED(vLookup) ve RELATEDTABLE

RELATED fonksiyonu, "many" yani Çok tablo tarafındaki tabloda satır bazlı bir işlem yaparken, ilgili satır için ilişkinin "one" tarafındaki kaydı bulur ve bu kaydın sütun değerlerine erişmemizi sağlar.Tablolar arasındaki ilişkilerin kurulmasını saglar.Related Tablolarda tabloların ilişkili olması gerekir.İlişkiler birebir ilişki yada Bire Çok ilişkide Çok olan tablodan işlem yapılır.

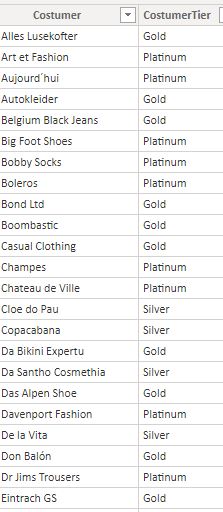
Aşagıda CostumerMapping Tablosunun içindeki Costumer ve CostumerTier Sutunları

bulunuyor.



Related tablolar dedigimiz gibi Çok olan tabloda işlem yapar.Tabloda gördügünüz yıldız işareti(\*)Çok’u temsil eder. Yani Bir Ürünün birden fazla kategorisi olabilir, anlamına gelir.

Related tablodada Product Tablosundan related fonksiyonunu çagırırız.



Buradaki amacımız ise CostumerMapping Tablosunun içindeki CostumerTier Sütunundaki değerleri

Aşağıdaki Customer Tablosundaki Sütunda Göstermektir.

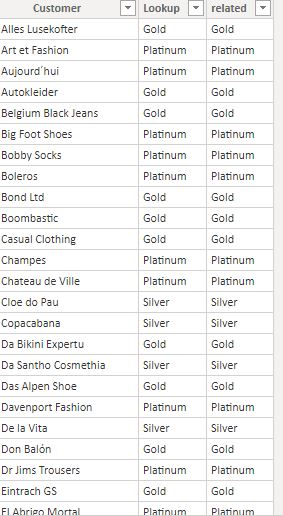


Yazgımız Kod ise



Son aldıgı deger bu şekildedir.

Related=RELATED(CostumerMapping[CostumerTier]) Şeklindedir.



LOOKUPVALUE KULLANIMI

Related fonksiyonuyla aynı şekilde çalışır.Farkı ise tabloların ilişkili olması gerekmiyor.

Kodu aşagıdaki gibidir;

Lookup = LOOKUPVALUE(CostumerMapping[CostumerTier];CostumerMapping[Costumer];Costumer[Customer])

----------------------------------------------------------------------------------------

Açıklama=1.parametrede Getirecegimiz tablodaki sutunun degerleri,2. Parametrede ise CostumerMapping tablosudaki ilişki kuracagımız Costumer sütunu,3.parametre ise Costumer tablosundaki ilişki kurulacak olan Costumer Sutunudur.

--------------------------------------------------------------------------------------------------------------------

e)CALCULATETABLE(<expression>,<filter1>,<filter2>,…)=Filtreleme işlemi yapar.

örn:=SUMX( CALCULATETABLE('InternetSales\_USD', 'DateTime'[CalendarYear]=2006) , [SalesAmount\_USD])

previous=önceki gün,ay,yıl

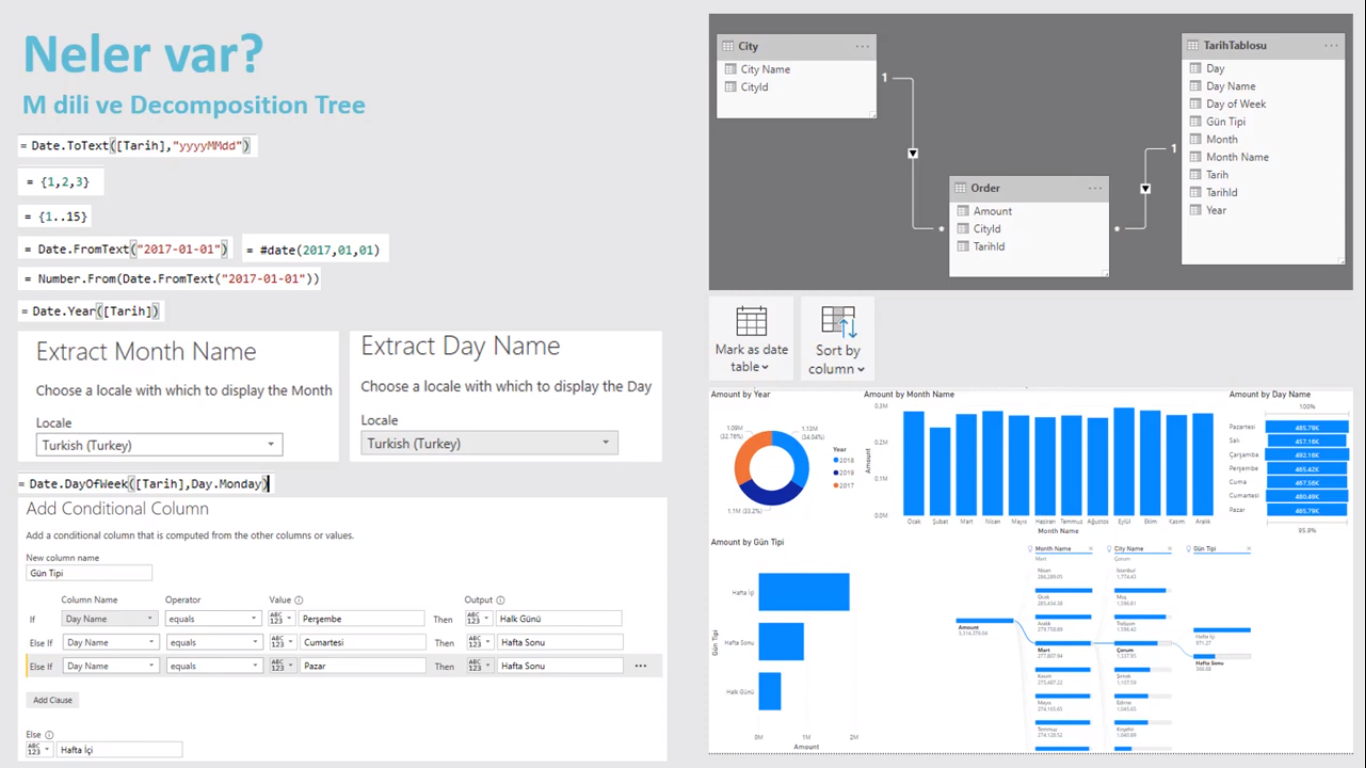
next=sonraki gün,ay,Yıl

COUNTROWS=Toplam Satır Sayısını ifade eder.

Örn: Siyah Ürün Sayısı:= COUNTROWS(FILTER('Ürünler'; 'Ürünler'[Renk] = "Siyah"))

Acıklama=Ürünler tablosunda Renk sutunundaki siyah satır sayısını gösterir.

Time İntellegent 1 Demo



Date.ToText([Tarih],”yyy,MM,ddd”) tarih formatını sayı formatına dönüştürmek için kullanılır.

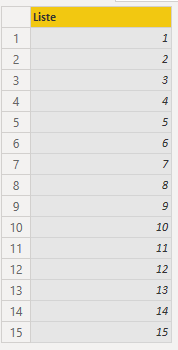
Amacımız ayrı bir tarih tablosu açıp Order tablosuyla ilişkilendirmek.

Sorgu tablosuna girmek için Giriş sekmesinden veri dönüştürme sekmesine tıklıyoruz.

Sorgu sayfası açıldıktan sonra sol tarafta tabloların olduğu kısmı görünecek.Oradaki boş kısmına sağ tıklayıp yeni sorgu dedikten sonra Boş sorgu kısmına tıklıyoruz.

Sonrasında list kısmından bahsedeceğim.

Örn:List kısmını yazmak için {1..15}dedigimiz zaman 1 den 15 e kadar olan ardısık sayıları getirir.



Çıktısı Liste halinde gözükecektir.

..=iki nokta işareti ise baslangıctan sona kadar tüm rakam,sayı,Tarihi list formatında gösterir.

Date.FromText()=Bu fonksiyon Tarih formatına çevirir.

{

Number.From(Date.FromText("2017-01-01"))

..

Number.From(Date.FromText("2019-12-31"))

}

Koşullu ifade yazmak için başlangıç olarak analiz kısmındaki Giriş sekmesine tıklayıp oradan sonra verileri dönüstürme sekmesine tıklanır. Sonrasında ise Sütun ekle sekmesine tıkladıktan sonra Koşullu Sütun sekmesine basarız.

Tabloları Birlestirmek için ise yukarıdaki adımlar uygulanır.Farklı olarak Sütun ekle sekmesi değilde

Dönüştür sekmesine tıklanır.Sonrasında Sütun Birleştir sekmesine tıklanır.

SET ANALİZLERİ

Toplam Tutar = SUM('Order'[Amount])—Order tablosundaki Toplam Tutar

-----------------------------------------------

Ortalama = AVERAGE('Order'[Amount]) –Order tablosundaki Ortalama

-----------------------------------------------

Bölme İşlemi=Divide(4;2) Bölme işlemi sonucunda 2 sonucunu verecektir.

------------------------------------------------

2017 yılına ait Toplam adet = CALCULATE([Toplam Adet];Tarih[Yıl]=2017)

Kasım ve Aralık Toplam Miktar = CALCULATE([Toplam Adet];'Order'[Ay]="Kasım" || 'Order'[Ay]="Aralık")

-------------------------------------------------

İstanbul ve Ankara ve İzmir 2017 ve 2018 Toplam Satışları = CALCULATE([Toplam Adet];'Order'[Yıl]=2017, || 'Order'[Yıl]=2018; City[City Name] IN {"İstanbul";"Ankara"; "İzmir"})

-------------------------------------------------

SamePeriodLastYear= CALCULATE([Toplam Adet];SAMEPERIODLASTYEAR(Takvim[Tarih]) )

--------------------------------------------------

DATEADD(dates, -1, year)=SamePeriodLastYear aynı şekilde Çalışır.

---------------------------------------------------

PreviousDay=Önceki Yılı Gösterir.

---------------------------------------------------

SamePeriodLastYear=Previous yeardan farkı önceki yıllardaki tüm toplamı gösterir.

---------------------------------------------------

TotalMTD=(Kümülatif Toplam) 2019 yılının Son Ayındaki Toplam Adeti Gösterir. = TOTALMTD([Toplam Adet];Takvim[Tarih])

TOTALYTD=2019 yılındaki Toplam adeti Döndürür.Yani Son Yıldaki Toplamı Döndürür.

DATESINPERİOD=Son 7 günün,Haftanın,Ayın Toplamını getirir.

CALCULATE([ToplamAdet];DATESINPERIOD(Takvim[Tarih];Max(Takvim[Gün]));-7;DAY )

1. parametre tablonun tarihi 2.parametre döndürecek gün tablosu,3.parametre ise -7 yani max tarihten 7 gün öncesinden en son tarihe kadar olan toplam degerini getirir.

--------------------------------------------------

Aylık Kümülatif = CALCULATE([Satışlar] ; DATESMTD('Tarih'[Tarih]))

--------------------------------------------------------------------------------------------

Yıllık Kümülatif= CALCULATE([Satışlar] ; DATESYTD('Tarih'[Tarih]))

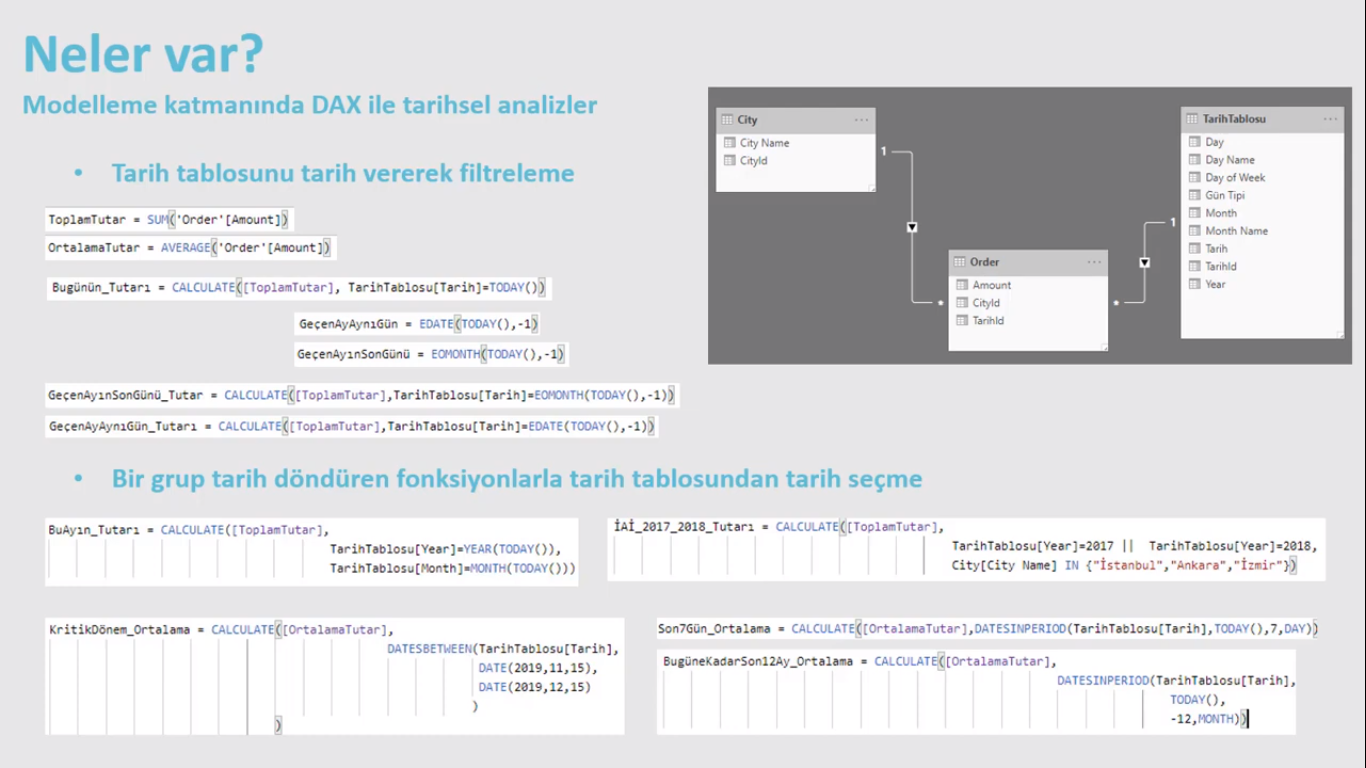
--------------------------------------------------------------------------------------------

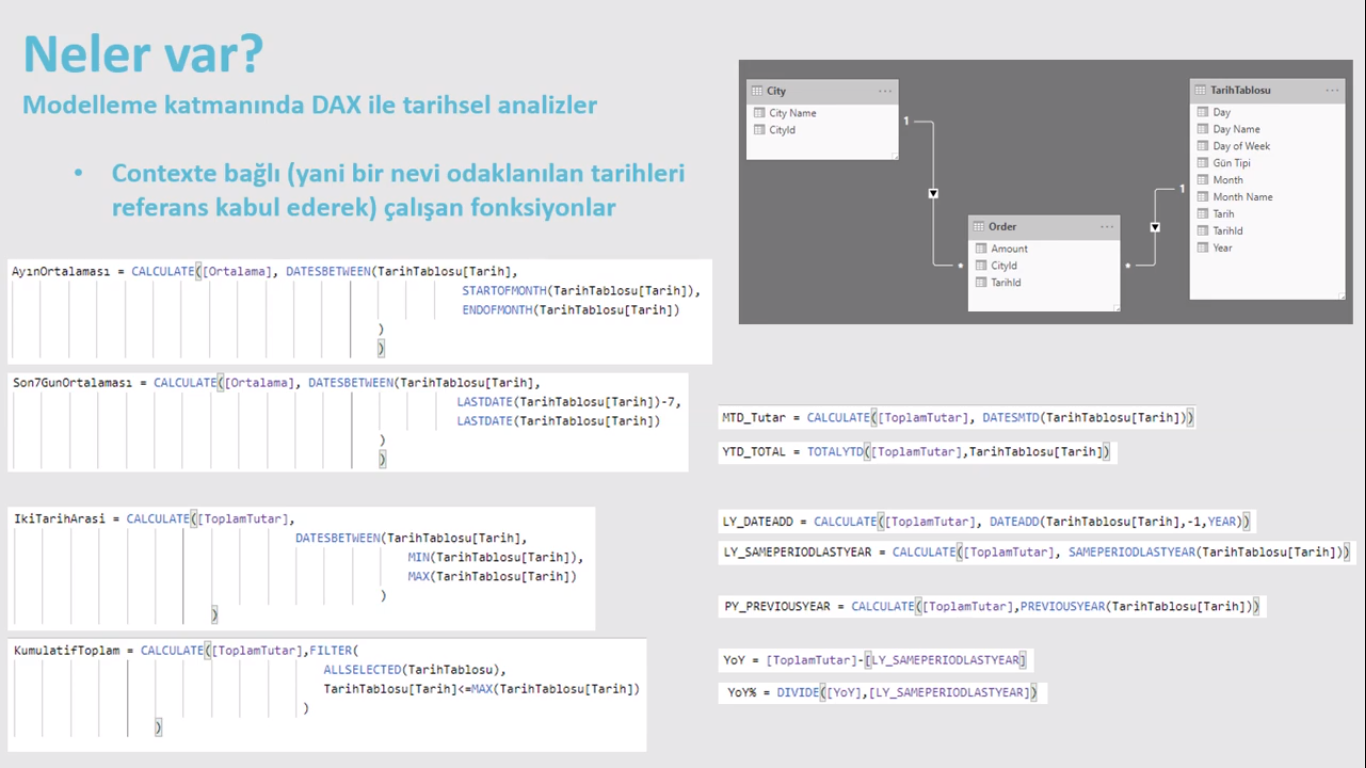
2019 Yılındaki Toplam = TOTALYTD([Toplam Adet];Takvim[Tarih])

---------------------------------------------------

ALL = CALCULATE([Toplam Adet];ALL(Takvim[Tarih];Takvim[Ay];Takvim[Yıl])) All fonksiyonunda

Takvimin içindeki yıl ay ve tarihten birini seçtigimizde KPI etkilenmez.Diğer filtrelerden etkilenir.





SWİTCH KULLANIM:

SWITCH   (deyim;

         değer  ; sonuç ;

         değer  ; sonuç ;

         ...         ;

         hiçbiri değilse sonuç)

Açıklama=Tek bir deyimi  (expression)  farklı değerler altında test edip, hangi "condition" doğru ise (true) ise o duruma ait sonucu döndürmemizi sağlıyor.

Bütçe Değerlendirme :=

SWITCH (

    TRUE();

    [Realizasyon %] >= 0,8; "İyi";

    [Realizasyon %] >= 0,7; "Orta";

    [Realizasyon %] >= 0,6; "Kötü";

    "Çok Kötü"

Yada İF’le kullanılır.

Bütçe Değerlendirme :=

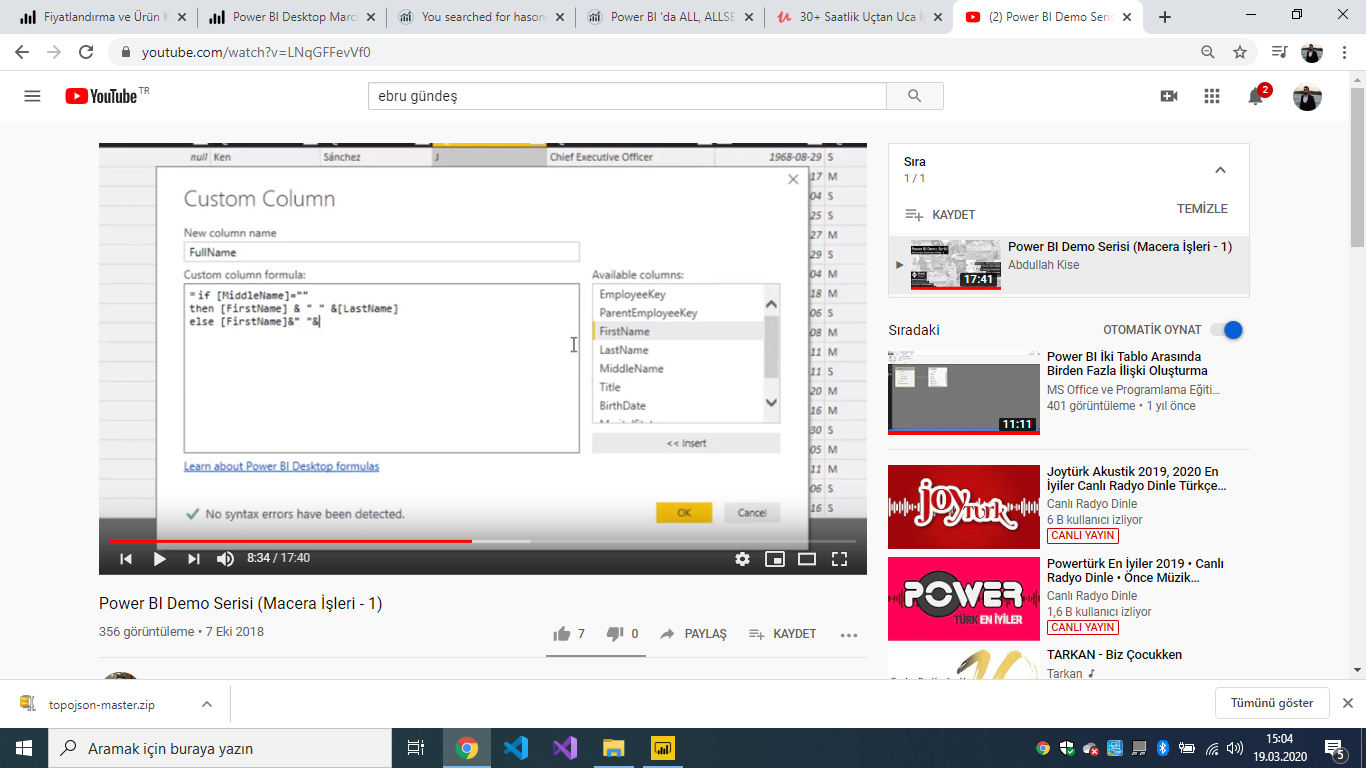
IF ([Realizasyon %] >= 0,8; "İyi";

   IF ( [Realizasyon %] >= 0,7; "Orta";

      IF ( [Realizasyon %] >= 0,6; "Kötü";

                                "Çok Kötü" ) ))

Diger if else durumu iki string ifadeyi birlestirmek icin kullanılır.



Bütçe Durum :=

SWITCH (

    TRUE();

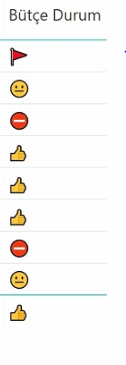
    [Realizasyon %] >= 0,8; UNICHAR(128077);

    [Realizasyon %] >= 0,7; UNICHAR(128528);

    [Realizasyon %] >= 0,6; UNICHAR(128681);

    UNICHAR(9940)

)



Yıldız :=

REPT ( UNICHAR ( 9733 ); [Realizasyon %] \* 5 ) realizasyon =Yüzdelik oran

    & REPT ( UNICHAR ( 9734 ); 10 - [Realizasyon %] \* 5 )

Rept(İlk Parametre,ikinci Parametre)

Rept=ilk parametresinde verilen karakteri, ikinci parametresinde bulduğu sayı kadar tekrarlayan bir fonksiyondur.

SELECTEDVALUE:

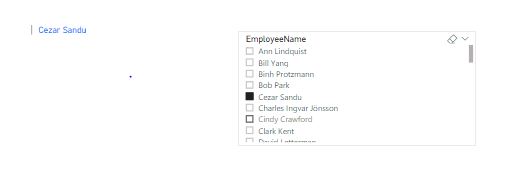
Kullanıcının hangi değeri seçtiğini bulmak için "SELECTEDVALUE" fonksiyonunu kullanabiliriz.

Not: Kullanıcı "tablodaki sütundan" tek bir değer seçerse seçilen değeri döndürür

  Birden fazla seçim yapıldıysa veya hiç değer seçimi yapılmadıysa  ya "blank" yani boş, döndürür ya da eğer girildiyse varsayılan değeri döndürür.

Örn: SelectedValue = SELECTEDVALUE(Employees[EmployeeName])

Açıklama=Burada Filtre bölmesinden Cesar Sanduya tıkladıgımızda sol taraftaki tabloda secilen degeri gösterir.



UNION İŞLEMİ

Tüm Müşteriler :=

UNION (

    DISTINCT ( 'Tablo 1'[Müşteri Kodu] );

    DISTINCT ( 'Tablo 2'[Müşteri Kodu] );

    DISTINCT ( 'Tablo 3'[Müşteri] )

)

Değişken Kullanımı

Degiskenler icin var degiskeni kullanılır.

Var SUM(),Count(),AVG() fonksiyonları yazılır.Sonrasında

Return,

Calculate() fonksiyonuyla birlikte devam eder.

FILTER KULLANIMI

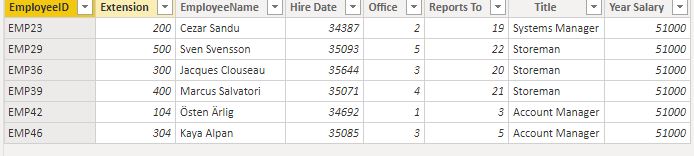
FILTER3 = FILTER(Employees;Employees[Year Salary]=51000 && Employees[EmployeeName]="Cezar Sandu")

//Filtrelenmiş tablo döndürür.sqldeki where gibi düsünebiliriz. Sanal tablolardır.

// Ve(&&)= komutunda ise maaşı 51000 ve employeeName Cezar Sandu olan tek kayıt getirdi.

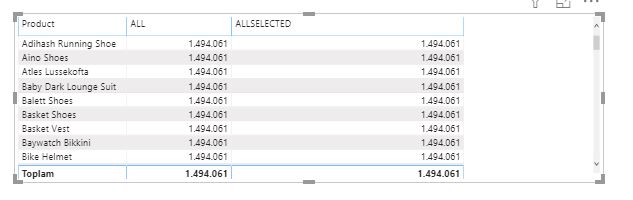


//veya(||)= Dedigimizde yıllık maasları 51000 olanları ve EmployeeName sütunundaki Cezar Sandu’yu getirdi



ALL VE ALLSELECTED FONKSİYONLARI

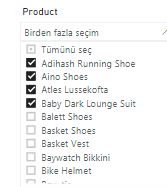
Filtreleme yapılmadan önce All ve Allselected tabloları aynı sekilde göstermektedir.



ALL Fonksiyonu

ALL(Tablo İsmi veya Sütun İsmi) = CALCULATE(Orders[Ürünün Maliyeti];ALL(Products))

Aralarındaki fark ALL fonksiyonunda tüm filtreleri yok sayarak toplam değeri o sütunun tüm satırlarında göstermesidir.Yüzdelik pay için kullanılır.

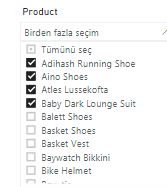


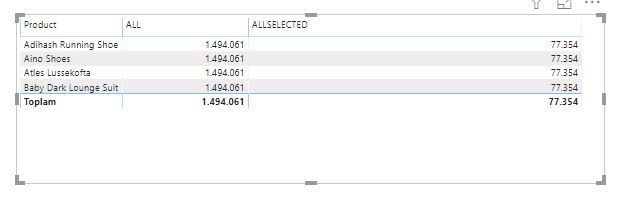
ALLSELECTED Fonksiyonu(Tablo İsmi veya Sütun İsmi)

ALL fonksiyonuyla AllSelected fonksiyonundan fark olarak

ALL fonksiyonunda filtreleri yok sayarak toplam ürün sayısını gösterecektir. ALLSELECTED fonksiyonu ise filtrelerde seçime göre ALL fonksiyonu gibi çalışır.

ALLSELECTED = CALCULATE(Orders[Ürünün Maliyeti];ALLSELECTED(Products))





Açıklama=Yukarıdaki seçimlerde ALL fonksiyonu hiçbir filtreye aldırış etmeden tüm ürün sayısını gösterirken,ALLSelected fonksiyonu ise filtrelemeleri aldırış eder,onlara ALL fonksiyonu gibi davranır.

SUMMARİZECOLOWNS FONKSİYONU

SQL deki groupby fonksiyonuna benzer çalışır.

Örn:

SummarizeColumns fonksiyonu =

**Kullanım= SUMMARIZECOLUMNS(Gruplayacagımız tablonun ismi;eklecegimiz tablo1;tablo2;tablo3…;”yeni hesaplanacak olan tablonun İsmi”;hesaplanacak ifade)**

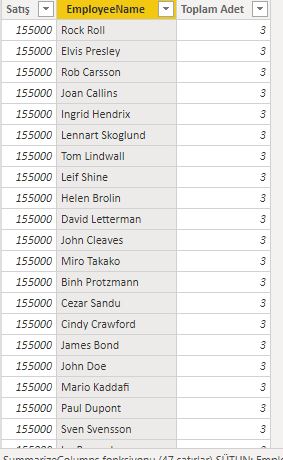
Açıklama=İlk parametre neye göre gruplama yapılacagı,En sonda yeni hesaplayacagımız tablonun ismi ve hesaplama olan yeni bir tablo dönrürecektir.Sanal tablolardır.

Örn:

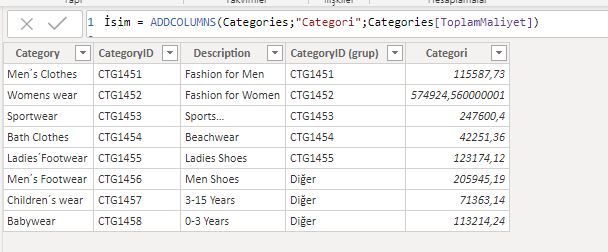
SummarizeColumns fonksiyonu = SUMMARIZECOLUMNS(Employees[EmployeeName];"Satış";[Toplam Maaş])

Açıklama=Yukarıdaki ifadede ise Çalışanİsimlerine

Gruplama yapar ve maaşlarıda ona göre hesaplama yapılır.



ADD COLUMNS



Addcolumns sanal bir tablodur.İstedigimiz tablonun tamamını döndürür.Sonrasında ise 2.parametre olarakta yine o tablodan yeni bir tablo daha bekler."tablonun ismi";istedigimiz tablonun ismini yazarız. Addcolumns’ta groupby gibi çalışır.

SUMMARİZE İLE ADDCOLUMNS FARKLARI

SUMMARİZECOLUMNS'tan farklı olarak ADDCOLUMNS tüm tabloyu döndürür.SummarizeColumns ise istedigimiz tabloları döndürür.

İF ELSE KOMUTU:

Her programlama dilinde oldugu gibi dax dilindede if else bulunur.

IF ( <LogicalTest>, <ResultIfTrue> )

Açıklama=İf koşulu parantezle koşul yazılır koşul sağlanırsa ilk parametre döner.Virgülden sonraki(,)

Deger ise ilk deger olmazsa öteki sonucu döndür.

Not()

Degil yani haricinde olarak kullanılabilir.

2 türlü yazılabilir;

1-Siyah haricindeki Renkler ALL Olan Not in = CALCULATE([Toplam Adet];FILTER(ALL(Cihaz[Cihaz Renk]);NOT(Cihaz[Cihaz Renk] in {"Siyah"})))

2-Siyah haricindeki Renkler ALL Olan NOT in = CALCULATE([Toplam Adet];FILTER(ALL(Cihaz[Cihaz Renk]);NOT(Cihaz[Cihaz Renk] in {"Siyah"})))

Açıklama=İki formülde aynı çıktıyı verecektir.

Burada anlatılan şey ise calculate fonksiyonunu yazarız.Fılter fonksiyonuyla ALL fonksiyonu oldugu için hangi seçim yaparsa yapsın hiçbir seçime riayet etmicek.Bütün satırlarda cihaz rengi siyah olmayan renklerin sayısını gösterir.

//Dilimleyici filtremede CihazRenk filtrelemesine riayet etmez.Ama marka veya firma filtrelemesi yaptıgımızda rakamlar degisiklik gösterecektir.

Value=Bir sayıyı temsil eden metin dizisini sayıya dönüştürür.

Values Fonksiyonu

Aynı ALL() gibi bir tablo fonksiyonu: Yani döndürdüğü şey bir tablo. Dolayısıyla bizden parametre olarak bir tablo isteyen diğer fonksiyonlarla birlikte kullanılabilir.

Kullanımı: Values(Tablo adı veya Sütun Adı)

Hem ALL( Sütun Adı ) hem de VALUES( Sütun Adı ), bize, verdiğimiz sütun neyse bu sütundaki  tekil değerleri içeren, tek sütunluk bir tablo döndürür .

ALL VE VALUES İN FARKI:

Aradaki en büyük fark ise -bir metrik içerisinde kullanıldığında ALL 'un  context gözetmeksizin  verdiğimiz sütundaki tüm tekil değerleri döndürmesi,

-VALUES 'un ise  filter context'te gördüğü tekil değerleri  döndürmesidir.

VALUES’IN TABLO OLARAK KULLANIMI:

Values fonksiyonu Distinct fonksiyonuna benzer.Values fonksiyonu tekil yani benzersiz(unique) yani tekrarsız(yineleme) degerleri döndürür.

Distinct ile Valuesin Farkı;

Distinct ile yaparken boş(blank) degerleri saymaz.Values fonksiyonu ise boş(Null) degerleride sayar.

ALL ve ALLNOBLANKROW Farkı;

ALL fonksiyonunda boş değerleride sayar.ALLNOBLANKROW ise sütunun içindeki boş satırları saymaz.All fonksiyonu gibi davranır.