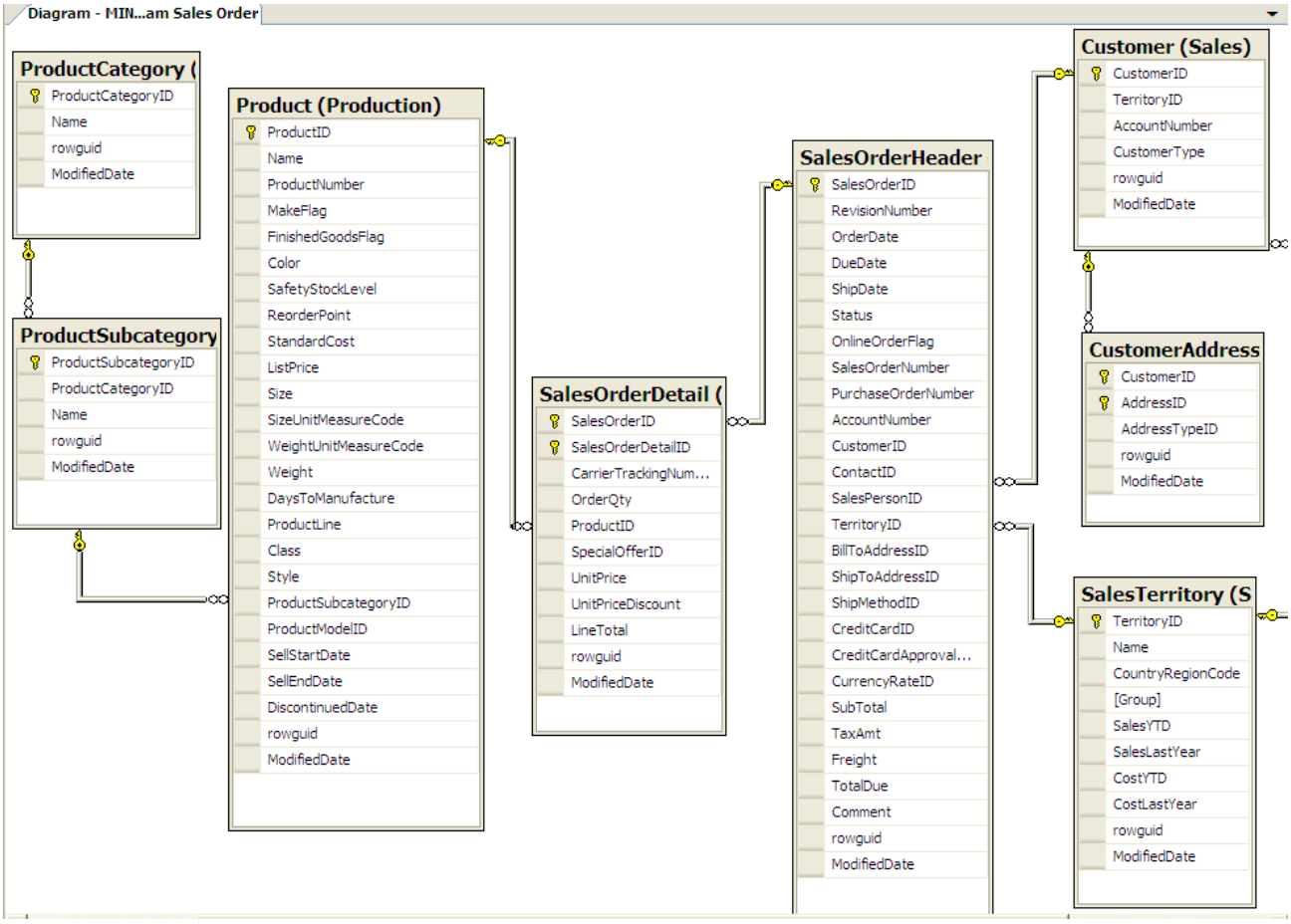
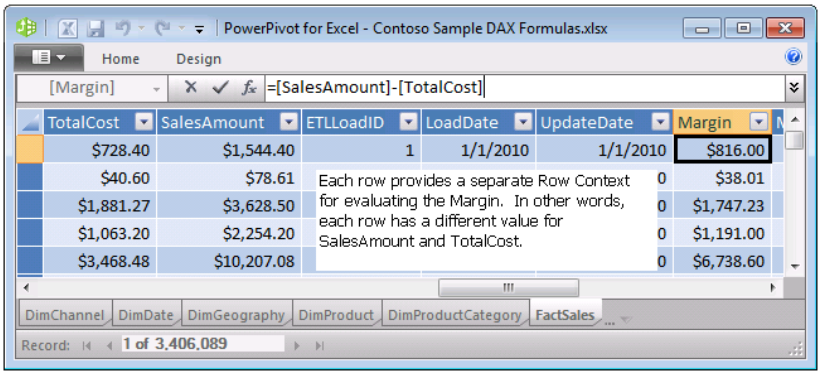
QuickReference PowerPivot DAX (Data Analysis eXpressions)



Calculated Columns

Row Context




```
= [Quantity] * [UnitPrice]
= [FirstName] & " " & [LastName]

= RELATED(Subcategory[Name])
= LOOKUPVALUE(Product[StockLevel], [ProductName], "Mountain-400-W Silver, 46")

= SUMX(
              RELATEDTABLE(SalesOrderDetail),
              SalesOrderDetail[UnitPrice] * SalesOrderDetail[Quantity])

= YEAR([OrderDate])
= MONTH([OrderDate])
= FORMAT([OrderDate],"mmm dd yyyy ddd hh:mm")
```

Operators

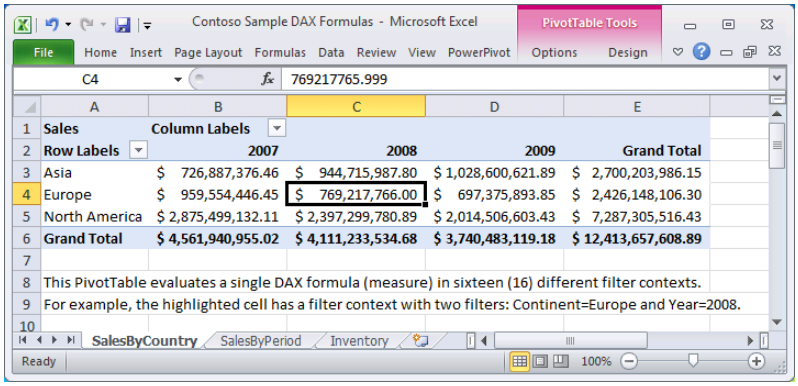
```
  AND              &&
  OR               ||
  CONCATENATE       &
```

```
= IF([EnglishDayNameOfWeek] = "Saturday"
              ||
     [EnglishDayNameOfWeek] = "Sunday", "Weekend","Working Day")

= IF( RELATED(ProductSubcategory[Name]) = BLANK(), "Unknown", RELATED(ProductSubcategory[Name]))
```

Measures

Filter Context




```
SalesTotal              := SUM( SalesOrderDetail[LineTotal] )
SalesTotal2             := SUMX( SalesOrderDetail ,    [OrderQty] *  [UnitPrice]    )

HighestPrice            := MAX(SalesOrderDetail[UnitPrice])
LowestPrice             := MIN(SalesOrderDetail[UnitPrice])
NumberOfSubCat          := COUNT(ProductSubcategory[ProductSubcategoryID])
NumberOfRows            := COUNTROWS(SalesOrderDetail)


CALCULATE( <expression>, <filter1>, <filter2>,…)

USASales                := CALCULATE( [SalesTotal], Region[Country] = "USA")
(All Filters on Region table stay active)

USASales                := CALCULATE( [SalesTotal], ALL(Region[Country]), Region[Country] = "USA")
(Reset Filter Context on Country column, all other filters stay active f.e.: Province, City!)

USASales                := CALCULATE( [SalesTotal], ALL(Region) , Region[Country] = "USA")
(Reset Filter Context on all columns in Region table)

HighMarginSales         := CALCULATE( [SalesTotal], FILTER(Product, Product[UnitPrice] >= Product[UnitCost]*2))
(User Filter function to fiter on two columns)

HighMarginSales         := CALCULATE( [SalesTotal], FILTER(ALL(Product),
(Reset all Product filters)                                 Product[UnitPrice] >= Product[UnitCost]*2))
```

VALUES

```
CountActiveSubCat       := COUNTROWS( VALUES(Product[SubCat]) )
SubCatIsActive          := ISFILTERED(Product[SubCat])

PercSalesOfCatSales     := [SalesTotal] / CALCULATE( [SalesTotal], ALL(Product) ,  VALUES(Product[Cat]) )
       ""               := [SalesTotal] / [SalesTotal]( ALL(Product) ,  VALUES(Product[Cat]) )
       ""               := DIVIDE([SalesTotal] , [SalesTotal]( ALL(Product) ,  VALUES(Product[Cat]) ) )
(DIVIDE: Performs division and returns alternate result or BLANK() on division by 0.)

PercSalesOfCatSalesTot:= IF([SubCatIsActive], [PercSalesOfCatSales} , BLANK())
```

USERELATIONSHIP

```
Total value Shipped     := CALCULATE( SUMX('OrderPipeline', [Total value] ) ,
                                      USERELATIONSHIP( OrderPipeline[BookDate], DimDate[Date] )      )
```

Time Intelligence

```
PreviousMonthSales      := CALCULATE(    [SalesTotal],
                                         All(SalesOrderHeader[OrderDate]),
                                         PREVIOUSMONTH( VALUES(SalesOrderHeader[OrderDate] ) ) )

PreviousYearMonthSales:= CALCULATE(    [SalesTotal],
                                         All(SalesOrderHeader[OrderDate]),
                                         PARALLELPERIOD( VALUES(SalesOrderHeader[OrderDate] ), -12, MONTH ) )

YTDSales                := CALCULATE(    [SalesTotal],
                                         All(SalesOrderHeader[OrderDate]),
                                         DATESYTD( VALUES(SalesOrderHeader[OrderDate] ) ) )

   "                    := TOTALYTD(     [SalesTotal],
                                         All(SalesOrderHeader[OrderDate]),
                                         VALUES(SalesOrderHeader[OrderDate] ) )

YTDPreviousYearSales    := CALCULATE(    [SalesTotal],
                                         All(SalesOrderHeader[OrderDate]),
                                         SAMEPERIODLASTYEAR(
                                                   DATESYTD( VALUES(SalesOrderHeader[OrderDate] ) ) ) )
```

CALCULATETABLE: Evaluates a table expression in a context modified by the given filters.

```
CALCULATETABLE(<expression>,<filter1>,<filter2>,…)
```

```
AvgSubCatSales        := AVERAGEX( CALCULATETABLE(
                              DISTINCT(Product[SubCat]), ALL(Product) , VALUES(Product[Cat] ) )   , [SalesTotal] )
```

SUMMARIZE: Returns a summary table for the requested totals over a set of groups.

```
SUMMARIZE(<table>, <groupBy_columnName>[, <groupBy_columnName>]…[, <name>, <expression>]…)
```

```
SUMMARIZE(ResellerSales_USD
      , DateTime[CalendarYear]
      , ProductCategory[ProductCategoryName]
      , "Sales Amount (USD)", SUM(ResellerSales_USD[SalesAmount_USD])
      , "Discount Amount (USD)", SUM(ResellerSales_USD[DiscountAmount])
      )
```

DateTime[CalendarYear]	ProductCategory[ProductCategoryName]	[Sales Amount (USD)]	[Discount Amount (USD)]
2008	Bikes	12968255.42	36167.6592
2005	Bikes	6958251.043	4231.1621
2006	Bikes	18901351.08	178175.8399
2007	Bikes	24256817.5	276065.992
2008	Components	2008052.706	39.9266
2005	Components	574256.9865	0
2006	Components	3428213.05	948.7674
2007	Components	5195315.216	4226.0444
2008	Clothing	366507.844	4151.1235
2005	Clothing	31851.1628	90.9593

```
MA3Months := AVERAGEX(
                SUMMARIZE(
                        CALCULATETABLE(
                              SalesOrderHeader ,
                              ALL(SalesOrderHeader),
                              DATESBETWEEN(
                                 SalesOrderHeader[OrderDate]    ,
                                 FIRSTDATE( DATEADD( VALUES(SalesOrderHeader[OrderDate]) ,-2, MONTH ) ),
                                 LASTDATE( VALUES( SalesOrderHeader[OrderDate])) )
                               ) ,
                        SalesOrderHeader[Jaar],
                        SalesOrderHeader[Maand],
                        "SalesPerMonth",
                        SUM(SalesOrderDetail[LineTotal])
                        ),
                    [SalesPerMonth]
            )
```

Role Based Security:

- **Row Filters**

Table	DAX expression
Region	= Region[Country] = "USA"
ProductCategory	= ProductCategory[Name] = "Bicycles"
Transactions	= Transactions[Year] = 2008

- **Dynamic Filters**

```
LOOKUPVALUE(<result_columnName>,
            <search_columnName>, <search_value>[,
            <search_columnName>, <search_value>]…)
```

1 to many

```
=       dimDepartment[DepartmentId] = LOOKUPVALUE( dimEmployees[DepartmentId],
dimEmployees[LoginId], USERNAME() )
```

dimDepartment

DepartmentId	DepartmentName
1	Corporate
2	Executive General and Administration
3	Inventory Management
4	Manufacturing

dimEmployees

LastName	FirstName	LoginID	DepartmentId
Brown	Kevin	Adventure-works\kevin0	7
Bradley	David	Adventure-works\david0	7
Dobney	JoLynn	Adventure-works\JoLynn0	4
Baretto DeMattos	Paula	Adventure-works\Paula0	2

many to many

```
=       ProductCategory[ProductCategoryID]
        =
        LOOKUPVALUE( SecurityEmployeeCategory[CategoryID],
              SecurityEmployeeCategory[CategoryID], ProductCategory[ProductCategoryID],
              SecurityEmployeeCategory[Login], USERNAME())
```