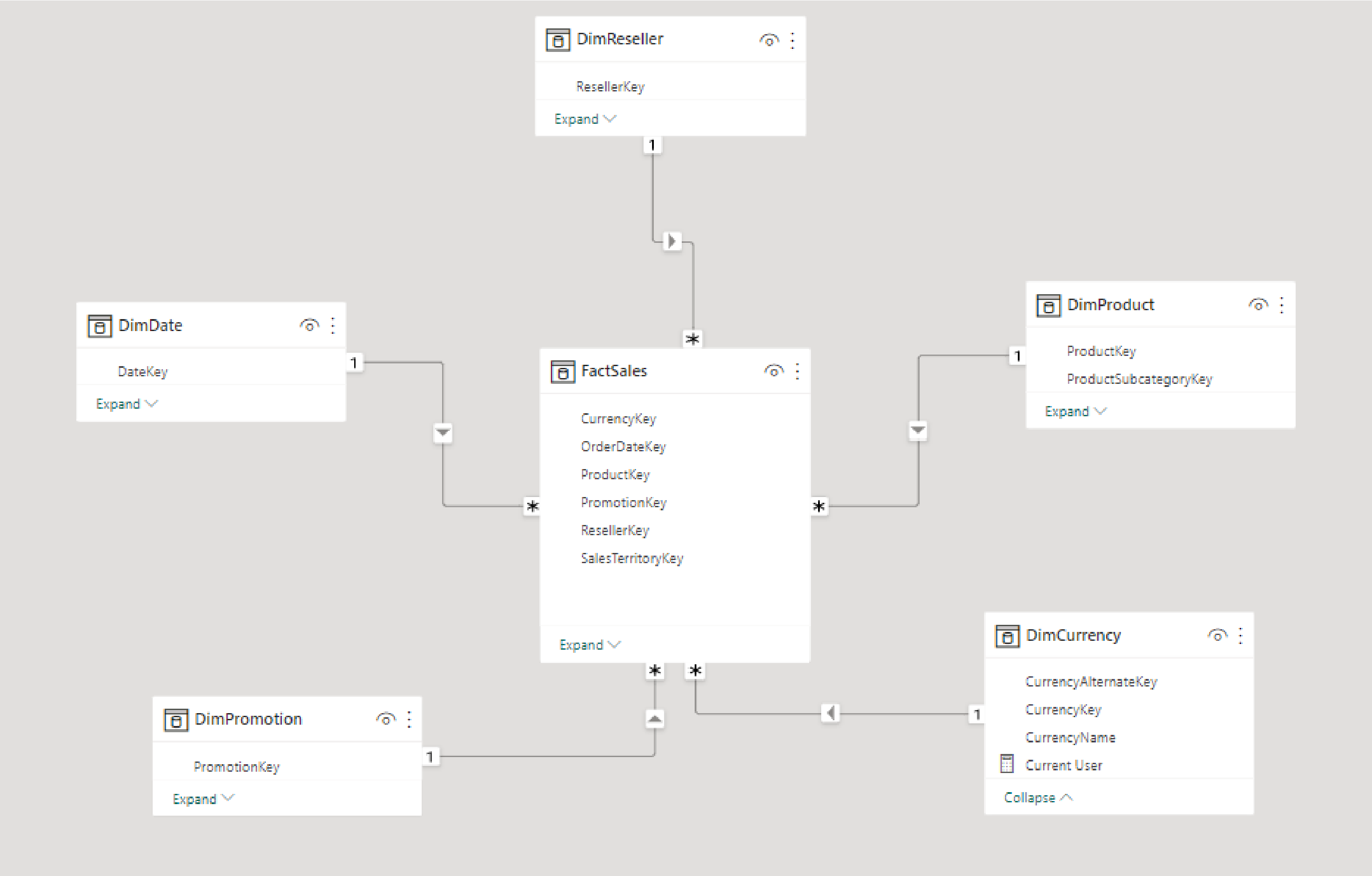
DAX Field Guide

Viktor Radu
Sr. Cloud Solutions Architect

Data Model

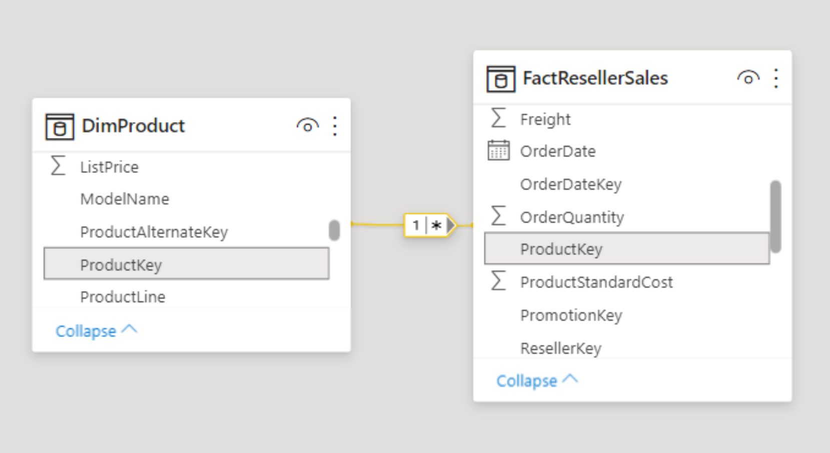
- Tables
- Relationships
- Metadata
  - Formatting
  - Categories
  - Hierarchies
  - Sorting

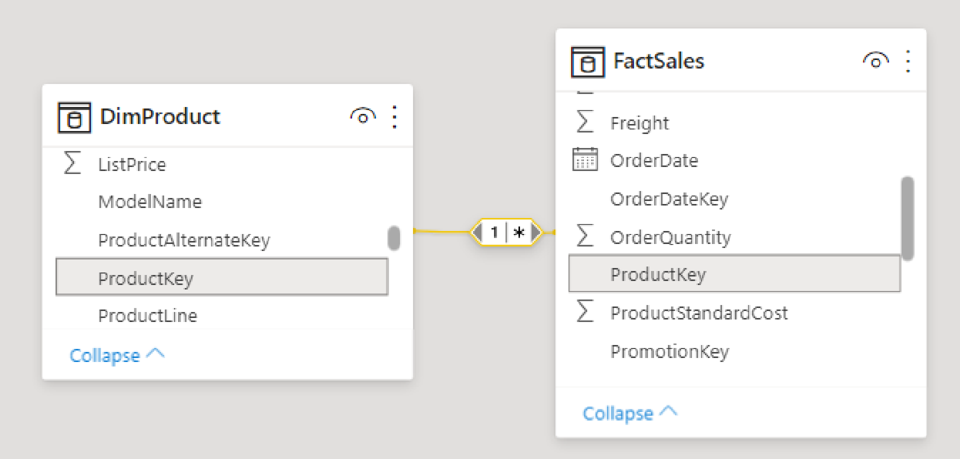


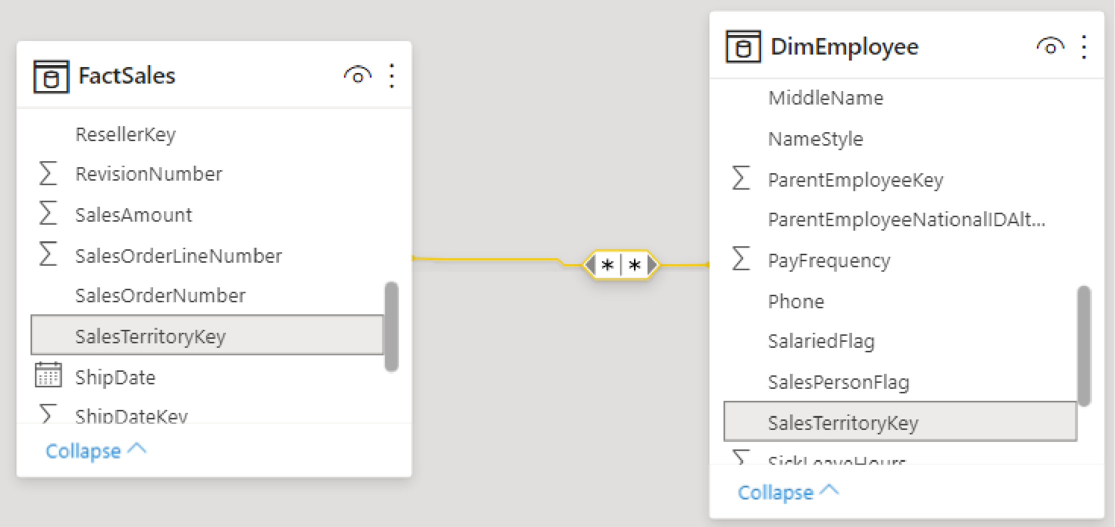
Start schema is a good way to structure the data model

Data Model - Navigation

- Cardinality
- Cross-filer direction





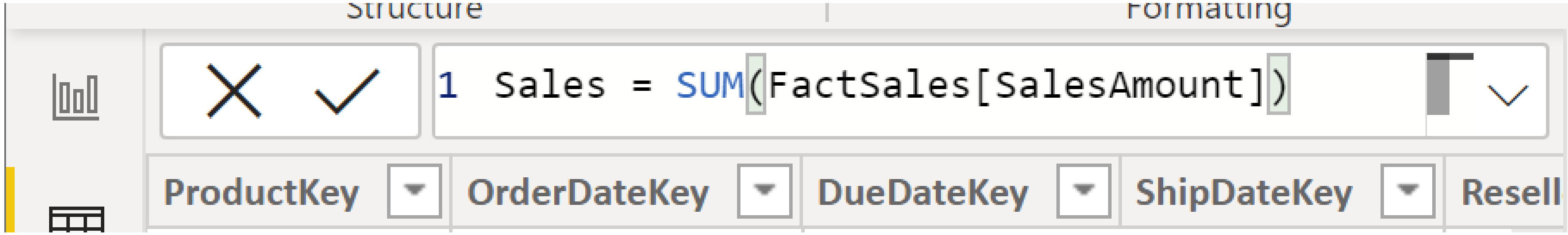


DAX in Power BI Desktop

- Calculated Columns
- Calculated Tables

- Measures
- Row-level security rules

Formula Bar



```
Sales = SUM(FactSales[SalesAmount])
```

DAX in Power BI Desktop - Syntax

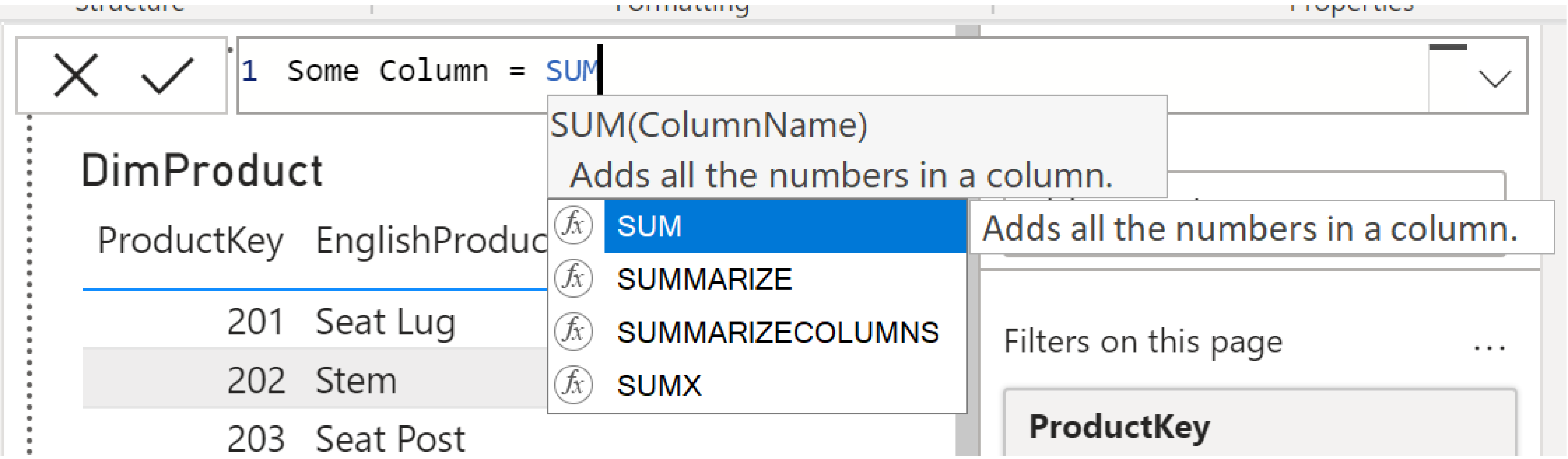
Calculations apply functions to tables/columns and combine the results into a single scalar or table

```
Some Column = FactSales[Other Column] + [Some Measure]

Some Measure = SUM(FactSales[Other Column]) + [Another Measure]

Ratio = DIVIDE(FactSales[Some Column], FactSales[Other Column])
```

Intellisense looks up columns/function options and function definitions



```
Some Column = SUM
```

SUM(ColumnName)

Adds all the numbers in a column.

Adds all the numbers in a column.

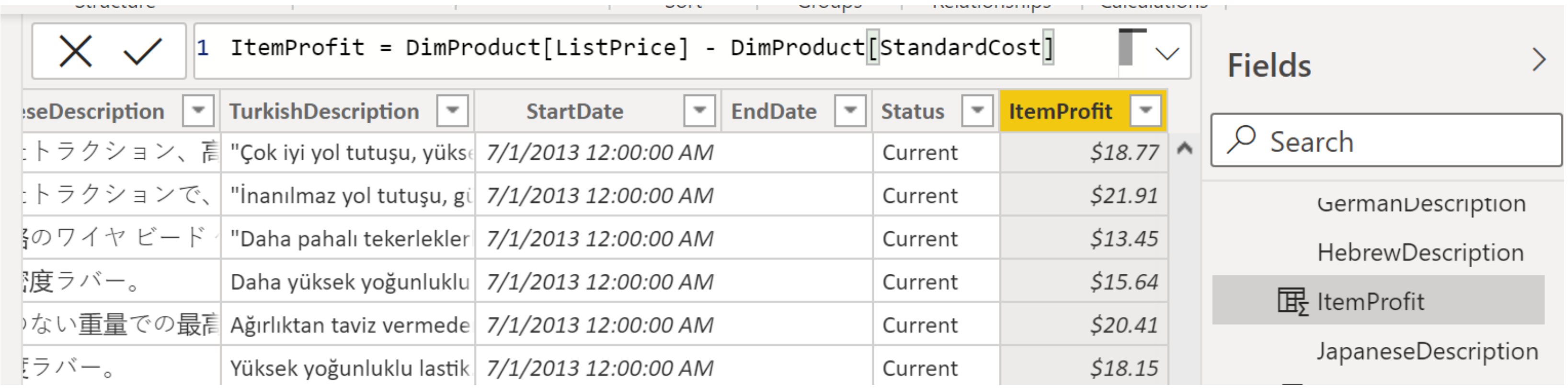
DimProduct

ProductKey	EnglishProduc
201	Seat Lug
202	Stem
203	Seat Post

Calculated columns vs Measures

Calculated Columns

- Persisted
- Evaluated row-by-row

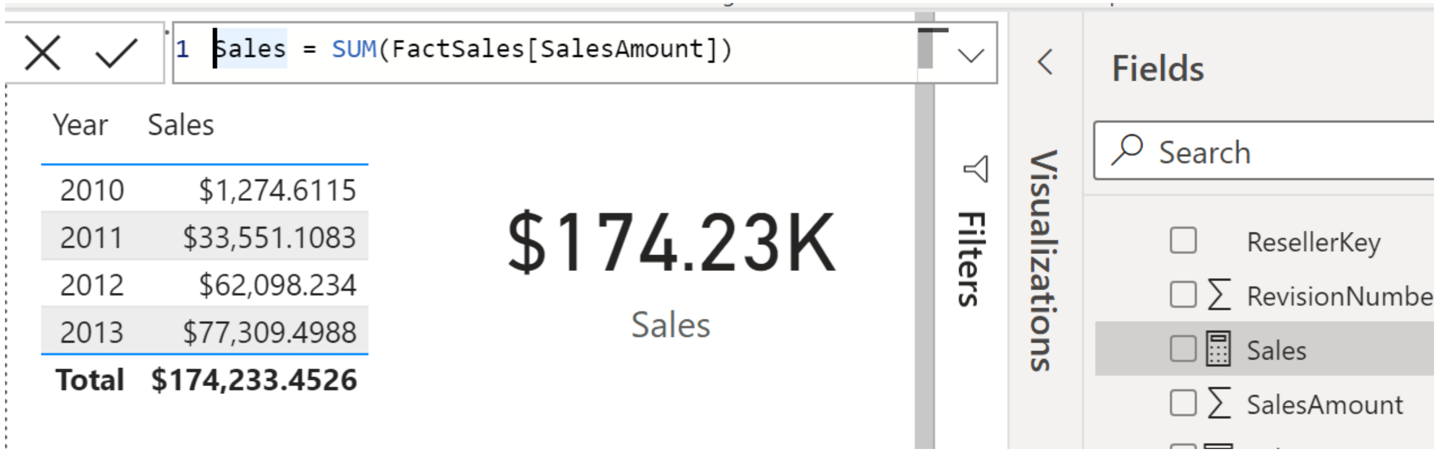


```
ItemProfit = DimProduct[ListPrice] - DimProduct[StandardCost]
```

seDescription	TurkishDescription	StartDate	EndDate	Status	ItemProfit
トラクション、高	"Çok iyi yol tutuşu, yüks	7/1/2013 12:00:00 AM		Current	$18.77
トラクションで、	"İnanılmaz yol tutuşu, gi	7/1/2013 12:00:00 AM		Current	$21.91
のワイヤ ビード	"Daha pahalı tekerlekler	7/1/2013 12:00:00 AM		Current	$13.45
度ラバー。	Daha yüksek yoğunluklu	7/1/2013 12:00:00 AM		Current	$15.64
ない重量での最高	Ağırlıktan taviz vermede	7/1/2013 12:00:00 AM		Current	$20.41
ラバー。	Yüksek yoğunluklu lastik	7/1/2013 12:00:00 AM		Current	$18.15

Measures

- Not persisted
- Evaluated for each cell/card/bar/point independently



```
Sales = SUM(FactSales[SalesAmount])
```

Year	Sales
2010	$1,274.6115
2011	$33,551.1083
2012	$62,098.234
2013	$77,309.4988
Total	**$174,233.4526**

$174.23K

Sales

DAX Functions - examples

- Aggregation

  SUM(), MAX(), MIN(), COUNT(), PRODUCT()

- Logical

  IF(), SWITCH(), AND(), OR()

- Financial

  ACCRINT(), NOMINAL(), PRICE(), YIELD()

- Statistical

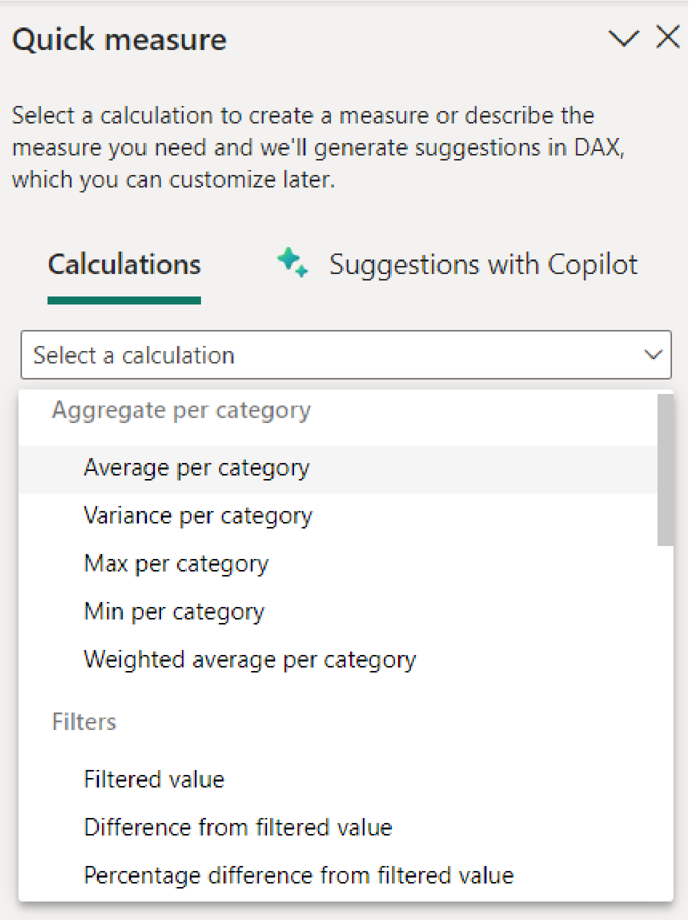
  AVERAGE(), POISSON.DIST(), NORM.DIST(), RANK.EQ(), STDEV.S()

- …
- DAX function reference – DAX

  https://learn.microsoft.com/en-us/dax/dax-function-reference

Quick Measures

- Library of common calculations
- Suggestions with Copilot



Quick measure

Select a calculation to create a measure or describe the measure you need and we'll generate suggestions in DAX, which you can customize later.

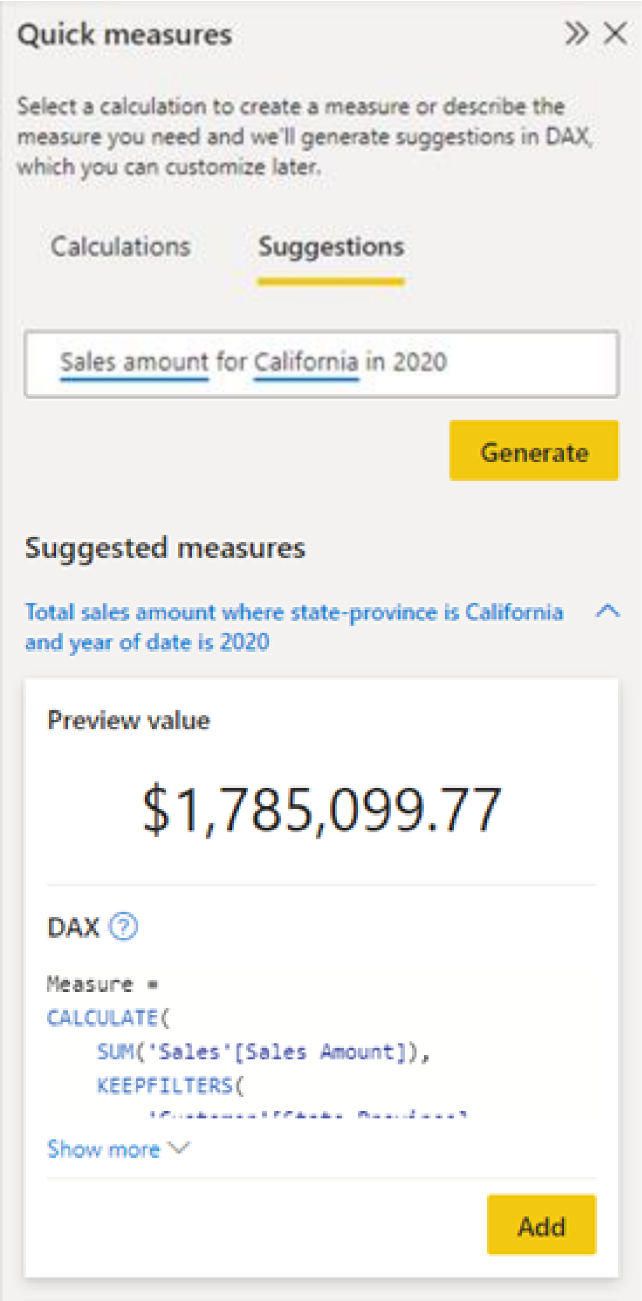
Select a calculation

Aggregate per category

- Average per category
- Variance per category
- Max per category
- Min per category
- Weighted average per category

Filters

- Filtered value
- Difference from filtered value
- Percentage difference from filtered value



Quick measures

Select a calculation to create a measure or describe the measure you need and we'll generate suggestions in DAX, which you can customize later.

Sales amount for California in 2020

Suggested measures

Total sales amount where state-province is California and year of date is 2020

Preview value

$1,785,099.77

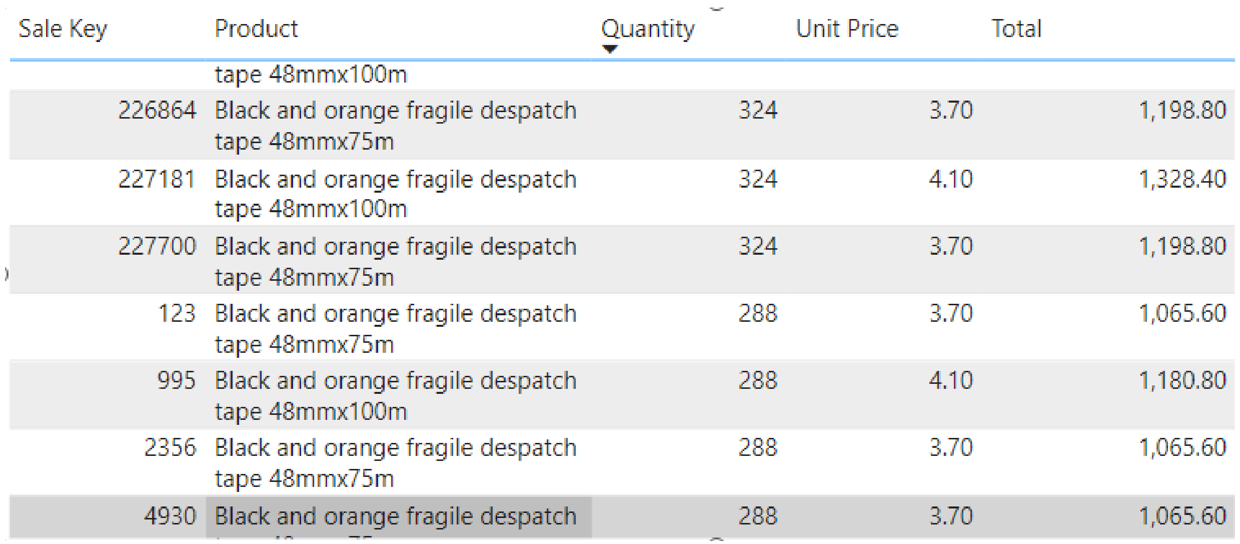
DAX

```
Measure =
CALCULATE(
    SUM('Sales'[Sales Amount]),
    KEEPFILTERS(
```

Evaluation context

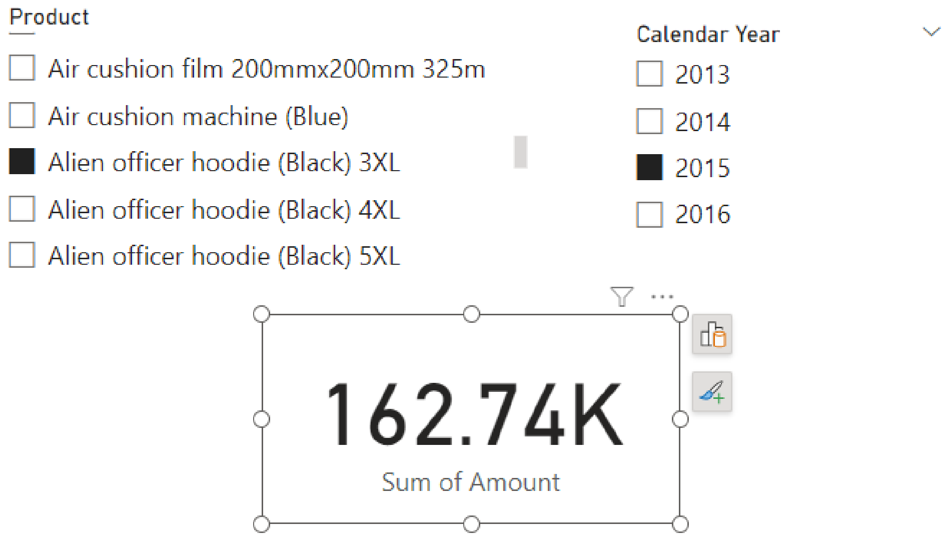
```
Total = FactSale[Unit Price] * FactSale[Quantity]
```

Row context



Sale Key	Product	Quantity	Unit Price	Total
	tape 48mmx100m			
226864	Black and orange fragile despatch tape 48mmx75m	324	3.70	1,198.80
227181	Black and orange fragile despatch tape 48mmx100m	324	4.10	1,328.40
227700	Black and orange fragile despatch tape 48mmx75m	324	3.70	1,198.80
123	Black and orange fragile despatch tape 48mmx75m	288	3.70	1,065.60
995	Black and orange fragile despatch tape 48mmx100m	288	4.10	1,180.80
2356	Black and orange fragile despatch tape 48mmx75m	288	3.70	1,065.60
4930	Black and orange fragile despatch	288	3.70	1,065.60

Filter context



Product

- ☐ Air cushion film 200mmx200mm 325m
- ☐ Air cushion machine (Blue)
- ■ Alien officer hoodie (Black) 3XL
- ☐ Alien officer hoodie (Black) 4XL
- ☐ Alien officer hoodie (Black) 5XL

Calendar Year

- ☐ 2013
- ☐ 2014
- ■ 2015
- ☐ 2016

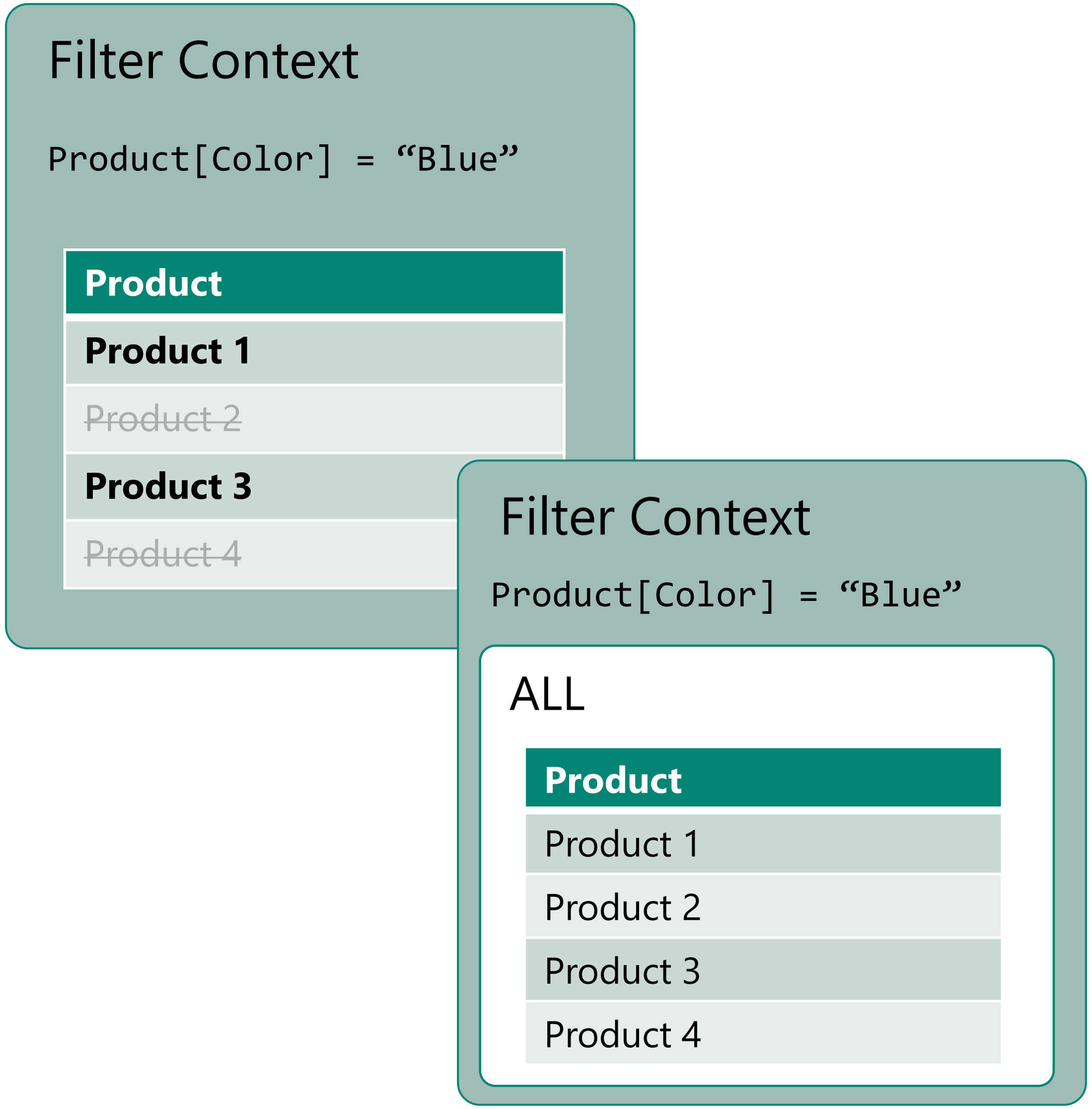
162.74K

Sum of Amount

Context manipulation

- Global
  - ALL
  - ALLNONBLANKROW
  - ALLEXCEPT

- Local
  - ALLSELECTED
  - ALLCROSSFILTERED
  - KEEPFILTERS



Filter Context

`Product[Color] = “Blue”`

Product
Product 1
~~Product 2~~
Product 3
~~Product 4~~

Filter Context

`Product[Color] = “Blue”`

ALL

Product
Product 1
Product 2
Product 3
Product 4

Practice Problems

Refer to practice.md. DAX basics 1 - 4

Calculated tables

- Persisted
- Generated by table value functions – examples:
  - SUMMARIZE
  - SELECTCOLUMNS
  - FILTER
  - UNION
  - DATATABLE
  - ALL
- Calculated columns can be defined

Variables

```
var totalSales = SUM(Sales)
return IF(totalSales > 0, totalSales, 0)
```

Variables can be used across contexts

```
Sales Amount =
var unitPrice = MAX(FactSales[UnitPrice])
var result = SUMX(FactSales, FactSales[OrderQuantity] * unitPrice)
return result
```

Conditional functions

-
  ```
  IF(Sales[Quantity] > 0, Sales[Quantity], 0)
  ```

-
  ```
  COALESCE([Sales Amount], 0)
  ```

-
  ```
  AND, OR, NOT
  ```

  ```
  OR(A, AND(B, C)) is equivalent to A || (B && C)
  ```

-
  ```
  SWITCH([Selected Category], “Clothing”, “C”, “Bikes”, “B”, “…”)
     SWITCH(TRUE(),
            [Selected Category] = “Clothing”, “C”,
             [Selected Region] = “East” && [Total] > 100, “B”,
            “…”)
  ```

Calculate function

```
Sales of Size M = CALCULATE(SUM(FactSale[Amount]), 'Product'[Size] = "M")
```

- ALL, ALLNONBLANKROW, ALLEXCEPT
- ALLSELECTED, ALLCROSSFILTERED
- KEEPFILTERS
- Operators && and ||

```
Price Average = CALCULATE(AVERAGE('Product'[Unit Price]), 'Product'[Color] = "Blue" && 'Product'[Brand] = "Contoso")
```

- CALCULATETABLE

Calculate function

```
Measure = SUM(FactSales[Amount])
```

Filter Context	Result
Empty (no filters applied)	All Sales records included in the calculation
Product[Color] = “Yellow”	Only Sales records related to “Yellow” products are included in the calculation

```
Measure = CALCULATE(SUM(FactSales[Amount]), Product[Color] = “Yellow”)
```

Filter Context	Result
Empty (no filters applied)	Only Sales records related to “Yellow” products are included in the calculation
Product[Color] = “Yellow”	Only Sales records related to “Yellow” products are included in the calculation
Product[Size] = “M”	Sales records related to “Yellow” products of size M are included in the calculation

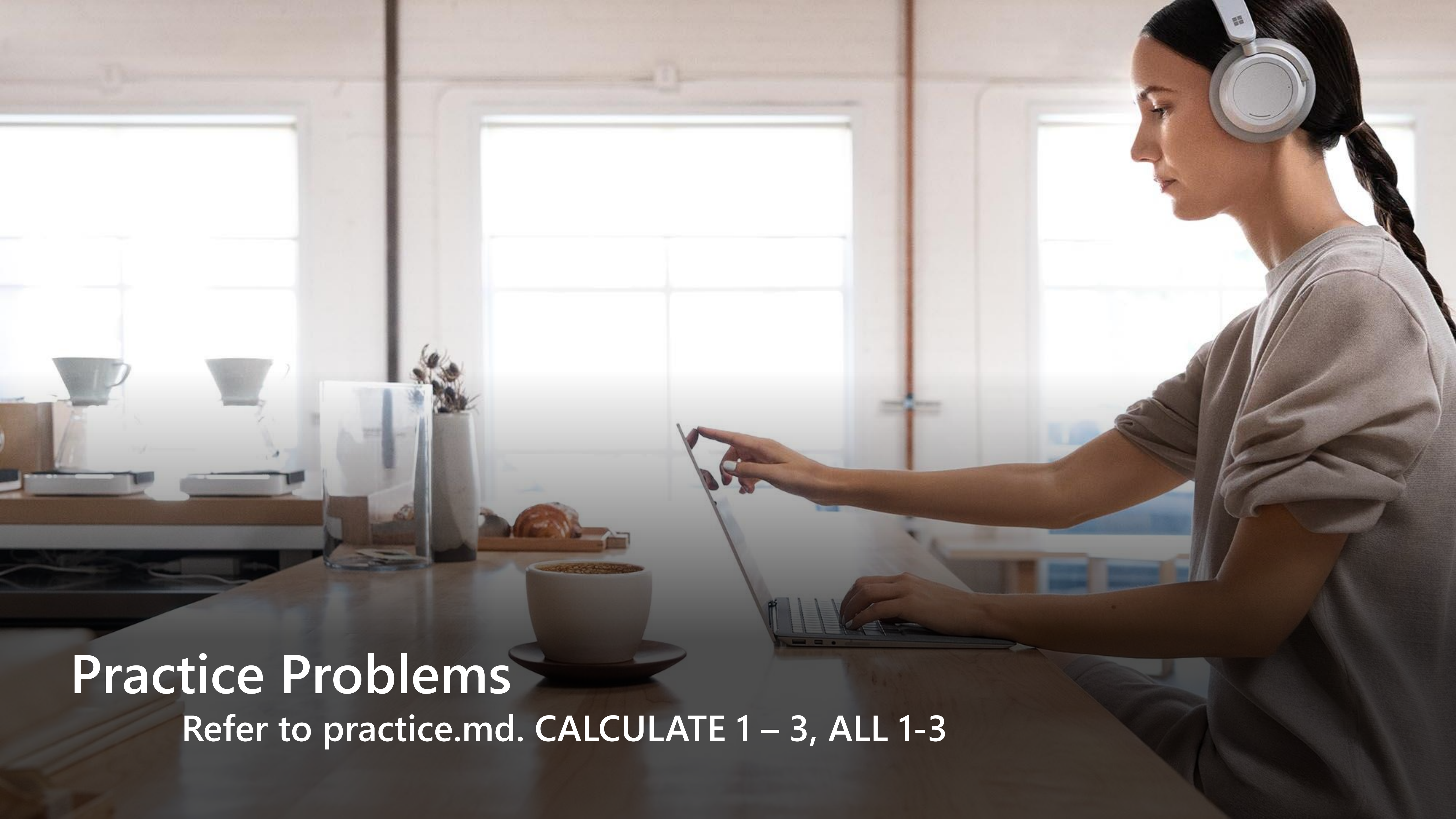
Calculate function - continued

```
Measure = CALCULATE(SUM(Sales[Amount]), KEEPFILTERS(Product[Color] = “Yellow”))
```

Filter Context	Result
Empty (no filters applied)	Only Sales records related to “Yellow” products are included in the calculation
Product[Color] = “Blue”	No Sales records satisfy both filter criteria, so the result is (Blank)

```
Measure = CALCULATE(SUM(Sales[Amount]), ALL(Product[Color]))
```

Filter Context	Result
Empty (no filters applied)	All Sales records included in the calculation
Product[Color] = “Red”	All Sales records included in the calculation
Product[Size] = “M”	Sales records related to products of size M are included in the calculation



Practice Problems

Refer to practice.md. CALCULATE 1 – 3, ALL 1-3

Microsoft

DAX Field Guide

Viktor Radu
Sr. Cloud Solutions Architect

Time intelligence – date table

- Date table needs to meet these requirements:
  - Date or DateTime type (same time stamp for DateTime)
  - Only unique date values
  - No blanks
  - Date values are contiguous

- Auto Date/Time
- Mark table as Date

Time intelligence

- Most time intelligence function are designed for use with CALCULATE
  - DATESBETWEEN, DATESINPERIOD
  - SAMEPERIODLASTYEAR
  - DATEADD
  - DATESYTD, DATESQTD, DATESMTD
  - STARTOFMONTH, ENDOFMONTH
- Shortcuts exist for common scenarios
  - TOTALYTD, TOTALQTD, TOTALMTD

Fiscal year

- Date functions include a YearEnd parameter

```
SalesYTD = TOTALYTD(FactSales[Sales], DimDate[Date],"6/30")
```

- Fiscal year attribute can be added to the calendar table

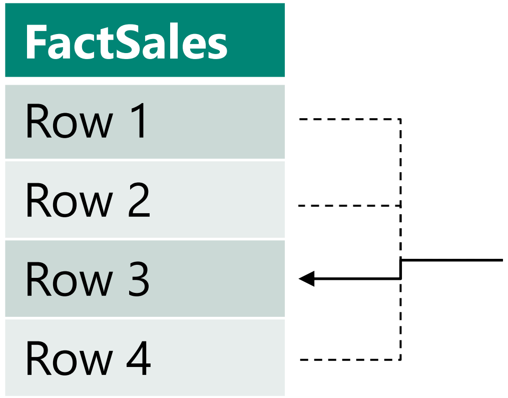
```
FiscalYear = YEAR(DimDate[Date]) + IF(MONTH(DimDate[Date]) >= 6, 1, 0)
```

Iterator functions

```
SUMX, RANKX, AVERAGEX, FILTER …..
```

```
Sales Amount = SUMX(FactSales, FactSales[OrderQuantity] * FactSales[UnitPrice])
```

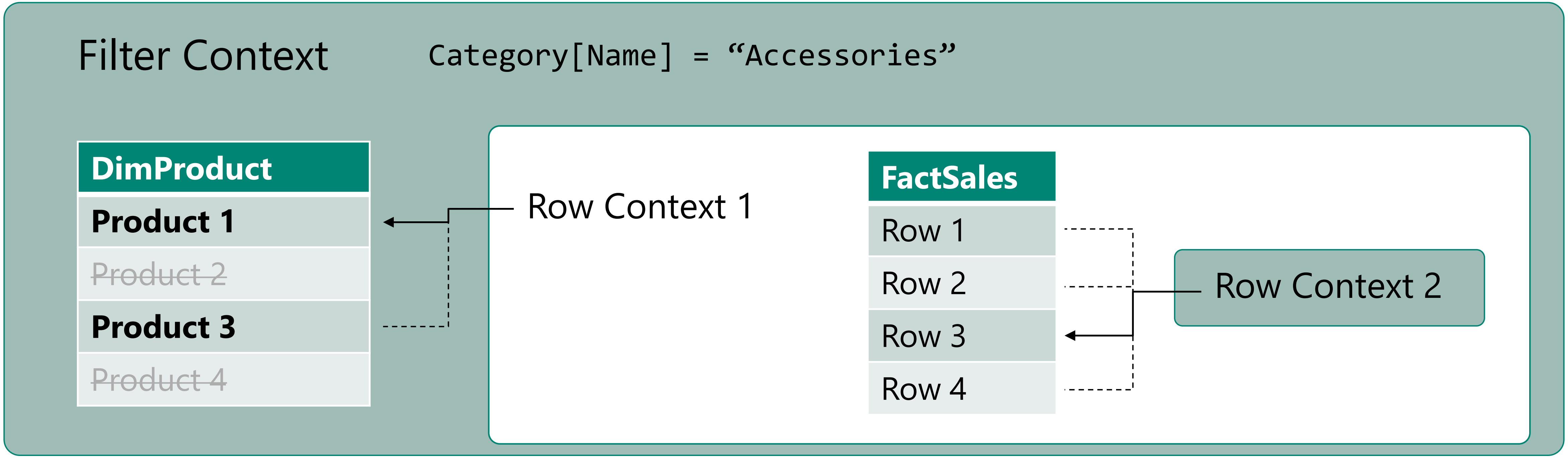
- Iterator functions create a Row Context
- Row Contexts can be nested within Filter Contexts and other Row Contexts



Row Context is like a cursor pointing at rows one at a time

Nested evaluation contexts

```
SalesOfTopProduct = MAXX(
        DimProduct,
        SUMX(
                RELATEDTABLE(FactSales),
                FactSales[OrderQuantity] * FactSales[UnitPrice] - FactSales[DiscountAmount]
        )
)
```



Filter Context Category[Name] = “Accessories”

DimProduct
Product 1
~~Product 2~~
Product 3
~~Product 4~~

Row Context 1

FactSales
Row 1
Row 2
Row 3
Row 4

Row Context 2

Practice Problems

Refer to practice.md. Time Intelligence 1-3

Table functions

- FILTER

```
FILTER (FactSale, FactSale [Package] = “Bag")
```

- RELATED

```
FILTER (FactSale, RELATED(Product[Color] ) = “Blue")
```

- VALUES
- SELECTEDVALUE

```
SelectedCategory = SELECTEDVALUE(Category[Name], “Multiple”)
```

- MAX vs. SELECTEDVALUE

Table summary functions

- ADDCOLUMNS

  ADDCOLUMNS(<Table>, <Name>, <Expression>, [<Name>, <Expression>], …)

- SELECTCOLUMNS

  SELECTCOLUMNS(<Table>, <Name>, <Expression>, [<Name>, <Expression>], …)

- SUMMARIZE

  SUMMARIZE(<Table>, <GroupBy_column>[, <GroupBy_column>]…[, <Name>, <Expression>]…)

- SUMMARIZECOLUMNS

  SUMMARIZECOLUMNS(<GroupBy_column> [, < GroupBy_column >]…, [<FilterTable>]…[, <Name>, <Expression>]…)

  Can't be used in measures

Cumulative totals

- Order by category

```
Sales RT = CALCULATE([Sales],
    FILTER(
        ALLSELECTED(DimCategory[Name]),
        DimCategory[Name] <= MAX(DimCategory[Name])
    )
)
```

- Order by value

```
Sales Pareto =
var categoryRank = RANKX(ALLSELECTED(DimCategory), [Sales], [Sales], DESC)
var result = SUMX(TOPN(categoryRank, ALLSELECTED(DimCategory), [Sales]), [Sales])
return result
```

Practice Problems

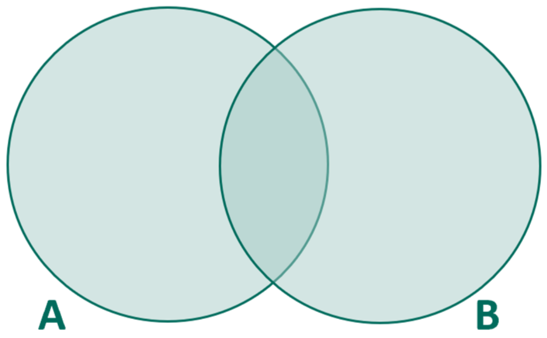
Refer to practice.md. Cumulative Total 1-3

Set operations

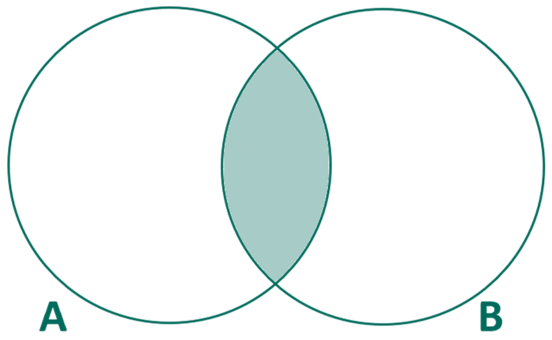
- CROSSJOIN

CROSSJOIN (<table1>, <table2>, [<table3>] …)

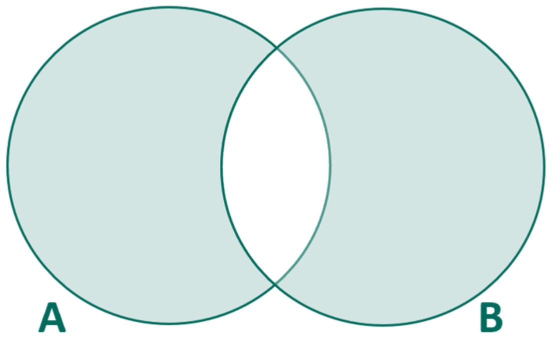
- UNION



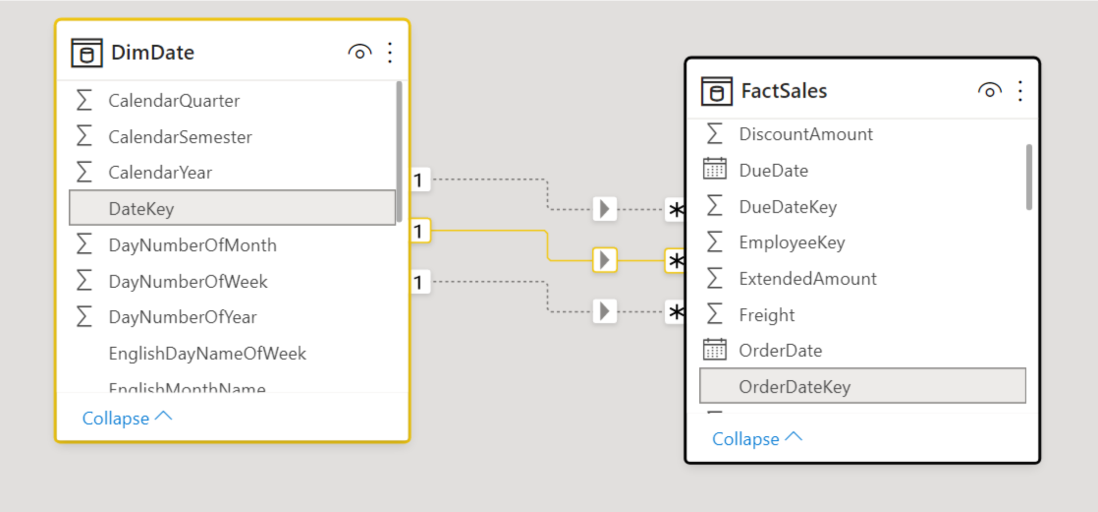
- INTERSECT



- EXCEPT



Alternative navigation paths



- USERELATIONSHIP

```
SalesByDueDate =
CALCULATE(
    SUM(FactSale[Amount]),
    USERELATIONSHIP(FactSales[DueDateKey], DimDate[DateKey])
)
```

- TREATAS

```
SalesByDueDate =
CALCULATE(
       SUM(FactSale[Amount]),
       TREATAS(VALUES(FactSales[DueDateKey]), DimDate[DateKey])
)
```

Disconnected table pattern

- The disconnected table can be imported from a data source or calculated
- What-if scenarios
- Dynamic measure selection
- Dynamic axis selection

Forecasting

- Linear regression
  - LINESTX (<table>, <expressionY>, <expressionX>[, …][, <const>])

```
var inputTable = SUMMARIZE(ALL(FactSales), Date[Date], "Sales", [Sales])

var regression = LINESTX(inputTable, [Sales], Date[Date])
```

- Slope1, Slope2, …, SlopeN
- Intercept
- StandardErrorSlope1, StandardErrorSlope2, StandardErrorSlopeN
- StandardErrorIntercept, CoefficientOfDetermination, StandardError, Fstatistic, DegreesOfFreedom, RegressionSumOfSquares, ResidualSumOfSquares

Practice Problems

Refer to practice.md. Virtual Relationships 1-2, Disconnected Table

Error handling

- ISERROR(expression)

  returns True if expression results in an error

- IFERROR(expression, fallback value)

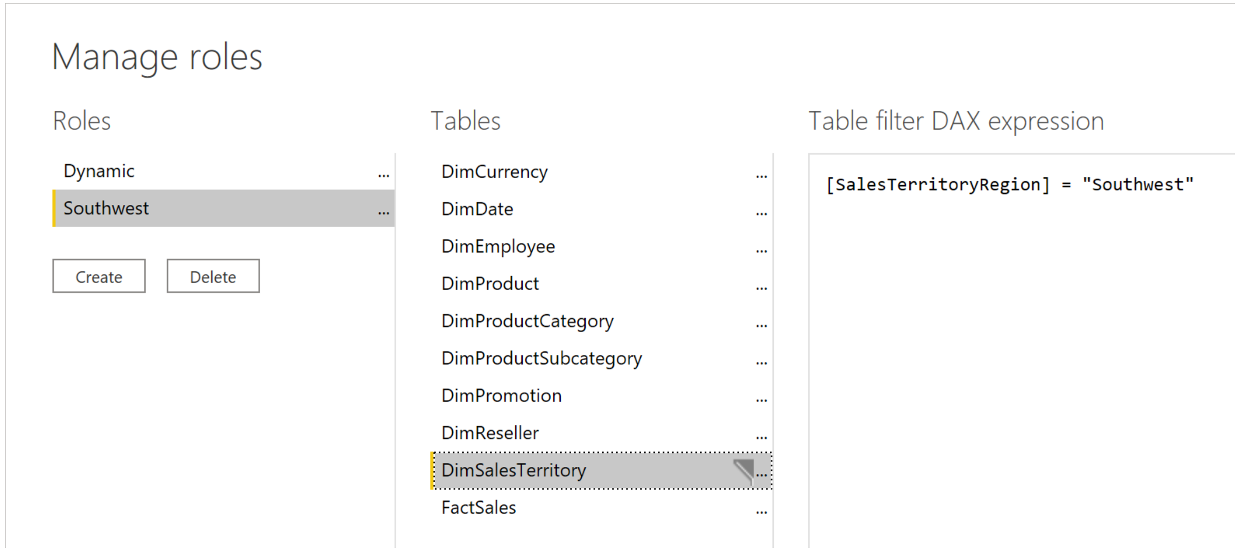
  returns a fallback value if expression results in an error

- Avoid using error-handling functions
  - When potential error conditions are known, check for those with IF
  - When potential error conditions are not known, handling an error could hide a problem with data or calculation logic
- DIVIDE(expression1, expression2, 0)
- ISBLANK, ISNUMBER, ISTEXT, ISNONTEXT

Row Level Security

- Static

  Mapping users to roles



Manage roles

Roles

Dynamic ...

Southwest ...

Tables

DimCurrency ...

DimDate ...

DimEmployee ...

DimProduct ...

DimProductCategory ...

DimProductSubcategory ...

DimPromotion ...

DimReseller ...

DimSalesTerritory ...

FactSales ...

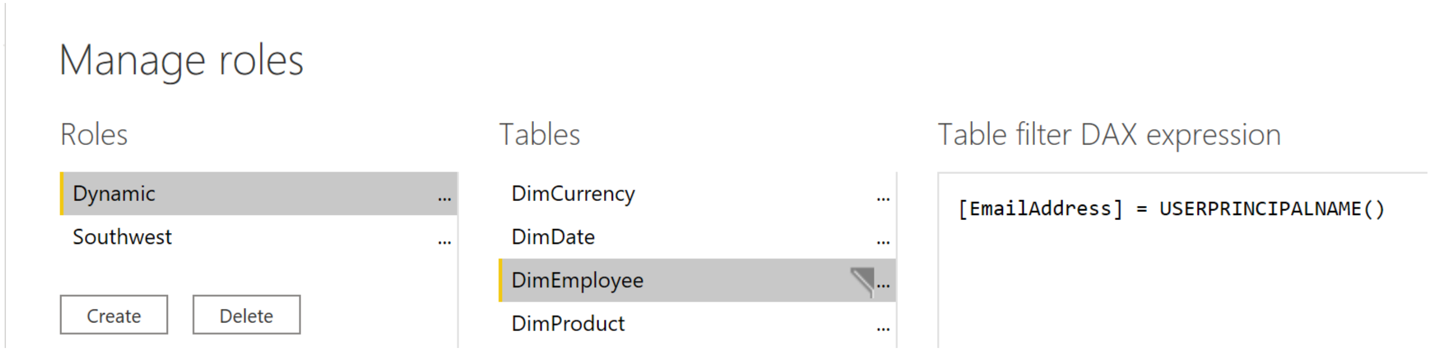
Table filter DAX expression

```
[SalesTerritoryRegion] = "Southwest"
```

- Dynamic

  Mapping users to data directly inside the model

  USERPRINCIPALNAME()



Manage roles

Roles

Dynamic ...

Southwest ...

Tables

DimCurrency ...

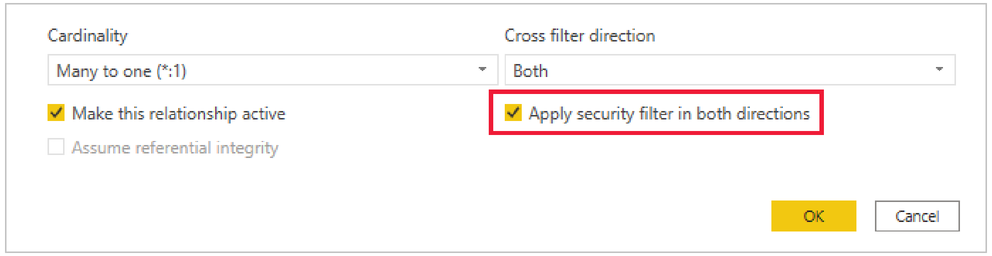
DimDate ...

DimEmployee ...

DimProduct ...

Table filter DAX expression

```
[EmailAddress] = USERPRINCIPALNAME()
```



Cardinality: Many to one (*:1)

Cross filter direction: Both

☑ Make this relationship active

☑ Apply security filter in both directions

☐ Assume referential integrity

DAX debugging

- Define measures to extract intermediary step values
- Define steps in a complex calculation as variables and return steps during debugging
- CONCATENATEX(DimProduct, FactSales[UnitPrice], “|”)

DAX performance

- Reduce overall data model size
  - Unused columns
  - Inefficient data types
  - Row order during load
  - Consider column granularity
- Avoid iterator functions where possible (e.g. SUMX or FILTER)
- Avoid replacing BLANK values with 0
- Use COUNTROWS instead of COUNT
- Use variables instead of repeating the same expression multiple times

DAX in Power BI Desktop - Tools

- Performance Analyzer
  - Provides a view into all the queries produced by a page
  - Selectively refresh visuals
  - Get native query text

- DAX Studio
  - Execute DAX and DMV queries against a local or remote Tabular model
  - Monitor queries
  - Analyze query execution times (SE vs. FE) and query plans
  - Examples:
    - EVALUATE DimProduct
    - EVALUATE SELECTCOLUMNS(DimProduct, "Key", DimProduct[ProductKey])

DAX Studio for performance tuning

- Enable Server Timings
- Use Storage Engine (SE) and Formula Engine (FE) timings to understand where the time is spent
- Clear Cache to get a more accurate time measurements
- Performance data can be imported from Power BI Desktop
- Use Advanced tab to review the VertiPaq storage metrics

Practice Problems

Refer to practice.md. Linear Regression