DAX Bootcamp

Mitchell Pearson

PRAGMATIC WORKS

About Mitchell

- Training Content Manager
- Pragmatic Works 8 years
- Author
- Runs Power BI User Group
- Blogs at MitchellPearson.com

- Twitter: @MitchellSQL
- LinkedIn: MitchellPearson1
- YouTube: MitchellSQL
- mpearson@pragmaticworks.com

PRAGMATIC WORKS

Agenda

Data Modeling

Calculated Columns

Calculated Measures

CALCULATE

Time Intelligence

Table Functions

Working with Totals

Evaluation Context

Semi-Additive Measures

Dynamic Security

Role-Playing Tables

Dimensional Modeling

What is a Data Model?

PRAGMATIC WORKS

Things to consider….

1) What are you measuring?
2) What types of business problems are you trying to solve?
3) How much data are you working with?
4) What are your data sources?

PRAGMATIC WORKS

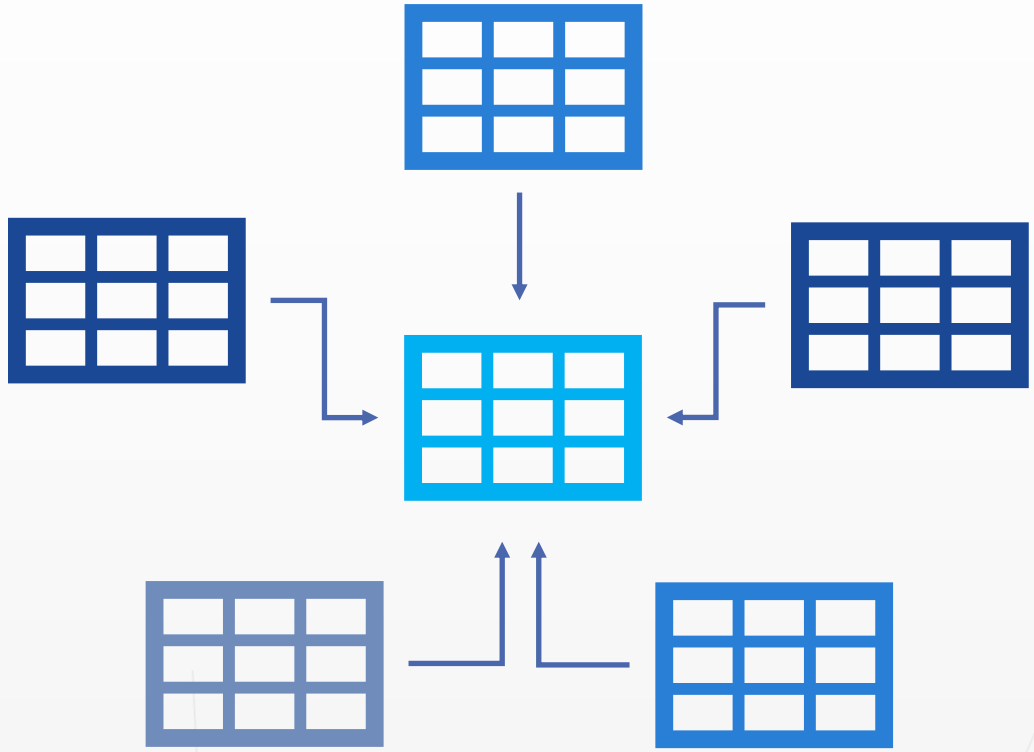
Attributes of a Good Data Model

Can be easily understood and consumed

Large data changes are scalable

Provides **predictable performance**

Is **flexible and adaptable,** but not at the expense of the other attributes

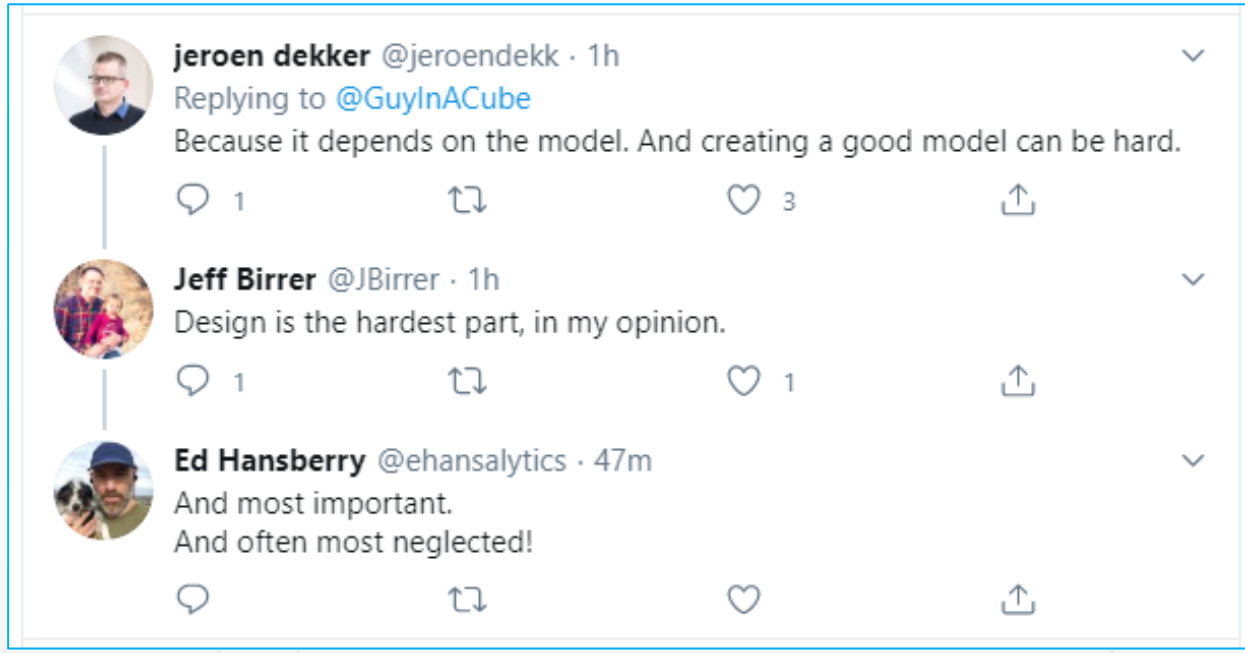


PRAGMATIC WORKS

Importance of the Data Model

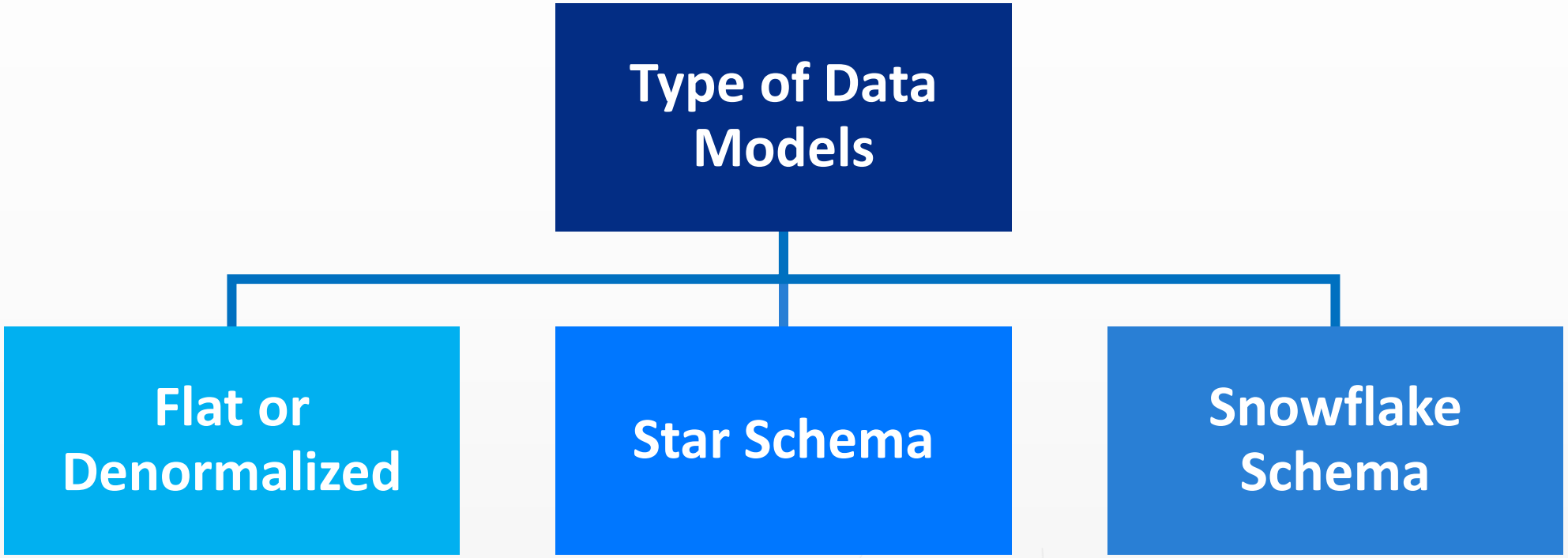
Performing ETL is more difficult with a bad data model

Without a well-built data model, **everything** becomes more of a struggle to implement



PRAGMATIC WORKS

Types of Power BI Data Models



PRAGMATIC WORKS

Flat or Denormalized Schema

	Screening Date	State	Division	Region	Gender	Line of Business	Policy Type	Market <>
1	1/7/2019	OH	Northeast	East Central	F	M	I	OH OTHER <> L
2	1/11/2019	OH	Northeast	East Central	M	M	I	OH OTHER <> L
3	1/15/2019	OH	Northeast	East Central	M	M	I	OH OTHER <> L
4	1/18/2019	OH	Northeast	East Central	M	M	I	OH OTHER <> L
5	1/25/2019	OH	Northeast	East Central	F	M	I	OH OTHER <> L
6	2/1/2019	OH	Northeast	East Central	M	M	I	OH OTHER <> L
7	2/11/2019	OH	Northeast	East Central	F	M	I	OH OTHER <> L
8	2/12/2019	OH	Northeast	East Central	M	M	I	OH OTHER <> L
9	2/15/2019	OH	Northeast	East Central	M	M	I	OH OTHER <> L
10	2/20/2019	OH	Northeast	East Central	F	M	I	OH OTHER <> L
11	2/20/2019	OH	Northeast	East Central	M	M	I	OH OTHER <> L
12	3/1/2019	OH	Northeast	East Central	M	M	I	OH OTHER <> L
13	3/4/2019	OH	Northeast	East Central	M	M	I	OH OTHER <> L
14	3/5/2019	OH	Northeast	East Central	M	M	I	OH OTHER <> L
15	3/8/2019	OH	Northeast	East Central	F	M	I	OH OTHER <> L

All attributes for model exist in a single table

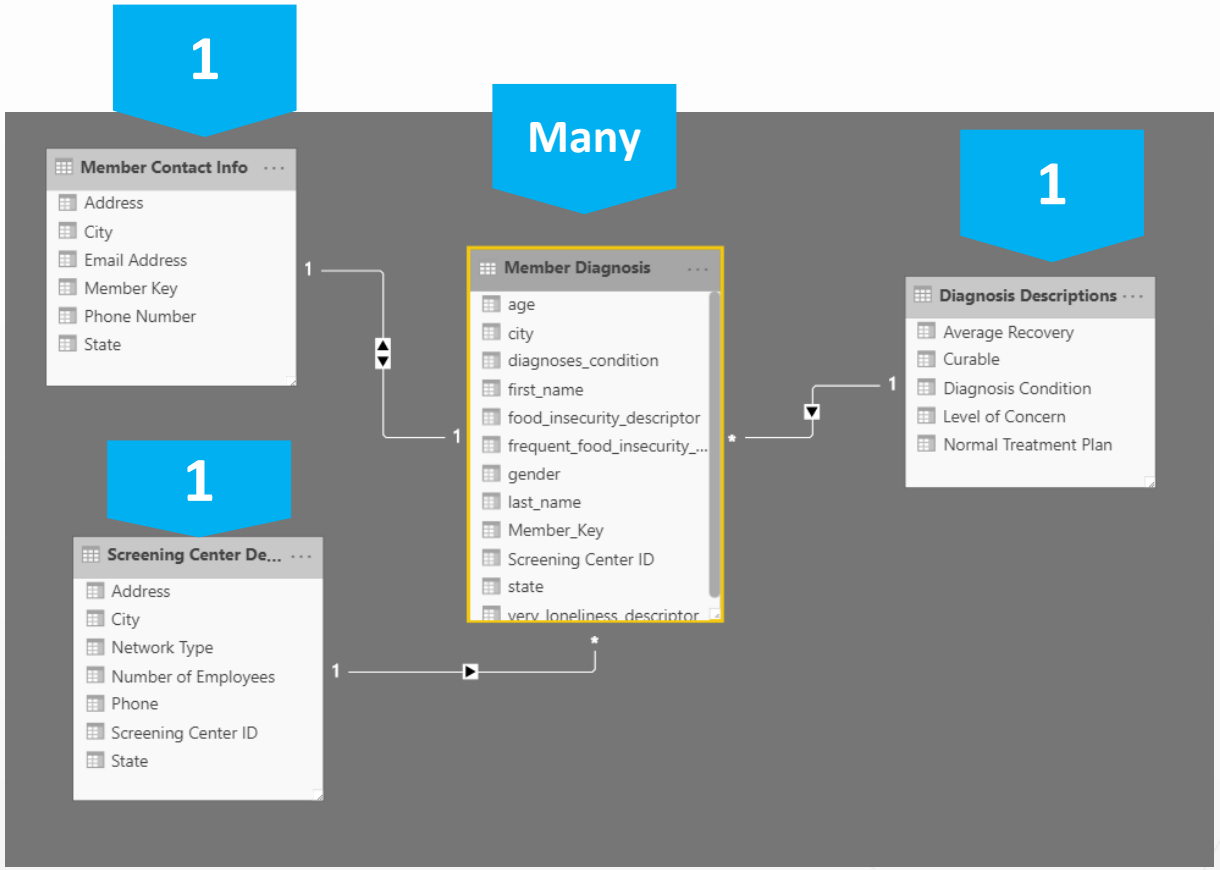
Highly inefficient

Model has extra copies of data = slow performance

Size of a flat table can blow up really quickly as data model becomes complex

PRAGMATIC WORKS

Star Schema



PRAGMATIC WORKS

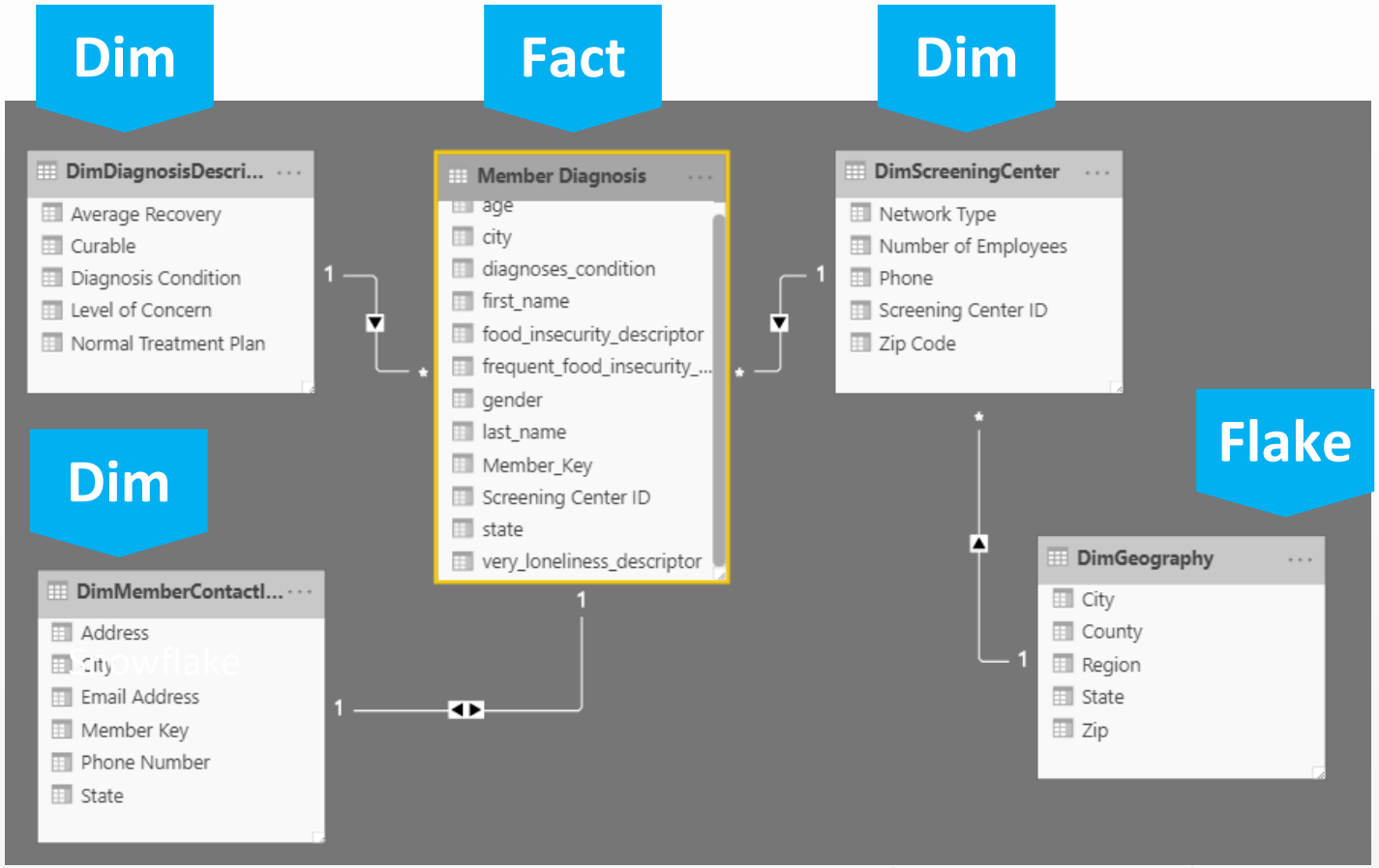
Fact table in the middle

Surrounded by Dims

Looks like a ‘Star’

Fact table is the “Many” side of the (one to many) relationship

Snowflake Schema

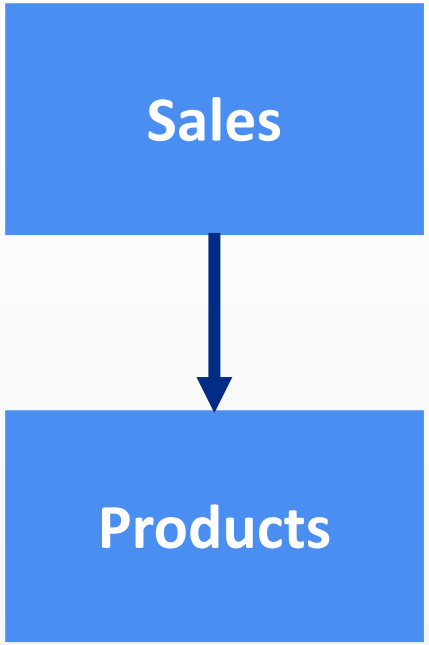


PRAGMATIC WORKS

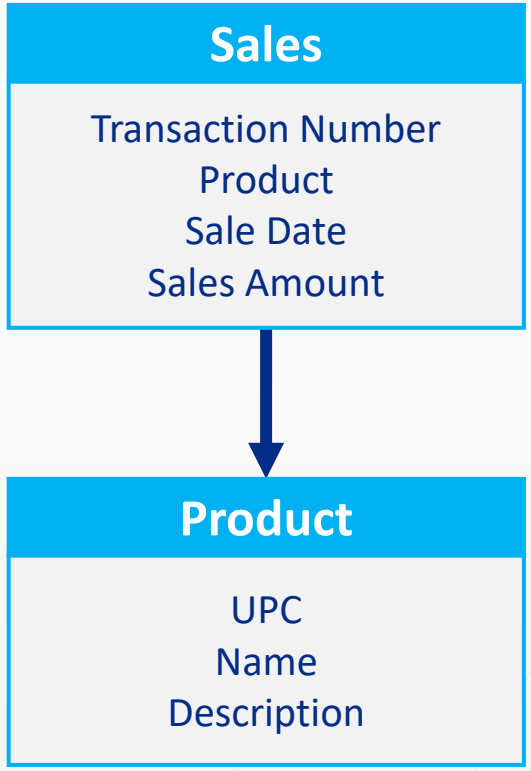
- Center is a Star schema
- Fact table in middle
- Surrounded by Dims
- Dims “snowflake” off of other Dims
- If you have many, it looks like a ‘Snowflake’
- Dim or Fact tables can be the “Many” side of the relationship

Model Types

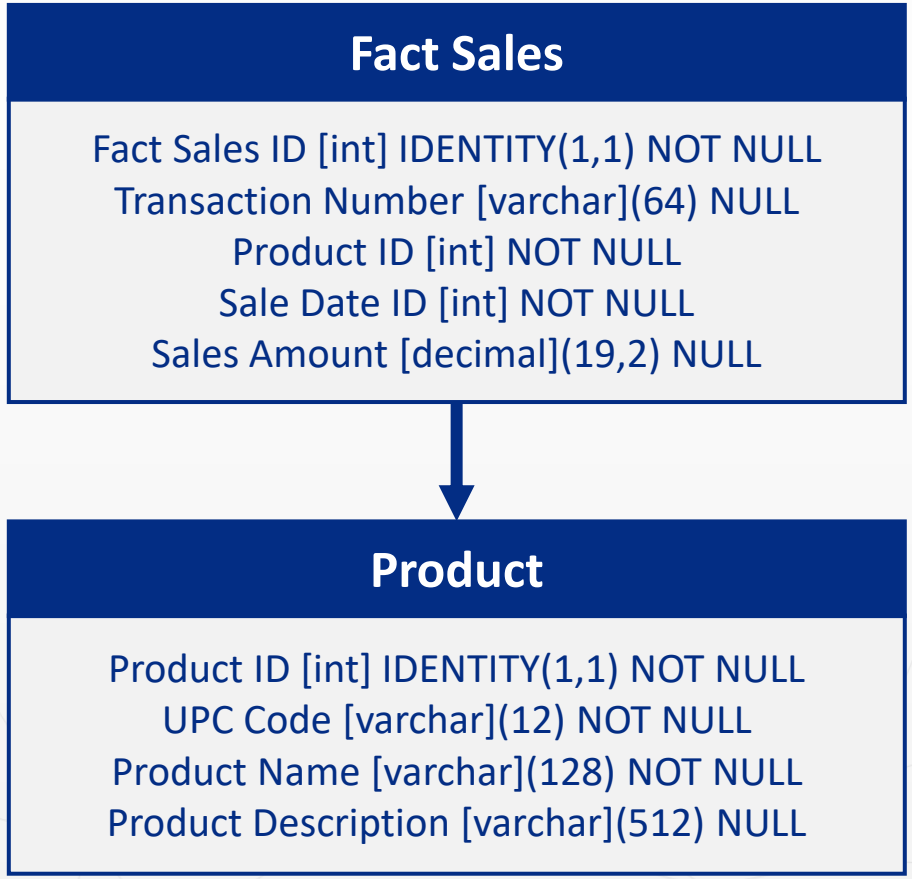
Conceptual



Logical

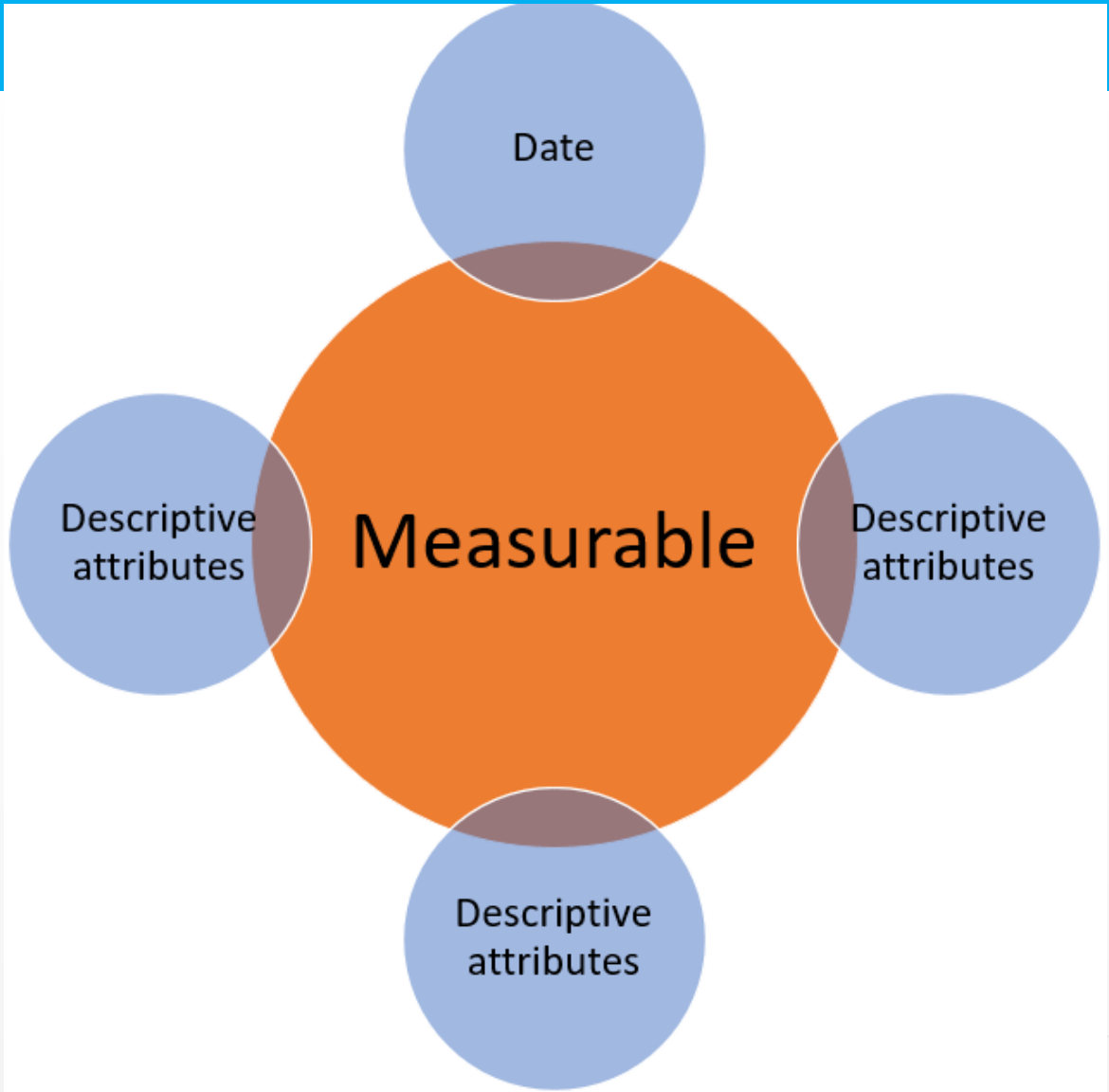


Physical



PRAGMATIC WORKS

Conceptual Model



PRAGMATIC WORKS

Dimensional Model Design

Dimensional Model Basics | Model Types

PRAGMATIC WORKS

Dimensional Model – Terminology

Dimensional Model

Organizes the data so it is easy to retrieve for reporting purposes

Fact Table

A fact is an event that may or may not include measures.

Dimension Table

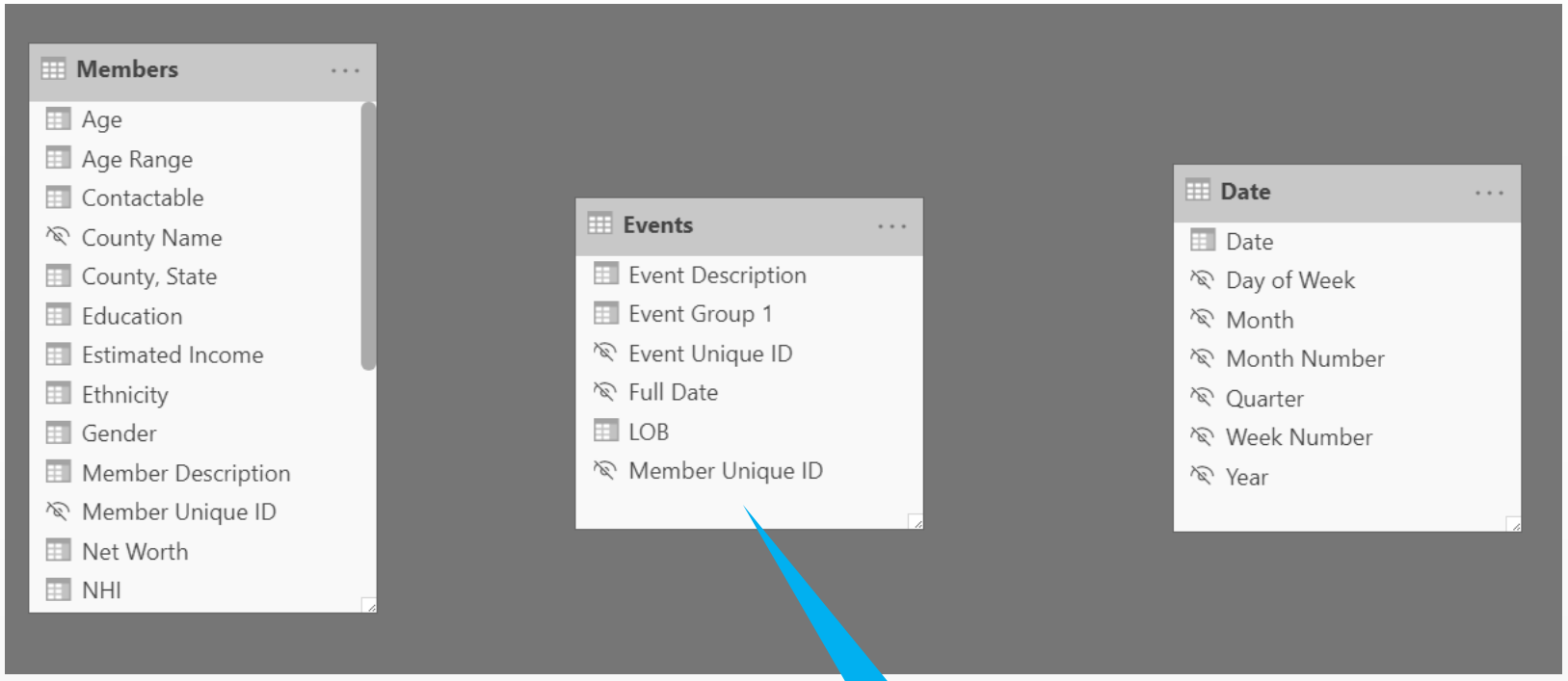
Category of information, or a noun, descriptive

Attribute *(column in dimension table)*

Descriptor of the object

PRAGMATIC WORKS

Fact Tables



Fact Table

Contains Measures
(or items to be aggregated)
of a business process

Examples

Claim Amount, Screenings,
Total Claims, Cost

Measures

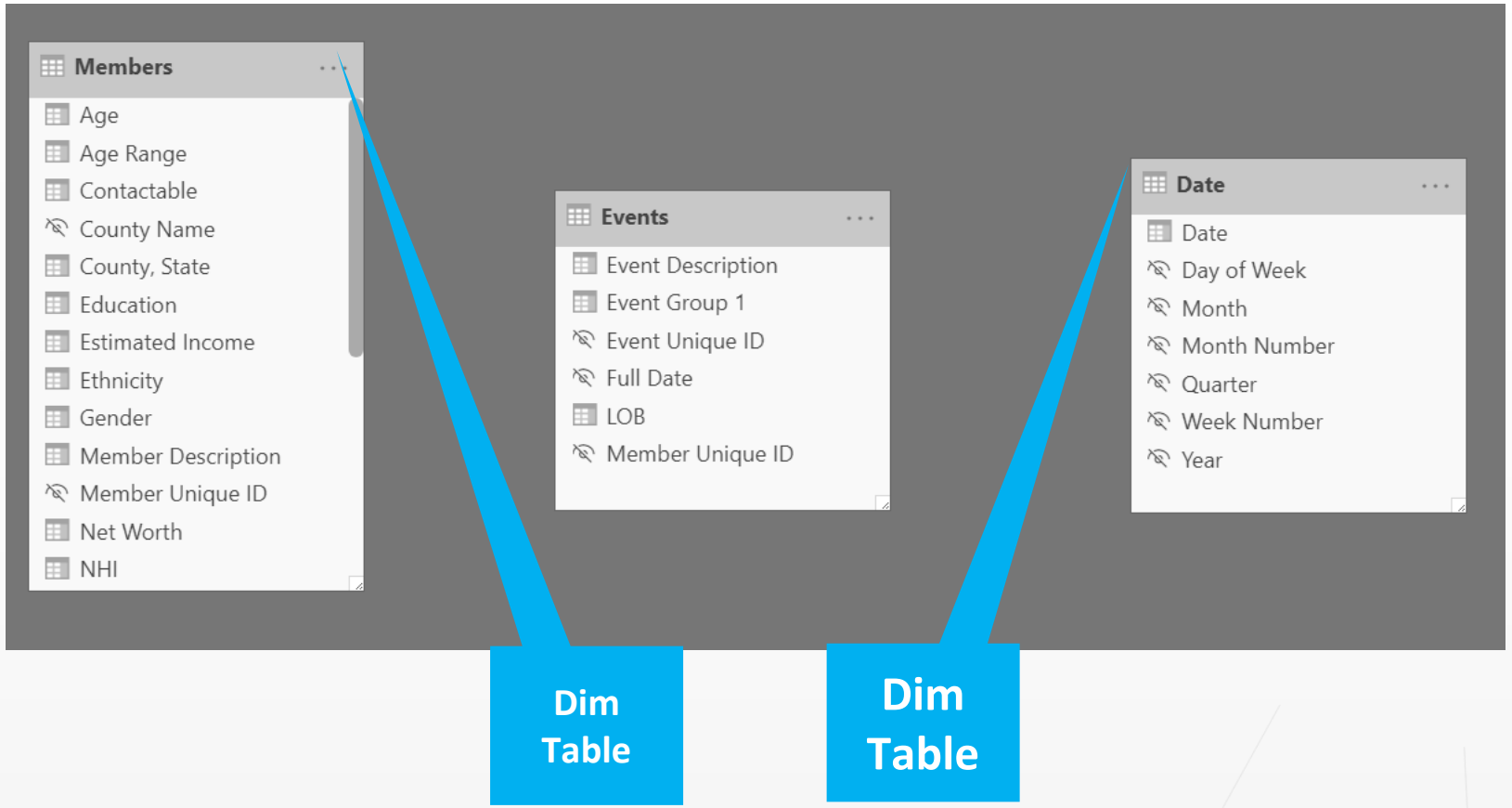
Usually sliceable

Examples:

By Month, By Member

PRAGMATIC WORKS

Dimensions



Dim Table

A Dim (or Dimension) table contains descriptive attributes that define how a fact should roll up

Examples:

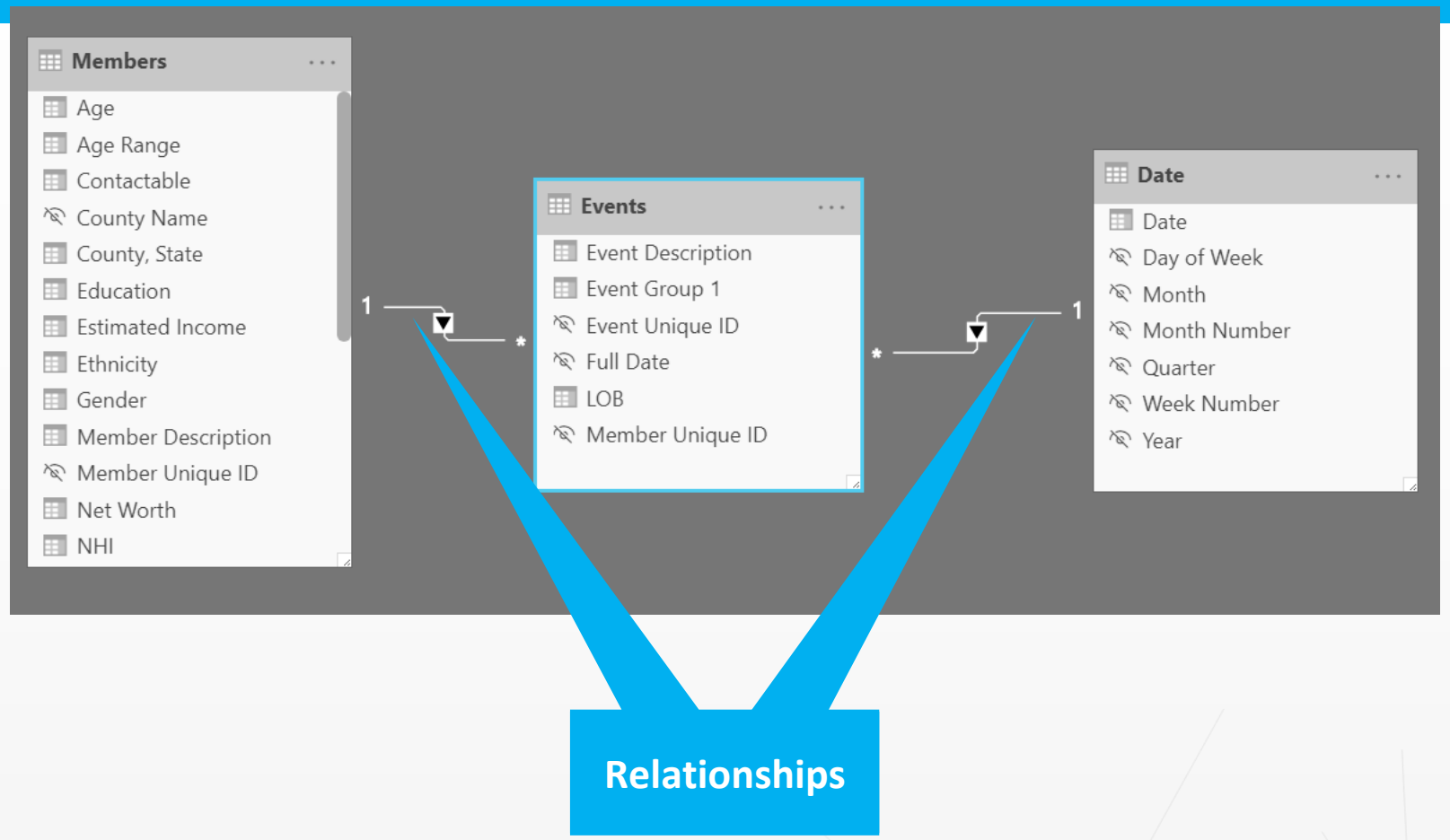
By Month

By Customer

By Geo

PRAGMATIC WORKS

Relationships



Relationships

Connection between 2 tables (usually fact & Dim tables) using columns from each

Types of Relationships

1 to Many

1 to 1

Many to Many
(with a bridge table)

PRAGMATIC WORKS

Fact Tables

Fact

A fact is an **event** that may or may not include measures

Granularity

Lowest level of information that will be stored in the fact table, or the values that would make the row distinct compared to all other rows

PRAGMATIC WORKS

Dimensional Model - Structure

Highly Denormalized

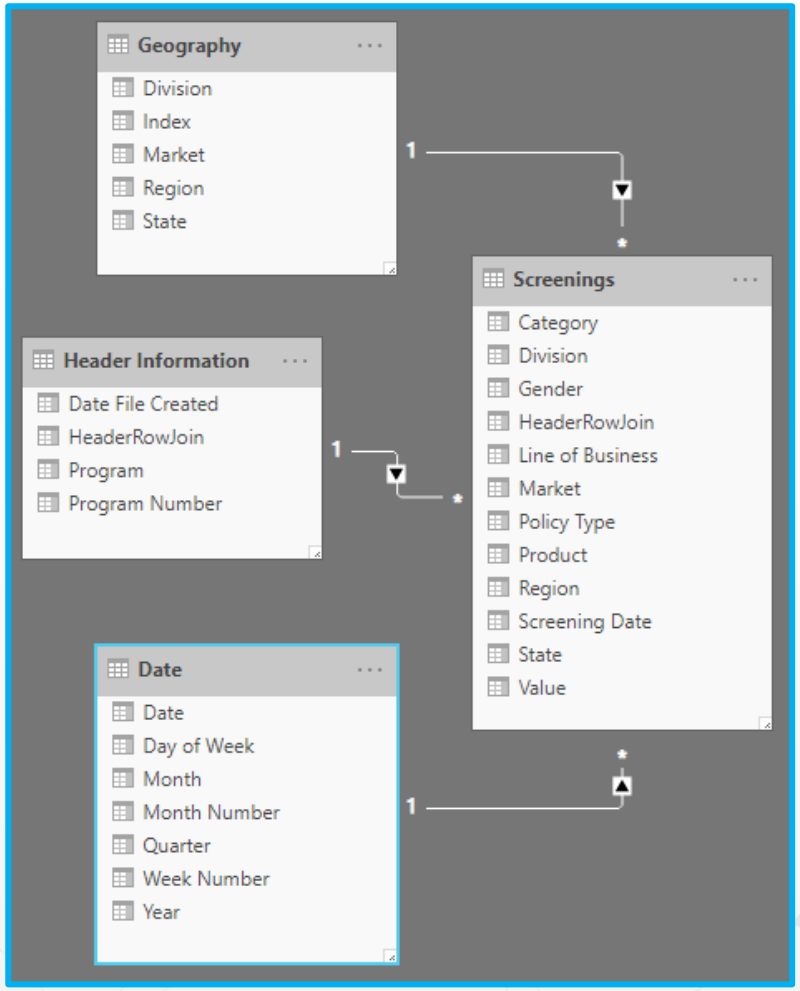
Tables merged logically for reporting

Table Types

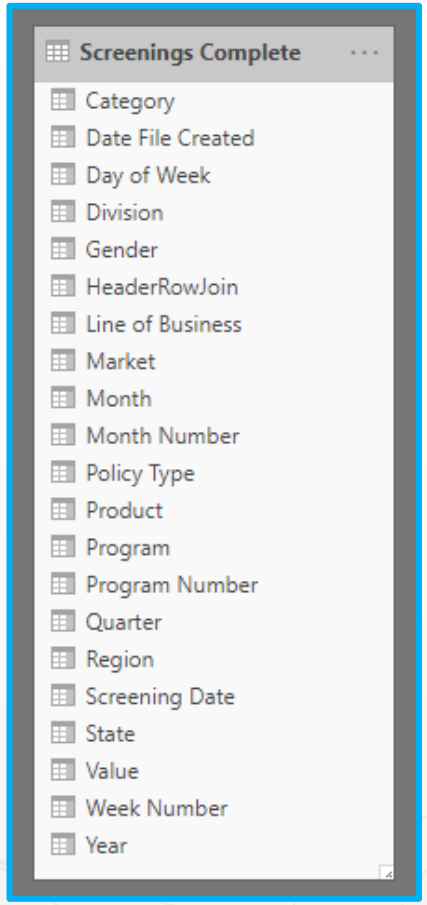
Only Facts and Dimensions

Necessary Fields Only

No Unnecessary Attributes



OR

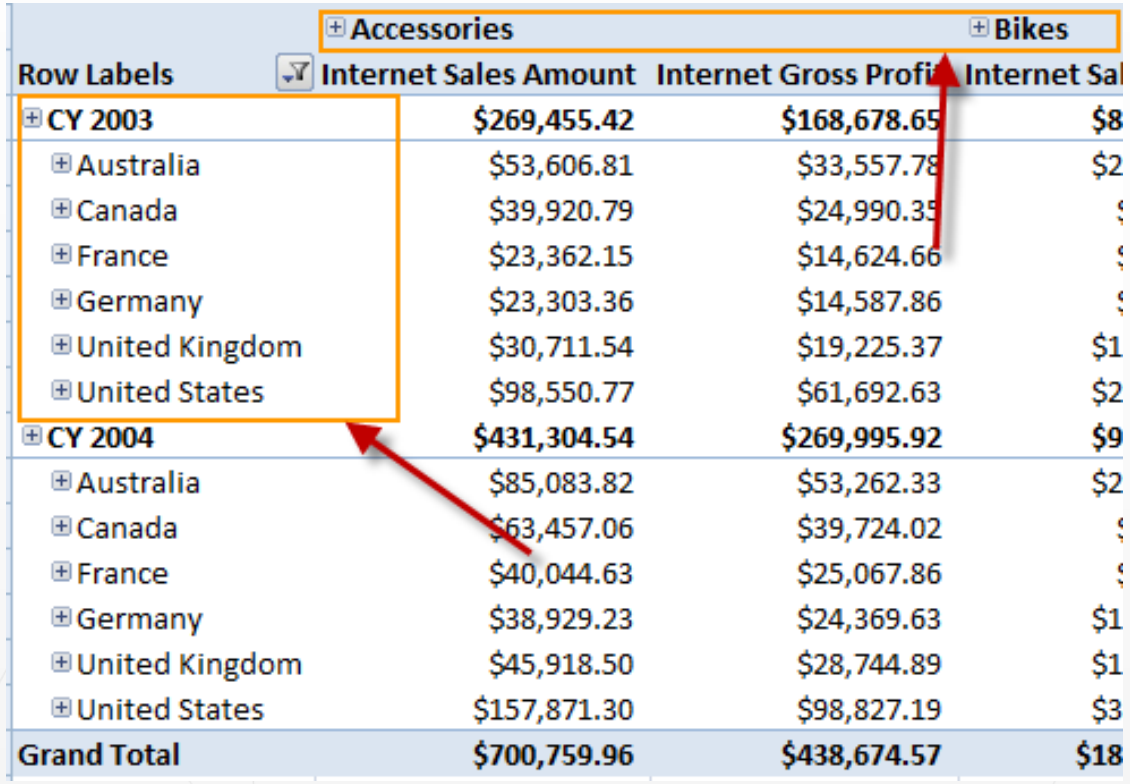


PRAGMATIC WORKS

Defining Dimension

"Dimensions provide the "who, what, where, when, why, and how" context surrounding a business process event.

- Ralph Kimball



	Accessories		Bikes
Row Labels	Internet Sales Amount	Internet Gross Profi	Internet Sal
CY 2003	$269,455.42	$168,678.65	$8
Australia	$53,606.81	$33,557.78	$2
Canada	$39,920.79	$24,990.35	
France	$23,362.15	$14,624.66	
Germany	$23,303.36	$14,587.86	
United Kingdom	$30,711.54	$19,225.37	$1
United States	$98,550.77	$61,692.63	$2
CY 2004	$431,304.54	$269,995.92	$9
Australia	$85,083.82	$53,262.33	$2
Canada	$63,457.06	$39,724.02	
France	$40,044.63	$25,067.86	
Germany	$38,929.23	$24,369.63	$1
United Kingdom	$45,918.50	$28,744.89	$1
United States	$157,871.30	$98,827.19	$3
Grand Total	$700,759.96	$438,674.57	$18

PRAGMATIC WORKS

Dimension Architecture

- Wide Table
- Surrogate Key (Unique ID)
- Natural Key
- Best Attributes are Desciptive
- Start Date / End Date
- Flags

PRAGMATIC WORKS

Understanding and Defining Relationships

What is a Data Model | Creating Relationships | Filtering

PRAGMATIC WORKS

What is a Data Model?

Data Model

Semantic Layer used for Power BI Reports

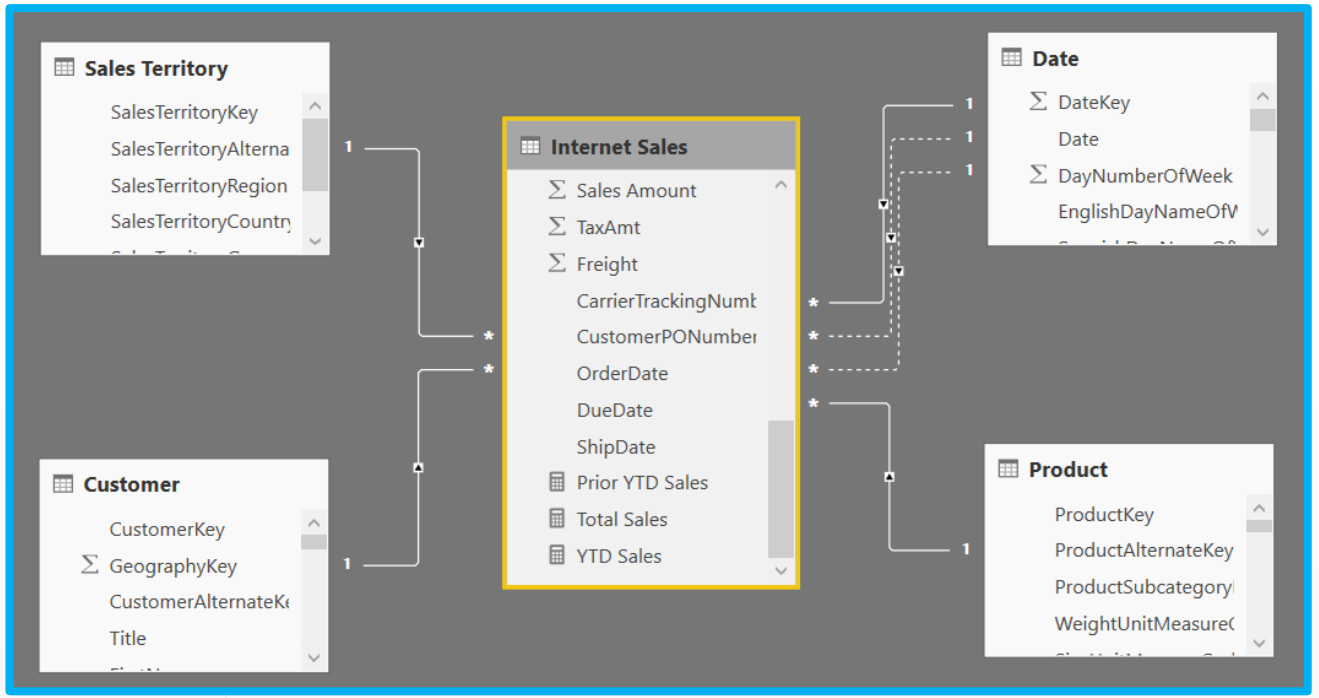
Logical Model vs Physical Model

Logical Model - How Business Looks at the Data

Physical Model - How the Data is Actually held in the Database

Tables and Relationships

Defines Tables and How They Relate to each Other



PRAGMATIC WORKS

Creating Table Relationships

Relationship Auto Detection

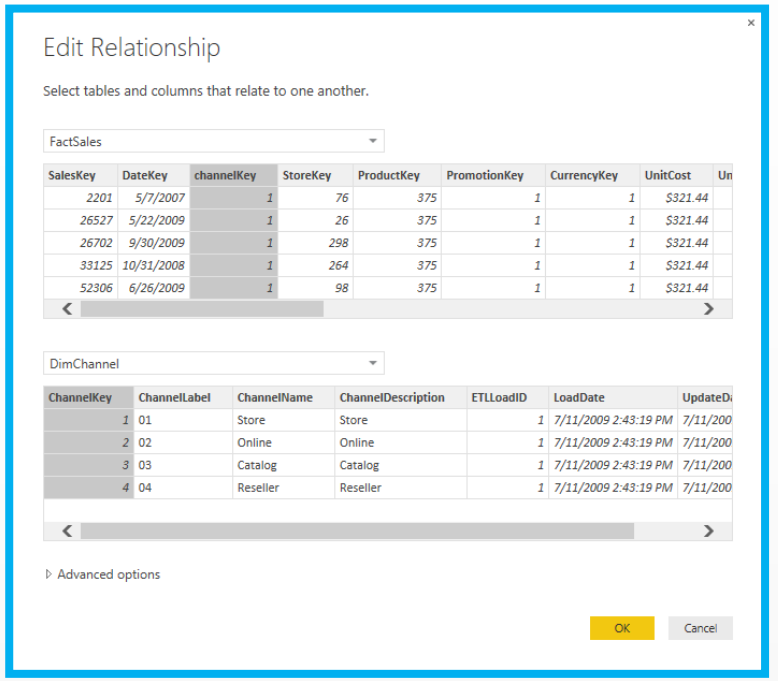
- Verify Auto Detected Relationships

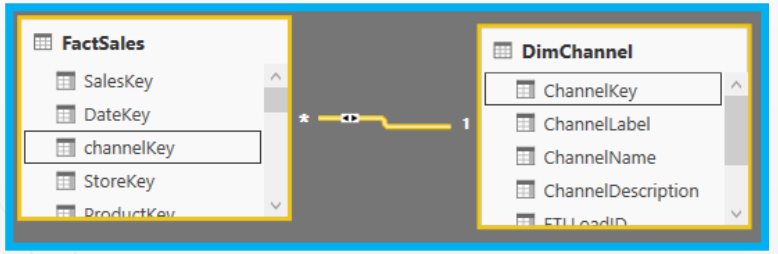
Manually Define Relationships

- Data Types
- Concatenated Key Relationships

Relationship Types

- Role Playing Tables
- Active and Inactive Relationships





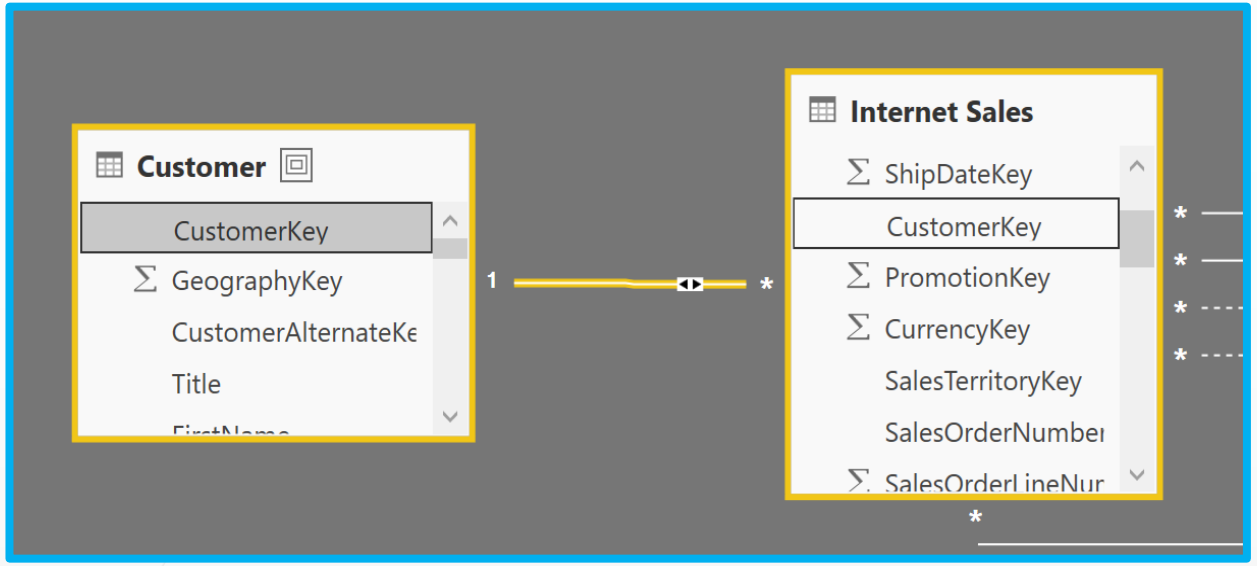
PRAGMATIC WORKS

Relationships in Context

Filtering

Descriptions filter metrics

Metrics do **NOT** filter descriptions



Bi-Directional Relationships

Filter Context

PRAGMATIC WORKS

DAX Fundamentals

How Calculations are Stored | Columns vs Measures
Query Editor vs DAX | Best Practices

PRAGMATIC WORKS

Data Model Engine

Performance

Uses In-Memory technology that provides unmatched performance

Short Development Cycles

Quick and easy to create solutions

End User Capable

Uses query language called DAX that is similar to Excel formulas

PRAGMATIC WORKS

Introduction to DAX

What is DAX?

- Data Analysis Expression Language
- Expression Language for Power BI

Why DAX?

- Designed to Support a Larger User Base
- Simpler than Traditional Technical Languages to Learn
- Leverage Existing Knowledge of Excel Formulas
- Less of a Learning Curve for Analyst

PRAGMATIC WORKS

What is DAX Used For?

Calculated Columns

- Creates New Columns in the Model
- Method for Connecting Disparate Data Sources with Multiple Key Columns

Calculated Measures

- Creates Aggregate Calculations for Reporting
- Handling Complex Relationships (Role-Playing)
- Time Intelligence

PRAGMATIC WORKS

DAX vs. M

Calculated Columns

- Adding new columns to the data model
- Customer Age / Age Breakdown / Full Name / Address

DAX vs. M

- DAX happens after sorting and compression
- M happens before

Creating Calculated Columns

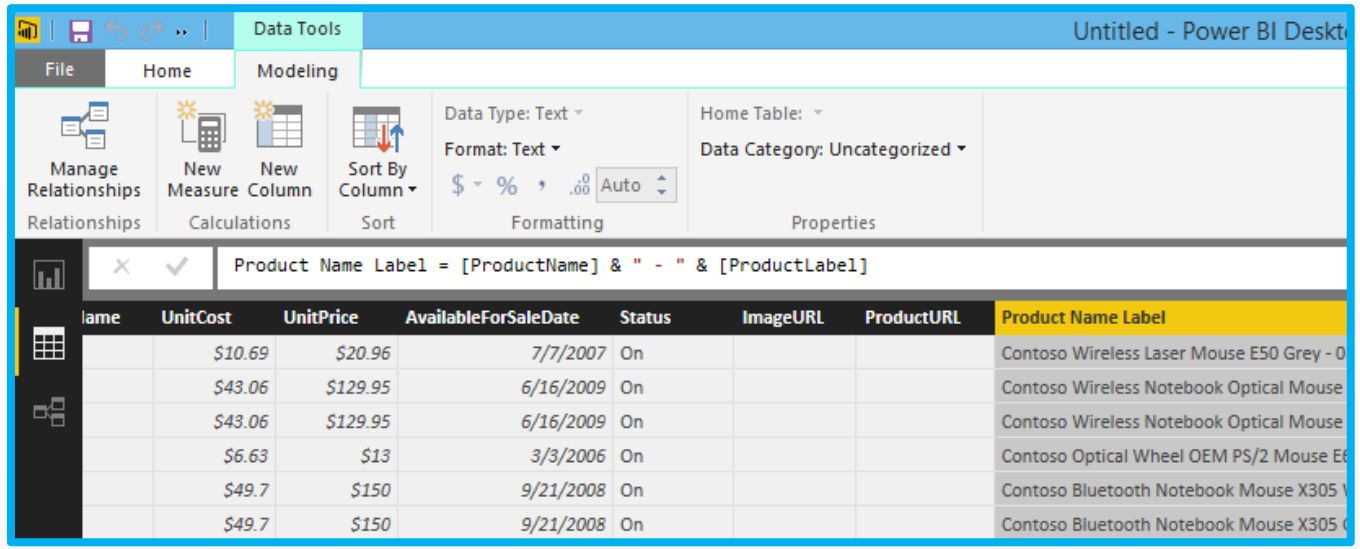
Functions for String Manipulation

PRAGMATIC WORKS

Calculated Columns

Creating Calculated Columns

- Uses DAX Expressions
- See Results Immediately when Defined



Data Types

- Whole Number
- Decimal Number
- Currency
- True/False
- Text
- Date
- Binary

PRAGMATIC WORKS

Functions for String Manipulation

Formatting
- FORMAT

Concatenating
- CONCATENATE

Casing
- LOWER
- UPPER

Trimming
- LEFT
- RIGHT
- TRIM

Splitting
- MID

Searching
- SEARCH
- FIND

Replacing
- REPLACE
- SUBSTITUTE

PRAGMATIC WORKS

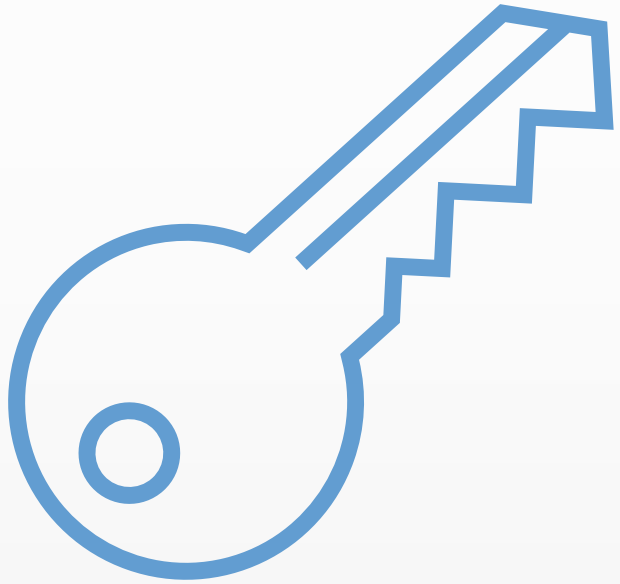
Concatenated Keys

Connecting Disparate Data Sources

Concatenate multiple columns to create key column

Bridge Table

Alternative to Concatenated Key



PRAGMATIC WORKS

Calculated Columns

Demonstration

Let's Review

Creating Calculated Columns

String Manipulation Functions

Creating Calculated Measures in DAX

Creating Aggregates | Time Intelligence | Role Playing Dimensions

PRAGMATIC WORKS

Calculated Measures

Implicit Measures

Default aggregation that occurs on numeric data

Explicit Measures

A user defined measure created by writing a DAX formula

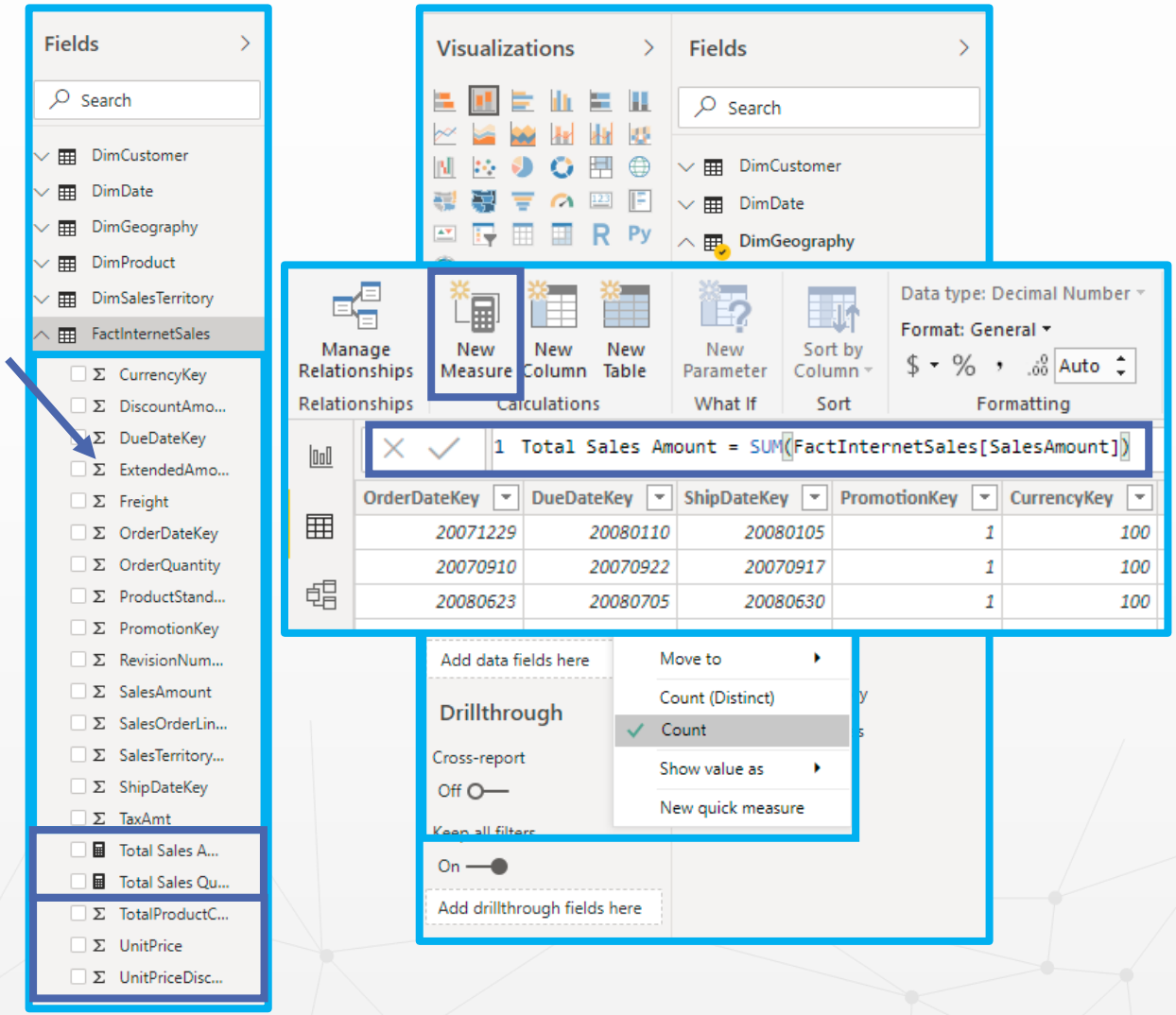
Benefits of Explicit Measures

Can build on one another

Encapsulate Code, making logic changes easier to implement

Can centrally define the formatting of results

Results are dynamic



PRAGMATIC WORKS

Creating Aggregates

Statistical

AVERAGE

AVERAGEA

AVERAGEX

COUNT

COUNTA

COUNTAX

COUNTBLANK

COUNTROWS

COUNTXMAX

MAXA

MAXX

MIN

MINA

MINX

SUM

SUMX

PRAGMATIC WORKS

Filter Context

CALCULATE

PRAGMATIC WORKS

Calculate

Why Calculate?

Only Function in DAX that can change a filter context.

Apply a filter(s) to an expression.

Ratios / Percent of Totals

Syntax

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

Expression is an aggregate

Filters Add to or Override Filter Context

```
ResellerSales:=CALCULATE([TotalSales],Store[StoreType]="Reseller")
```

```
AllProductSales:=CALCULATE([TotalSales], ALL(Product))
```

PRAGMATIC WORKS

ALL

ALL() –

Returns all the rows in a table, or all the values in a column ignoring any filters that might have been applied.

ALLEXCEPT() –

Exceptions to specific columns

ALLSELECTED() –

Like ALL(), excepts keeps *current* context

PRAGMATIC WORKS

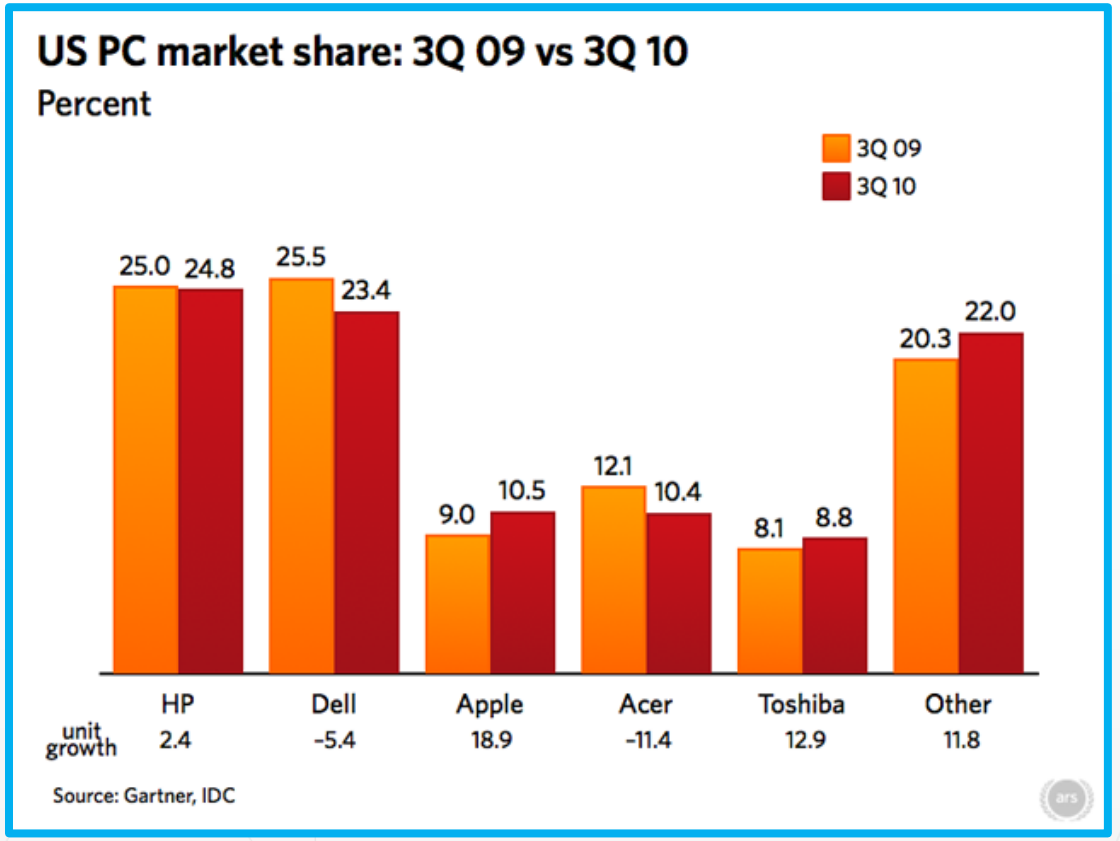
Built-In Time Intelligence

PRAGMATIC WORKS

Using Time Intelligence

Comparing Data over Time

- Year-to-Date
- Year Over Year Growth
- Using a Date Table



PRAGMATIC WORKS

Incorporating Time Intelligence

Time Intelligence

CLOSINGBALANCEMONTH
CLOSINGBALANCEQUARTER
CLOSINGBALANCEYEAR
DATESINPERIOD
DATESBETWEEN
DATEADD
FIRSTDATE
LASTDATE
LASTNONBLANKDATE
STARTOFMONTH
STARTOFQUARTER
STARTOFYEAR
ENDOFMONTH
ENDOFQUARTER
ENDOFYEAR
PARALLELPERIOD
PREVIOUSDAY
PREVIOUSMONTH
PREVIOUSQUARTER
PREVIOUSYEAR
NEXTDAY
NEXTMONTH
NEXTQUARTER
NEXTYEAR
DATESMTD
DATESQTD
DATESYTD
SAMEPERIODLASTYEAR
OPENINGBALANCEMONTH
OPENINGBALANCEQUARTER
OPENINGBALANCEYEAR
TOTALMTD
TOTALQTD
TOTALYTD

PRAGMATIC WORKS

Creating Date Tables

Date Table Requirements

One Row for Every Date

Span Range of Possible Dates

Import From Source or Create a new table in DAX

Mark as a Date Table in Model (Tabular)

Date Column Passed to Time Intelligence Functions

Datekey	FullDateLabel
1/1/2005 12:00:00 AM	2005-01-01
1/2/2005 12:00:00 AM	2005-01-02
1/3/2005 12:00:00 AM	2005-01-03
1/4/2005 12:00:00 AM	2005-01-04
1/5/2005 12:00:00 AM	2005-01-05
1/6/2005 12:00:00 AM	2005-01-06
1/7/2005 12:00:00 AM	2005-01-07
1/8/2005 12:00:00 AM	2005-01-08
1/9/2005 12:00:00 AM	2005-01-09
1/10/2005 12:00:00 AM	2005-01-10
1/11/2005 12:00:00 AM	2005-01-11

PRAGMATIC WORKS

Cross Filtering

Filtering

Automatically occurs from the single side of the relationship

Products

ProdID	Name	ListPrice
A	Bike	250
B	Socks	10
C	Shoes	120
D	Helmet	150
E	Gloves	45

Internet Sales

DateSK	ProdID	Amt	QTY
20080101	B	250	1
20080101	A	20	2
20080102	A	45	1
20080103	E	45	1
20080103	D	150	1
20080104	D	150	1
20080105	A	250	1

Date

DateSK	Date
20080101	1/1/2008
20080102	1/2/2008
20080103	1/3/2008
20080104	1/4/2008
20080105	1/5/2008

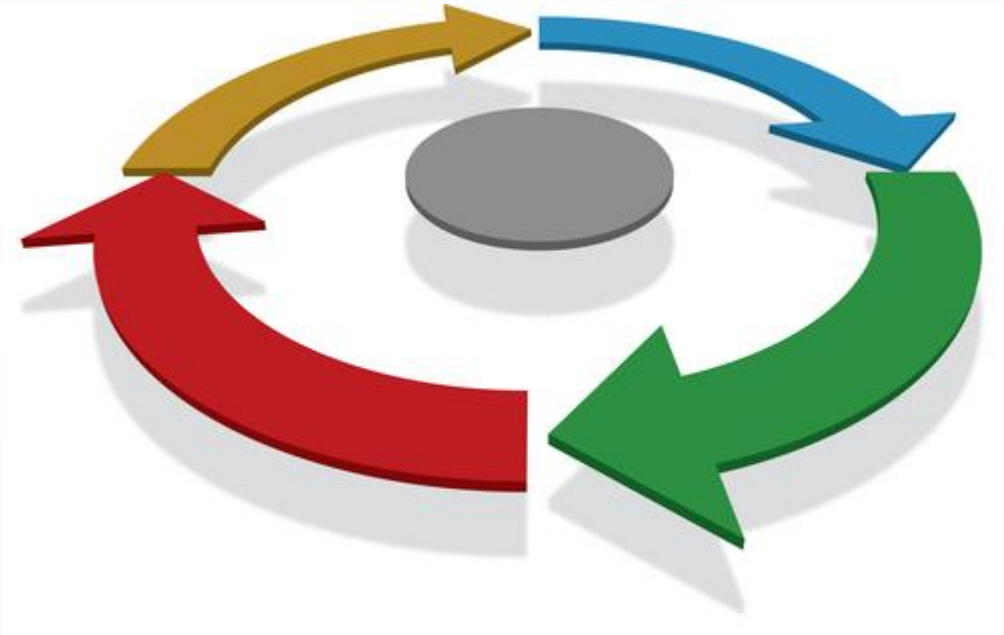
PRAGMATIC WORKS

Iterative Functions

X-Functions | Filter | Working with Totals

PRAGMATIC WORKS

Iterative Functions



Iterators

- Table or Table Expression
- Create Row Context
- Row by Row operation
- X-Functions
- Filter Function

PRAGMATIC WORKS

DAX Calculation

```
xxxX(
        'Table',
        [Calculation]
)
```

PRAGMATIC WORKS

Table Functions

Calendar | Filter

PRAGMATIC WORKS

Creating Date Tables

CALENDAR

Accepts two parameters

CALENDARAUTO

Determined by data set

PRAGMATIC WORKS

Calculated Tables

Calculated Tables

- Validation and debugging of Code
- Role Playing Dimensions
- Simplify DAX formulas

Limitation

- Memory

PRAGMATIC WORKS

Filter

FILTER

Returns a table (Table Function)

Iterates a table

FILTER(<table>, <condition>

PRAGMATIC WORKS

Filter vs. CalculateTable

Filter

Returns a table that represents a subset of another table or expression.

```
Filter(
        <table>,
        <filter>
        )
```

CalculateTable

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

```
CalculateTable(
        <expression>,
        <filter>
        )
```

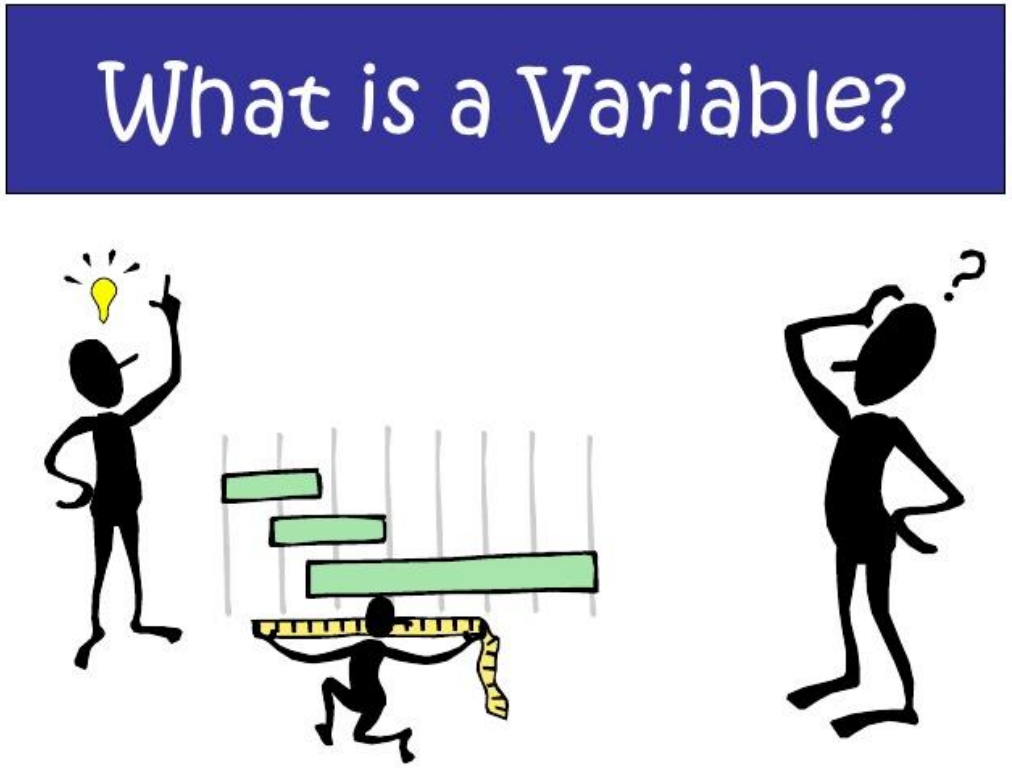
PRAGMATIC WORKS

Working with Variables

Variables

PRAGMATIC WORKS

Variables



Reusable Code

Easier to write and maintain

Increased readability

Performance

PRAGMATIC WORKS

More about Variables!

Syntax

VAR

<Variable Name> = <expression>

RETURN

Where can variables be used?

Scalar Expressions

Table Expressions

SQL Server 2016

Working with Variables

Demonstration

Totals

Filter Context | HASONEVALUE | BLANK()

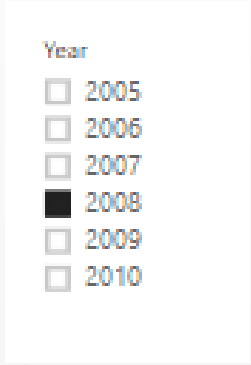
PRAGMATIC WORKS

Confusing Totals

Questions

Why does Forecast YTD Sales and Dynamic Measure have the same total?

How come the totals do not add up to the sum of the rows?



Year	Month	YTD Sales	Forecast YTD Sales	Dynamic Measure
2008	January	$1,340,244.95	$877,730.34	$1,340,244.95
2008	February	$2,802,724.78	$1,855,911.01	$2,802,724.78
2008	March	$4,283,629.96	$2,827,060.60	$4,283,629.96
2008	April	$5,892,380.49	$3,839,859.13	$5,892,380.49
2008	May	$7,770,698.00	$4,965,404.26	$7,770,698.00
2008	June	$9,062,685.56	$6,075,002.72	$6,075,002.72
Total		**$9,062,685.56**	**$19,582,120.60**	**$19,582,120.60**

PRAGMATIC WORKS

Working with Totals

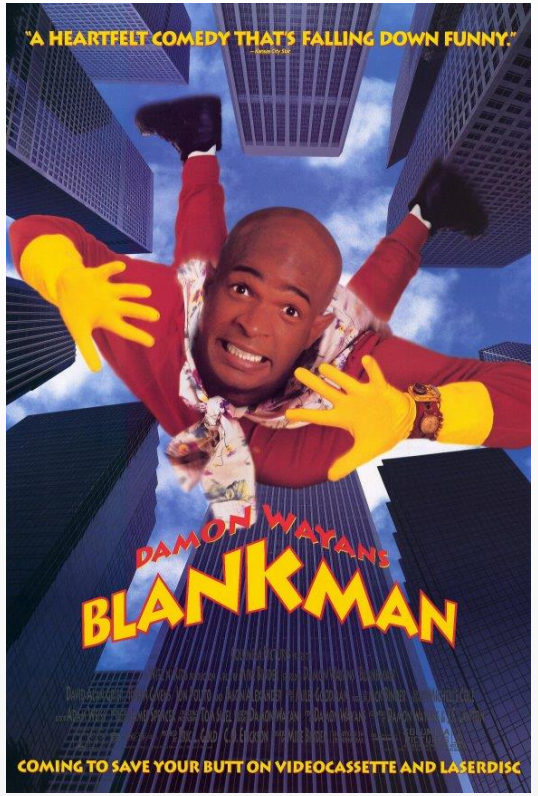
HASONEVALUE

Returns **TRUE** when the context for *columnName* has been filtered down to one distinct value only. Otherwise is **FALSE**.

BLANK()

Returns a blank

```
IF(
    HASONEVALUE('Table'[Column],
    (Perform Calculation),
    BLANK()
)
```



PRAGMATIC WORKS

Working with Totals

Demonstration

Filter Context

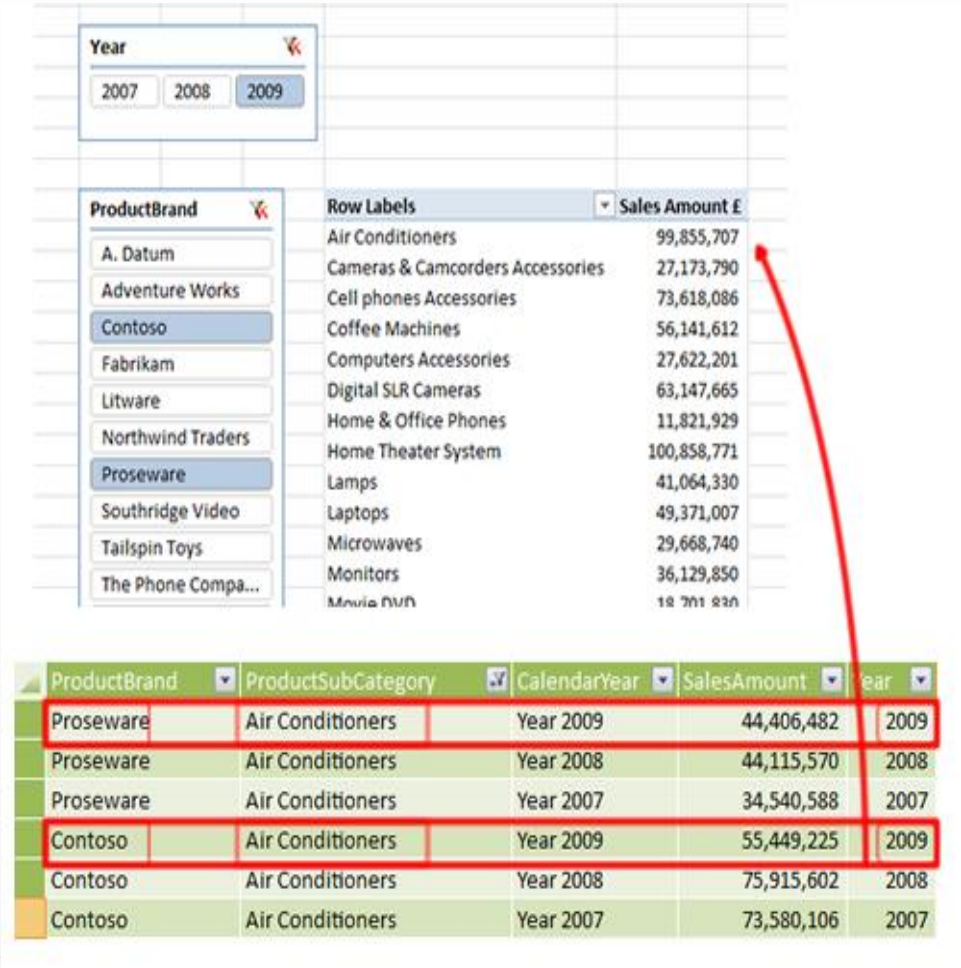
What is Filter Context?

PRAGMATIC WORKS

Filter Context

Applied by Filter Constraints

- Row and Column Headers
- Slicers and Filters
- DAX Formula Filters



PRAGMATIC WORKS

Row Context

Calculated Columns | Iterators | Nested Row Context

PRAGMATIC WORKS

Row Context

Row Context

Filter applied at the row level

Automatic Filtering does not occur

Calculated Columns

Iterator Functions

Context Transition

Context Transition

Row context transitions into a filter context

CALCULATE

Automatically

Explicit

Implicit

Row Context

Demonstration

PRAGMATIC WORKS

Semi-Additive Measures

FIRSTDATE | FIRSTNONBLANK | OPENINGBALANCE

PRAGMATIC WORKS

Semi-Additive Measures

Inventory Levels

Account Balances

PRAGMATIC WORKS

NONBLANK Functions

FIRSTDATE

LASTDATE

FIRSTNONBLANK

LASTNONBLANK

FIRSTDATE and LASTDATE

Demonstration

Role Playing Dimensions

PRAGMATIC WORKS

Working with Multiple Dates

Date Table

Date

Sales Data

Order Date

Shipped Date

Due Date

PRAGMATIC WORKS

Working with Role Playing Dimensions

PRAGMATIC WORKS

Navigating Role Playing Dimensions with DAX

Use DAX USERELATIONSHIP Function

Create a Calculated Measure for Every Value you Need to use Inactive Relationship

Alternative to DAX

Import Same Table Multiple Times

Rename Each Table Appropriately

Create a Single Relationship Between Tables

Role Playing – Multiple Tables

Demonstration

Advanced Data Modeling

Role Playing Tables

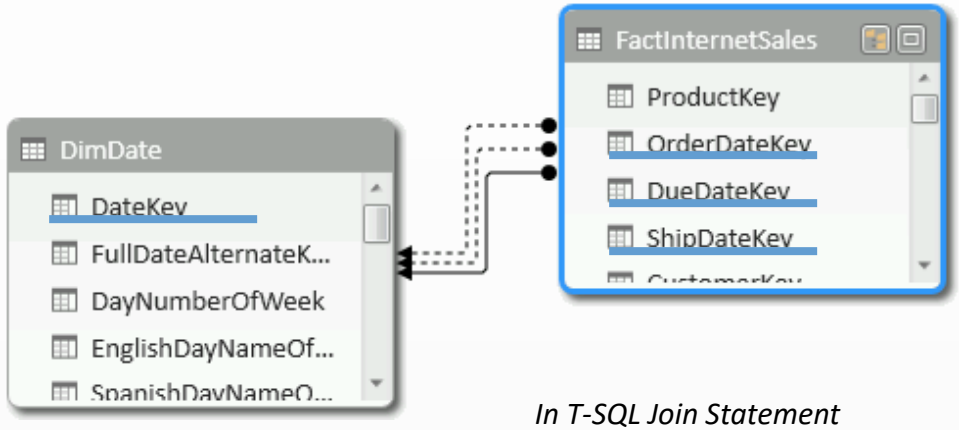
PRAGMATIC WORKS

Working with Role Playing Tables

What is a Role Playing Table?

Two Tables that have Multiple Relationship with Each Other

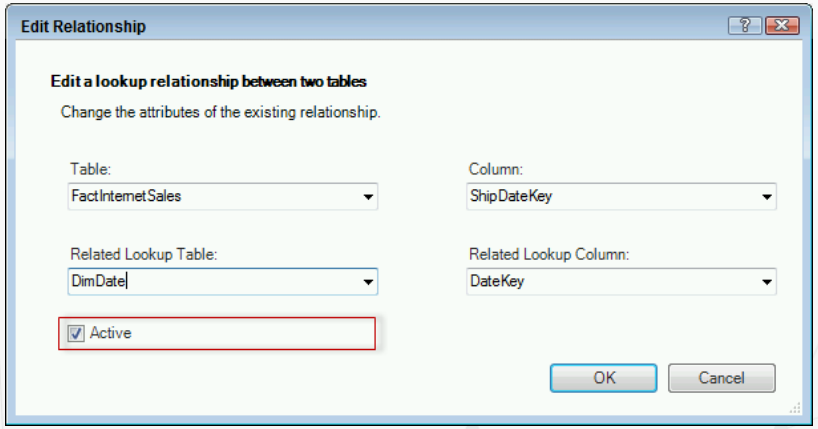
Reduces Redundancy in Database



In T-SQL Join Statement Determines Which Date to Use

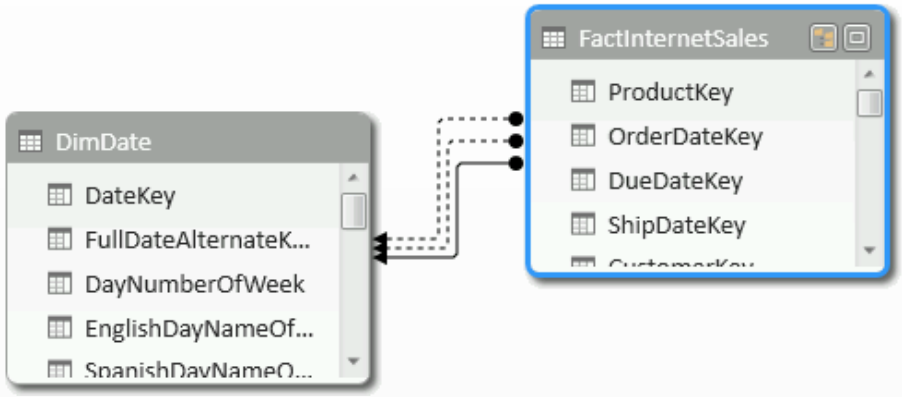
Role Playing in Power Pivot

Active/Inactive Relationships

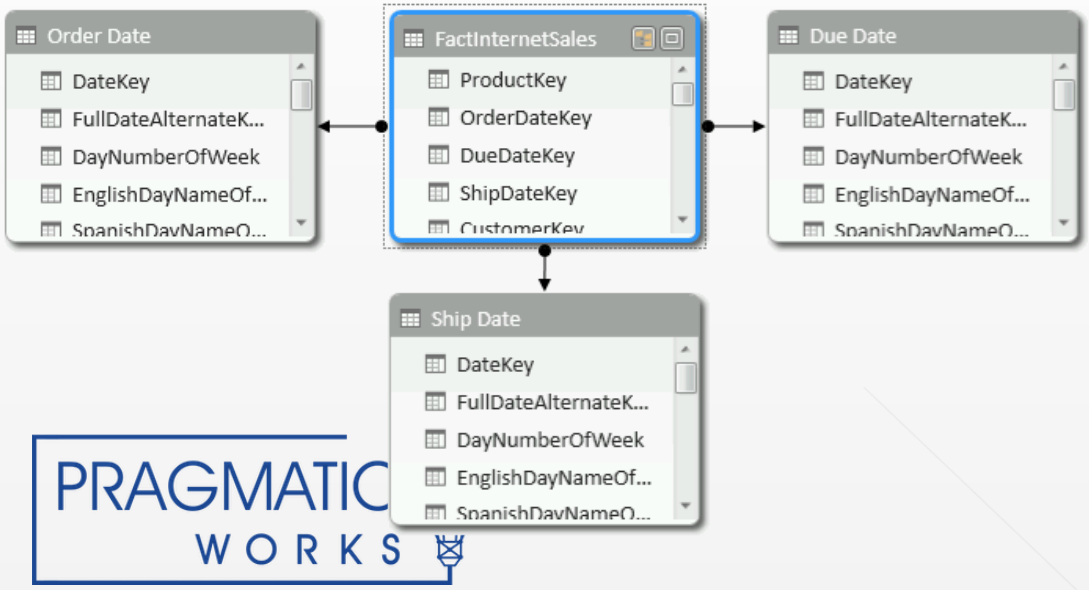


PRAGMATIC WORKS

Working with Role Playing Tables




```
=CALCULATE([SalesAmount], USERELATIONSHIP(DimDate[DateKey],
FactInternetSales[ShipDate])
```



PRAGMATIC WORKS

Navigating Role Playing Tables with DAX

Use DAX USERELATIONSHIP Function

Create a Calculated Measure for Every Value you Need to use Inactive Relationship

Alternative to DAX

Import Same Table Multiple Times

Rename Each Table Appropriately

Create a Single Relationship Between Tables

Role Playing Tables with DAX

Demonstration

PRAGMATIC WORKS

Advanced Data Modeling

Importing Multiple Date Tables

PRAGMATIC WORKS

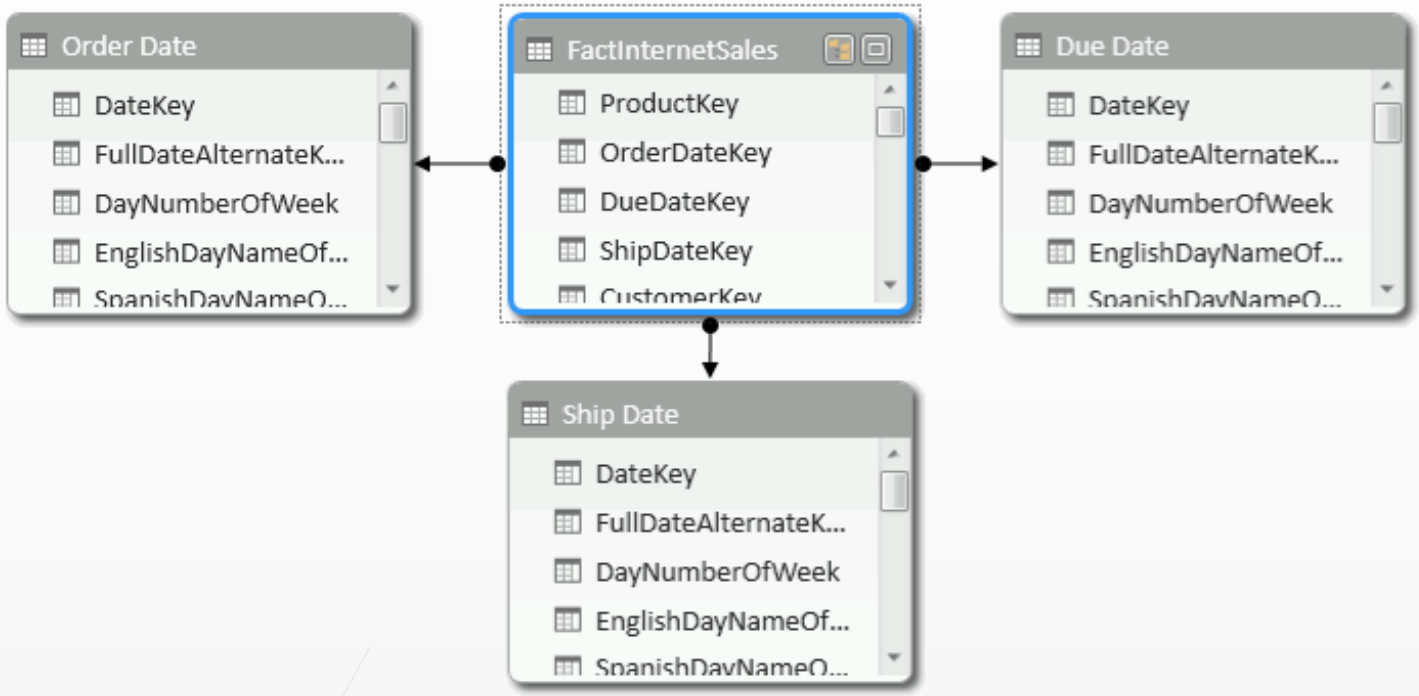
Working with Role Playing Tables

Alternative to DAX

Import Same Table Multiple Times

Rename Each Table Appropriately

Create a Single Relationship Between Tables



PRAGMATIC WORKS

Importing Multiple Date Tables

Demonstration

PRAGMATIC WORKS

Advanced Data Modeling

Weighted Allocation | Many to Many

PRAGMATIC WORKS

Weighted Allocation

Many to Many

- Double
- Over Counting
- Inaccurate reports

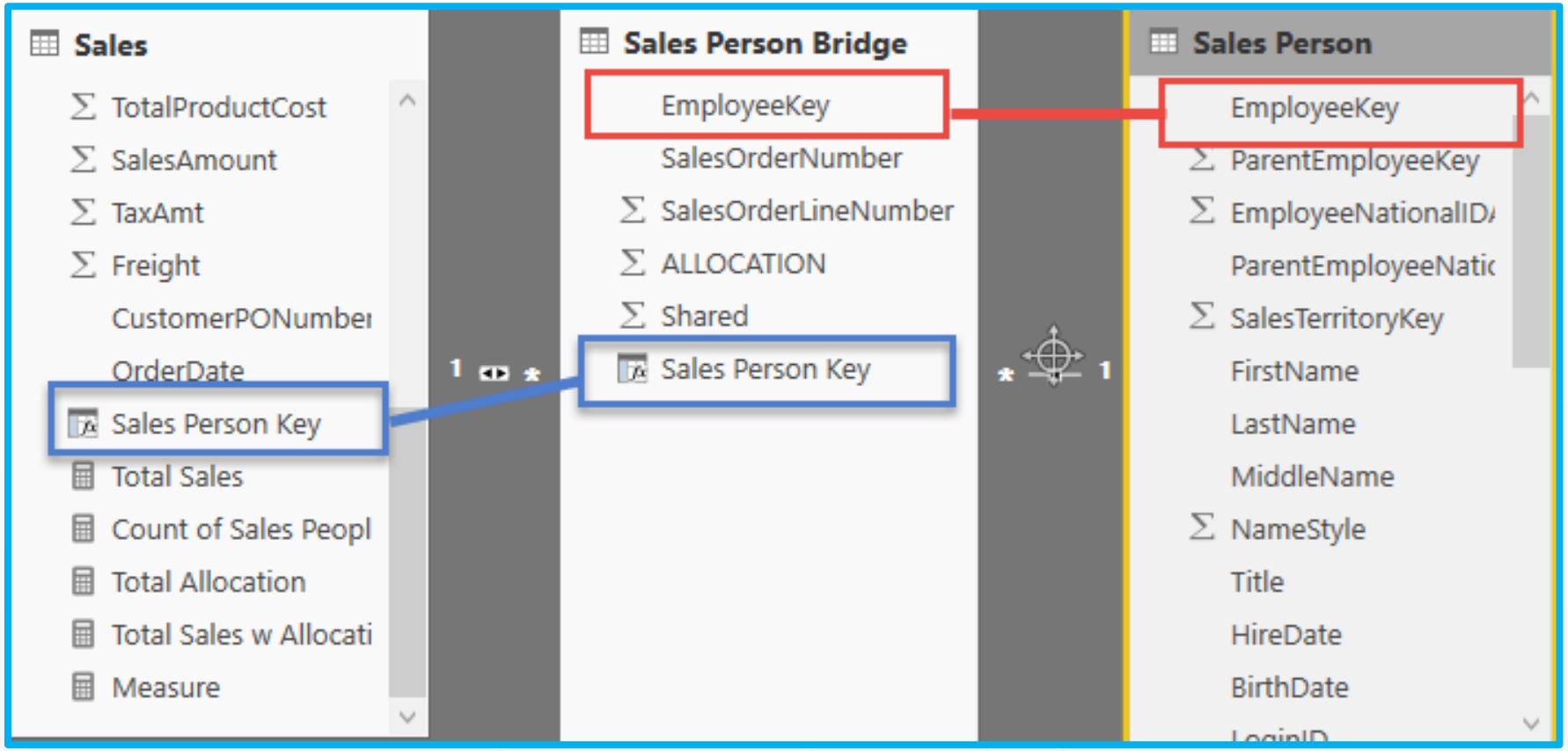
PRAGMATIC WORKS

Bridge Table

EmployeeKey	SalesOrderNumber	SalesOrderLineNumber	ALLOCATION	Shared
282	SO69471	11	0.3	1
282	SO69532	2	0.3	1
282	SO69532	6	0.3	1
282	SO69561	18	0.3	1
282	SO69561	42	0.3	1
282	SO71783	7	0.3	1
282	SO71783	29	0.3	1
282	SO71796	7	0.3	1
282	SO71902	20	0.3	1
272	SO43898	18	0.7	1
272	SO44129	19	0.7	1
272	SO46614	21	0.7	1
272	SO46638	34	0.7	1
272	SO46640	13	0.7	1

PRAGMATIC WORKS

Relationship



PRAGMATIC WORKS

DAX Calculation

```
SUMX(
        'Sales Person Bridge',
        [Total Sales] *
        'Sales Person Bridge'[Allocation])
```

PRAGMATIC WORKS

Weighted Allocation

Demonstration

PRAGMATIC WORKS

xVelocity Engine *(Vertipaq)*

PRAGMATIC WORKS

Data Model Engine

Performance

Uses In-Memory technology that provides unmatched performance

Short Development Cycles

Quick and easy to create solutions

End User Capable

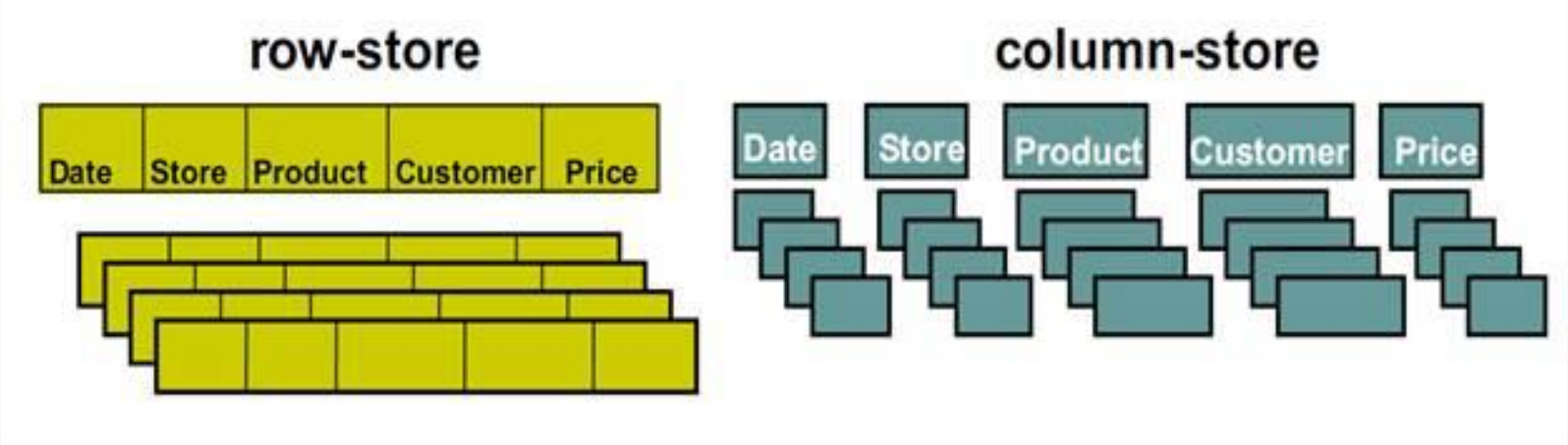
Uses query language called DAX that is similar to Excel formulas

PRAGMATIC WORKS

The xVelocity Engine

What Makes xVelocity Special?

- Column-Store Data
- Data Compression



PRAGMATIC WORKS

Compression

xVelocity Compression

Column Compression

Much higher ratio than traditional compression techniques

Value / Dictionary / Run Length Encoding

Benefits of improved compression:

Better use of existing hardware

Faster query performance

How xVelocity works

Columnar database

Loads and reads from memory

High CPU usage lower IO

Columnar storage

Date	Name	Product	Price
1/1/2017	Devin	Bike	14.99
1/1/2017	Devin	Helmet	19.56
1/4/2017	Manuel	Bike	20.15

Each column has its own data structure

Value Encoding

Only works on integer columns

Does not work on floats or text

Account Number
5000001
5000002
5000003
5000004
5000005
5000799

Account Number
1
2
3
4
5
799

PRAGMATIC WORKS

Dictionary Encoding

Category
0
1
1
2
2
2
3
0

Category_ID	Category
0	Bikes
1	Accessories
2	Clothing
3	Components

Category
Bikes
Accessories
Accessories
Clothing
Clothing
Clothing
Components
Bikes

PRAGMATIC WORKS

Run Length Encoding

Complimentary

Avoid Repeated Values

Depends on repetition pattern of column.

Sorting of data

Run Length Encoding does not always occur.

Run Length Encoding

Age Breakdown	Count of Rows

Age Breakdown
18-34
18-34
18-34
18-34
…
65+
18-34
65+
…
35-64
35-64
35-64

Rows 1-15 are all 18-34

Rows 16 is 65+

Rows 17 is 18-34

Rows 18-35 are all 65+

Rows 36-38 are all 35-64

PRAGMATIC WORKS

DAX vs. M

Calculated Columns

Adding new columns to the data model

Customer Age / Age Breakdown / Full Name / Address

DAX vs. M

DAX happens after sorting and compression

M happens before

PRAGMATIC WORKS

xVelocity

Vertipaq Analyzer

PRAGMATIC WORKS

Analyzing compression

VertiPaq Analyzer

Power Pivot model

Uses existing DMVs

Analyze Vertipaq storage structures for a data models

SQLBI.com

PRAGMATIC WORKS

Role Playing Tables with DAX

Demonstration

PRAGMATIC WORKS

xVelocity

Column Cardinality

PRAGMATIC WORKS

Cardinality

Column Uniqueness

- Address
- Full Name

To remove or not to remove?

PRAGMATIC WORKS

Column Cardinality

Demonstration

PRAGMATIC WORKS