www.sqlbi.com

sqlbi

a brand of Loader

Microsoft Partner
Gold Business Intelligence
Gold Data Platform

MVP Microsoft Most Valuable Professional

SSAS MAESTRO by Microsoft

Alberto Ferrari
alberto.ferrari@sqlbi.com
twitter @FerrariAlberto

Advanced Modeling with Analysis Services Tabular

sqlbi

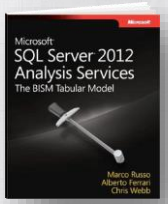
Who's Speaking?

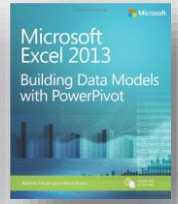
sqlbi

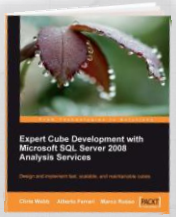
a brand of

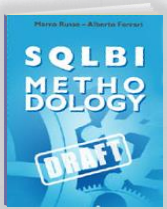
Loader

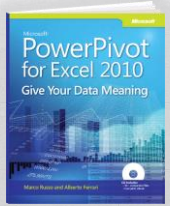
- BI Expert and Consultant
- Founder of www.sqlbi.com
  - Problem Solving
  - Complex Project Assistance
  - Data Warehouse Assessments and Development
  - Courses, Trainings and Workshops
- Book Writer
- Microsoft Business Intelligence Partner











Microsoft Partner
Gold Data Platform
Gold Business Intelligence

MVP
Microsoft
Most Valuable
Professional

SSAS
MAESTRO
by Microsoft

sqlbi

Tabular and Dimensional Models

- Multidimensional
  - Based on Kimball methodology
  - Facts, Dimensions
- Tabular
  - Based on relational modeling
  - Tables, relationships
  - Calculations (calc columns, measures)

sqlbi

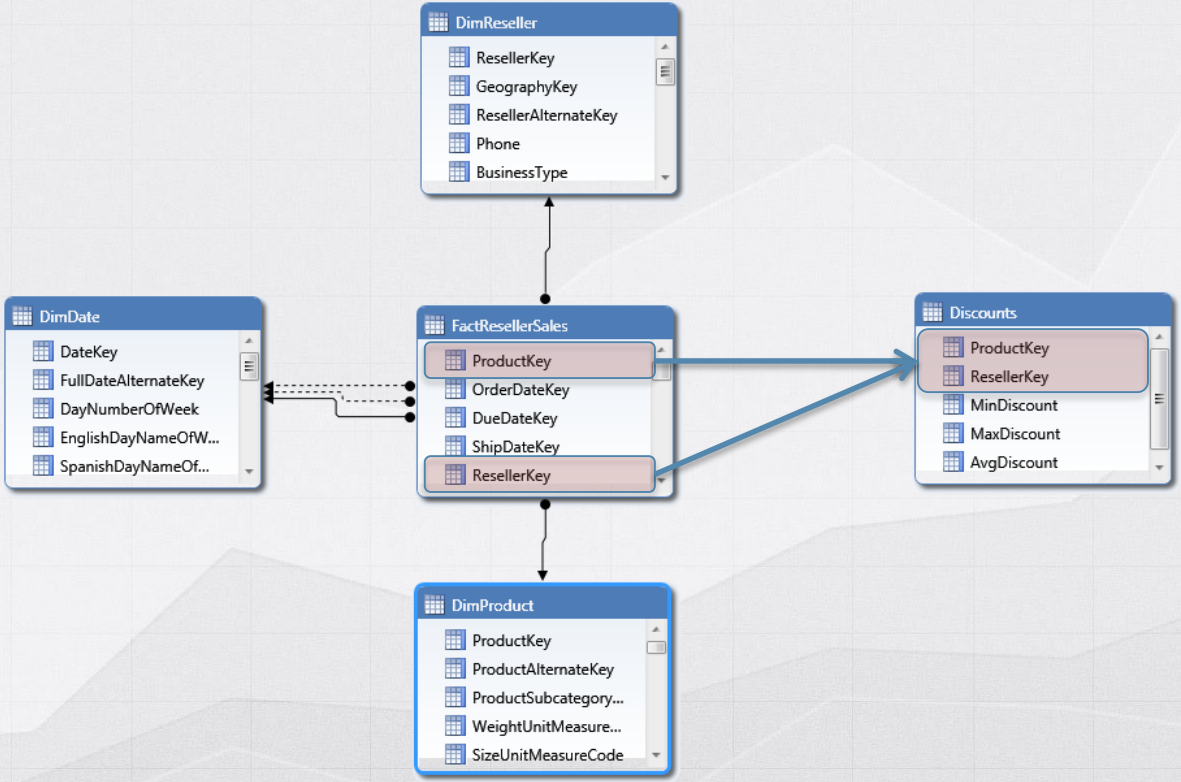
Build a new mindset

- Classical patterns, different solutions
  - Static and Dynamic Segmentation
  - Survey Analysis
  - Basket Analysis
  - Currency Conversion
- These scenarios have different solutions in Tabular and Multidimensional

sqlbi

Relationships

Relationships with many columns



sqlbi

Build a Relationship

String concatenation to build a new key

```
ProductAndReseller =

Discounts[ProductKey] & "-" & Discounts[ResellerKey]
```

ProductKey	ResellerKey	MinDiscount	MaxDiscount	AvgDiscount	ProductAndReseller
470	2	0.02	0.02	0.02	470-2
214	3	0.02	0.02	0.02	214-3
224	3	0.02	0.02	0.02	224-3
231	3	0.02	0.02	0.02	231-3
233	3	0.02	0.02	0.02	233-3
236	3	0.02	0.02	0.02	236-3
327	3	0.02	0.02	0.02	327-3
333	3	0.02	0.02	0.02	333-3

sqlbi

Calculated Column

LOOKUPVALUE to denormalize attributes in the fact table

```
Discount =

LOOKUPVALUE (
    Discounts[MaxDiscount],
    Discounts[ProductKey], FactResellerSales[ProductKey],
    Discounts[ResellerKey], FactResellerSales[ResellerKey]
)
```

sqlbi

Segmentation

Segmenting the prices

- Price changes over time
  - Discounts
  - Price variations
- Continuous dimension
- High fragmentation
- Segmentation
  - From 0 to 100 USD
  - From 101 to 500

Segment	FromPrice	ToPrice
VERY LOW	0	5
LOW	5	30
MEDIUM	30	100
HIGH	100	500
VERY HIGH	500	9999

sqlbi

Banding: The data model

NO RELATIONSHIP CAN BE CREATED

sqlbi

Banding With DAX

Leverage DAX and create the relationship inside the formula.

```
PriceBand =

CALCULATE (
    VALUES ( PriceBands[PriceBand] ),
    FILTER (
        PriceBands,
        Sales[UnitPrice] >= PriceBands[MinPrice]
     && Sales[UnitPrice] < PriceBands[MaxPrice]
    )
)
```

sqlbi

Dynamic Segmentation

sqlbi

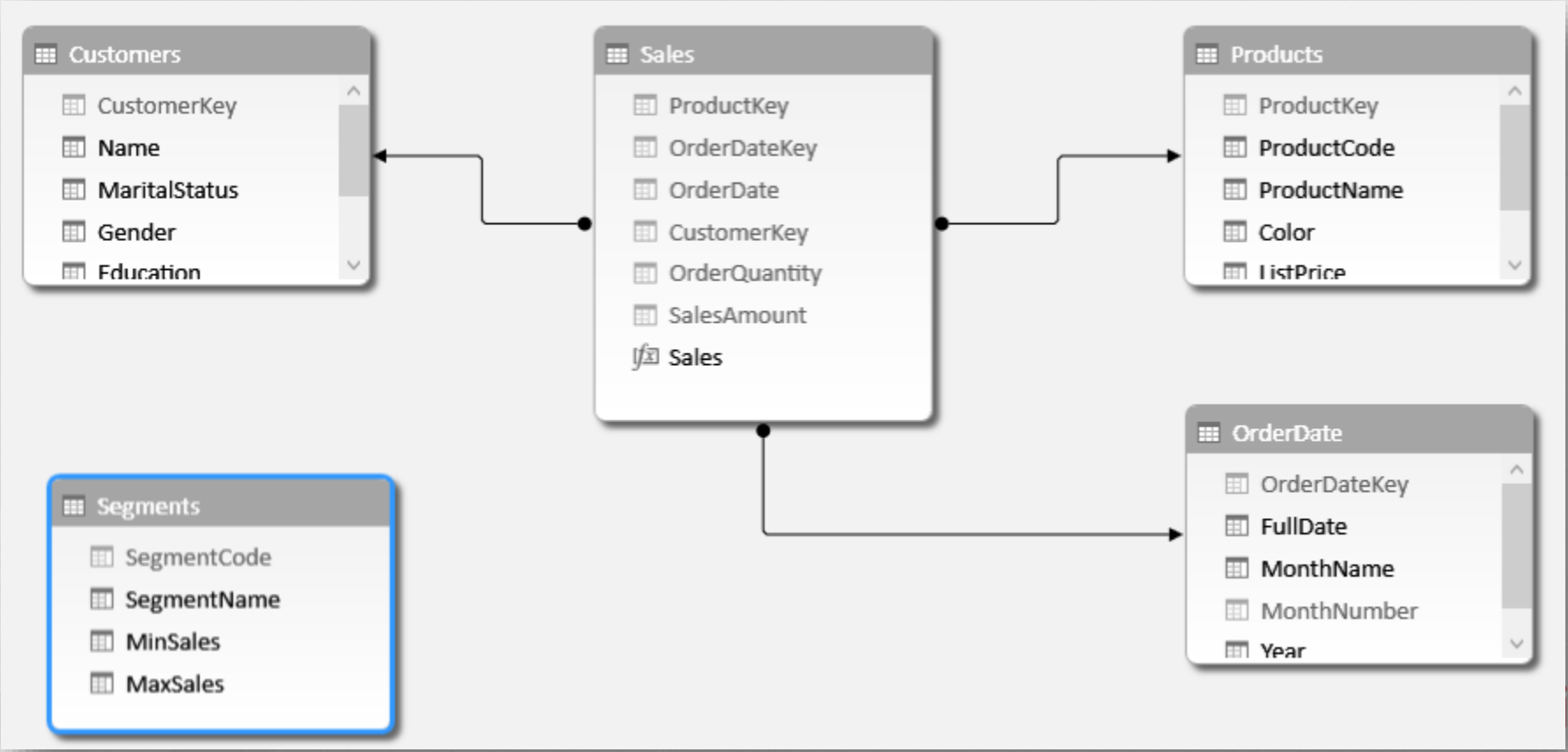
Dynamic Segmentation

SegmentCode	SegmentName	MinSales	MaxSales
1	Very Low	0	100
2	Low	100	1000
3	Medium	1000	5000
4	High	5000	10000
6	Very High	10000	9999999

CustInSegment	Column Labels				
Row Labels	2005	2006	2007	2008	Grand Total
Very Low			3810	5040	7676
Low	113	401	1805	2539	3112
Medium	900	2276	3639	3792	5964
High			54	6	1703
Very High			1		29
Grand Total	**1013**	**2677**	**9309**	**11377**	**18484**

sqlbi

The Data Model



Dynamic Segmentation

This time, the relationship is “virtual”.
It does not exists outside of the formula

```
CustInSegment :=

CALCULATE (
    COUNTROWS ( Customers ),
    FILTER (
        Customers,
        [Sales] > MIN ( Segments[MinSales] ) &&
        [Sales] <= MAX ( Segments[MaxSales] )
    )
 )
```

sqlbi

Survey

sqlbi

Survey: the data

AnswerKey	Question	Answer
1	Sport Practiced	Baseball
2	Sport Practiced	Football
3	Sport Practiced	Tennis
4	Sport Practiced	Karate
5	Job	IT Pro
6	Job	Teacher
7	Job	Consultant
8	Yearly Net Income	< 10,000
9	Yearly Net Income	10,000 - 20,000
10	Yearly Net Income	20,000 - 40,000
11	Yearly Net Income	40,000 - 80,000
12	Yearly Net Income	> 80,000
13	Gender	Male
14	Gender	Female
15	Movies Preferences	Horror
16	Movies Preferences	Comedy
17	Movies Preferences	Cartoons

CustomerKey	AnswerKey
4	5
5	5
11	5
12	5
19	5
25	5
26	5
29	5
34	5
39	5

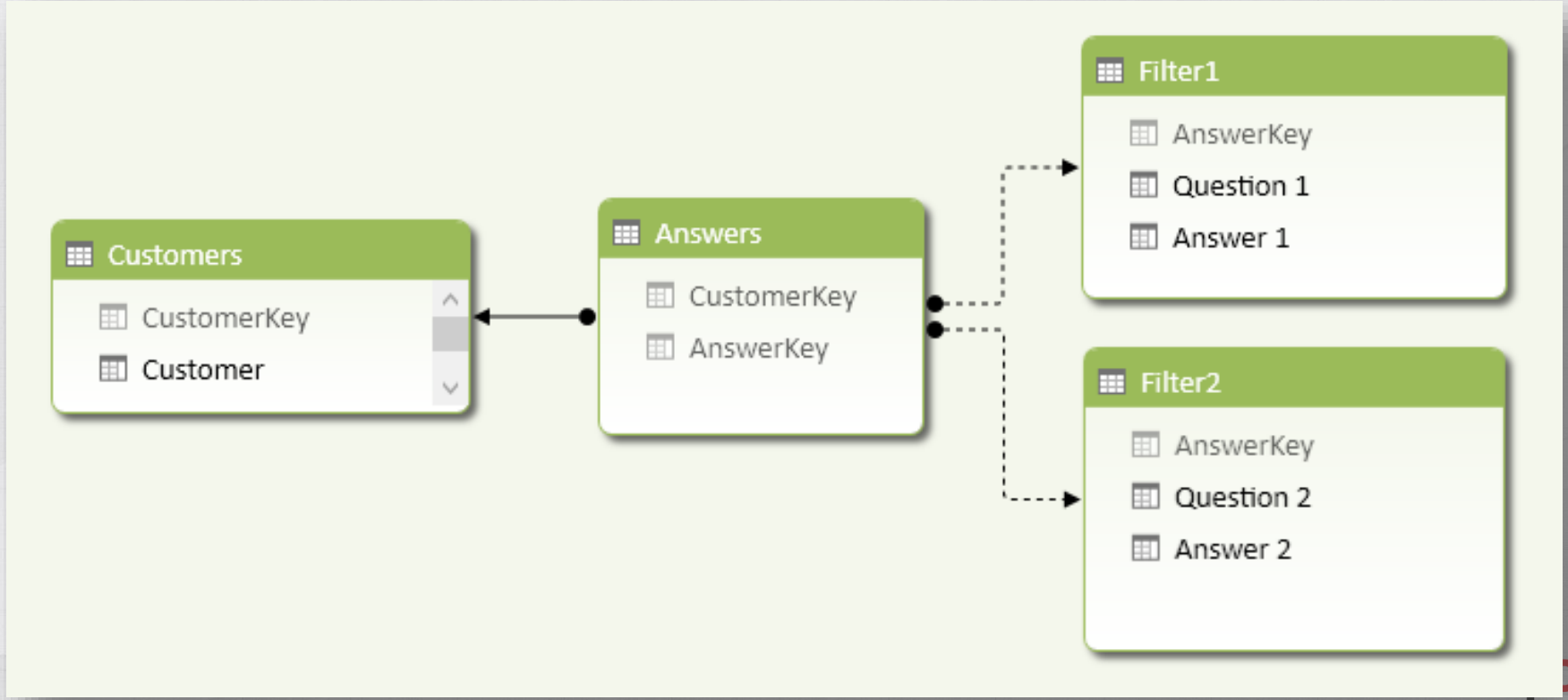
CustomerKey	Customer	Category
1	Regina Mc Gee	Power
2	Dewayne Forbes	Power
3	Jolene Nunez	Power
4	Tricia Ingram	Sports & leisure Infrastructure
5	Shane Anderson	Giftware, Jewellery & Tableware

The question

- How many users answered a specific question?
- How many answered «A» to one question and «B» to another one?
- e.g. how many consultants like horror movies?

sqlbi

Survey: the data model



Disabled Relationships

- Relationships created and deactivated
- Can be enabled on-demand
- USERELATIONSHIP
  - Enables a relationship
  - Disables any conflicting one

sqlbi

Customers Q1

Number of users who answered a specific question

```
=IF (
    HASONEVALUE ( Filter1[Question 1] ),
    CALCULATE (
        COUNTROWS ( Customers ),
        CALCULATETABLE (
            Answers,
            USERELATIONSHIP (
                Answers[AnswerKey],
                Filter1[AnswerKey]
            )
        )
    )
)
```

sqlbi

Customers Q1 and Q2

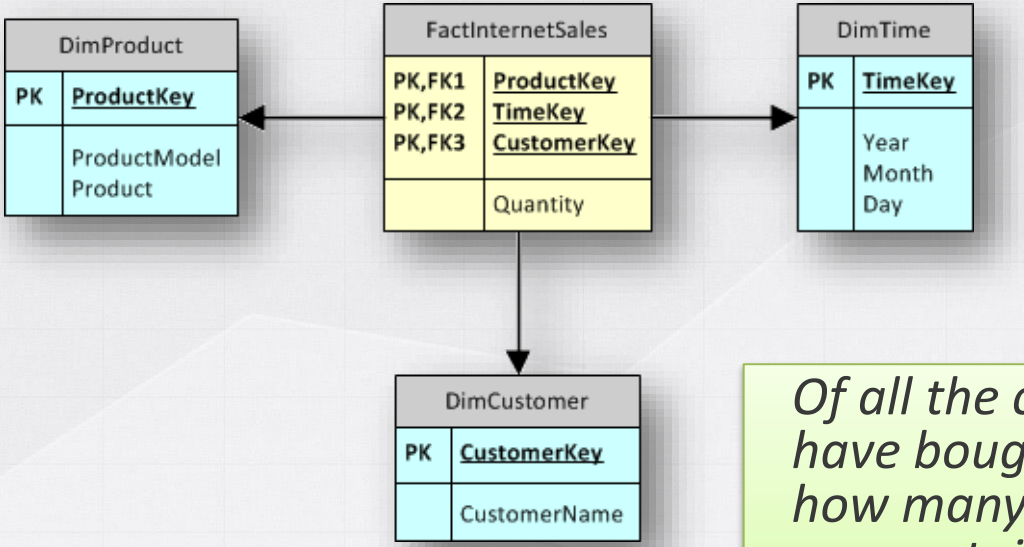
Number of users who answered two questions

```
=IF (
    HASONEVALUE ( Filter1[Question 1] ) && HASONEVALUE ( Filter2[Question 2] ),
    IF (
        VALUES ( Filter2[Question 2] ) <> VALUES ( Filter1[Question 1] ),
        CALCULATE (
            COUNTROWS ( Customers ),
            CALCULATETABLE (
                Answers,
                USERELATIONSHIP ( Answers[AnswerKey]; Filter2[AnswerKey] )
            ),
            CALCULATETABLE (
                Answers,
                USERELATIONSHIP ( Answers[AnswerKey]; Filter1[AnswerKey] )
            )
        )
    )
)
```

sqlbi

Basket Analysis

Basket Analysis: The Scenario

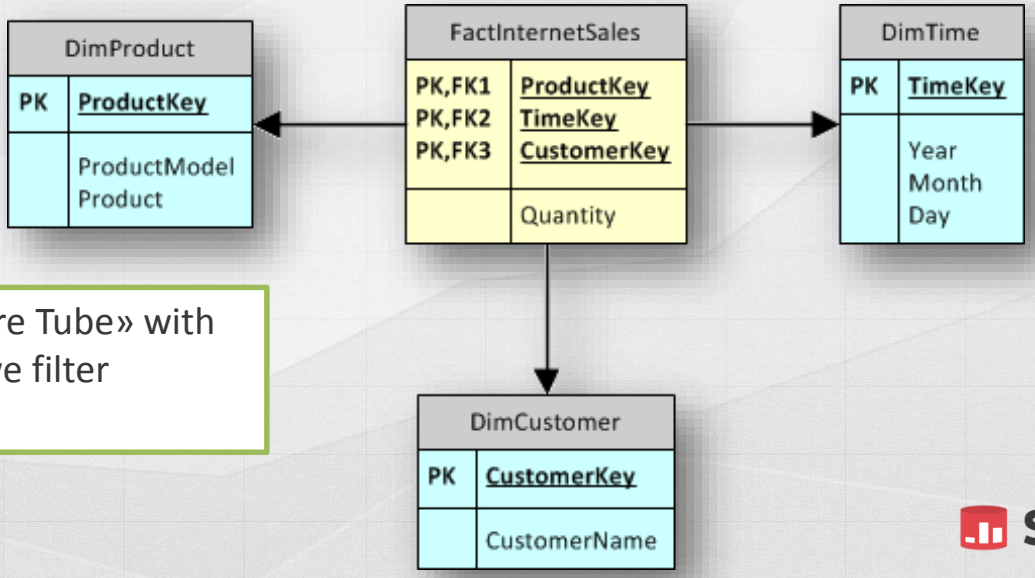


Of all the customers who have bought a Mountain Bike, how many have never bought a mountain tire tube?

sqlbi

Basket Analysis: The Data Model

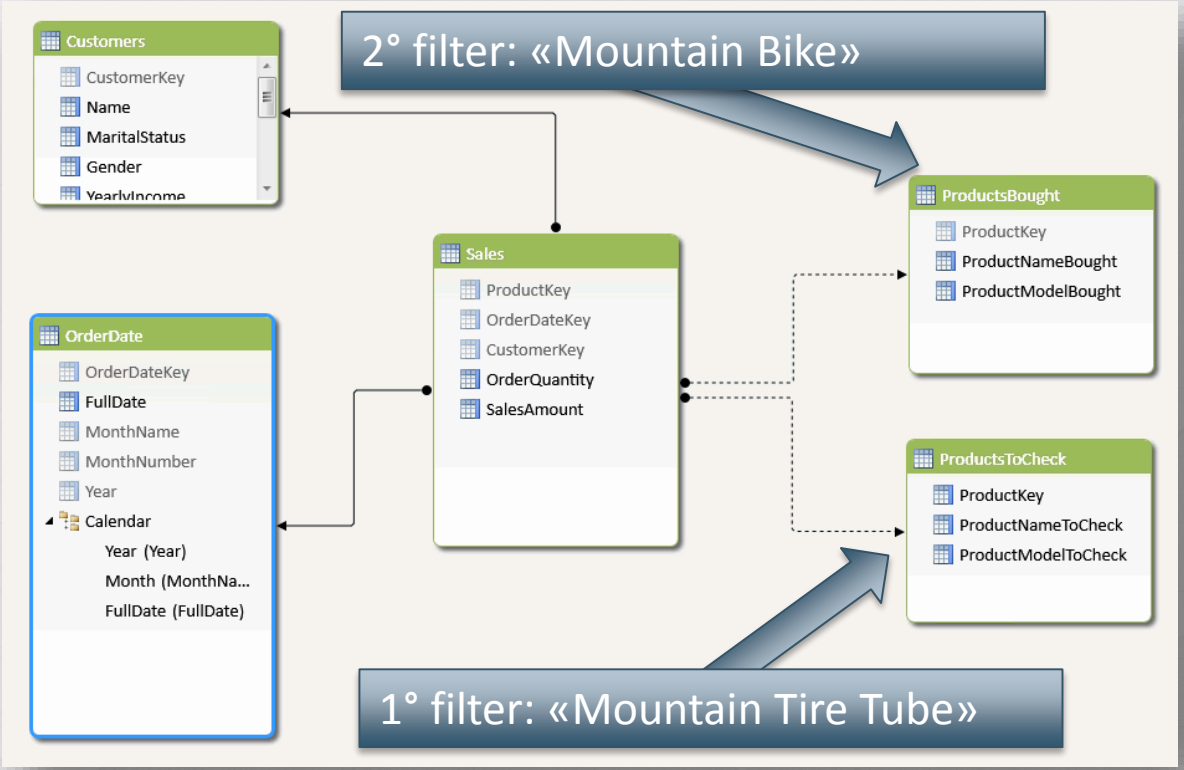
*Of all the customers who have bought a **Mountain Bike**, how many have never bought a **mountain tire tube**?*



We can filter «Mountain Tire Tube» with this table but… where do we filter «Mountain Bike»?

sqlbi

Basket Analysis: The Data Model



sqlbi

The Final Result

2° filter: «Mountain Tire Tube»

ProductModelToCheck

Men's Sports Shorts	Minipump	ML Bottom Bracket	ML Crankset	ML Fork	ML Headset
ML Mountain Frame	ML Mountain Frame-W	ML Mountain Front Wheel	ML Mountain Handlebars	ML Mountain Pedal	ML Mountain Rear Wheel
ML Mountain Seat/Saddl...	ML Mountain Tire	ML Road Frame	ML Road Frame-W	ML Road Front Wheel	ML Road Handlebars
ML Road Pedal	ML Road Rear Wheel	ML Road Seat/Saddle 2	ML Road Tire	ML Touring Seat/Saddle	Mountain Bike Socks

ProductModelBo...

- ML Road Rear ...
- ML Road Seat/...
- ML Road Tire
- ML Touring Se...
- Mountain Bike...
- Mountain Bott...
- Mountain Pump
- Mountain Tire ...
- Mountain-100
- Mountain-200
- Mountain-300
- Mountain-400...
- Mountain-500
- Patch kit
- Racing Socks
- Rear Brakes

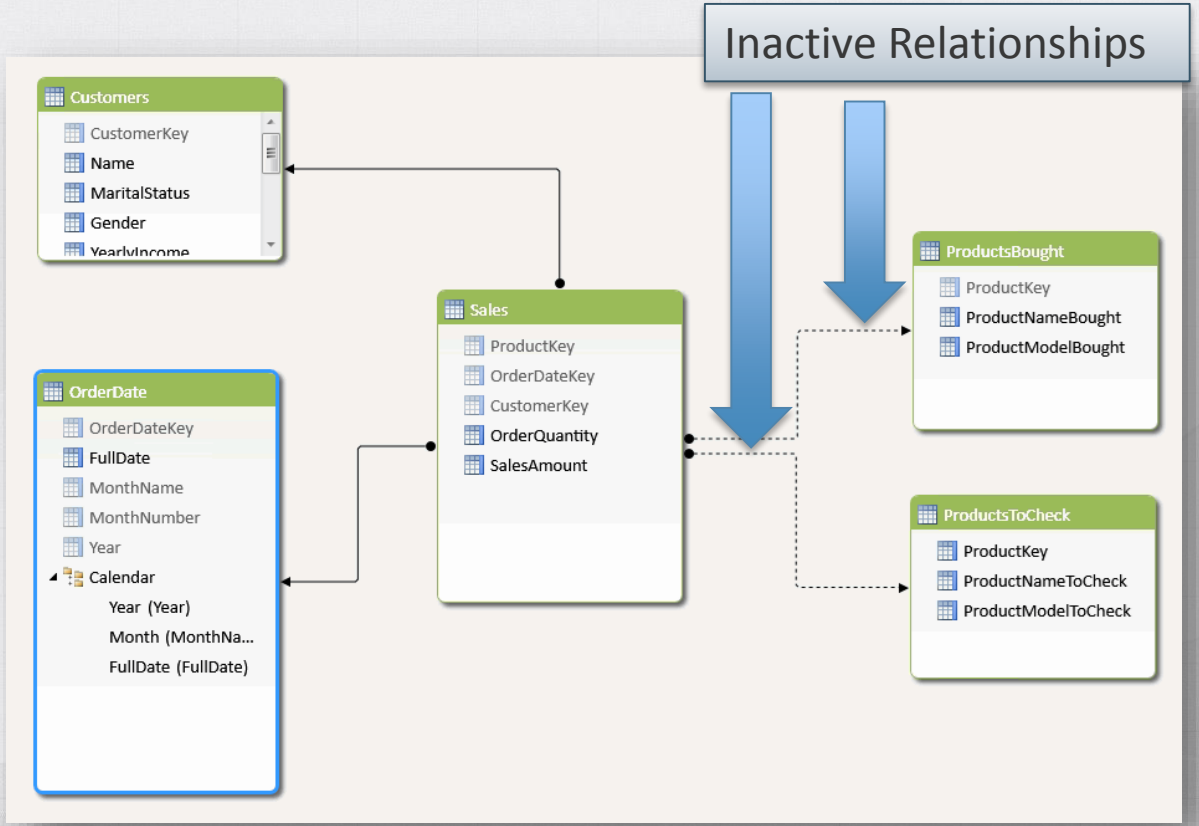
	Column Labels			
	Mountain Tire Tube			
Row Labels	HavingProduct	NotHavingProduct	Total HavingProduct	Total NotHavingProduct
2007	**33**	**174**	**33**	**174**
Mountain-400-W Silver, 38	9	48	9	48
Mountain-400-W Silver, 40	8	42	8	42
Mountain-400-W Silver, 42	7	41	7	41
Mountain-400-W Silver, 46	9	43	9	43
2008	**83**	**460**	**83**	**460**
Mountain-400-W Silver, 38	26	122	26	122
Mountain-400-W Silver, 40				
Mountain-400-W Silver, 42				
Mountain-400-W Silver, 46				
2009				
Mountain-400-W Silver, 38				
Mountain-400-W Silver, 40	20	108	20	108
Mountain-400-W Silver, 42	21	108	21	108
Mountain-400-W Silver, 46	16	122	16	122
2010	**83**	**460**	**83**	**460**
Mountain-400-W Silver, 38	26	122	26	122
Mountain-400-W Silver, 40	20	108	20	108
	21	108	21	108
	16	122	16	122
	83	**460**	**83**	**460**

HavingProduct = Bought Both
NotHavingProduct = Bought Bike, No Tire

1° filter: «Mountain Bike»

sqlbi

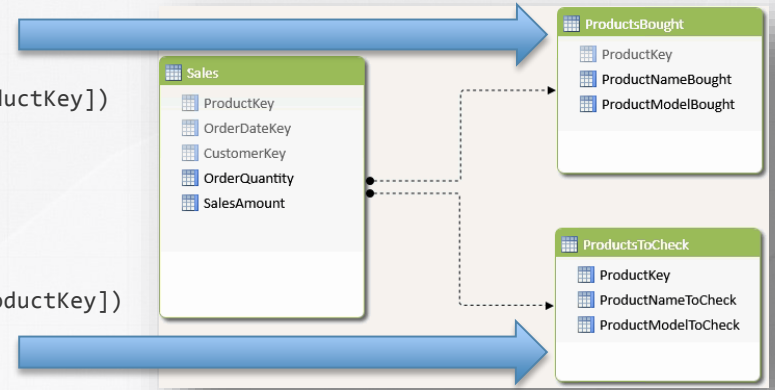
Inactive Relationships



sqlbi

The formula for HavingProducts

```
=COUNTROWS (
    CALCULATETABLE (
        FILTER (
            Customers,
            CALCULATE (
                COUNTROWS (Sales),
                USERELATIONSHIP (
                    Sales[ProductKey],
                    ProductsBought[ProductKey])
            ) > 0
            &&
            CALCULATE (
                COUNTROWS (Sales),
                USERELATIONSHIP (
                    Sales[ProductKey],
                    ProductsToCheck[ProductKey])
            ) > 0
        ),
        FILTER (
            ALL (OrderDate),
            OrderDate[FullDate] <= MAX (OrderDate[FullDate])
        )
    )
)
```

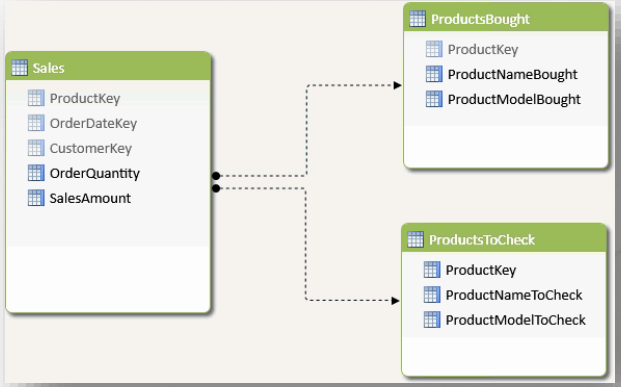


sqlbi

Not HavingProducts

```
=COUNTROWS (
    CALCULATETABLE (
        FILTER (
            Customers,
            CALCULATE (
                COUNTROWS (Sales),
                USERELATIONSHIP (
                    Sales[ProductKey],
                    ProductsBought[ProductKey])
            ) > 0
            &&
            CALCULATE (
                COUNTROWS (Sales),
                USERELATIONSHIP (
                    Sales[ProductKey],
                    ProductsToCheck[ProductKey])
            ) = 0

        FILTER (
            ALL (OrderDate),
            OrderDate[FullDate] <= MAX (OrderDate[FullDate])
        )
    )
)
```

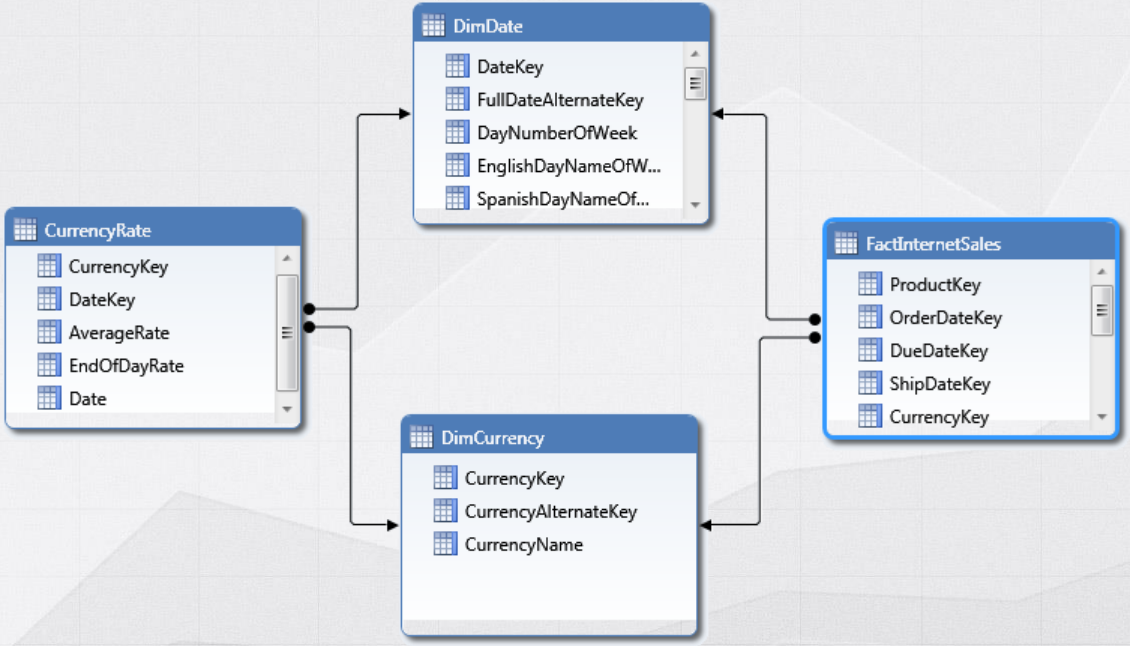


sqlbi

Currency Conversion

sqlbi

Currency Conversion



sqlbi

Currency Conversion

- All amounts are in USD
- Reports generated
  - In the order currency
  - With exchange rate at the date of the order
  - With last available exchange rate
- Exchange rate must be computed dynamically

sqlbi

Currency Conversion Naïve

The naïve solution works fine for small amounts of data but LOOKUPVALUE is very expensive if iterated over the fact table

```
OriginalCurrencyAmount :=

SUMX (
    Sales,
    Sales[SalesAmount] *
        LOOKUPVALUE (
            CurrencyExchange[EndOfDayRate],
            CurrencyExchange[DateKey], Sales[OrderDateKey],
            CurrencyExchange[CurrencyKey], Sales[CurrencyKey]
        )
)
```

sqlbi

Reduce the number of loops

Iterating over DimCurrency and DimDate highly reduces the number of loops executed inside the formula engine

```
OriginalCurrencyAmount :=

SUMX (
    Currencies,
    SUMX (
        OrderDate,
        CALCULATE (
            VALUES ( CurrencyExchange[EndOfDayRate] )
          * SUM ( Sales[SalesAmount] )
        )
    )
)
```

sqlbi

Further optimize the loops

Using CROSSJOIN the full computation is performed inside a single loop, reducing the need to spool temporary data

```
OriginalCurrencyAmount :=

SUMX (
    CROSSJOIN (
        Currencies,
        OrderDate
    ),
    CALCULATE (
        VALUES ( CurrencyExchange[EndOfDayRate] )
      * SUM ( Sales[SalesAmount] )
    )
)
```

sqlbi

Last Available Exchange Rate

Finding the last available exchange rate is pretty easy: just create a filter context that shows only the correct date

```
OriginalCurrencyAmountLastDate :=

CALCULATE (
    SUMX (
        Currencies,
        CALCULATE (
            VALUES ( CurrencyExchange[EndOfDayRate] )
          * SUM ( Sales[SalesAmount] )
        )
    ),
    FILTER (
        VALUES ( CurrencyExchange[DateKey] ),
        CurrencyExchange[DateKey] = MAX ( CurrencyExchange[DateKey] )
    )
)
```

sqlbi

Conclusions

- Tabular requires different solutions to common problems
- DAX is the key to handle complex relationships
- Complexity is in the code, not always in the model

sqlbi

Thank you!

Check our new articles on

www.sqlbi.com