

## Different types of many-to-many relationships in Power BI

Marco Russo  
marco@sqlbi.com



1

## Sponsors



2

## Raffle prizes from the sponsors



3

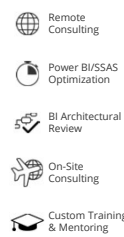
## We write Books



## We teach Courses



## We provide Consulting



## We are recognized BI Experts



4

## Agenda

- Many-to-many relationships between dimensions
- Relationships at different granularity
- Compare different types of relationships



5

## Types of relationships

- Many-to-many relationships between dimensions
  - Implement a pattern with two relationships:
    - one-to-many + many-to-one
- Types of relationship cardinality in Power BI:
  - One-to-one
  - One-to-many (or many-to-one)
  - Many-to-many (new – October 2018)
    - Implement the pattern many-to-one + one-to-many



6

What is a many-to-many relationship in the dimensional model world

## Many-to-many relationships between dimensions



7

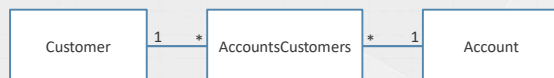
## Many-to-many relationships

- Typically involve two business entities (dimensions)
- Examples
  - Bank current account and holders
  - Companies and shareholders
  - House and householders



8

## Many-to-many requires a bridge table



• Marco  
• Paul  
• Robert  
• Luke

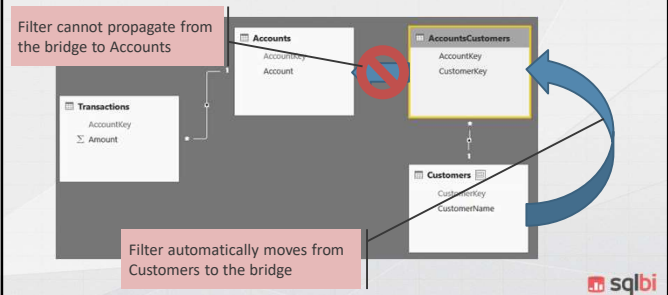
• Marco, 1  
• Paul, 2  
• Robert, 3  
• Luke, 4  
• Marco, 5  
• Paul, 5  
• Marco, 6  
• Paul, 6

• 1  
• 2  
• 3  
• 4  
• 5  
• 6



9

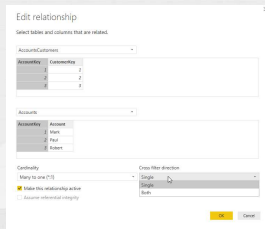
## Filter propagation in Tabular



10

## Bidirectional filtering

- Enabled at the relationship level
- Let the filter context propagate both ways
- Works with any measure in the model: fewer coding means fewer errors



CustomerName	Amount	SumOfAmount
Luke	\$5,000.00	\$800.00
Mark	\$5,000.00	\$2,800.00
Paul	\$5,000.00	\$1,700.00
Robert	\$5,000.00	\$1,700.00
<b>Total</b>	<b>\$5,000.00</b>	<b>\$5,000.00</b>



11

## Using CROSSFILTER

Changes the direction of a relationship for the duration of a CALCULATE statement. This pattern must be used in every measures requiring the many-to-many behavior.

```
SumOfAmount CrossFilter =
CALCULATE (
    SUM ( Transactions[Amount] ),
    CROSSFILTER (
        AccountsCustomers[AccountKey],
        Accounts[AccountKey],
        BOTH
    )
)
```



12

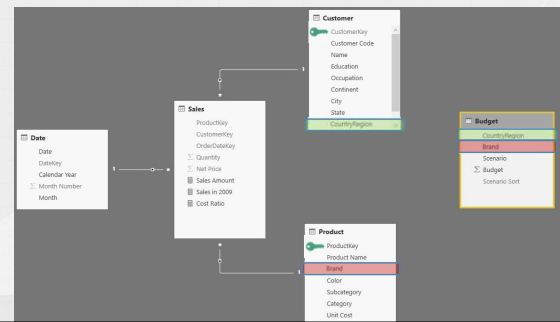
Relationships at different granularities are a challenge

## Relationships at different granularity



13

## Analyzing budget data



14

## Missing relationship

- Without the relationship, the model does not work
- The relationship exists, but at a different granularity
- In fact, dimensions have granularity too
- Need to build a relationship at a different granularity

Brand	Sales in 2009	Budget
A. Datum	1,823,681.13	44,855,187.00
Adventure Works	4,878,941.52	44,855,187.00
Contoso	9,113,675.42	44,855,187.00
Fabrikam	7,933,936.37	44,855,187.00
Litware	4,668,613.86	44,855,187.00
Northwind Traders	826,993.38	44,855,187.00
Proseware	3,664,900.11	44,855,187.00
Southridge Video	1,892,420.79	44,855,187.00
Tailspin Toys	606,558.34	44,855,187.00
The Phone Company	1,891,590.92	44,855,187.00
Wide World Importers	3,317,561.02	44,855,187.00
<b>Total</b>	<b>40,618,872.86</b>	<b>44,855,187.00</b>



15

## Problems to solve

- Budget is at the year level, needs to slice by month too
- Brand is not a key in Product
- CountryRegion is not a key in Customer
- We will see several solutions
  - DAX code to simulate relationships
  - Creation of new tables to slice
  - Weak relationships



16

## Using TREATAS

TREATAS can change the data lineage of a column, transforming the data lineage of Product and Customer columns in Budget ones.

```
Budget 2009 :=
CALCULATE (
    SUM ( Budget[Budget] ),
    TREATAS (
        VALUES ( 'Product'[Brand] ),
        Budget[Brand]
    ),
    TREATAS (
        VALUES ( Customer[CountryRegion] ),
        Budget[CountryRegion]
    )
)
```



17

## Use DAX to move the filters

You can use DAX to move the filter from the Product[Brand] column to the Budget[Brand] one, and repeat the same operation for the CountryRegion pair of columns.

```
Budget 2009 :=
CALCULATE (
    SUM ( Budget[Budget] ),
    Budget[Brand] IN VALUES ( 'Product'[Brand] ),
    Budget[CountryRegion] IN VALUES ( Store[CountryRegion] )
)
```



18

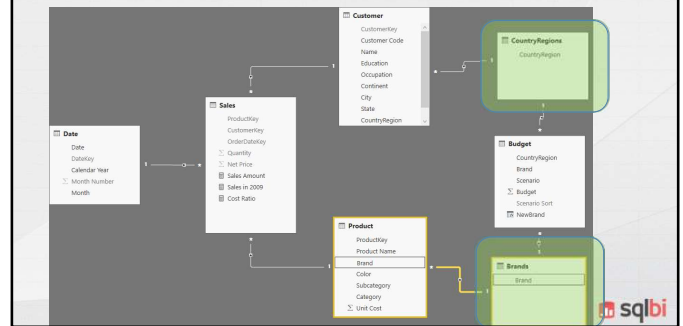
## Using DAX to move filter

- Flexibility
  - You change the filter context in a very dynamic way
  - Full control over the functions used
- Complexity
  - Every measure need to be authored using the pattern
  - Error-prone
- Speed
  - Using DAX to move a filter is sub-optimal
  - Leverages the slower part of the DAX engine (FE)



19

## Calculated tables to slice dimensions



20

## Using calculated tables

In Power BI and Analysis Services the intermediate tables can be built as calculated tables.

```
Brands =
DISTINCT (
    UNION (
        DISTINCT ( Product[Brand] ),
        DISTINCT ( Budget[Brand] )
    )
)

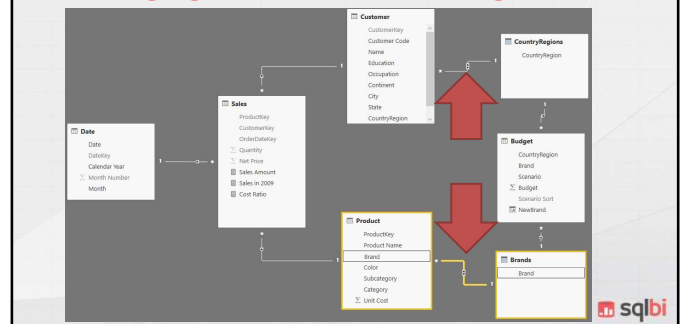
CountryRegions =
DISTINCT (
    UNION (
        DISTINCT ( Customer[CountryRegion] ),
        DISTINCT ( Budget[CountryRegion] )
    )
)
```

Use ALLNOBLANKROW or DISTINCT to avoid circular dependency issues. In fact, if you use ALL or VALUES, the result depends from the existence of the blank row.



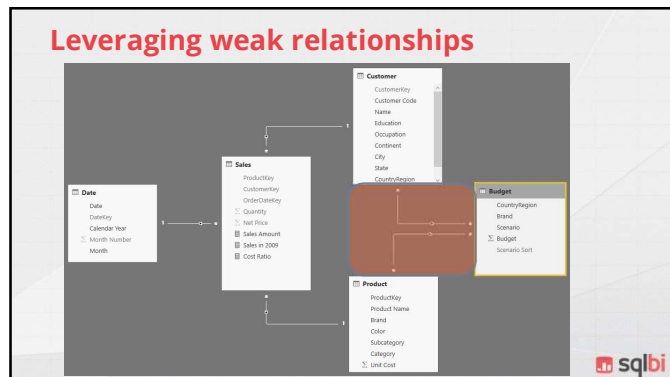
21

## Leveraging bidirectional filtering



22

## Leveraging weak relationships



23

## Many-to-many cardinality in a relationship



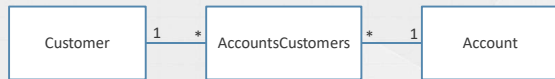
- Contoso Rechargeable Battery E100 Black
- Contoso USB Cable M120 White
- Litware Refrigerator L1200 Orange
- Litware Microwave E090 Grey
- Fabrikam Microwave M1150 Grey
- Fabrikam Microwave M1250 Grey
- ...
- Adventure Works
- Contoso
- Fabrikam
- Litware
- Northwind Traders
- ...
- Adventure Works, USA, 3000
- Adventure Works, UK, 1000
- Contoso, USA, 10000
- Contoso, UK, 3500
- Fabrikam, USA, 4000
- Fabrikam, UK, 1200
- ...



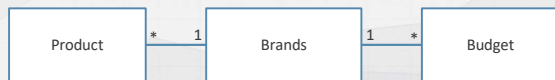
24

## Different types of many-to-many

"Classic" many-to-many relationships between dimensions



Relationship at different granularities → can be implemented using a weak relationship



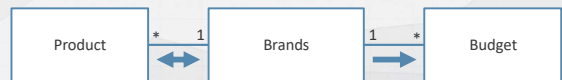
25

## Check correct filter direction

The filter direction of a weak relationship should be "single" (default is "both")



This corresponds to a bidirectional filter only in one of the two physical relationships



26

## Conclusions

- Different type of many-to-many relationships
- One-Many / Many-One
  - Bridge table contains data coming from data source
- Many-One / One-Many
  - Bridge table is a normalized attribute
  - Similar to a snowflake schema
  - No additional data
  - Can be implemented with many-to-many cardinality

27

Check our Articles, Books, Videos, and Courses on

[www.sqlbi.com](http://www.sqlbi.com)

28