



Who we are

- o BI Experts and Consultants
- o Problem Solving
- o Complex Project Assistance
- o DataWarehouse Assesments and Development
- o Courses, Trainings and Workshops
- o Microsoft Business Intelligence Partners
- o Book Writers



sqlbi

Our latest conferences

- o PASS Europe 2009 – Neuss – Germany
- o PASS 2009 – Seattle – USA
- o SQL Conference 2010 – Milan – Italy
- o Teched 2010 – New Orleans – USA
- o 24 Hours of PASS 2010 – Online
- o PASS 2010 – Seattle – USA

sqlbi

Road to BISM: PowerPivot

- o PowerPivot for Excel
 - Free AddIn for Microsoft Excel 2010
 - Available since June 2010
 - Different releases for 32/64 bits
 - Contains the Vertipaq Engine
 - o SSAS running in process with Excel
- o PowerPivot for SharePoint
 - Vertipaq running server side
 - Integrated with SharePoint

sqlbi

Road to BISM: Denali

- o Denali
 - New release of SQL Server
 - SSAS / Vertipaq running server side
 - Works standalone (PowerPivot for SP cannot)
 - Vertipaq / SSAS selected at instance level
 - New release of PowerPivot
- o PowerPivot 1.0
 - **Is the road to learn new Denali features**

sqlbi

Vertipaq: change your mind!

- o Vertipaq in Denali
 - Not a simple evolutionary step
 - It is a complete new environment
 - New calculation paradigms
 - New data modeling techniques
 - New formula authoring experience
- o Want to learn BISM?
 - Forget MDX
 - Forget SQL
 - Then, Learn DAX. ☺

What is changing?

DENALI	SQL 2008 R2
BISM	UDM
DAX	MDX
RELATIONAL	MULTIDIMENSIONAL
IN MEMORY	ON DISK
FAST VERTIPAQ ENGINE	AGGREGATIONS
LEAF LEVEL COMPUTATION	CALCULATED MEMBERS
FILTER / ROW CONTEXT	MDX FUNCTIONS, TUPLES ETC
SELF SERVICE + CORPORATE	CORPORATE ONLY
EASY DATA INTEGRATION	BI PROFESSIONALS NEEDED

Relational vs Multidimensional

- o UDM
 - Facts, Dimensions, Bridges, Snowflakes
 - Relationships: Regular, Referenced, M2M
 - Hierarchies, Attribute Relationships
- o BISM
 - Tables
 - Relationships (1:N, One Column only)

Complexity in BISM

- o BISM: simple data model
- o Complexity is moved into:
 - Calculated columns
 - Measures
 - DAX, DAX, DAX, DAX, DAX and... **DAX!**
- o Every column is a dimensional attribute
- o Every column is a measure

A SMALL EXAMPLE OF HOW TO AGGREGATE DATA IN DAX

Semi Additive Measures

Semi Additive Measures

- o Additive Measure
 - SUM over all dimensions
- o Non Additive Measure
 - Different function over all dimensions
 - Example: average of the sale price
- o Semi Additive Measure
 - SUM over some dimensions
 - Different function over other dimensions
 - Time is the standard exception for aggregations
 - Examples
 - o Warehouse stocking
 - o Current account balance

Current Account Balance

Name	Occupation	Country	Date	Quarter	Balance
Kate Jordan	Farmer	USA	1/1/2010 Q1/2010		1,470.00
Luis Bonifaz	IT Consultant	Argentina	1/1/2010 Q1/2010		1,470.00
Maurizio Musagno	IT Consultant	Italy	1/1/2010 Q1/2010		1,500.00
Kate Jordan	Farmer	USA	2/1/2010 Q1/2010		2,812.50
Luis Bonifaz	IT Consultant	Argentina	2/1/2010 Q1/2010		2,812.50
Maurizio Musagno	IT Consultant	Italy	2/1/2010 Q1/2010		2,500.00
Kate Jordan	Farmer	USA	3/1/2010 Q1/2010		3,917.50
Luis Bonifaz	IT Consultant	Argentina	3/1/2010 Q1/2010		3,410.00

Row Labels	Kate Jordan	Luis Bonifaz	Maurizio Musagno	Grand Total
Q1/2010	6,412.50	7,290.00	7,500.00	23,202.50
1/1/2010	1,470.00	1,470.00	1,500.00	4,457.50
2/1/2010	2,812.50	2,450.00	2,500.00	7,762.50
3/1/2010	3,917.50	3,410.00	3,500.00	10,827.50
Q2/2010	6,975.00	6,070.00	6,200.00	19,255.00
4/1/2010	2,950.00	1,960.00	2,000.00	6,910.00
5/1/2010	2,025.00	1,784.00	1,800.00	5,589.00
6/1/2010	2,700.00	2,352.00	2,400.00	7,452.00
Q3/2010	11,475.00	9,096.00	10,200.00	31,471.00
7/1/2010	5,940.00	3,136.00	3,200.00	9,596.00
8/1/2010	5,062.50	4,410.00	4,500.00	13,972.50
9/1/2010	2,812.50	2,450.00	2,500.00	7,762.50
Q4/2010	6,962.50	5,978.00	6,100.00	19,040.50
10/1/2010	2,950.00	1,960.00	2,000.00	6,910.00
11/1/2010	2,881.25	1,813.00	1,850.00	5,744.25
12/1/2010	2,525.25	2,205.00	2,250.00	6,985.25
Grand Total	33,250.00	29,400.00	30,000.00	93,350.00

- Month level **correct**
- Quarter level **wrong**
- Year level **wrong**

Semi Additive Measures in UDM

- Aggregation function:
 - LastChild
 - LastNonEmpty
- Performances not very impressive
- Aggregations always needed
- The aggregation is handled by the data model
- In DAX, there is no default aggregation function. Thus, DAX is needed.

Semi Additive Measures

- Create a calendar table
- Add a new measure:
 - CALCULATE: to set the filter
 - LASTDATE: to find the last child

```
LastBalance = CALCULATE (
    SUM( Balances[Balance] ),
    LASTDATE(Date[Date])
)
```

Opening and Closing Balance

- Special functions for closing balance
 - ClosingBalanceMonth = CLOSINGBALANCEMONTH(SUM(Balances[Balance]), BalanceDate[Date])
 - ClosingBalanceQuarter = CLOSINGBALANCEQUARTER(SUM(Balances[Balance]), BalanceDate[Date])
 - ClosingBalanceYear = CLOSINGBALANCEYEAR(SUM(Balances[Balance]), BalanceDate[Date])

Row Labels	LastBalance	ClosingBalanceMonth	ClosingBalanceQuarter	ClosingBalanceYear
2010	6,486.00	6,486.00	6,486.00	6,486.00
Q1	10,667.00	10,667.00	10,667.00	6,486.00
01 - January	4,537.00	4,537.00	10,667.00	6,486.00
02 - February	7,762.00	7,762.00	10,667.00	6,486.00
03 - March	10,667.00	10,667.00	10,667.00	6,486.00
Q2	7,452.00	7,452.00	7,452.00	6,486.00
04 - April	6,210.00	6,210.00	7,452.00	6,486.00
05 - May	5,589.00	5,589.00	7,452.00	6,486.00
06 - June	7,452.00	7,452.00	7,452.00	6,486.00
Q3	7,762.00	7,762.00	7,762.00	6,486.00
07 - July	9,396.00	9,396.00	7,762.00	6,486.00
08 - August	13,972.00	13,972.00	7,762.00	6,486.00
09 - September	7,762.00	7,762.00	7,762.00	6,486.00
Q4	6,985.00	6,985.00	6,985.00	6,486.00
10 - October	6,210.00	6,210.00	6,985.00	6,486.00
11 - November	5,744.00	5,744.00	6,985.00	6,486.00
12 - December	6,985.00	6,985.00	6,985.00	6,486.00
Grand Total	6,486.00	6,486.00	6,486.00	6,486.00

In Memory vs On Disk

- UDM
 - Aggregations
 - Usage Based Optimization
 - Aimed to optimize Storage Engine
- BISM
 - No Storage Engine: everything in memory
 - No aggregations, fast in memory computation
 - Non standard aggregations = Measures

Calculated Columns / Measures

- Measures
 - Similar to MDX Calculated Members
 - Work on aggregations
 - Might go leaf-level, if needed
- Calculated columns
 - New kind of calculation
 - Computed once in the data model
 - Can access the whole data model
 - Different from calc columns in the UDM DSV

Calculated Columns / Measures

- o $GrossMargin = SalesAmount - ProductCost$
 - Calculated column
- o $GrossMargin\% = GrossMargin / SalesAmount$
 - Cannot be computed row by row
- o Measures needed

HOW TO COMPUTE MOVING AVERAGES WITH DAX

Moving Averages

Stocks Analysis

- o Trading technique: moving averages
- o Two moving averages
 - 50 working days
 - 200 working days
- o Analysis of crossing of moving averages
 - Indicate buy/sell points
- o Issue: compute moving average
 - Date range different for each stock
 - Calculated columns come to help us!

The final result



Stocks Analysis

- o Compute moving average
- o Different date period for different stocks
- o For each date:
 - Compute starting date
 - Compute average over moving period
- o Then plot data on a chart

SOME NEW METHODS TO HANDLE SLOWLY CHANGING DIMENSIONS

Slowly Changing Dimensions in DAX

SCD with calculated columns

- Slowly Changing Dimensions (Type 2)
 - Historical Value
 - Current Value
 - On the same table!
- UDM / Relational Solution
 - Updates on dimensions
 - Problems with locking / deadlock
 - Slow ETL
- BISM Solution
 - No update needed on relational engine
 - All computations carried on in memory



SCD Handling

- Example – "current" (most recent) price for each product version
- Requires a new Calculated Column:

```
=CALCULATE (
    VALUES (DimProduct[ListPrice]),
    DimProduct[AltKey] = EARLIER (DimProduct[AltKey]),
    DimProduct[Status] = "Current",
    ALL(DimProduct)
)
```



In Memory vs On Disk

- Vertipaq and DAX
 - Fast in memory computations
 - Leaf level calculations
 - Calculated columns
- The data model can be different
- Less ETL code
- More DAX code
 - No locking, No I/O, very fast
 - Data is not persisted on disk for other clients



Self Service vs Corporate BI

- Self Service BI
 - Creative Solutions in the hand of Power Users
 - No need to ask for help to IT
 - Less work for IT
- Example
 - Banding of prices



BANDING IN THE SELF SERVICE BI WORLD

Banding

sqlbi.com



Analysis of product sell price

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

Row Labels	Reseller Order Quantity	Reseller Sales Amount
2.39	974	9923.21
4.99	2,571	57476.00
7.99	2,414	\$11,888.37
6.6642	3,289	\$16,779.84
8.99	6,284	\$52,826.52
6.5	1,197	\$6,379.39
19.99	1,130	\$13,518.69
20.24	774	\$6,377.71
23.9481	1,877	\$24,818.41
34.49	1,621	\$52,507.99
47	1,066	\$16,252.22
27.12	448	\$7,280.41
33.6642	6,692	\$11,508.29
44.2	70	\$1,368.40
34.99	6,409	\$127,204.64
37.99	6,055	\$128,847.18
39.14	918	\$14,498.62
40.49	1,117	\$11,996.20
40.49009	514	\$12,960.38
44.54	547	\$14,510.41
46.09	58	\$1,348.62
46.6675	6,587	\$197,402.11



The quick and dirty solution

```

= IF (
  FactResellerSales[DiscountedPrice] <= 5,
  "01 LOW",
  IF (
    FactResellerSales[DiscountedPrice] <=30,
    "02 MEDIUM",
    IF (
      FactResellerSales[DiscountedPrice] <=100,
      "03 MEDIUM",
      IF (
        FactResellerSales[DiscountedPrice] <= 500,
        "04 HIGH",
        "05 VERY HIGH"))))
  
```

Even if this works... a better data model would be welcome!

Banding: a Data Driven Model

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

```

SELECT
  P.BandName,
  SUM (S.ExtendedAmount)
FROM dbo.FactResellerSales S
JOIN PriceBands P
ON S.UnitPrice BETWEEN P.FromPrice AND
P.ToPrice
GROUP BY
  P.BandName
  
```

Data Model with PowerPivot

```

SELECT
  P.BandName,
  SUM (S.ExtendedAmount)
FROM dbo.FactResellerSales S
JOIN PriceBands P
ON S.UnitPrice = P.Price
GROUP BY
  P.BandName
  
```

BandName	Price
VERY LOW	1
VERY LOW	2
VERY LOW	3
VERY LOW	4
VERY LOW	5
LOW	6
LOW	7
...	...

Band Expansion

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

BandName	Price
VERY LOW	1
VERY LOW	2
VERY LOW	3
VERY LOW	4
VERY LOW	5
LOW	6
LOW	7
...	...

- o Improve the Data Model
- o Cannot do this in PowerPivot
- o Write VBA code to transform data
- o Needs some VBA knowledge

The VBA Code

```

For Each Row In ActiveSheet.ListObjects("PriceBands").ListRows
  Dim MinValue As Integer
  Dim MaxValue As Integer
  Dim Value As Integer
  Dim newRow As ListRow
  MinValue = Row.Range(1, 2).Value
  MaxValue = Row.Range(1, 3).Value - 1
  For Value = MinValue To MaxValue
    Set newRow = PriceBandsExpanded.ListRows.Add
    newRow.Range(1, 1) = Row.Range(1, 1)
    newRow.Range(1, 2) = Value
  Next
Next
  
```

BandName	FromPrice	ToPrice	Price	Count
VERY LOW	0	5	VERY LOW	8
LOW	5	30	VERY LOW	1
MEDIUM	30	100	VERY LOW	2
HIGH	100	500	VERY LOW	4
VERY HIGH	500	9999	LOW	5
			LOW	6
			LOW	7
			LOW	8
			LOW	9

Issues With Band Expansion

- o Need to know Visual Basic for Application
- o Refresh needs two steps
- o Macro Enabled WorkBooks are not safe
- o Search for a new solution
- o Entirely based on DAX

Banding With DAX

- Link only the configuration table
- No need to perform expansion
- The complexity moves to the expression
- Again, DAX programming needed

Banding With CALCULATE

- Works with the filter context
- Leverages
 - CALCULATE
 - VALUES
 - FILTER

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

SOME INSIGHTS INTO BISM TIME INTELLIGENCE FUNCTIONS

Time Intelligence

Aggregations Over Time

- Many interesting aggregations
 - YTD: Year To Date
 - QTD: Quarter To Date
 - MTD: Month To Date
- They all need a Calendar Table
- And some understanding of CALCULATE ☺
- Tool dimension is not affordable
 - No DateTool dimension!

Year To Date

```
CALCULATE(
  SUM ( SalesOrderDetail[LineTotal] ),
  DATESYTD ( OrderDate[Date] )
)
```

- DATESYTD: Returns a set of dates
 - From the start of the year
 - Up to the parameter date
- CALCULATE
 - Creates a filter contexts
 - Perform the SUM operation

PARALLELPERIOD

```
CALCULATE (
  SUM( SalesOrderDetail[LineTotal] ),
  PARALLELPERIOD
  (OrderDate[Date], -1, YEAR)
)
```

- Returns a set of dates (a table)
 - Shifted in time
 - The whole period is returned, regardless dates in the first parameter

Moving Annual Total

```

CALCULATE(
    SUM (SalesOrderDetail[LineTotal]),
    DATESBETWEEN (
        OrderDate[Date],
        NEXTDAY(
            SAMEPERIODLASTYEAR(
                LASTDATE( OrderDate[Date] )
            )
        ),
        LASTDATE( OrderDate[Date] )
    )
)
    
```

Counting Working Days

- How many working days in a date range?
 - Easily solved with Calendar table
 - Define a new column «WorkingDays»
 - Aggregate with SUM
- Handles any date range
- Works on periods with «holes»
- Difference between BISM and UDM
 - UDM solution requires ETL
 - BISM solution can be self-contained

Example: Check Delayed Orders

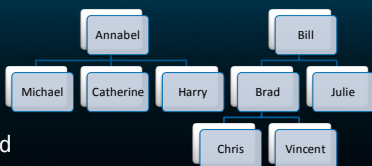
- How many orders were delayed?
 - Easy: «ShipDate Greater Than DueDate»
- How many working days of delay?
 - New Calculated Column
 - WorkingDayNumber
 - Incremental value from the start of table
 - Delta can be computed as difference

POWERPIVOT 1.0 CAN HANDLE P/C HIERARCHIES, WITH SOME EFFORTS.

Parent / Child Hierarchies

sqlbi.com

Parent / Child Hierarchies



- Unbalanced
- Variable Depth
- Data present for both leaves and nodes
- Handled in UDM, not handled in BISM

P/C Hierarchies in UDM

- UDM
 - Very slow, don't fully leverage aggregations
 - Unary operator might be a real pain
 - Especially with many hierarchies
- P/C Naturalizing
 - Available in BIDS Helper
 - Standard method to improve performances

P/C Hierarchies in BISM

- Simply... not present in the modeling options
- Naturalization is mandatory
 - But it is not enough
- Naturalization should be done in SQL
 - Denali will have new functions to do that in DAX

Parent/Child: Naturalization

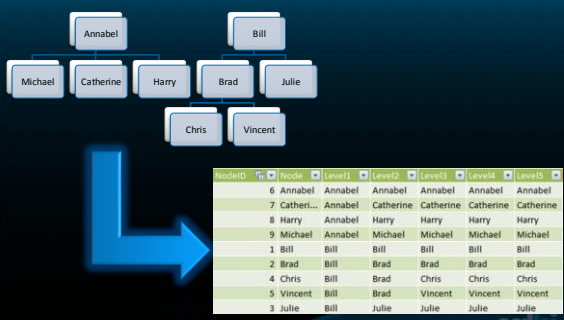


Parent/Child: First trial

NodeID	%	Node	Level1	Level2	Level3	Level4	Level5
6		Annabel	Annabel				
7		Catherine	Annabel	Catherine			
8		Harry	Annabel	Harry			
9		Michael	Annabel	Michael			
1		Bill	Bill				
2		Brad	Bill	Brad			
4		Chris	Bill	Brad	Chris		
5		Vincent	Bill	Brad	Vincent		
3		Julie	Bill	Julie			

Row Labels	Sum of Amount
Annabel	3,200.00
Catherine	600.00
Harry	1,200.00
Michael	800.00
Bill	1,600.00
Brad	1,300.00
Chris	400.00
Vincent	500.00
Julie	300.00
Grand Total	4,800.00

Naturalization not NULL



Parent/Child NOT NULL

NodeID	%	Node	Level1	Level2	Level3	Level4	Level5
6		Annabel	Annabel	Annabel	Annabel	Annabel	Annabel
7		Catheri...	Annabel	Catherine	Catherine	Catherine	Catherine
8		Harry	Annabel	Harry	Harry	Harry	Harry
9		Michael	Annabel	Michael	Michael	Michael	Michael
1		Bill	Bill	Bill	Bill	Bill	Bill
2		Brad	Bill	Brad	Brad	Brad	Brad
4		Chris	Bill	Brad	Chris	Chris	Chris
5		Vincent	Bill	Brad	Vincent	Vincent	Vincent
3		Julie	Bill	Julie	Julie	Julie	Julie

Row Labels	Sum of Amount
Annabel	3,200.00
Annabel	600.00
Catherine	1,200.00
Harry	800.00
Michael	600.00
Michael	600.00
Bill	1,600.00
Brad	1,300.00
Brad	400.00
Chris	400.00
Vincent	500.00
Julie	300.00
Julie	300.00
Grand Total	4,800.00

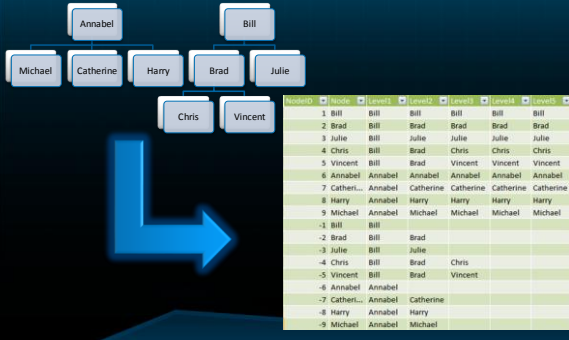
Who are those unwanted rows?

Row Labels	Sum of Amount
Annabel	3,200.00
Annabel	600.00
Catherine	1,200.00
Harry	800.00
Michael	600.00
Bill	1,600.00
Brad	1,300.00
Brad	400.00
Chris	400.00
Vincent	500.00
Julie	300.00
Julie	300.00
Grand Total	4,800.00

NodeID	%	Node	Level1	Level2	Level3	Level4	Level5
6		Annabel	Annabel	Annabel	Annabel	Annabel	Annabel
7		Catheri...	Annabel	Catherine	Catherine	Catherine	Catherine
8		Harry	Annabel	Harry	Harry	Harry	Harry
9		Michael	Annabel	Michael	Michael	Michael	Michael
1		Bill	Bill	Bill	Bill	Bill	Bill
2		Brad	Bill	Brad	Brad	Brad	Brad
4		Chris	Bill	Brad	Chris	Chris	Chris
5		Vincent	Bill	Brad	Vincent	Vincent	Vincent
3		Julie	Bill	Julie	Julie	Julie	Julie

Row Labels	Sum of Amount	Filter Context
Annabel	3,200.00	
Annabel	600.00	(Level1 = "Annabel")
Annabel	600.00	(Level1 = "Annabel", Level2="Annabel")
Annabel	600.00	(Level1 = "Annabel", Level2="Annabel", Level3="Annabel")

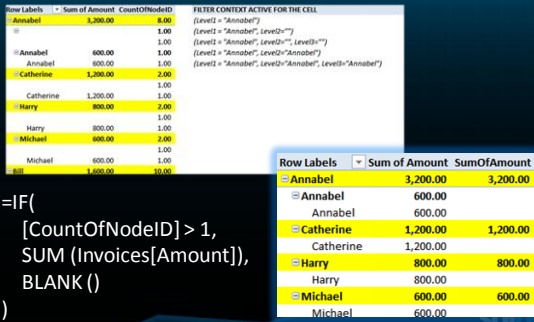
Double Naturalization



Double Naturalization



Double Naturalization



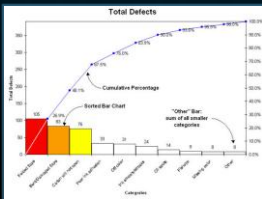
=IF(
[CountOfNodeID] > 1,
SUM (Invoices[Amount]),
BLANK ())

CALCULATED COLUMNS AT WORK WITH ABC PARETO ANALYSIS

ABC PARETO ANALYSIS

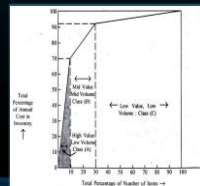
ABC and Pareto Analysis

- Pareto principle
 - 80% of effects come from 20% of the causes



ABC and Pareto Analysis

- ABC Analysis
 - Class A contains items for 70% of total value
 - Class B contains items for 20% of total value
 - Class C contains items for 10% of total value



ABC Analysis Usage

- The ABC class can be used to slice data

ABC Analysis with Excel

The formula is straightforward:

=IF(H6<=0,7;"A";IF(H6<=0,9;"B";"C"))

ABC Analysis

- Tables: Product, SalesOrderDetail

SalesAmountProduct =
CALCULATE(SUM(SalesOrderDetail[LineTotal]))

CumulatedProduct =
SUMX(
FILTER(Product,
Product[SalesAmountProduct]
>= EARLIER(Product[SalesAmountProduct])),
Product[SalesAmountProduct])

ABC Analysis

- Calculate the class

SortedWeightProduct =
Product[CumulatedProduct]
/ SUM(Product[SalesAmountProduct])

[ABC Class Product] =
IF(Product[SortedWeightProduct] <= 0.7, "A",
IF(Product[SortedWeightProduct] <= 0.9, "B", "C"))

ABC Analysis - Denormalized Attribute on Normalized Tables

- Example: Model column in Product table

SalesAmountProduct =
CALCULATE(SUM(SalesOrderDetail[LineTotal]))

SalesAmountModel =
SUMX(
FILTER(Product,
Product[Model] = EARLIER(Product[Model])),
Product[SalesAmountProduct])

ABC Analysis - Denormalized Attribute on Normalized Tables

- Cumulated value for attribute
 - Requires SUMX

CumulatedModel =
SUMX(
FILTER(Product,
Product[SalesAmountModel]
>= EARLIER(Product[SalesAmountModel])),
Product[SalesAmountProduct])

ABC Analysis - Denormalized Attribute on Normalized Tables

- Calculate the class
 - Use the SalesAmountProduct in the ratio denominator to get the right value

```
SortedWeightModel =
Product[CumulatedModel]
/ SUM( Product[SalesAmountProduct] )
```

```
[ABC Class Model] =
IF( Product[SortedWeightModel] <= 0.7, "A",
IF( Product[SortedWeightModel] <= 0.9, "B", "C" ) )
```

Product	Model	SalesAmountProduct	CumulatedModel	SortedWeightModel	ABC Class Model
Road-150 Red, 44	Road-150	1,940,459.90	2,913,701.93	71,772,543.62	65.34 % A
Road-150 Red, 62	Road-150	1,769,096.69	7,913,701.93	77,018,795.64	70.12 % B
Road-150-W Yellow, 48	Road-150-W	694,003.52	5,246,152.02	77,018,795.64	70.12 % B
Road-150-W Yellow, 44	Road-150-W	1,130,046.50	5,246,152.02	77,018,795.64	70.12 % B

A COMPLEX EXAMPLE INVOLVING MANY TO MANY RELATIONSHIPS

M2M and Audience

sqlbi.com

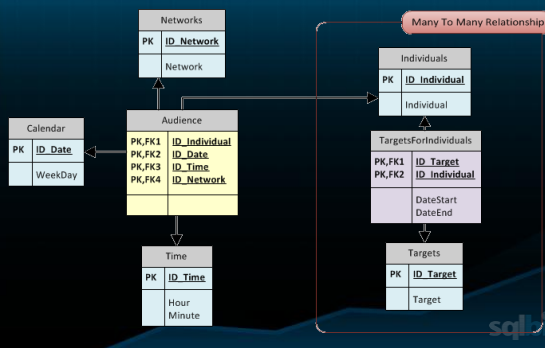
Audience

- Taken from a real world analysis
- Audience data of TV broadcasts
 - Simple data model
 - Pretty big database
 - Analysis at a very detailed level
- Current solution
 - Ad-hoc software
- Proof of Concept
 - Analysis with PowerPivot and Vertipaq

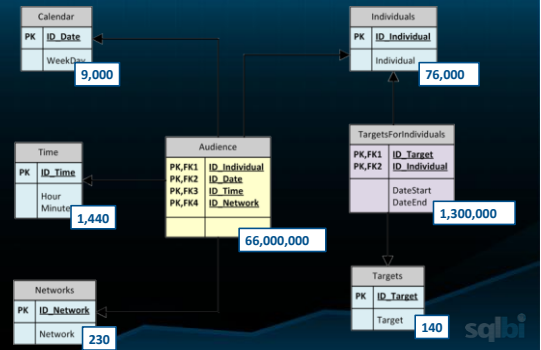
Audience: background

- Info taken from the remote control
 - Once a minute
 - Individual watching
 - Channel watched
 - Once a day
 - All the individuals watching
 - Detect the influence of each individual
 - Individual
 - Categories
 - Targets

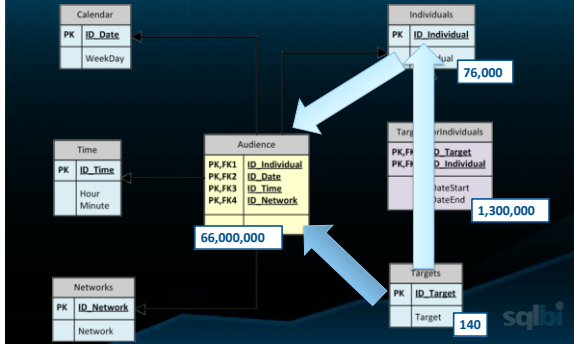
Audience: The Data Model



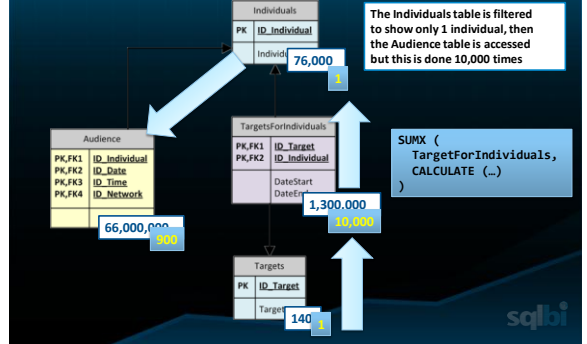
Audience: The Numbers



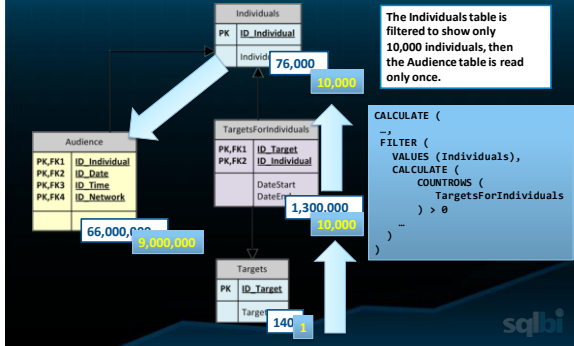
Audience: The Analysis



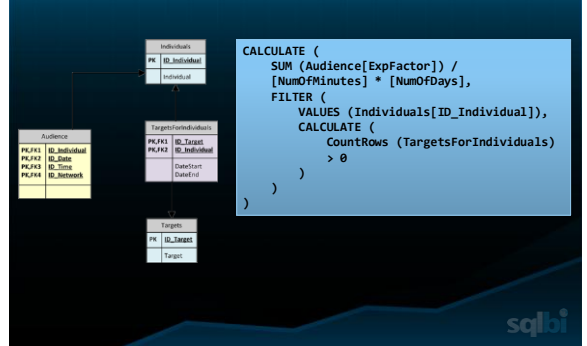
The Naif Formula



A different approach



Audience: The Final Formula



IS DIMENSIONAL MODELING STILL NECESSARY FOR POWERPivot / BISM?

Dimensional vs Relational Modeling

Dimensional vs Relational: UDM

- UDM
 - Strong separation between facts and dimensions
 - Advanced relationship types
 - Great aggregation level computations
 - Poor leaf level computation performances
- Designed, optimized for dimensional models
- No brain option: go dimensional

Dimensional vs Relational: BISM

- o BISM
 - No concept of fact and dimension: only tables
 - Very simple concept of relationship
 - No aggregations, in memory computation
 - Great leaf level computation performances
- o Optimized for simple models
- o Two options
 - Dimensional modeling
 - Relational modeling

Dimensional vs Relational: BISM

- o Relational Modeling
 - Does not mean third normal form!
 - Users like facts and dimensions
- o Denormalization
 - Needed for the user experience
 - And for performances
- o Each relationship has a cost, remember it!

BISM: Advanced Relationships

- o Advanced Relationships in BISM
 - Are not supported
 - But can be implemented through DAX
 - Between (Banding)
 - Many to many (Audience)
 - Parent / Child (Naturalization)
 - Reference Dimensions (RELATED)
 - LastChild (CALCULATE)
 - Fact Relationships (not needed)
- o Thus, nothing is really missing

BISM: New Power

- o Calculated columns
 - Materialize complex calculations
 - ABC Pareto is a good example
 - Have access to the whole data model
 - o Not like calculated columns in UDM DSV!
- o Linked tables
 - Two steps computation
- o Easy integration with many sources
- o Power to the end user

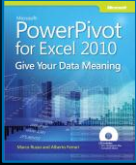
BISM: DAX

- o Brand new language
- o Mixes programming and querying
- o Much cleaner and easier than MDX
- o Designed for efficiency on modern CPU
- o Hard to enter the «DAX Mind»
 - For MDX and SQL programmers
 - Don't search same concepts, they are not there!
- o Once you master it... awesome!


What is changing?

DENALI	SQL 2008 R2
BISM	UDM
DAX	MDX
RELATIONAL	MULTIDIMENSIONAL
IN MEMORY	ON DISK
FAST VERTIPAQ ENGINE	AGGREGATIONS
LEAF LEVEL COMPUTATION	CALCULATED MEMBERS
FILTER / ROW CONTEXT	MDX FUNCTIONS, TUPLES ETC
SELF SERVICE + CORPORATE	CORPORATE ONLY
EASY DATA INTEGRATION	BI PROFESSIONALS NEEDED

A NEW BOOK FROM SQLBI



PowerPivot for Excel 2010: give your data meaning

sqlbi.com 

**Questions
and
Answers**

sqlbi.com 

Links

- SQLBI Website
www.sqlbi.com
- PowerPivot Workshop
www.powerpivotworkshop.com
- Marco Russo blog
www.sqlblog.com/blogs/marco_russo
- Alberto Ferrari blog
www.sqlblog.com/blogs/alberto_ferrari

For any question contact us at
info@sqlbi.com

sqlbi.com 

Thank you!

sqlbi.com 