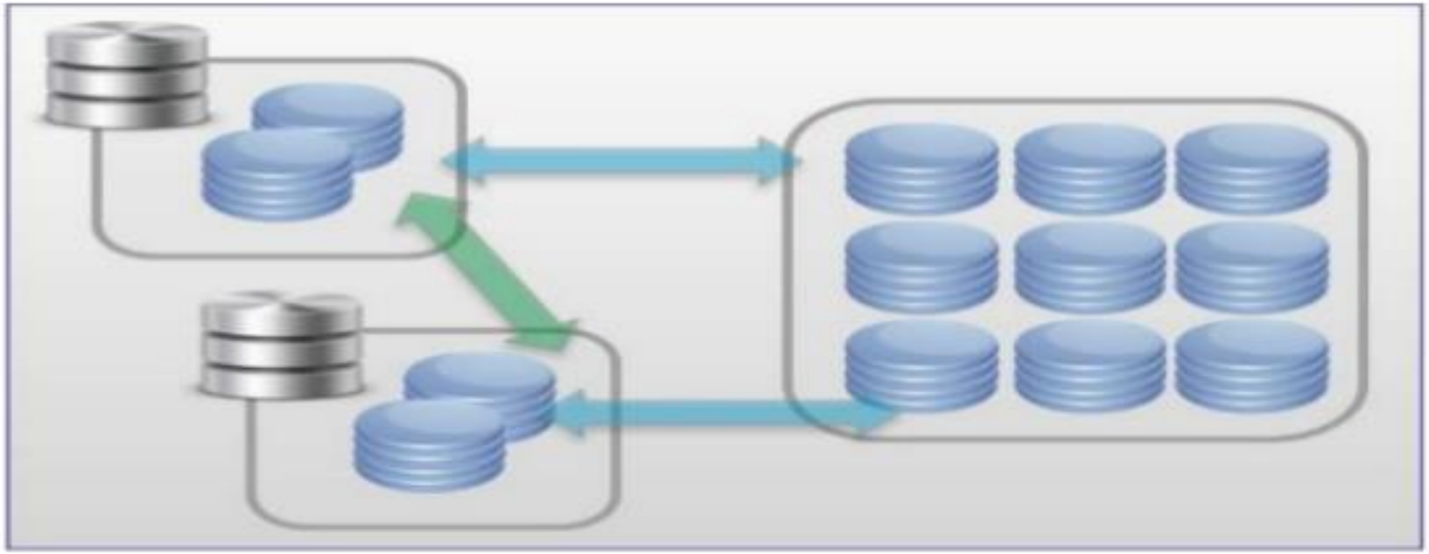
Outline

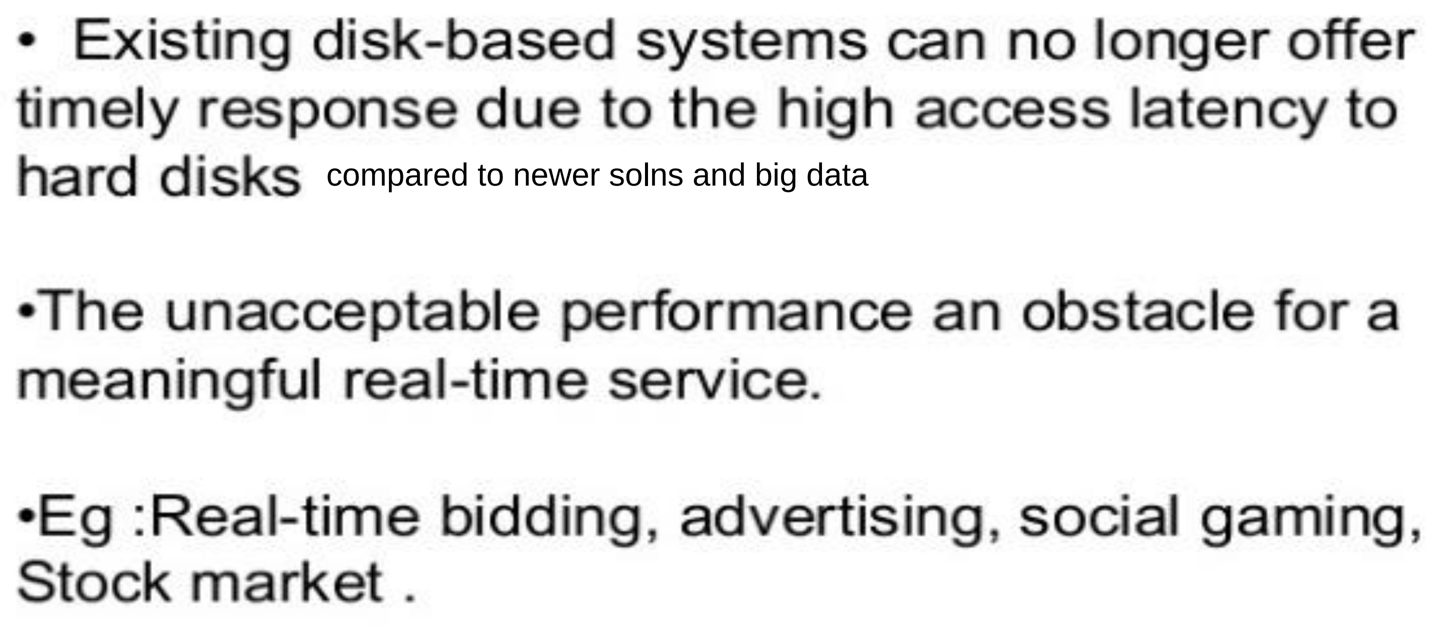
- Learn OLAP In-memory engine concepts.
- Develop a tabular project.

Data Store for Typical database

- Data resides on disk.
- Data maybe cached into memory for access.



Problems with current Databases

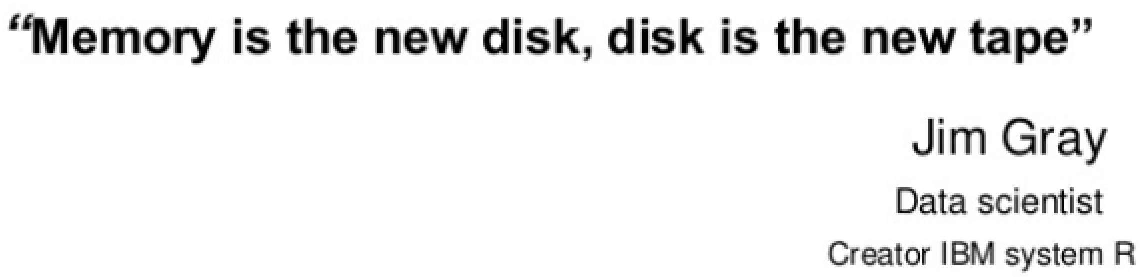


- Existing disk-based systems can no longer offer timely response due to the high access latency to hard disks compared to newer solns and big data

- The unacceptable performance an obstacle for a meaningful real-time service.

- Eg :Real-time bidding, advertising, social gaming, Stock market .

Shift in Data Storage



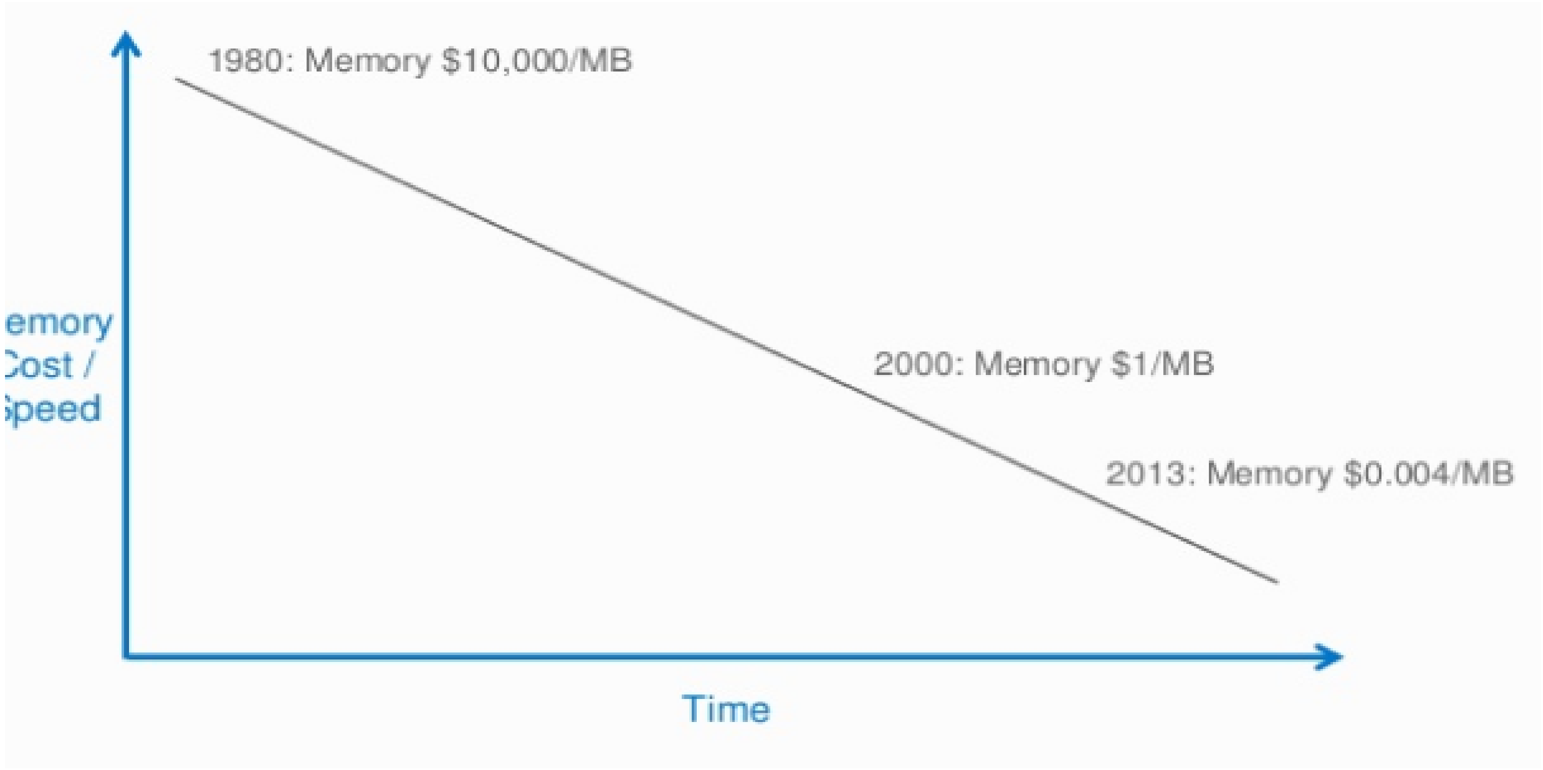
“Memory is the new disk, disk is the new tape”

Jim Gray

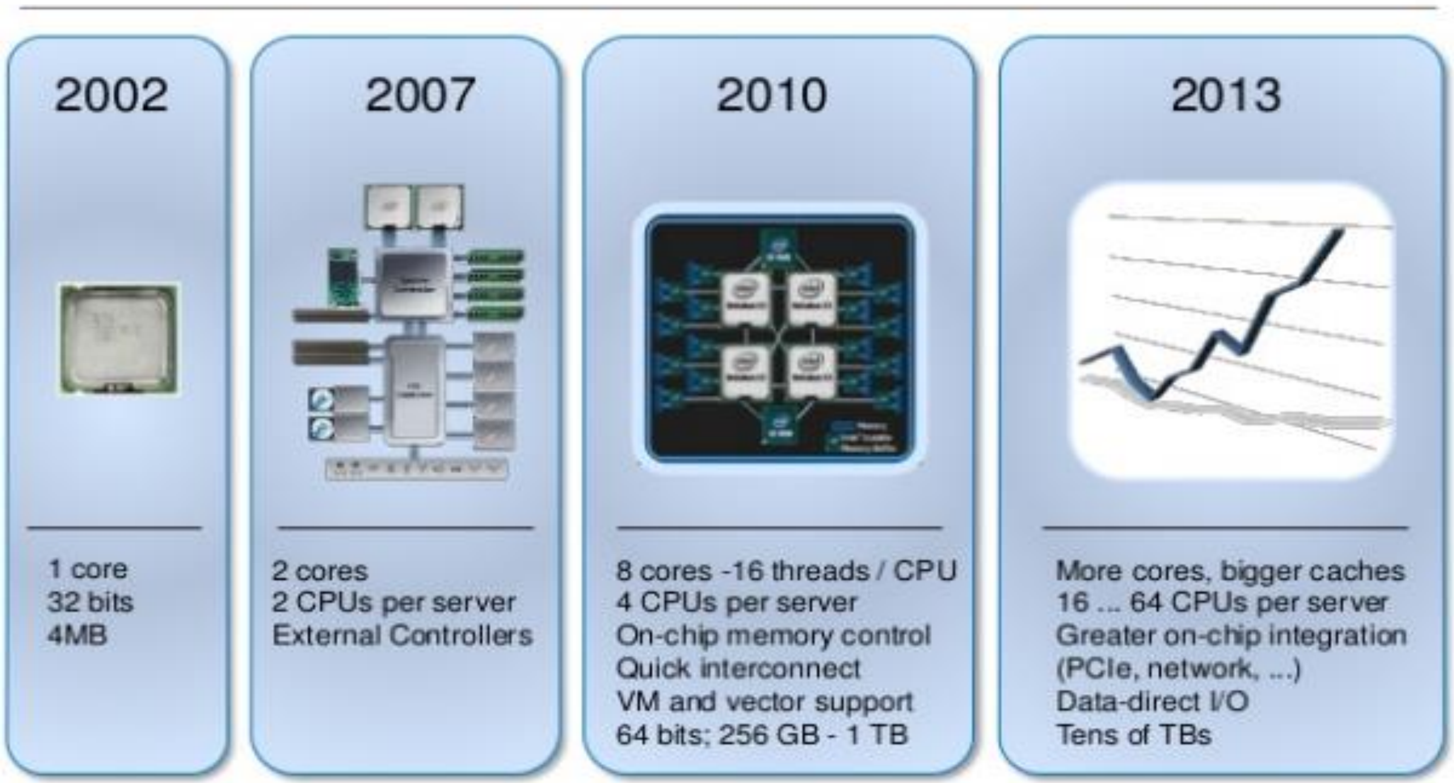
Data scientist

Creator IBM system R

Moore’s Law


1980: Memory $10,000/MB
2000: Memory $1/MB
2013: Memory $0.004/MB
emory
Cost /
Speed
Time

CPU Improvements



2002

1 core
32 bits
4MB

2007

2 cores
2 CPUs per server
External Controllers

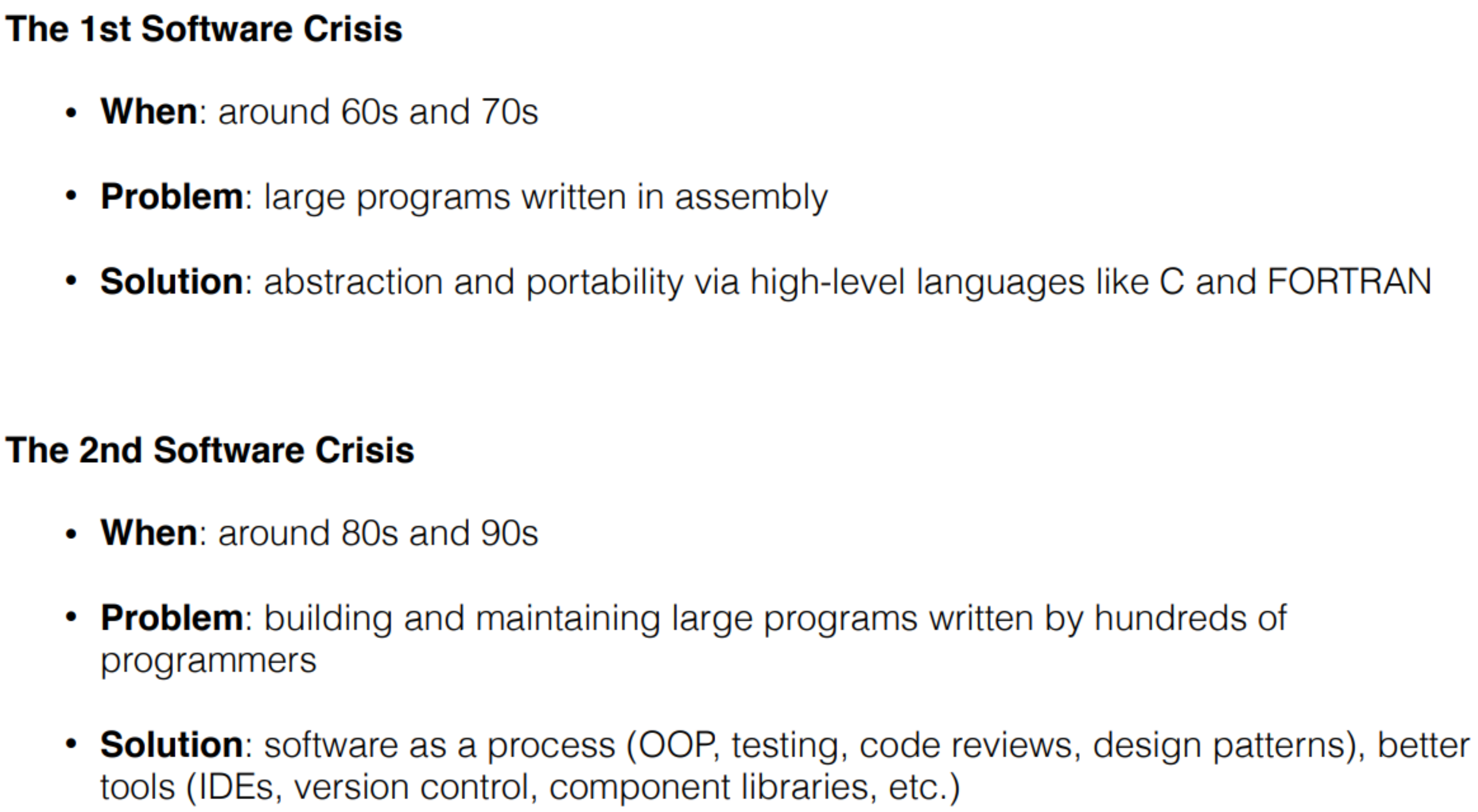
2010

8 cores -16 threads / CPU
4 CPUs per server
On-chip memory control
Quick interconnect
VM and vector support
64 bits; 256 GB - 1 TB

2013

More cores, bigger caches
16 ... 64 CPUs per server
Greater on-chip integration
(PCIe, network, ...)
Data-direct I/O
Tens of TBs

Crisis – brought on by technology



The 1st Software Crisis

- **When**: around 60s and 70s
- **Problem**: large programs written in assembly
- **Solution**: abstraction and portability via high-level languages like C and FORTRAN

The 2nd Software Crisis

- **When**: around 80s and 90s
- **Problem**: building and maintaining large programs written by hundreds of programmers
- **Solution**: software as a process (OOP, testing, code reviews, design patterns), better tools (IDEs, version control, component libraries, etc.)

Emerging crisis....

- **When**: 2005 and ...
- **Problem**: sequential performance is stuck
- **Required solution**: continuous and reasonable performance improvements
    - To process large datasets (BIG Data!)
    - To support new features
    - Without loosing portability and maintainability

SMP – Architecture

Limit of an SMP configuration is somewhere! Between 16 and 64 processors.

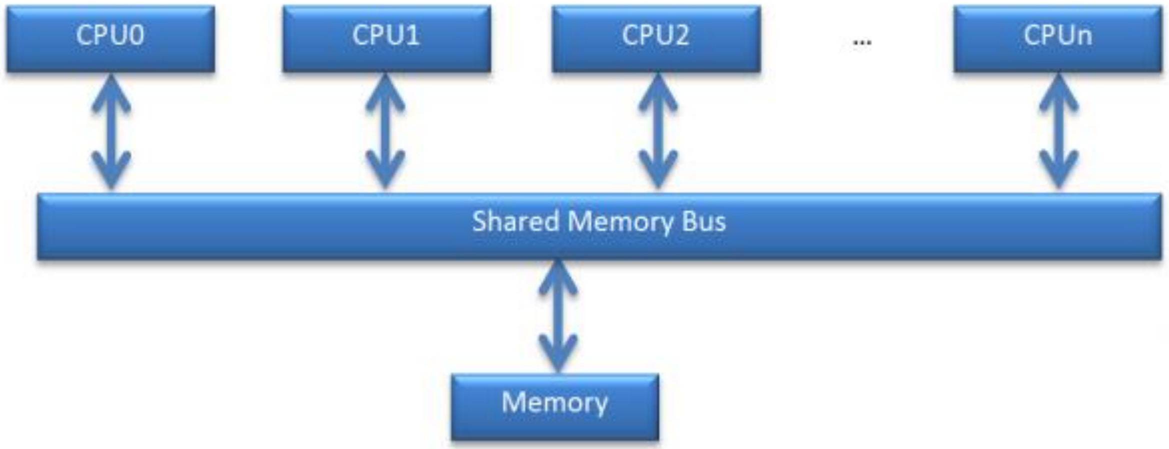
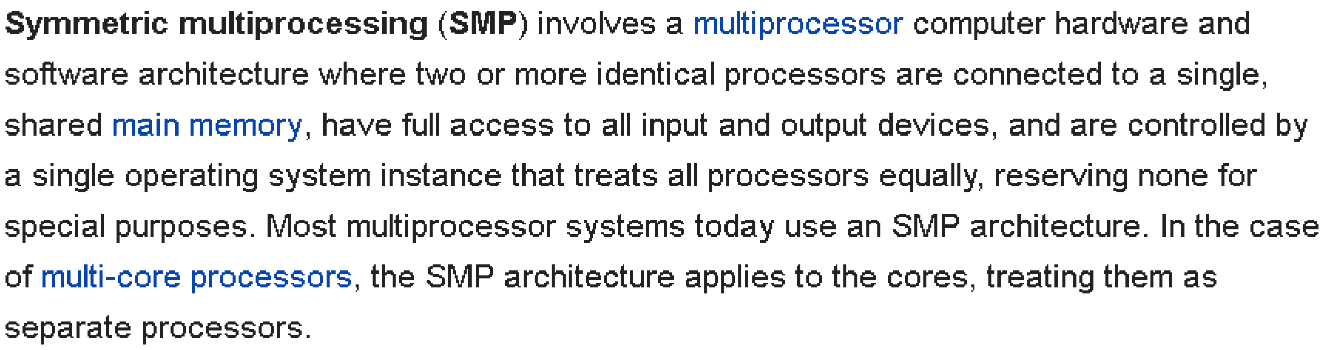


Figure 1 - The SMP Architecture



Symmetric multiprocessing (**SMP**) involves a multiprocessor computer hardware and software architecture where two or more identical processors are connected to a single, shared main memory, have full access to all input and output devices, and are controlled by a single operating system instance that treats all processors equally, reserving none for special purposes. Most multiprocessor systems today use an SMP architecture. In the case of multi-core processors, the SMP architecture applies to the cores, treating them as separate processors.

Numa – (non-uniform memory access) architecture

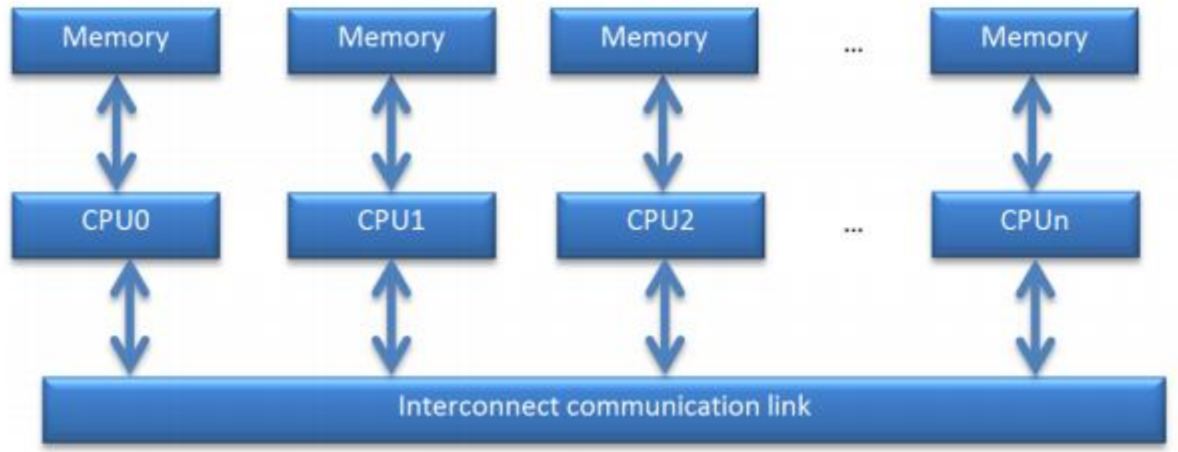
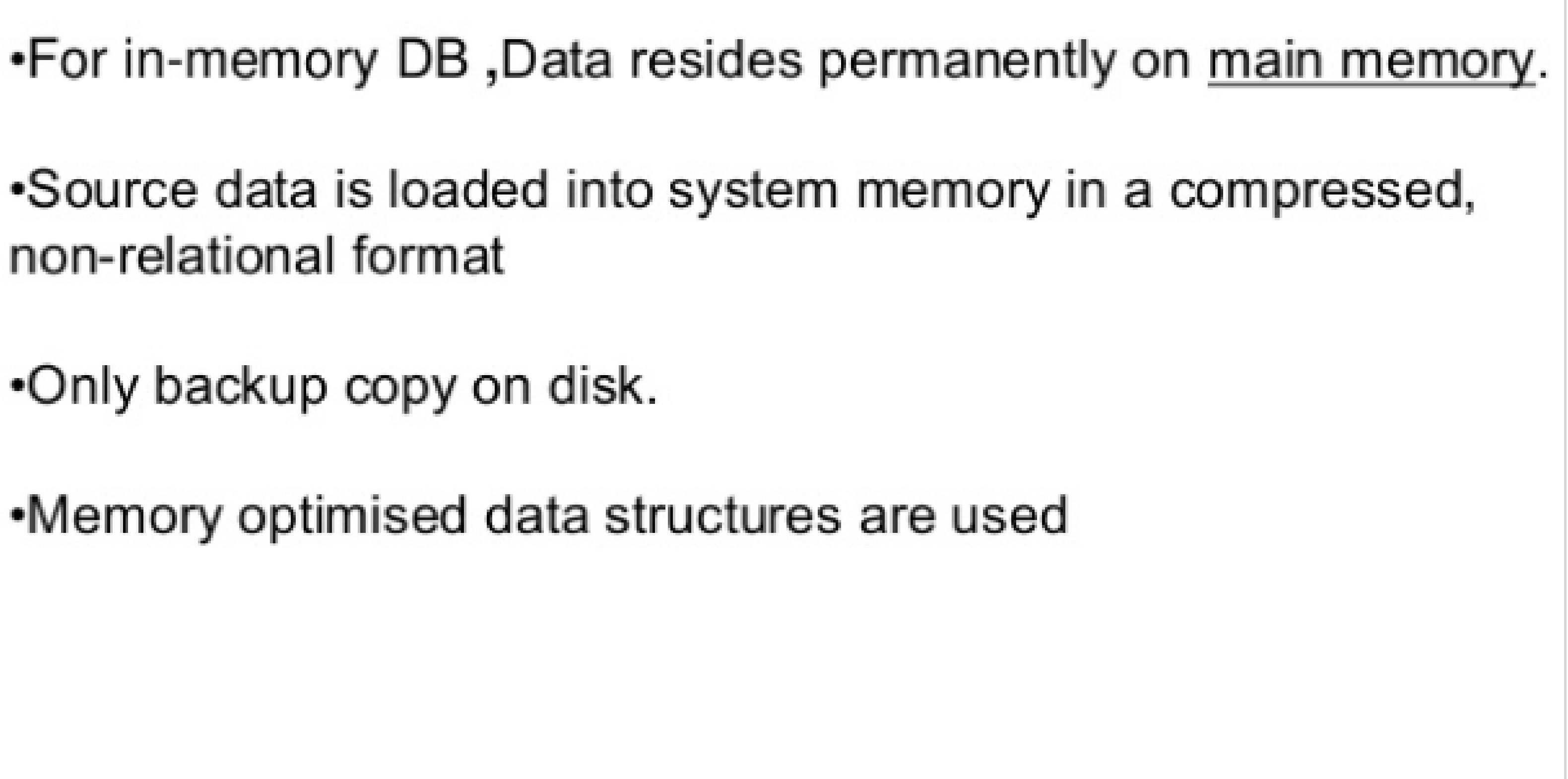


Figure 2 - The NUMA Architecture

In Memory Database Systems - characteristics



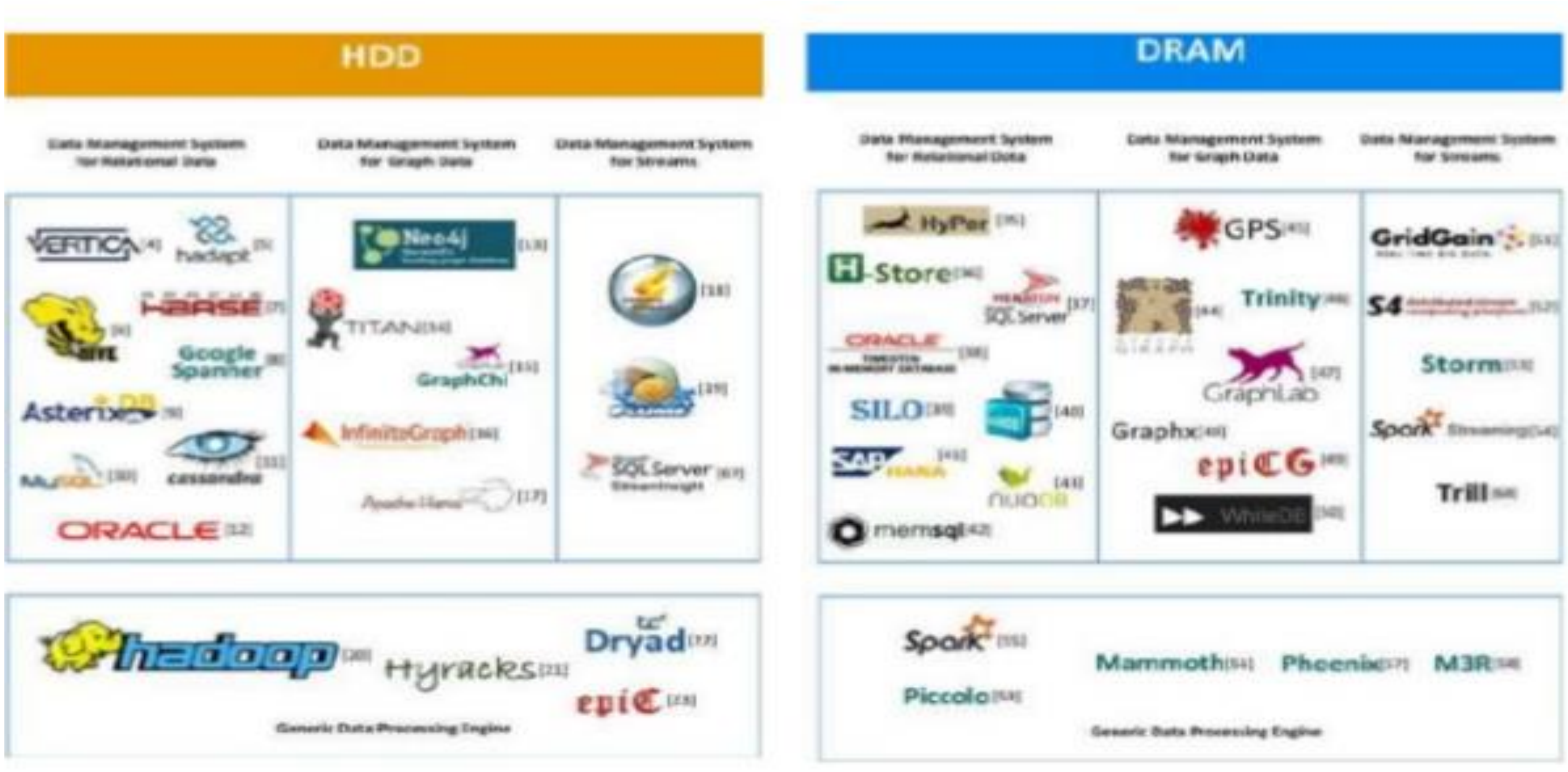
- For in-memory DB ,Data resides permanently on main memory.

- Source data is loaded into system memory in a compressed, non-relational format

- Only backup copy on disk.

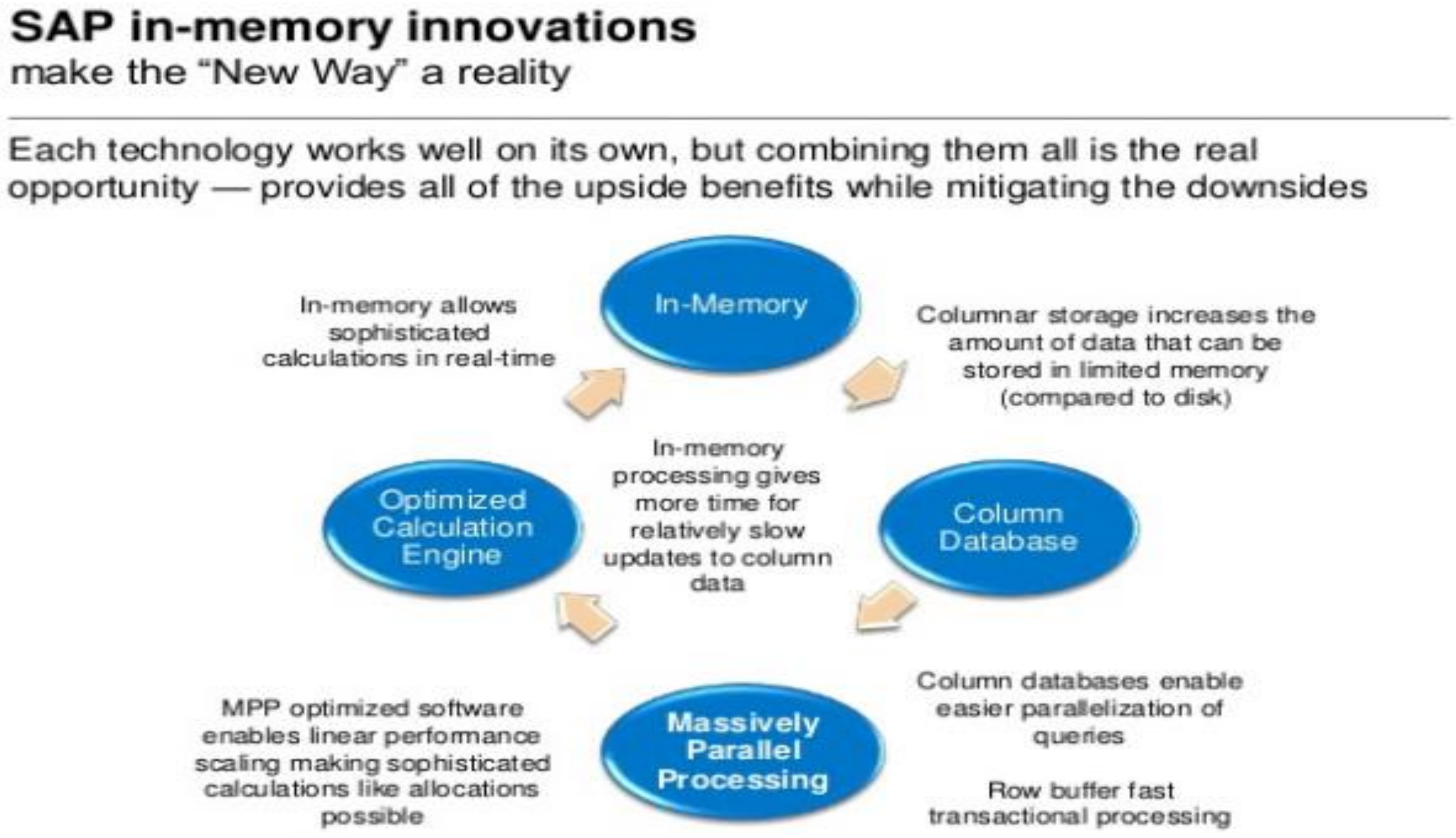
- Memory optimised data structures are used

Database Space



hadoop is probably somewhere half way in b/t

SAP in-memory innovations



SAP in-memory innovations

make the "New Way" a reality

Each technology works well on its own, but combining them all is the real opportunity — provides all of the upside benefits while mitigating the downsides

In-memory allows sophisticated calculations in real-time

Columnar storage increases the amount of data that can be stored in limited memory (compared to disk)

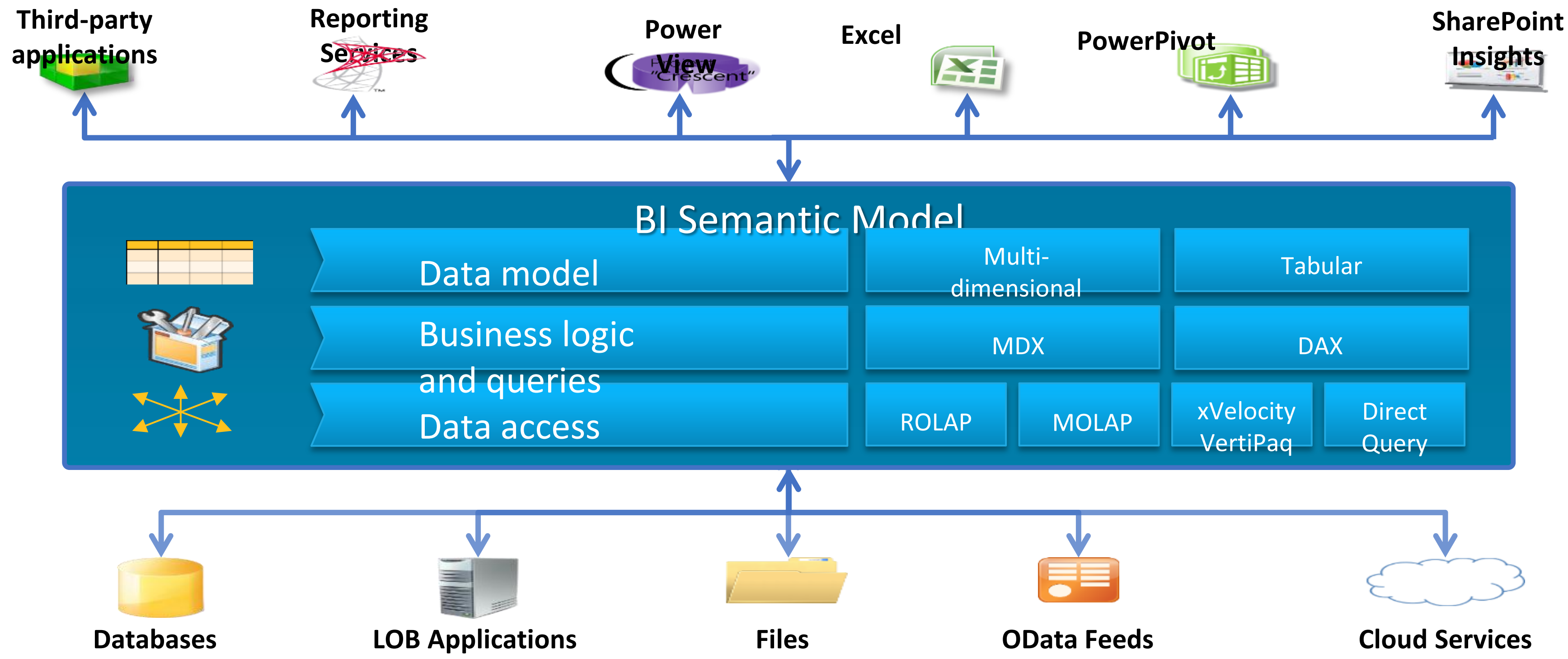
In-memory processing gives more time for relatively slow updates to column data

Column databases enable easier parallelization of queries

Row buffer fast transactional processing

MPP optimized software enables linear performance scaling making sophisticated calculations like allocations possible

BISM – BI Semantic Model



Third-party applications

Reporting Services

Power View "Crescent"

Excel

PowerPivot

SharePoint Insights

BI Semantic Model

Data model	Multi-dimensional	Tabular
Business logic and queries	MDX	DAX
Data access	ROLAP MOLAP	xVelocity VertiPaq Direct Query

Databases

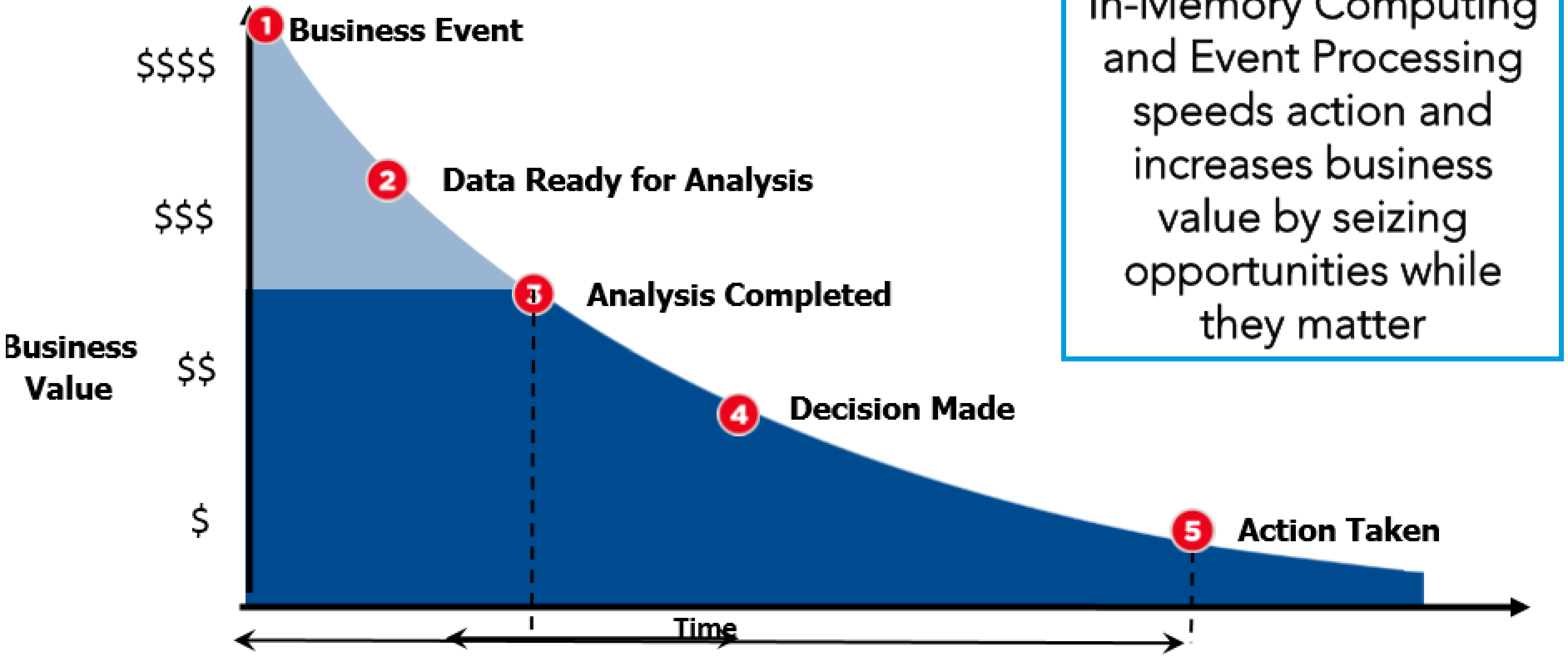
LOB Applications

Files

OData Feeds

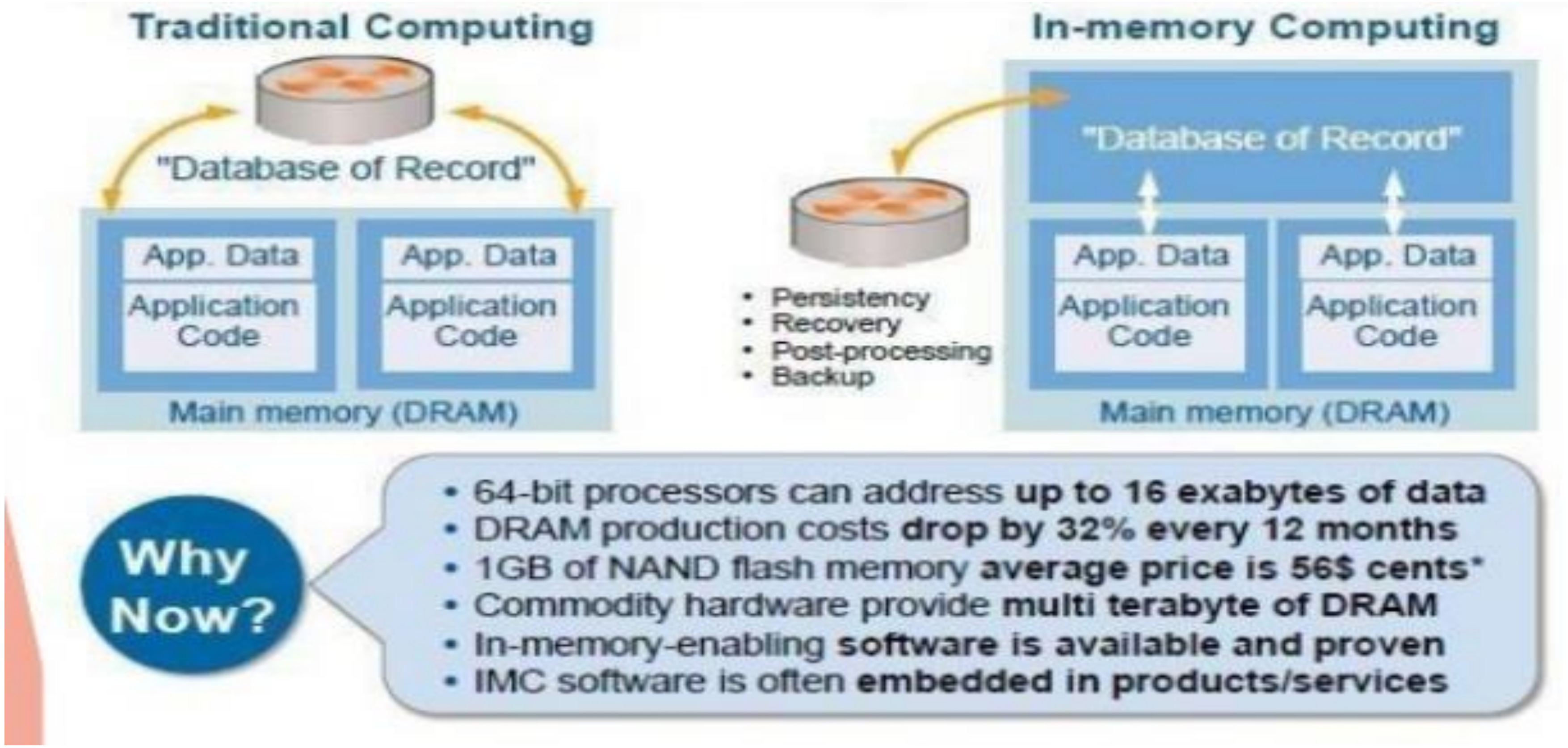
Cloud Services

Drivers for in memory computing



In-Memory Computing and Event Processing speeds action and increases business value by seizing opportunities while they matter

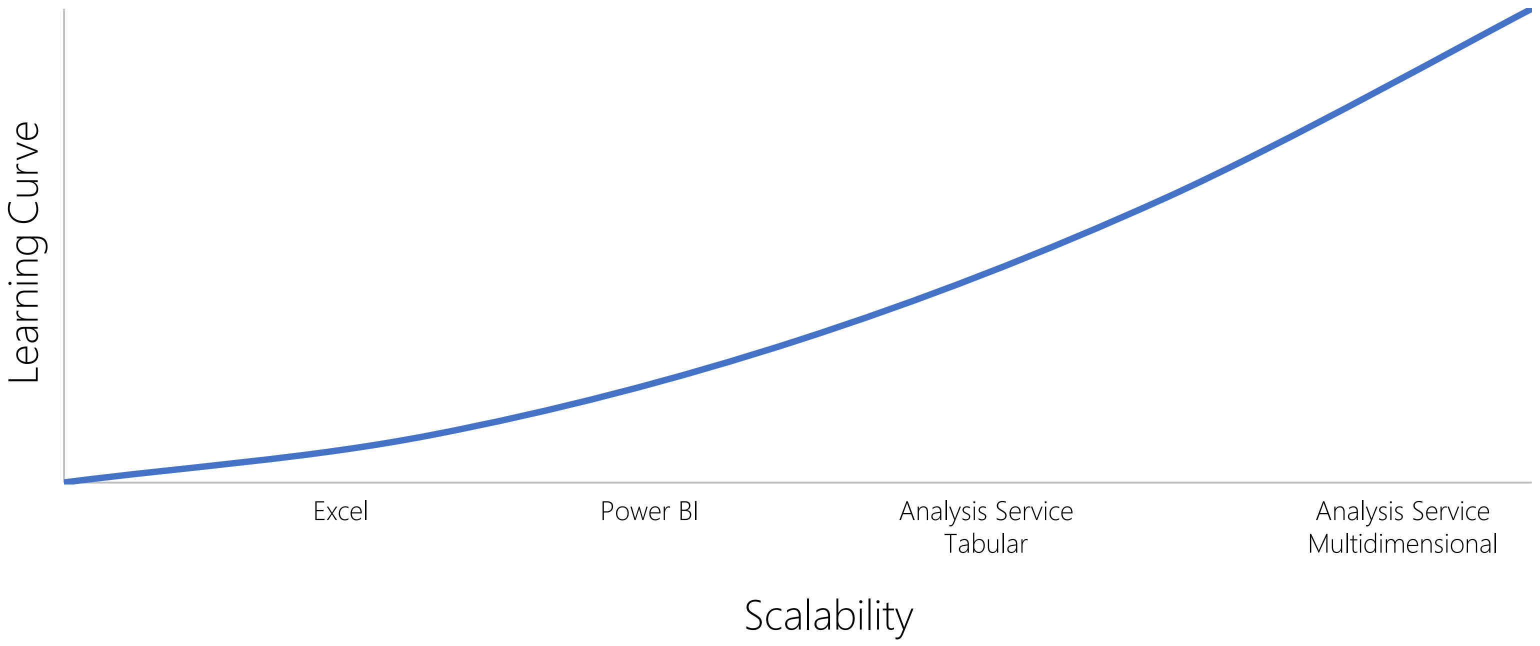
What is in memory computing



- Persistency
- Recovery
- Post-processing
- Backup

- 64-bit processors can address **up to 16 exabytes of data**
- DRAM production costs **drop by 32% every 12 months**
- 1GB of NAND flash memory **average price is 56$ cents***
- Commodity hardware provide **multi terabyte of DRAM**
- In-memory-enabling **software is available and proven**
- IMC software is often **embedded in products/services**

Different Flavors of Analysis



used to be called veripaq

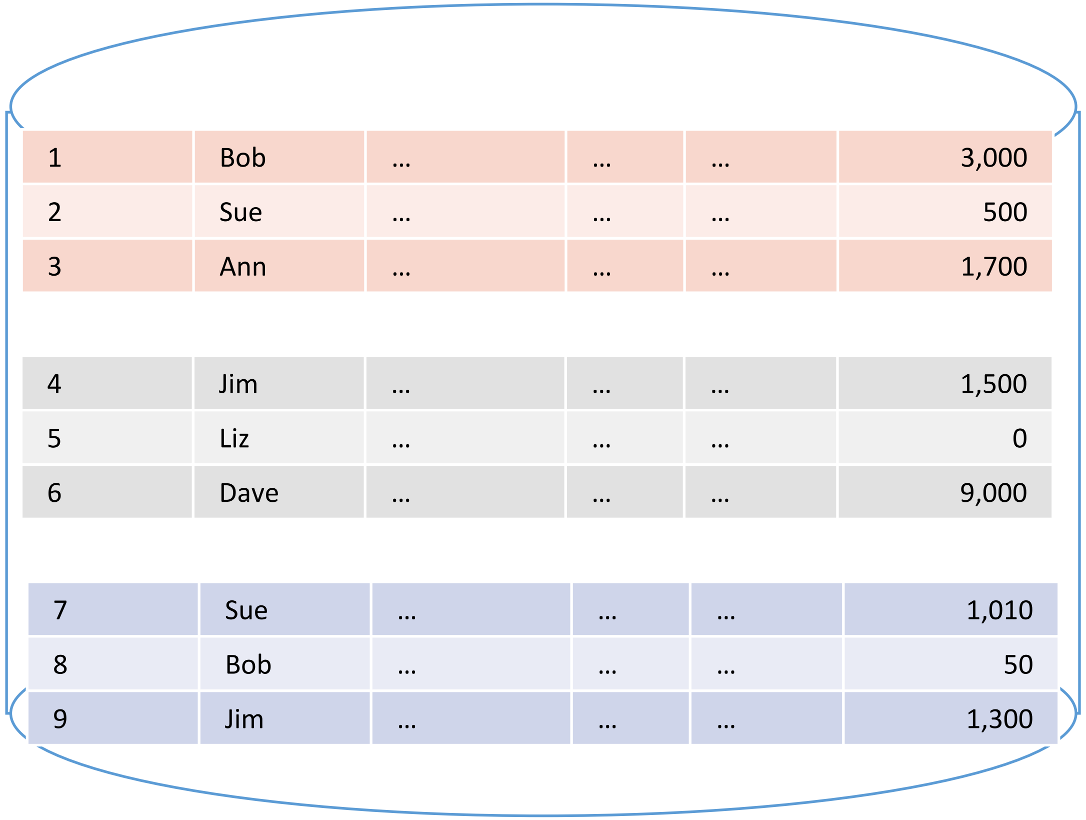
What is xVelocity in-memory?

- In-memory database
- Based on the relational methodology
- Column oriented database
- Data is stored in a compressed format

Row Storage Layout

Customers Table

ID	Name	Address	City	State	Bal Due
1	Bob	…	…	…	3,000
2	Sue	…	…	…	500
3	Ann	…	…	…	1,700
4	Jim	…	…	…	1,500
5	Liz	…	…	…	0
6	Dave	…	…	…	9,000
7	Sue	…	…	…	1,010
8	Bob	…	…	…	50
9	Jim	…	…	…	1,300



1	Bob	…	…	…	3,000
2	Sue	…	…	…	500
3	Ann	…	…	…	1,700

4	Jim	…	…	…	1,500
5	Liz	…	…	…	0
6	Dave	…	…	…	9,000

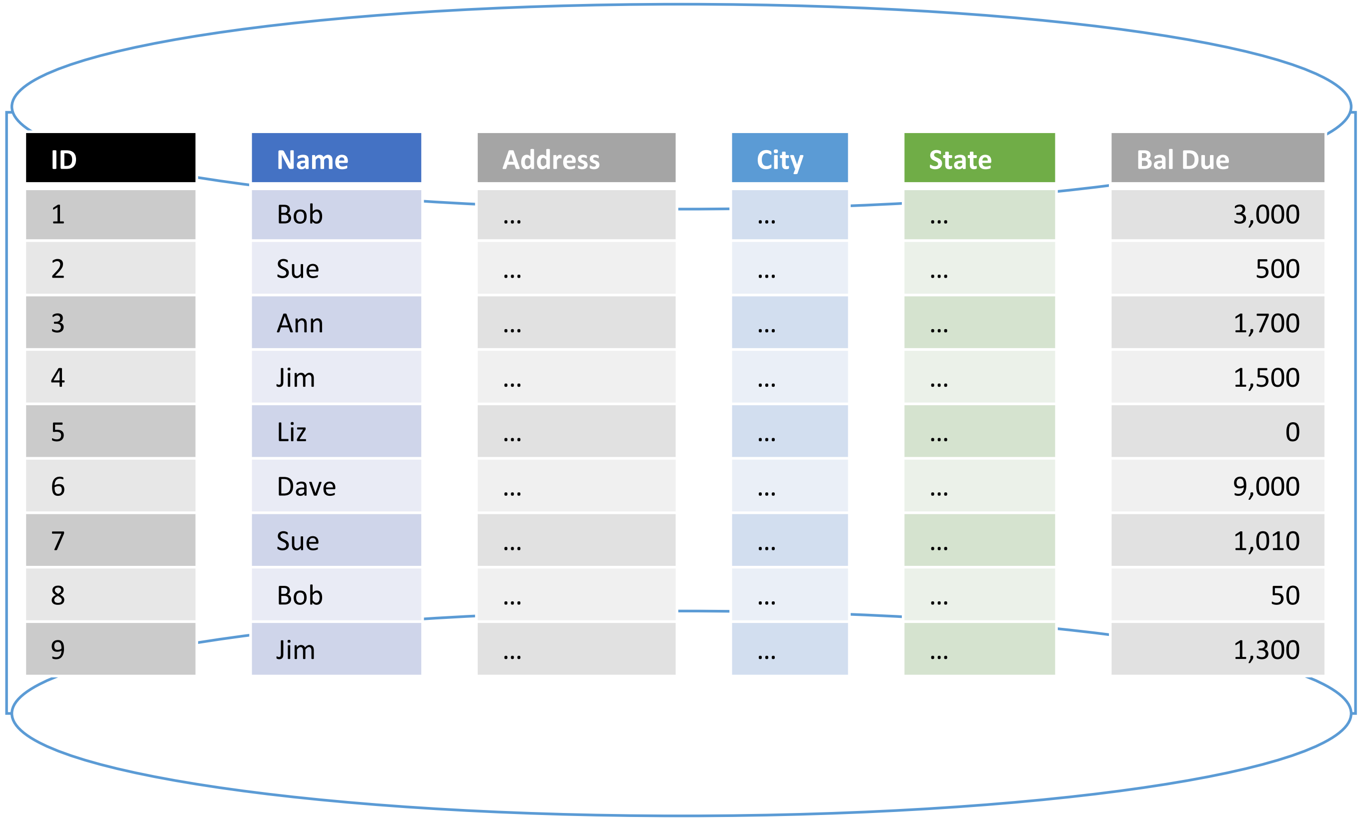
7	Sue	…	…	…	1,010
8	Bob	…	…	…	50
9	Jim	…	…	…	1,300

Nothing special here.
This is the standard way database systems have been laying out tables on disk since the mid 1970s.
Technically, it is called a “row store”

Column Storage Layout

Customers Table

ID	Name	Address	City	State	Bal Due
1	Bob	…	…	…	3,000
2	Sue	…	…	…	500
3	Ann	…	…	…	1,700
4	Jim	…	…	…	1,500
5	Liz	…	…	…	0
6	Dave	…	…	…	9,000
7	Sue	…	…	…	1,010
8	Bob	…	…	…	50
9	Jim	…	…	…	1,300



ID	Name	Address	City	State	Bal Due
1	Bob	…	…	…	3,000
2	Sue	…	…	…	500
3	Ann	…	…	…	1,700
4	Jim	…	…	…	1,500
5	Liz	…	…	…	0
6	Dave	…	…	…	9,000
7	Sue	…	…	…	1,010
8	Bob	…	…	…	50
9	Jim	…	…	…	1,300

Tables are stored “column-wise” with all values from a single column stored in a single block

Column vs Row Storage

- Column Storage
  - Quick access to a single column
  - Time needed to materialize (re-create) rows
  - Trade CPU vs I/O
- Row Storage
  - Quick access to a single row
  - No materialization needed
  - Trade I/O vs CPU

Run Length Encoding (RLE)

will be on the test

Quarter	ProdID	Price
Q1	1	100
Q1	1	120
Q1	1	315
Q1	1	100
Q1	1	315
Q1	2	198
…	2	450
Q2	2	320
Q2	…	320
Q2	1	150
Q2	1	256
Q2	1	450
Q2	1	192
Q2	1	184
Q2	2	310
Q2	2	251
…	2	266

Quarter	Start	Count
Q1	1	310
Q2	311	290
…	…	…

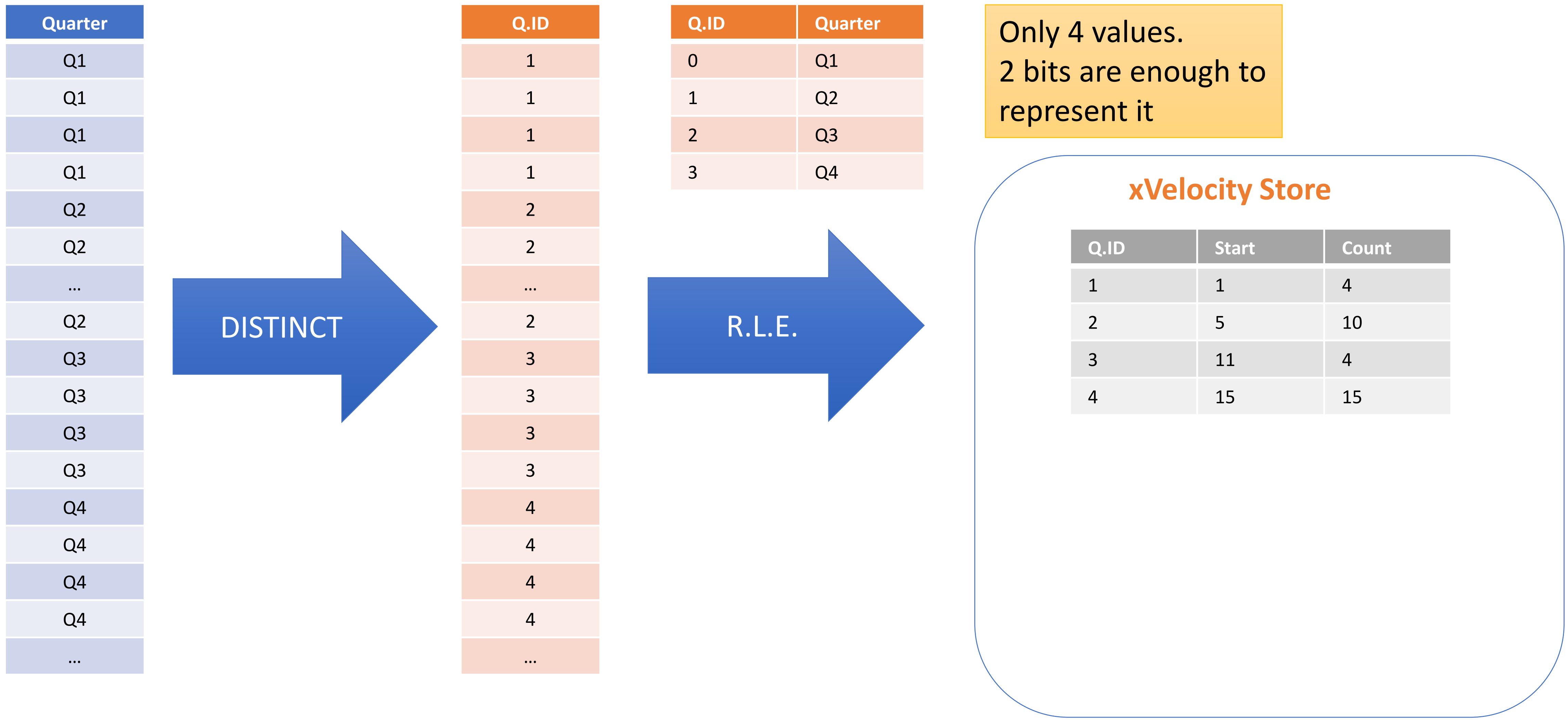
ProdID	Start	Count
1	1	5
2	6	3
…	…	…
1	51	5
2	56	3

Price
100
120
315
100
315
198
450
320
320
150
256
450
192
184
310
251
266

RLE Compression applied only when size of compressed data is smaller than original

Dictionary Encoding – Bits to represent values

Combined with RLE



Quarter
Q1
Q1
Q1
Q1
Q2
Q2
…
Q2
Q3
Q3
Q3
Q3
Q4
Q4
Q4
Q4
…

Q.ID
1
1
1
1
2
2
…
2
3
3
3
3
4
4
4
4
…

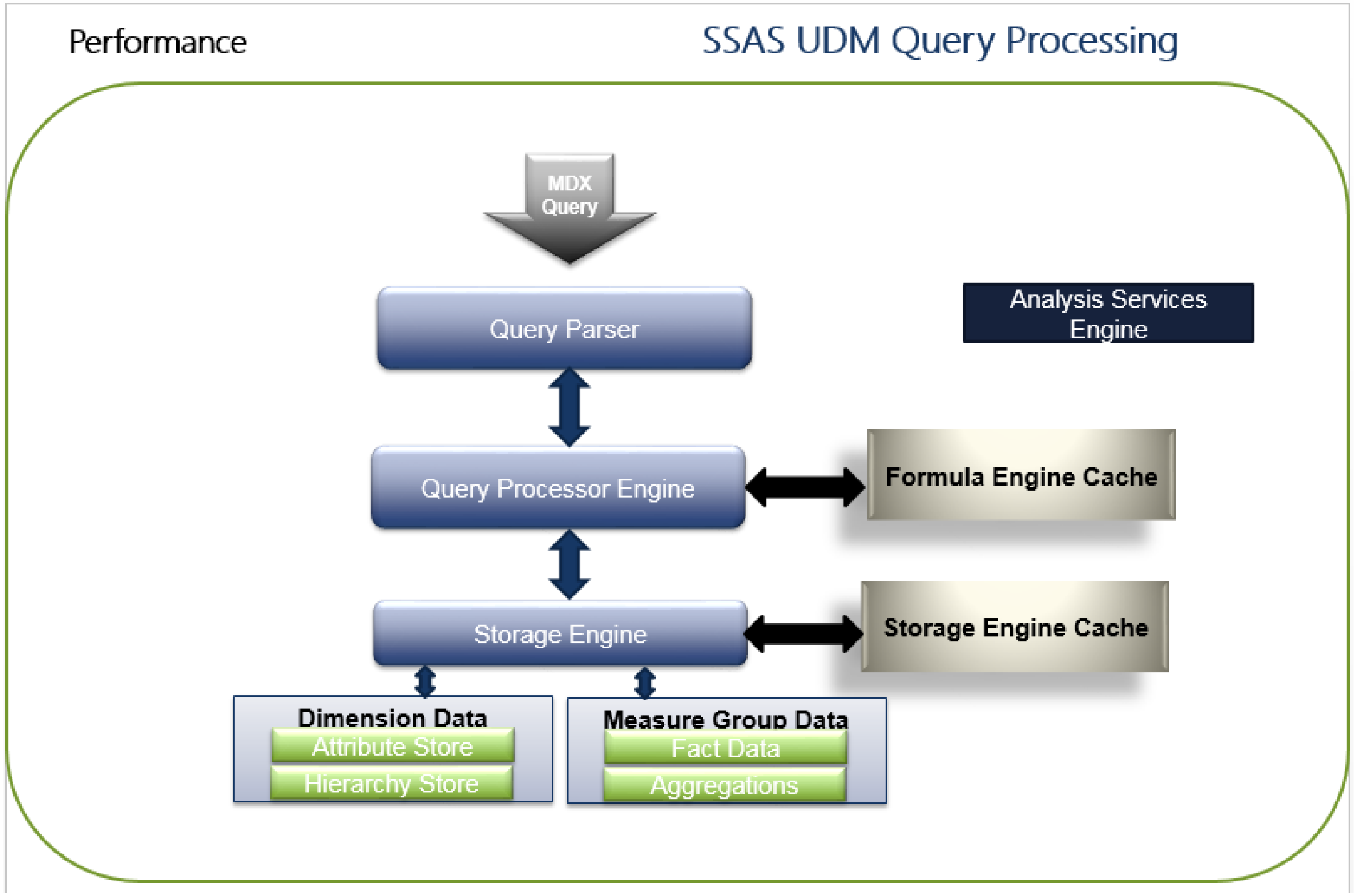
Q.ID	Quarter
0	Q1
1	Q2
2	Q3
3	Q4

Only 4 values.
2 bits are enough to represent it

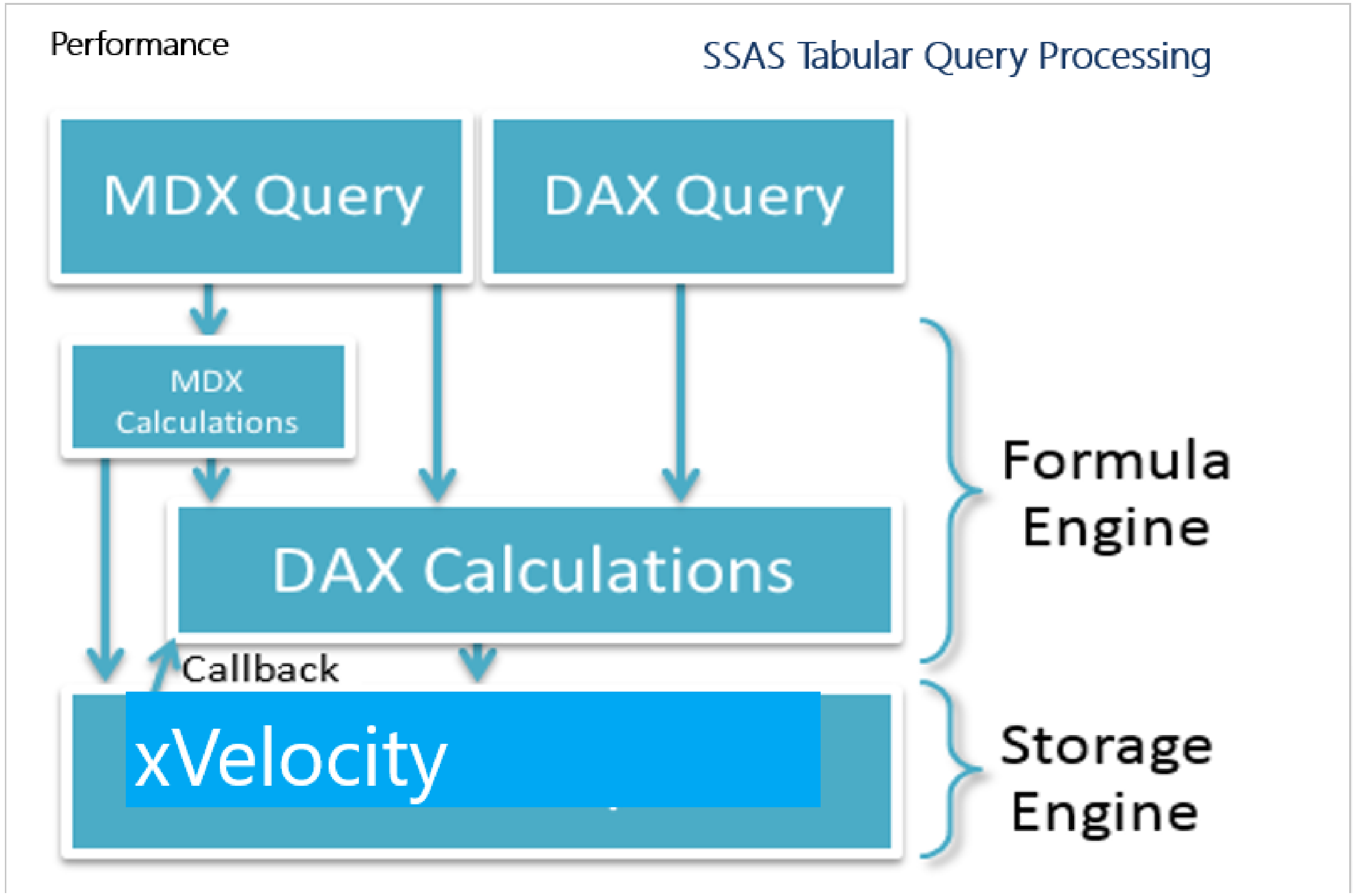
xVelocity Store

Q.ID	Start	Count
1	1	4
2	5	10
3	11	4
4	15	15

Multidimensional query processing – aggregations taken off the hard disk



The formula engine is single-threaded per query.
High degree of parallelism in the storage engine - multiple threads.



xVelocity works only in-memory

- Compressed data
  - Must fit in memory
  - Otherwise, it simply does not work
- On the hardware side this means
  - Very fast CPU
  - Very fast memory
  - Disks are not important at all

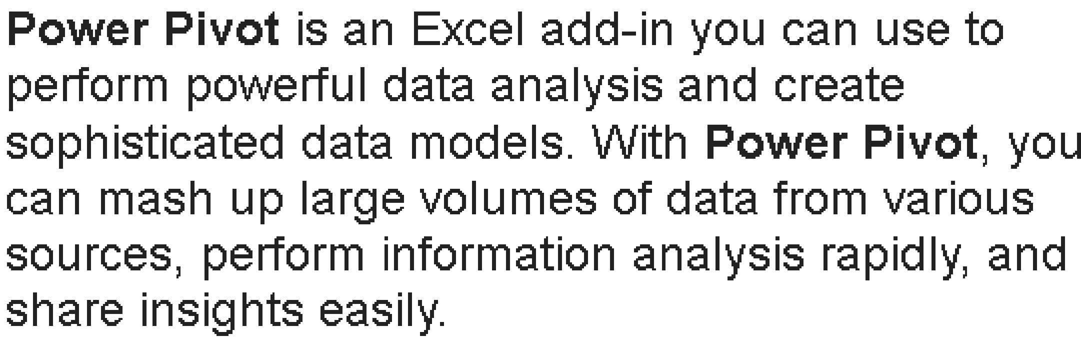
Which Server for SSAS?

Feature	Multidimensional	Tabular
RAM	Some (16/32 Gb)	A lot (64/128 Gb)
RAM Speed	Important	**Crucial**
Number of cores	4/8/16	4/8/16
Core speed	Less Important	**Crucial**
Disk speed	**Very Important**	Useless
SSD Disk Usage	**Strongly recommended**	Useless
Network speed	Important	Important
Concurrency	Pretty good ACID database transactions	Not enough experience…

Ideally, don't use the same server for both

So What’s Analysis Services TABULAR ANYWAY

- “Server Side version of PowerPivot v1”
- Development in Visual Studio, not in Excel
- Adding PowerPivot v2 features
- Adding Enterprise features like
  - security
  - partitions
  - management
  - …



Power Pivot is an Excel add-in you can use to perform powerful data analysis and create sophisticated data models. With **Power Pivot**, you can mash up large volumes of data from various sources, perform information analysis rapidly, and share insights easily.

What’s new compared to PowerPivot v1

- Richer Models
  - KPIs
  - Descriptions
  - Persisted formatting
  - Advanced sorting
  - Mark as Date Table
  - Distinct count
  - Drill-through
  - Perspectives
  - Hierarchies
  - Multiple relationships
  - Parent child

Optimized Usability

Improved Date and Text filtering

Diagram

Measure grid

Various usability enhancements

Reporting Properties

Representative Column

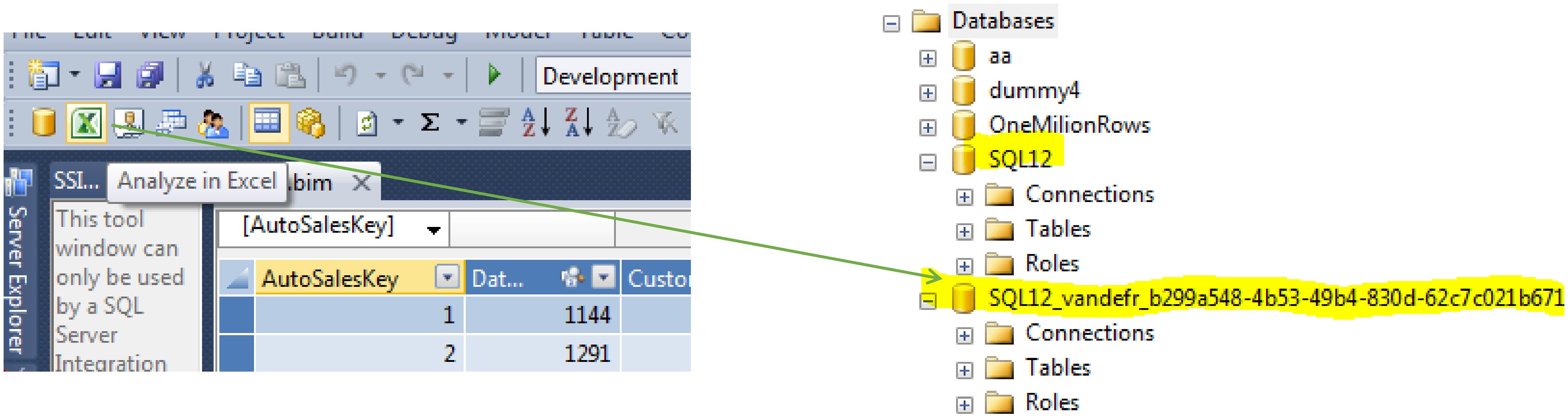
Table Detail

Table Identifier

ImageURL

The Workspace Database

- Can be local or remote server
- Holds in memory “development copy” of model
- Databasename = ProjectName + username + GUID
- Analyze in Excel connects to workspace DB
- No need to redeploy to see changes in Excel or Cube browser



This tool window can only be used by a SQL Server Integration

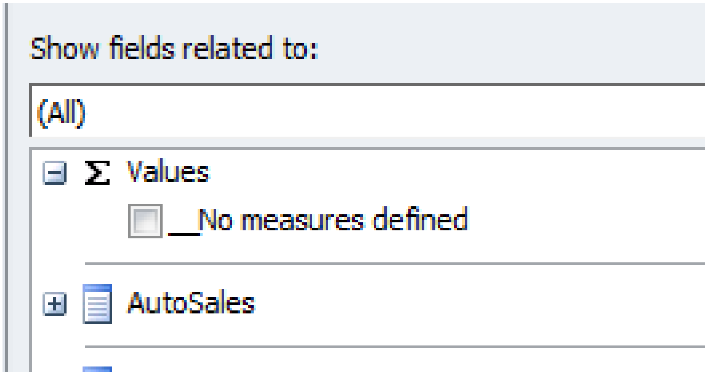
[AutoSalesKey]

AutoSalesKey	Dat...	Custo
1	1144	
2	1291	

- Databases
  - aa
  - dummy4
  - OneMilionRows
  - SQL12
    - Connections
    - Tables
    - Roles
  - SQL12_vandefr_b299a548-4b53-49b4-830d-62c7c021b671
    - Connections
    - Tables
    - Roles

Measures must be explicitly created

- PowerPivot will create implicit measures when you drag a column to the values area in a pivot table
- Not possible in BISM [BI Semantic model] (No_ measure defined)
- Like in SSAS OLAP – measures must be created explicitely
- Easiest way: AutoSum in SQL Data Tools



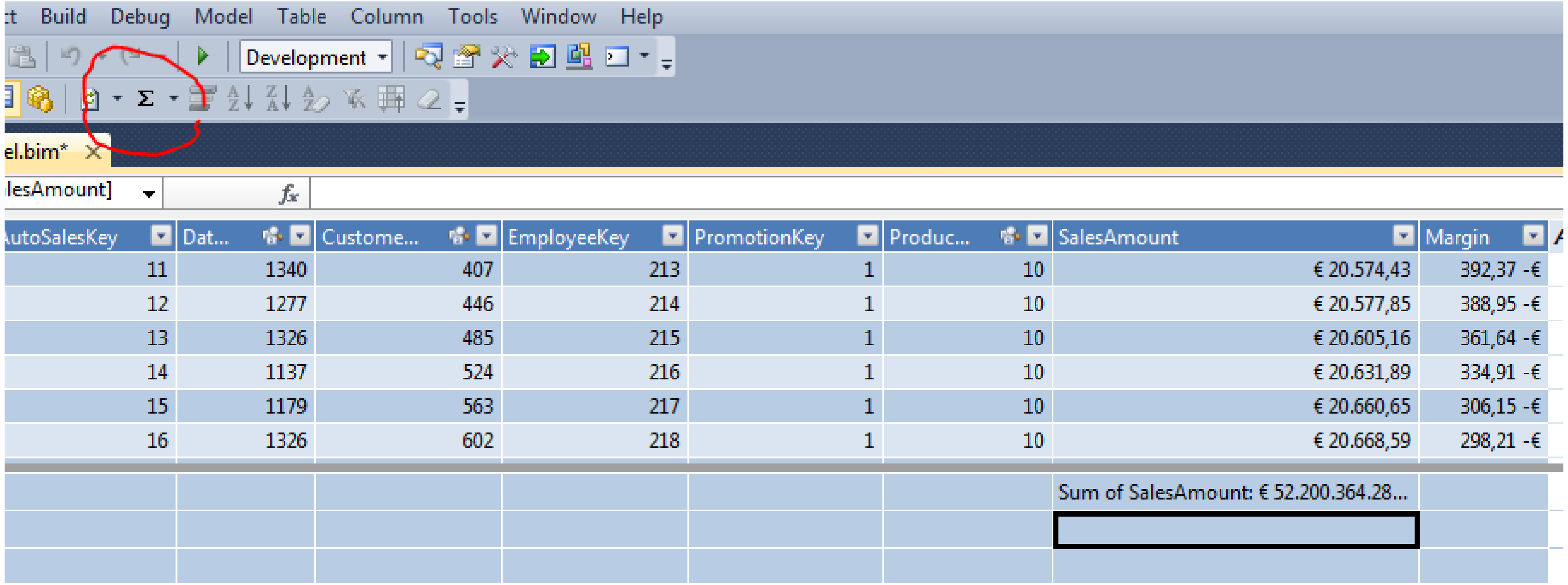
Show fields related to:

(All)

⊟ Σ Values

□ __No measures defined

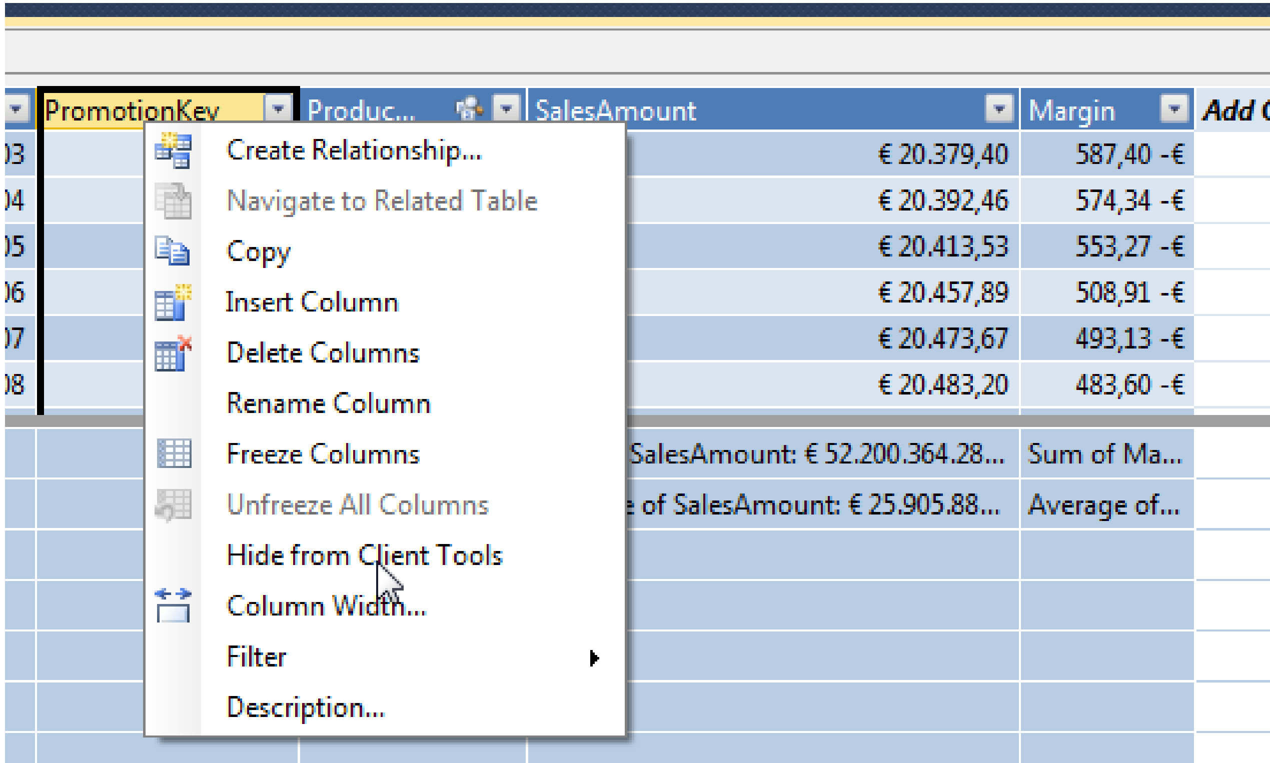
⊞ AutoSales



utoSalesKey	Dat...	Custome...	EmployeeKey	PromotionKey	Produc...	SalesAmount	Margin
11	1340	407	213	1	10	€ 20.574,43	392,37 -€
12	1277	446	214	1	10	€ 20.577,85	388,95 -€
13	1326	485	215	1	10	€ 20.605,16	361,64 -€
14	1137	524	216	1	10	€ 20.631,89	334,91 -€
15	1179	563	217	1	10	€ 20.660,65	306,15 -€
16	1326	602	218	1	10	€ 20.668,59	298,21 -€
						Sum of SalesAmount: € 52.200.364.28...	

Hide columns from client tools

- Make your model user friendly
- Hide (surrogate) key columns
- Hide columns used in formulas
    - Hide SalesAmount
    - Sum of Sales Amount visible



	PromotionKey	Produc...	SalesAmount	Margin	Add C
)3			€ 20.379,40	587,40 -€	
)4			€ 20.392,46	574,34 -€	
)5			€ 20.413,53	553,27 -€	
)6			€ 20.457,89	508,91 -€	
)7			€ 20.473,67	493,13 -€	
)8			€ 20.483,20	483,60 -€	
			SalesAmount: € 52.200.364.28...	Sum of Ma...	
			: of SalesAmount: € 25.905.88...	Average of...	

DAX Data Analysis (X)Expression

- DAX = Data Analysis Expressions
- Launched with PowerPivot
- DAX goals
  - Make data analysis really easy
  - Used relationships defined in PowerPivot/BISM model (no need for VLOOKUP Excel Database Lookup)
- Excel like Syntax
- Support for +/- 80 Excel function
- 35 Built in Time Intelligence functions
- + Additional functions

What CAN WE DO WITH DAX

- Create caclulated columns
- Create calculated measures
- Define security in the model
- Query the BI Semantic Model
  - out of scope of this presentation
  - Power View uses DAX as a query language

Reasons to Adopt Tabular

- Rapid, agile modeling relative to MOLAP
- Easier to learn language (DAX vs MDX)
- In-Memory, Columnar Performance
- Migrate Self-Service Models to Server
- Align with MSBI Roadmap (Power BI)
- Options to Mitigate Limitations
- DAX for Many to Many Relationships
- DAX for Role Playing; Semi-Additive
- Affinitize to single NUMA node
- Add-Ins: BIDS Helper, DAX Studio, DAX Editor, OLAP Pivot Extensions
- Model and DAX Design Tuning

Calculated Columns

- Just another column in a table
- Similar to named calculations in SSAS OLAP Datasource views
- Calculated after data has been loaded (cube processing)
- Can be placed in colums, rows, filters and slicers
- Evaluated in row context
- Examples
  - [Amount]   =[Qty] *[Price]
  - [Full Name] = [First Name] & “ “ & [Last Name]

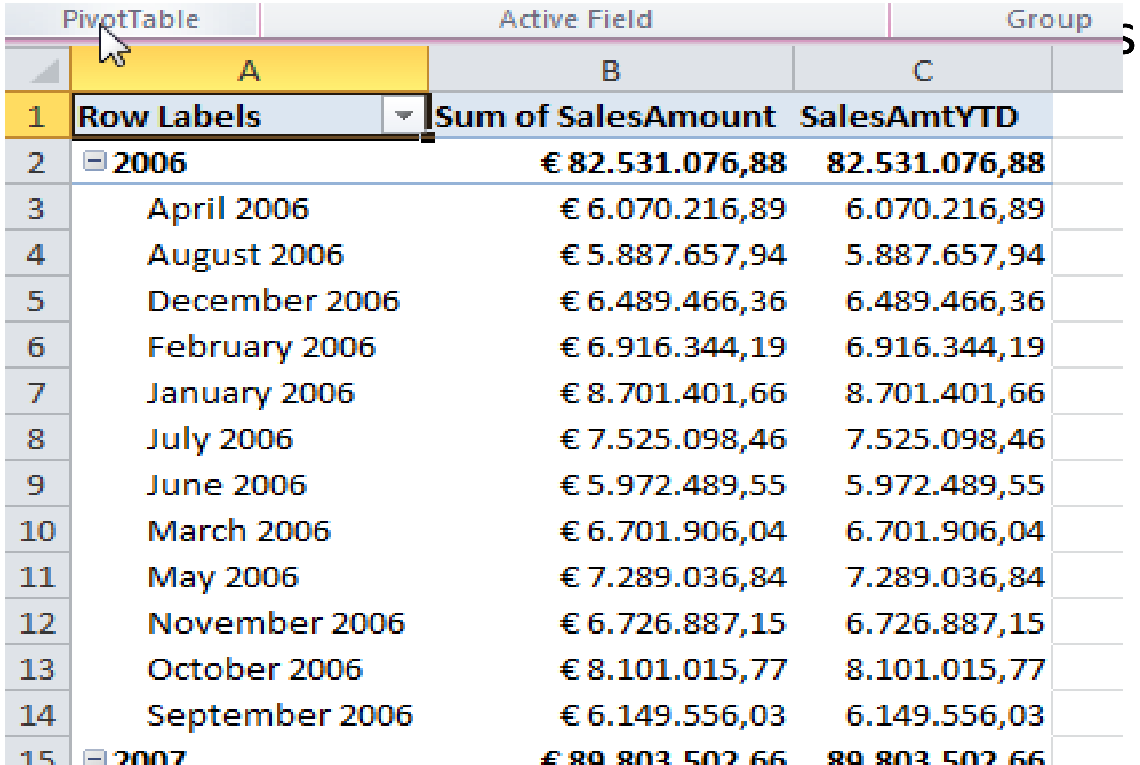
Calculated Measures

- Created in the measure grid
- Calculated “on the fly”
- Can only be placed in values area of PivotTables
- Evaluated in filter context
  - Rows/Columns/filters/Slicers
- Examples
  - [Sales]   = SUM (Sales[Amount])
  - [Average Sale] = AVERAGEX (Sales, Sales[Amount])

This is a year to date calculation but April is not rolled into August

Date Calculations

- Errors if not defined in dimension table
- TOTALYTD function
    - Calculated Year To Date Values



	A	B	C
1	Row Labels	Sum of SalesAmount	SalesAmtYTD
2	⊟2006	€ 82.531.076,88	82.531.076,88
3	April 2006	€ 6.070.216,89	6.070.216,89
4	August 2006	€ 5.887.657,94	5.887.657,94
5	December 2006	€ 6.489.466,36	6.489.466,36
6	February 2006	€ 6.916.344,19	6.916.344,19
7	January 2006	€ 8.701.401,66	8.701.401,66
8	July 2006	€ 7.525.098,46	7.525.098,46
9	June 2006	€ 5.972.489,55	5.972.489,55
10	March 2006	€ 6.701.906,04	6.701.906,04
11	May 2006	€ 7.289.036,84	7.289.036,84
12	November 2006	€ 6.726.887,15	6.726.887,15
13	October 2006	€ 8.101.015,77	8.101.015,77
14	September 2006	€ 6.149.556,03	6.149.556,03
15	⊟2007	€ 89.803.502,66	89.803.502,66

s[SalesAmount]);Dates[Date])

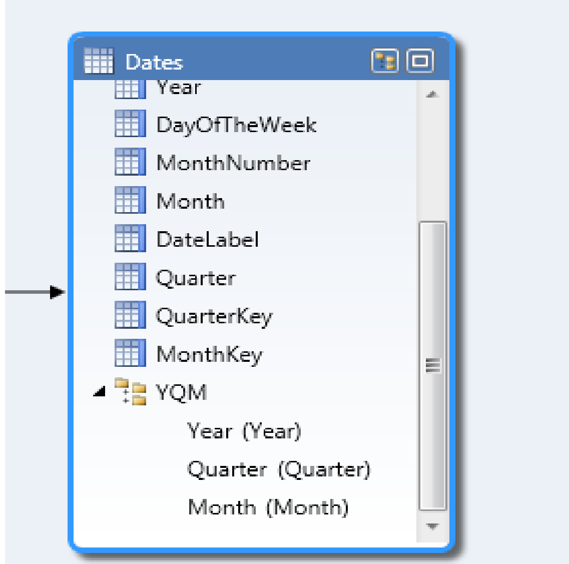
WRONG RESULTS

The monthly totals roll in correctly.

Create a DATE Dimension

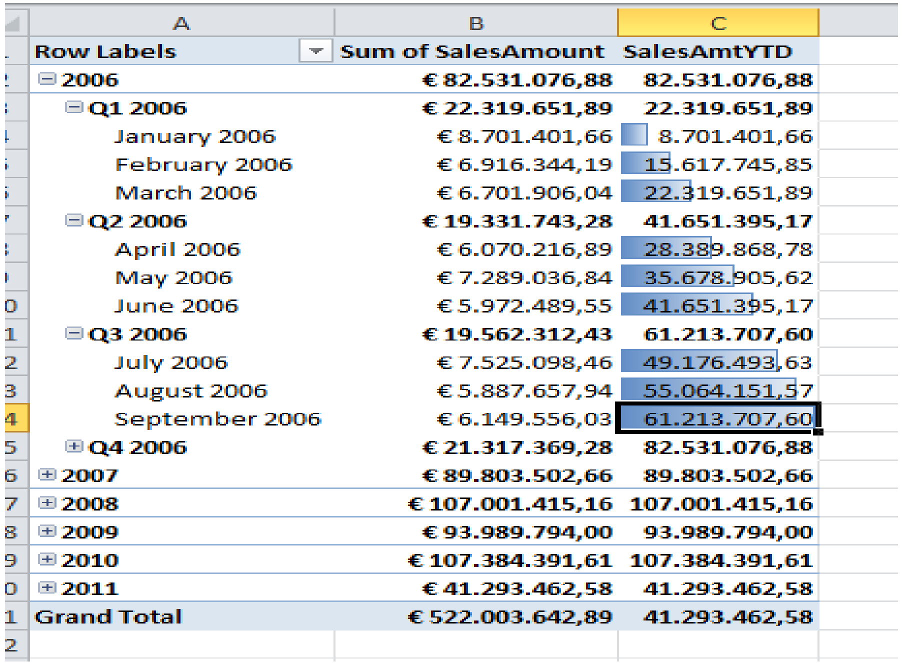
- Mark Time Dimension Table as Date
- Use data column from Time dimension in Calculations
- Sort Columns By key column
- Create Hierarchies (in Diagram view)

CORRECT RESULTS



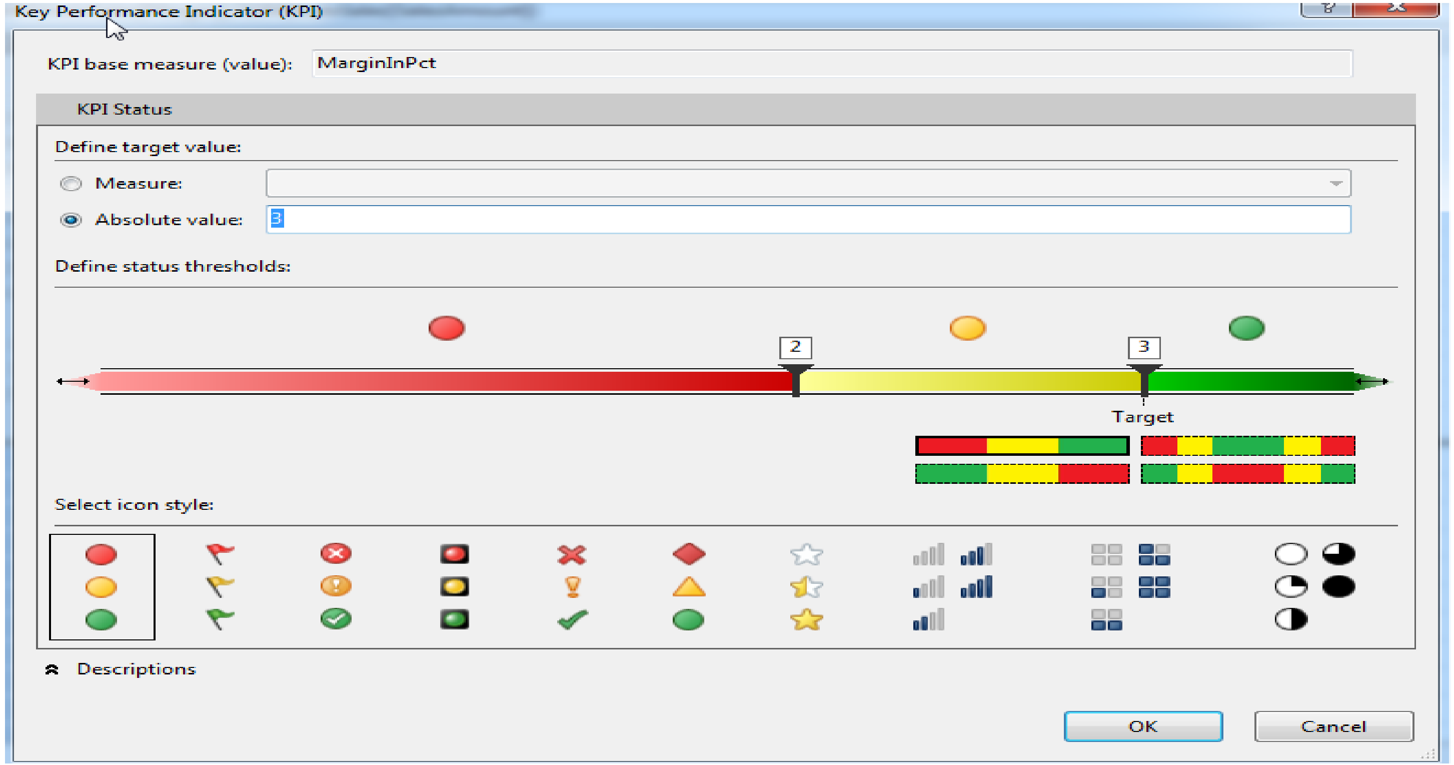
Dates

- Year
- DayOfTheWeek
- MonthNumber
- Month
- DateLabel
- Quarter
- QuarterKey
- MonthKey
- YQM
  - Year (Year)
  - Quarter (Quarter)
  - Month (Month)



Row Labels	Sum of SalesAmount	SalesAmtYTD
⊟ 2006	€ 82.531.076,88	82.531.076,88
⊟ Q1 2006	€ 22.319.651,89	22.319.651,89
January 2006	€ 8.701.401,66	8.701.401,66
February 2006	€ 6.916.344,19	15.617.745,85
March 2006	€ 6.701.906,04	22.319.651,89
⊟ Q2 2006	€ 19.331.743,28	41.651.395,17
April 2006	€ 6.070.216,89	28.389.868,78
May 2006	€ 7.289.036,84	35.678.905,62
June 2006	€ 5.972.489,55	41.651.395,17
⊟ Q3 2006	€ 19.562.312,43	61.213.707,60
July 2006	€ 7.525.098,46	49.176.493,63
August 2006	€ 5.887.657,94	55.064.151,57
September 2006	€ 6.149.556,03	61.213.707,60
⊞ Q4 2006	€ 21.317.369,28	82.531.076,88
⊞ 2007	€ 89.803.502,66	89.803.502,66
⊞ 2008	€ 107.001.415,16	107.001.415,16
⊞ 2009	€ 93.989.794,00	93.989.794,00
⊞ 2010	€ 107.384.391,61	107.384.391,61
⊞ 2011	€ 41.293.462,58	41.293.462,58
Grand Total	€ 522.003.642,89	41.293.462,58

KPIs



Key Performance Indicator (KPI)

KPI base measure (value): MarginInPct

KPI Status

Define target value:

○ Measure:

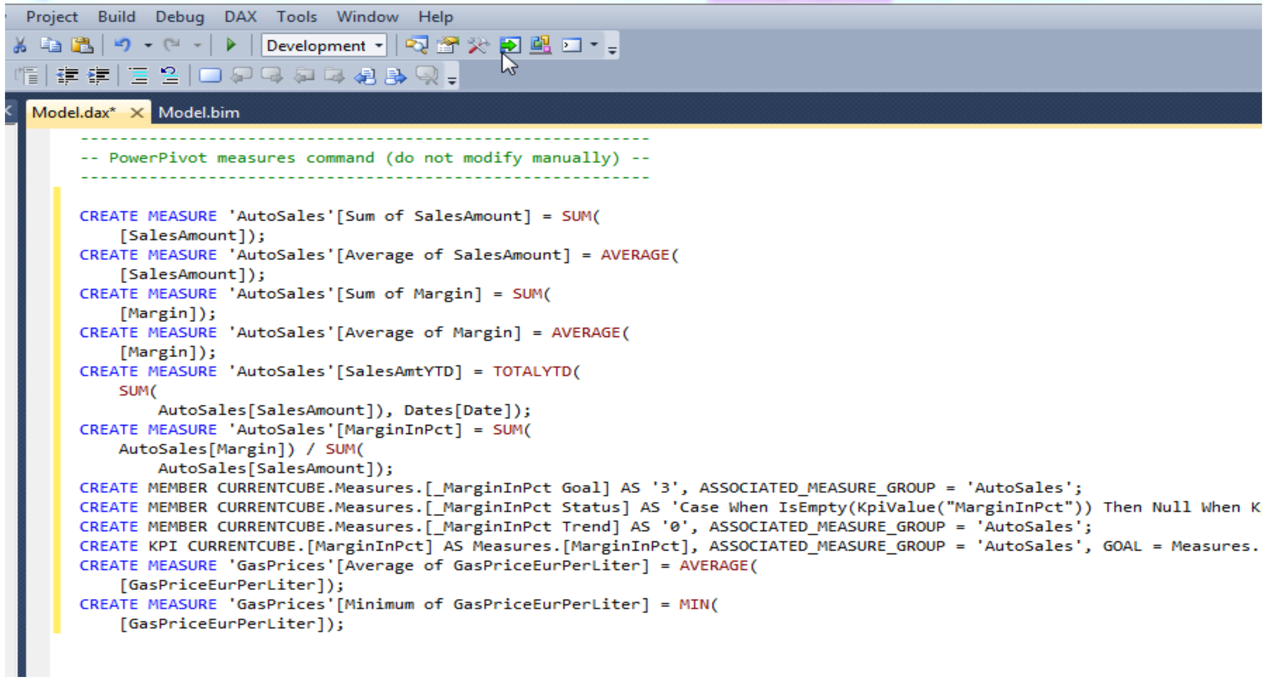
● Absolute value: 3

Define status thresholds:

Select icon style:

DAX Editor

- Code highlighting
- No intellisense
- Free download at http://daxeditor.codeplex.com/

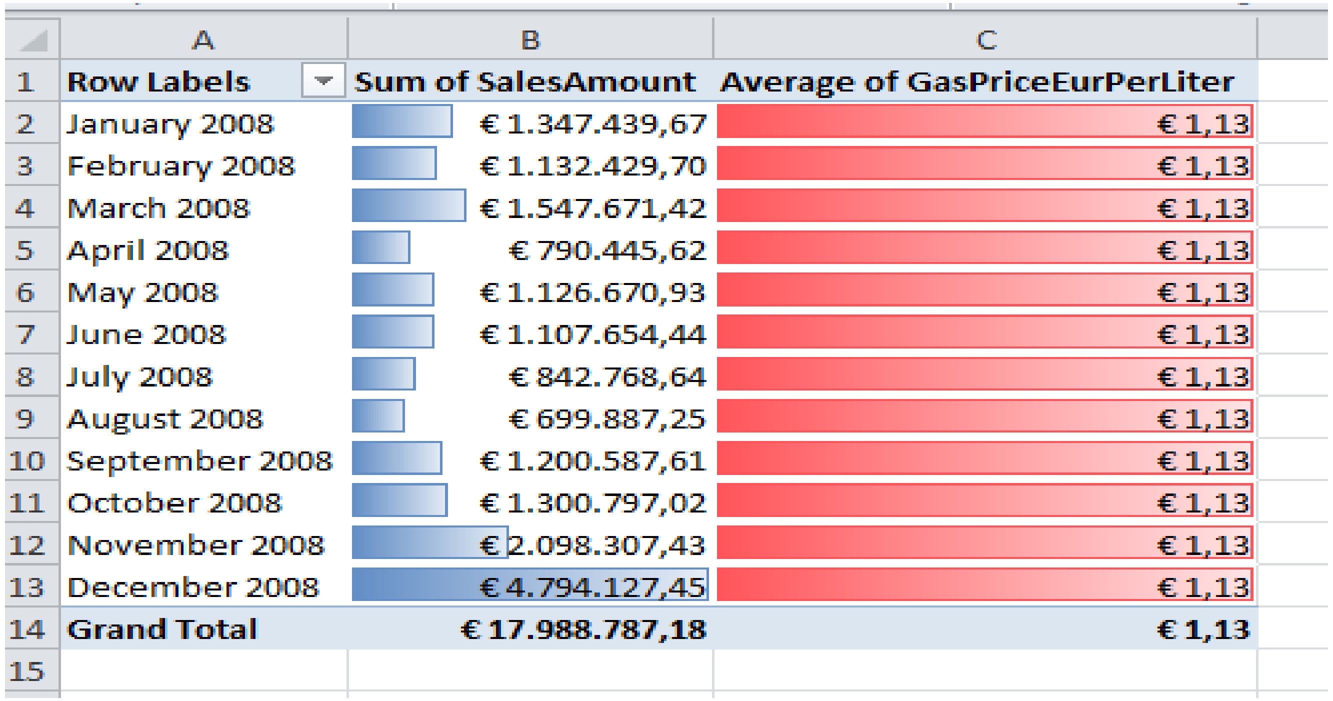


```
------------------------------------------------------------
-- PowerPivot measures command (do not modify manually) --
------------------------------------------------------------

CREATE MEASURE 'AutoSales'[Sum of SalesAmount] = SUM(
    [SalesAmount]);
CREATE MEASURE 'AutoSales'[Average of SalesAmount] = AVERAGE(
    [SalesAmount]);
CREATE MEASURE 'AutoSales'[Sum of Margin] = SUM(
    [Margin]);
CREATE MEASURE 'AutoSales'[Average of Margin] = AVERAGE(
    [Margin]);
CREATE MEASURE 'AutoSales'[SalesAmtYTD] = TOTALYTD(
    SUM(
        AutoSales[SalesAmount]), Dates[Date]);
CREATE MEASURE 'AutoSales'[MarginInPct] = SUM(
    AutoSales[Margin]) / SUM(
        AutoSales[SalesAmount]);
CREATE MEMBER CURRENTCUBE.Measures.[_MarginInPct Goal] AS '3', ASSOCIATED_MEASURE_GROUP = 'AutoSales';
CREATE MEMBER CURRENTCUBE.Measures.[_MarginInPct Status] AS 'Case When IsEmpty(KpiValue("MarginInPct")) Then Null When K
CREATE MEMBER CURRENTCUBE.Measures.[_MarginInPct Trend] AS '0', ASSOCIATED_MEASURE_GROUP = 'AutoSales';
CREATE KPI CURRENTCUBE.[MarginInPct] AS Measures.[MarginInPct], ASSOCIATED_MEASURE_GROUP = 'AutoSales', GOAL = Measures.
CREATE MEASURE 'GasPrices'[Average of GasPriceEurPerLiter] = AVERAGE(
    [GasPriceEurPerLiter]);
CREATE MEASURE 'GasPrices'[Minimum of GasPriceEurPerLiter] = MIN(
    [GasPriceEurPerLiter]);
```

No detection of missing relationships

- PowerPivot detects missing relationships
- No warning in SSAS Tabular
- Create relationships manual



	A	B	C
1	**Row Labels**	**Sum of SalesAmount**	**Average of GasPriceEurPerLiter**
2	January 2008	€ 1.347.439,67	€ 1,13
3	February 2008	€ 1.132.429,70	€ 1,13
4	March 2008	€ 1.547.671,42	€ 1,13
5	April 2008	€ 790.445,62	€ 1,13
6	May 2008	€ 1.126.670,93	€ 1,13
7	June 2008	€ 1.107.654,44	€ 1,13
8	July 2008	€ 842.768,64	€ 1,13
9	August 2008	€ 699.887,25	€ 1,13
10	September 2008	€ 1.200.587,61	€ 1,13
11	October 2008	€ 1.300.797,02	€ 1,13
12	November 2008	€ 2.098.307,43	€ 1,13
13	December 2008	€ 4.794.127,45	€ 1,13
14	**Grand Total**	**€ 17.988.787,18**	**€ 1,13**
15			

created tabular project in visual studio

Conclusions

- Tabular model vs traditional SSAS OLAP cubes
  - Easier learing curve
  - No MDX, but DAX
  - Not all SSAS OLAP features are available in BISM tabular
    - SCOPE statements
    - Write Back
    - Native support for many to many dimensions
    - …
- Personal BI -> Team BI – Corporate BI
- PowerPivot for Excel → PowerPivot for Sharepoint → BISM