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

Optimizing & Analyzing DAX Query Plans

sqlbi

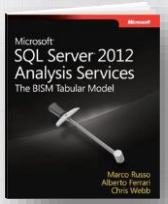
Who's Speaking?

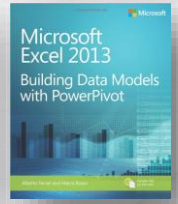
sqlbi

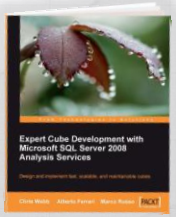
a brand of

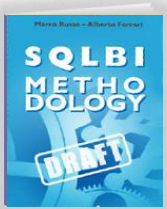
Loader

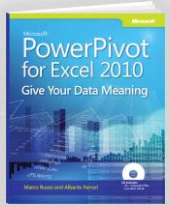
- BI Expert and Consultant
- Founder of www.sqlbi.com
  - Problem Solving
  - Complex Project Assistance
  - Data Warehouse Assessments and Development
  - Courses, Trainings and Workshops
- Book Writer
- Microsoft Business Intelligence Partner











Microsoft Partner
Gold Data Platform
Gold Business Intelligence

MVP
Microsoft
Most Valuable
Professional

SSAS
MAESTRO
by Microsoft

sqlbi

SSAS Multidimensional and MDX

- OLAP data structures
  - Proprietary and complex data structure
  - Complex design of solutions
  - Requires highly-optimized databases
  - Runs on expensive hardware
  - Design of aggregation is an obscure art
- MDX Language
  - Hard to learn, fast only with his closest friends
  - Requires complex modelling skills

sqlbi

SSAS Tabular and DAX

- xVelocity in-memory Analytics Engine
  - In-memory database
  - Columnar storage
  - Automatically compresses data
  - Already optimized out-of-the-box
  - Runs on commodity hardware
- DAX Language
  - Easy to learn, amazingly fast
  - Less modelling skills to achieve great results

sqlbi

Welcome to the real world

- Billions of rows are a challenge
  - Find optimal compression
  - Get prepared for fancy ideas
- Optimizing DAX
  - Understand how the query engine works
  - Force different query plans
  - Requires a lot of effort

sqlbi

Agenda

- Few words on the VertiPaq engine
- Optimizing the model
- Understanding the query engine
- Optimizing some queries

sqlbi

A few words about the internals of the VertiPaq engine

VertiPaq

sqlbi

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

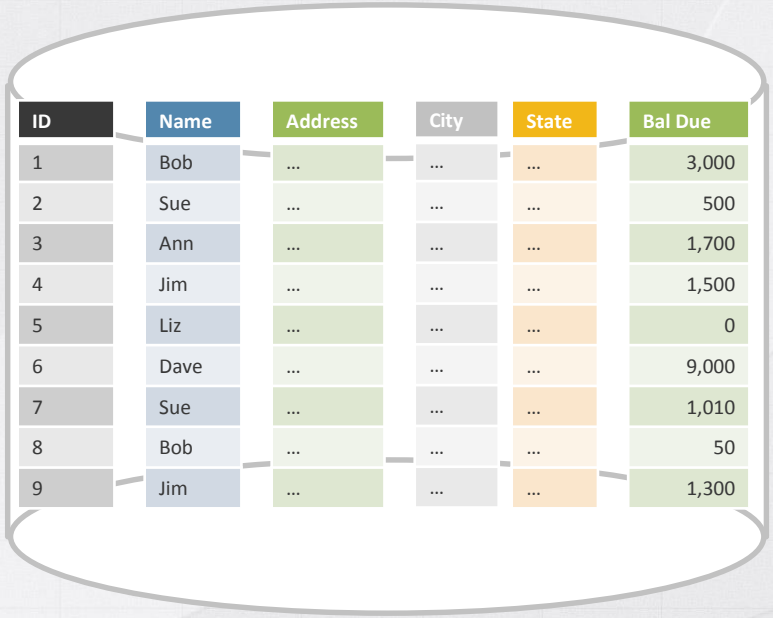
Nothing special here.
This is the standard way database systems have been laying out tables on disk since the mid 1970s.
Technically, it is called a “row store”

sqlbi

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



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

sqlbi

Run Length Encoding

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

sqlbi

Dictionary Encoding

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

DISTINCT

Q.ID
0
0
0
0
1
1
…
1
2
2
2
2
4
4
4
4
…

R.L.E.

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

Only 4 values.
2 bits are enough to represent it

xVelocity Store

Quarter	Start	Count
0	1	4
1	5	10
2	11	4
3	15	15

sqlbi

Cardinality

- Reducing the cardinality of columns
  - Reduces the memory usage
  - Increases scan speed
  - Increases filter operations
- Focus on high-cardinality columns
  - Primary keys in fact tables
  - Order number / reference numbers

sqlbi

Where's my memory?

BISMServerMemoryReport_2013 - Excel

	A	B	C
1		ModelRelated	True
2			
3		**Row Labels**	**Usage MB**
4		**AdventureWorks Tabular Model SQL 2012**	**36.27**
5		**Audience**	**13,763.44**
6		**Contoso Retail Analysis**	**583.93**
7		Channel	0.12
8		Currency	0.12
9		Customer	3.36
10		Date	0.83
11		ExchangeRate	0.14
12		Geography	0.24
13		Inventory	104.55
14		ITMachine	0.32
15		ITSLA	0.88
16		Machine	0.72

Introduction | Instructions | Analytics

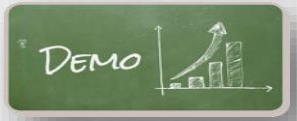
sqlbi

How does modeling affect performance?

Modeling

sqlbi

Modeling



How much does modeling affect performance?

```
EVALUATE

    ROW (
        "Result",
        SUM ( Sales[SalesAmount] )
    )
```

Modeling columns: the results

Column	Memory (MB)	Distinct Values	SUM	DISTINCTCOUNT
Age	203.55	96	130	103
Time	6,192.81	1,438	**1,391**	844
Date	0.37	1,588	13	17
Weight	2,690.35	1,467,394	700	**1,986**

Base query time depends on

- Number of distinct values of the column (bit encoding)
- Distribution of values (RLE encoding)

Do you prefer to optimize date or time in this model?

sqlbi

Reducing Distinct Count

- Reduce precision of numbers
- Split date and time in two tables
  - Never use DateTime
- Split high cardinality columns in two parts
  - Combined later during calculations
  - <Num> = <High> * 1000 + <Low>

sqlbi

What happens when you run a query?

Query Execution Flow

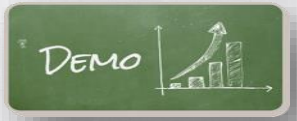
sqlbi

What happens in the engine

- FE executes the DAX query plan
- It asks SE to retrieve values from the storage
  - Many SE queries, one for each dataset
  - Each SE query is shown twice
    - What FE asked
    - What SE actually executed
- Each result set is stored in a data cache
- FE uses result sets to build the final answer to the query

sqlbi

SUMMARIZE



Let us perform a deep analysis of a simple SUMMARIZE statement, to discover how it is evaluated

```
EVALUATE
    SUMMARIZE (
        Audience,
        Individuals[AgeRange],
        "Weight",
        SUM ( Audience[Weight] )
    )
```

SUMMARIZE Analysis

```
EVALUATE
    SUMMARIZE (
        Audience,
        Individuals[AgeRange],
        "Weight",
        SUM ( Audience[Weight] )
    )
```

Retrieves SUM(Weight) grouped by AgeRange

EventClass	EventSubclass	Duration	CPUTime
DAX Query Plan	1 - DAX VertiPaq Logical Plan		
VertiPaq SE Query End	10 - VertiPaq Scan internal	1981	22938
VertiPaq SE Query End	0 - VertiPaq Scan	1981	22938
VertiPaq SE Query End	10 - VertiPaq Scan internal	628	7313
VertiPaq SE Query End	0 - VertiPaq Scan	628	7313
DAX Query Plan	2 - DAX VertiPaq Physical Plan		
Query End	3 - DAXQuery	2629	16

Retrieves AgeRange from the fact table

Two scans of the fact table are needed: one for the group-by columns, one for the values

sqlbi

Time to see the results of optimizing a query

Optimizing

sqlbi

Events in Progress

- How many orders are active each day?
- Very common pattern
  - Whenever you have a duration
  - i.e. start and end date of an event

sqlbi

Events in Progress: a first trial

Computing Events in Progress

```
EVALUATE
    ADDCOLUMNS (
        VALUES ( 'Date'[Date] ),
        "OpenOrders",
        COUNTROWS (
            FILTER (
                Sales,
                Sales[Order Date] < 'Date'[Date] &&
                Sales[Ship Date] > 'Date'[Date]
            )
        )
    )
```

sqlbi

Optimizing Events in Progress

- SQL Server Profiler is your best friend
- Build a template, saves a lot of time
- Understand Formula Engine and Storage Engine
- Remember that VertiPaq is a columnar database

- We will see 4 variations of the same query

sqlbi

Events in Progress: final results

Version	Formula Engine	Execution Time	Notes
Naïve	22,078	22,175	First query you will write
Column-wise	906	925	Query written by a seasoned DAX coder
DatesBetween	156	167	Original idea (from Chris Webb) you will find in on a blog post
Generate	78	80	Required 4/5 days of hard optimization

Optimizing DAX leads to great results

- But it requires a lot of time
- Stop when it is “good enough”

Extreme optimizations need to work on the whole model

sqlbi

Conclusions

- Learn the internals of the engine
  - Knowing it, is the key to any optimization
- Think in terms of
  - Formula engine
  - Storage engine
- Don’t trust any rule of thumb, including mine!

sqlbi

Thank you!

Check our new articles on

www.sqlbi.com