Francesco De Chirico

#sqlsat

PASS SQLSATURDAY

Il mio grosso, grasso Power BI model

inn0va community · altitudo · beanTech IT moves your business · fluentis · PALAZZETTI IL CALORE CHE PIACE ALLA NATURA · WÜRTHPHOENIX · PASS · Microsoft · CONSORZIO UNIVERSITARIO DI PORDENONE

Francesco De Chirico

in /fdechirico/ @fdechirico

Consultant, trainer and speaker

Specializing in Power BI, SSAS, MDX, DAX and M.
Using SSAS and MS BI Platform since 2001

Main certifications

MCT since 2008, SSAS Maestro in 2012 and
Microsoft Professional Program Data Science in 2017

Projects

Ideator and co-developer of ASQA tool

ASQA https://ssasqueryanalyzer.github.io/

inn0va
community

CONSORZIO
UNIVERSITARIO
DI PORDENONE

Microsoft

PASS

altitudo

beanTech
IT moves your business

fluentis

PALAZZETTI
IL CALORE CHE PIACE ALLA NATURA

WÜRTHPHOENIX

Agenda

- Why To optimize – The problem
  - Demo
- Vertipaq engine overview
- How To optimize – Best practices
- How to verify optimization – DMV
  - Demo
- Tools available
- Connect to .pbix data model
  - SSAS sandbox
  - Problem: how to get port number?
  - Demo
- Build a .pbix template
  - Power Vertipaq Demo

What we’ll talk about

- What increase .pbix size
- What we can do to reduce .pbix size
- How to identify causes

What we’ll NOT talk about

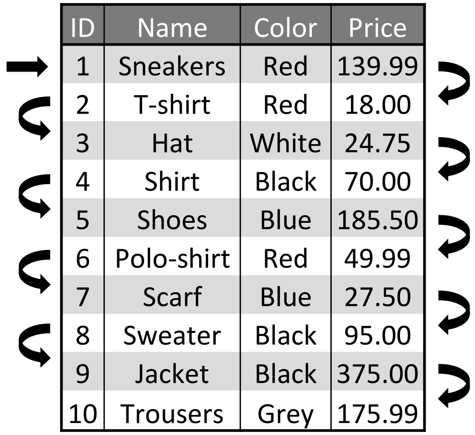
- Performance
- DAX
- Power Query and M
- …

DEMO

VertiPaq engine
in ~~3~~ … ~~4~~ … ok, 5 slides!

Vertipaq is an in-memory columnar database

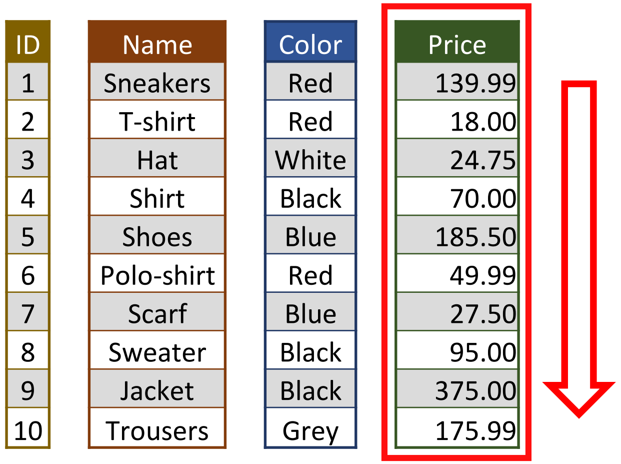
Row store



ID	Name	Color	Price
1	Sneakers	Red	139.99
2	T-shirt	Red	18.00
3	Hat	White	24.75
4	Shirt	Black	70.00
5	Shoes	Blue	185.50
6	Polo-shirt	Red	49.99
7	Scarf	Blue	27.50
8	Sweater	Black	95.00
9	Jacket	Black	375.00
10	Trousers	Grey	175.99

- Data is organized in rows
- Sum of Price:
  - Start reading the first row
  - Discard 75% of data (ID, Name.Color)
  - Retain the searched value (Price)
  - Move to the next row and repeat until the end of the table

Column store



ID	Name	Color	Price
1	Sneakers	Red	139.99
2	T-shirt	Red	18.00
3	Hat	White	24.75
4	Shirt	Black	70.00
5	Shoes	Blue	185.50
6	Polo-shirt	Red	49.99
7	Scarf	Blue	27.50
8	Sweater	Black	95.00
9	Jacket	Black	375.00
10	Trousers	Grey	175.99

- Data is organized in columns
  - Optimize vertical scanning
- Sum of Price:
  - scan the Price column ONLY

- Very fast on single-column access
- More columns require to reorganize the information
- The more columns you need the harder to obtain the result!

1nn0va community · altitudo · beanTech IT moves your business · fluentis · PALAZZETTI IL CALORE CHE PIACE ALLA NATURA · WÜRTHPHOENIX · PASS · Microsoft · CONSORZIO UNIVERSITARIO DI PORDENONE

Vertipaq compression in pills

Value encoding

Price
489
497
508
493
500
480
487
505
510
499

Max: 510
9 bits need

Price - 480

Price
9
17
28
13
20
0
7
25
30
19

Max: 30
5 bits need

Only for integer columns

Dictionary encoding

Color
Red
Red
White
Black
Blue
Red
Blue
Black
Black
Grey

Indexes

Color ID
0
0
1
2
3
0
3
2
2
4

Dictionary

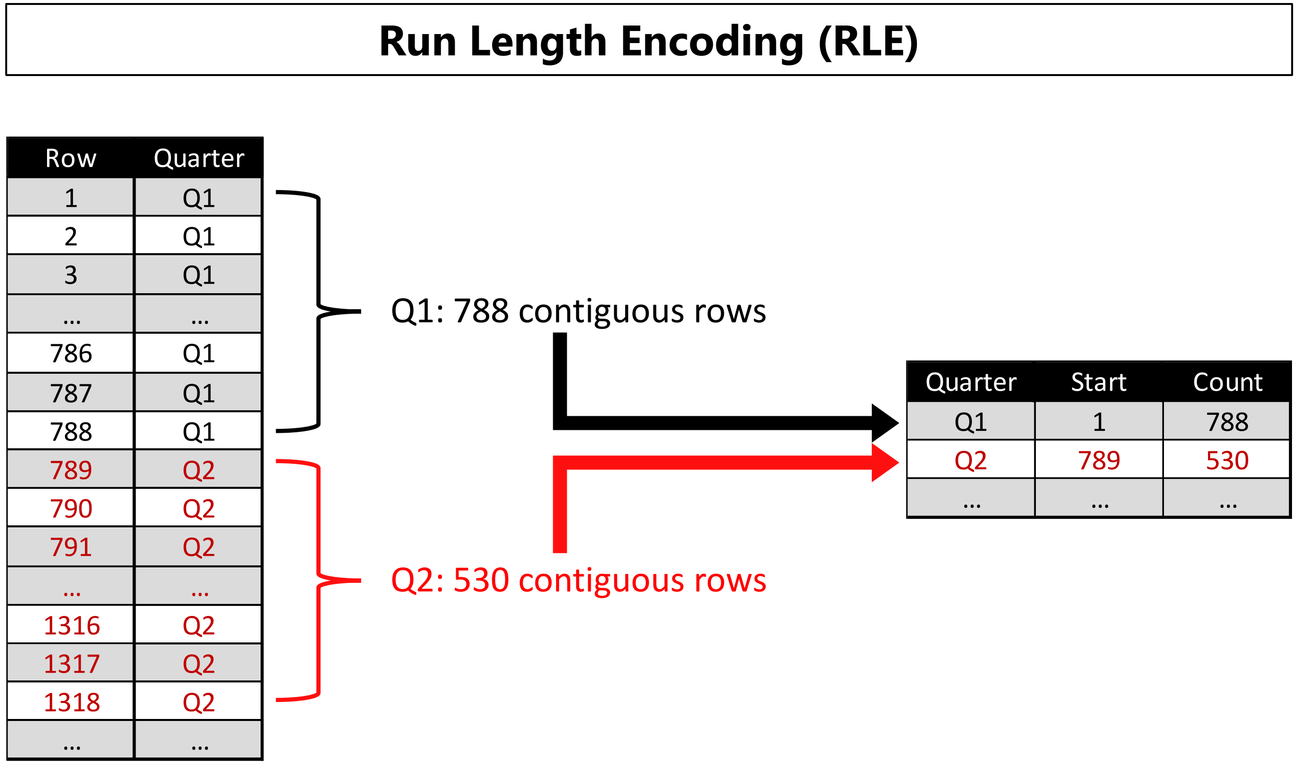
Color ID	Color
0	Red
1	White
2	Black
3	Blue
4	Grey

5 values = 3 bits need

Bits to store a single value = minimum bits of index entry
All columns contain ONLY integer values
It does not matter what the original data type is!

inn0va community altitudo beanTech IT moves your business fluentis PALAZZETTI IL CALORE CHE PIACE ALLA NATURA WÜRTHPHOENIX PASS Microsoft CONSORZIO UNIVERSITARIO DI PORDENONE

Vertipaq compression in pills



Run Length Encoding (RLE)

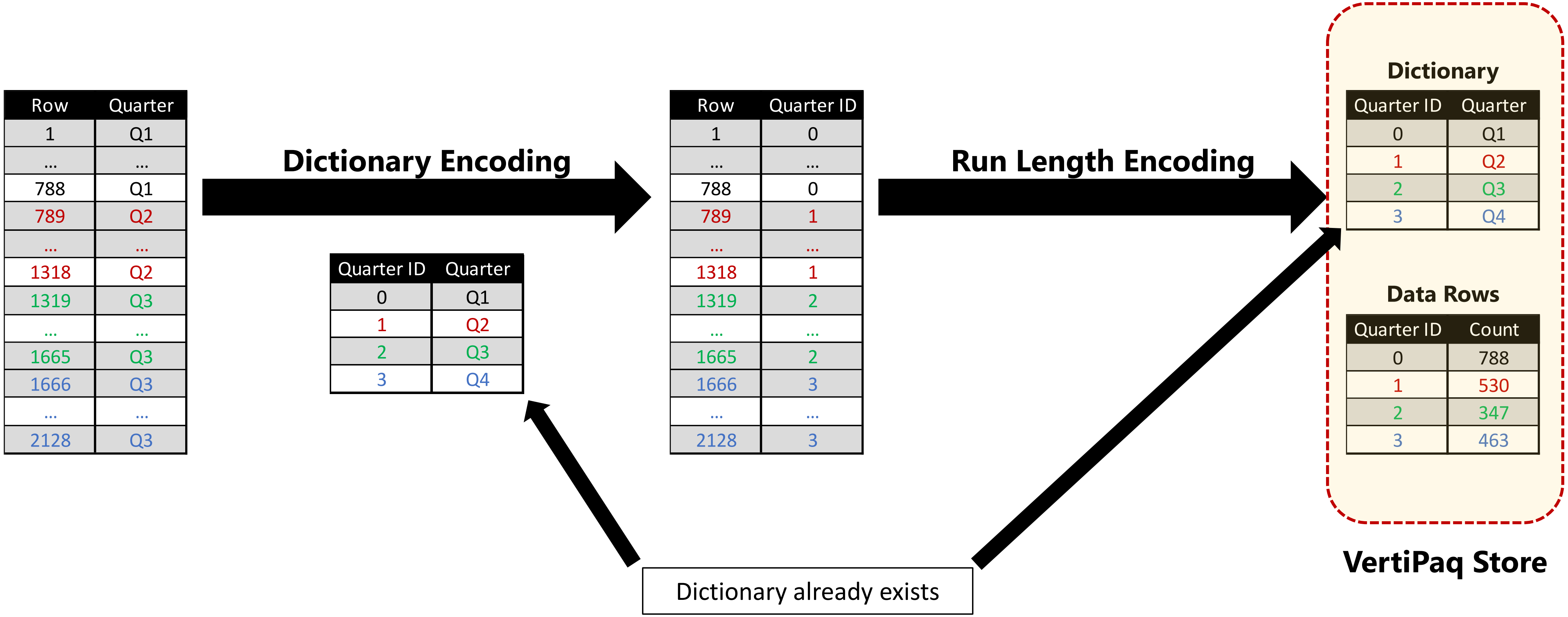
Row	Quarter
1	Q1
2	Q1
3	Q1
…	…
786	Q1
787	Q1
788	Q1
789	Q2
790	Q2
791	Q2
…	…
1316	Q2
1317	Q2
1318	Q2
…	…

Quarter	Start	Count
Q1	1	788
Q2	789	530
…	…	…

- RLE’s efficiency strongly depends on the repetition pattern of the column
- Two rows with the same cardinality can have very different compression ratio
- Sorting of data is extremely important to improve the compression ratio
- In columns with very high cardinality (i.e. primary keys) RLE is larger then the column itself.

Vertipaq compression in pills

Run Length Encoding (RLE) applied to the dictionary-encoded version of a column



Row	Quarter
1	Q1
…	…
788	Q1
789	Q2
…	…
1318	Q2
1319	Q3
…	…
1665	Q3
1666	Q3
…	…
2128	Q3

Quarter ID	Quarter
0	Q1
1	Q2
2	Q3
3	Q4

Row	Quarter ID
1	0
…	…
788	0
789	1
…	…
1318	1
1319	2
…	…
1665	2
1666	3
…	…
2128	3

Dictionary

Quarter ID	Quarter
0	Q1
1	Q2
2	Q3
3	Q4

Data Rows

Quarter ID	Count
0	788
1	530
2	347
3	463

Data models in pills of pills!

Denormalized

ID	Product Name	Product Code	Price
1	Sneakers	00-3475	139.99
2	T-shirt	17-8553	18.00
…	…	…	…

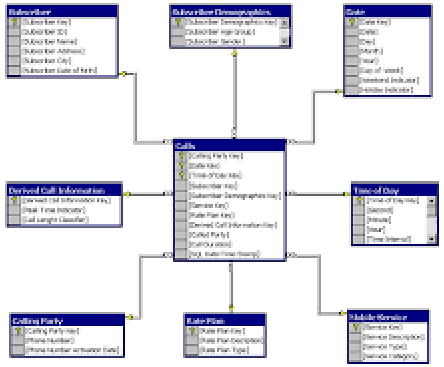
VS

Normalized

ID	Product Code	Price
1	00-3475	139.99
2	17-8553	18.00
…	…	…

Product Code	Product Name
00-3475	Sneakers
17-8553	T-shirt
…	…

Star schema



- Every relationship has an additional memory cost
  - Normalized Memory cost:

    ColumnCost(Price[Product Code]) + ColumnCost(Product[Product Code]) + ColumnCost(Product[Product Name]) + RelationshipCost(Price[Product Code])
  - Denormalized Memory cost:

    ColumnCost(Price[Product Code]) + ColumnCost(Price[Product Name])
- Theorically the "optimal" model:
  - One single table
- In reality it is less usable and would force to have a single granularity for all the measures

Inn0va community · altitudo · beanTech IT moves your business · fluentis · PALAZZETTI IL CALORE CHE PIACE ALLA NATURA · WÜRTHPHOENIX · PASS · Microsoft · CONSORZIO UNIVERSITARIO DI PORDENONE

VertiPaq compression ratio - summary

Factor to consider (in order of importance):

- The cardinality of the column
  - Determines the number of bits used to store a value
- The distribution of data in the column
  - Many repeated values = high compression ratio
  - Frequently changing values = low compression ratio
- The number of rows in the table
- The datatype of the column
  - It affects only the dictionary size

Well, now we know how VertiPaq engine works (more or less ☺), so …

Best practices: at least 6 tips

- Import only useful columns
- Remove unused fields
  - Especially those with high cardinality
  - Best candidate: ID’s
- Set correct data type
  - Typical issue: strings containing only numbers
- No datetime type!
  - Use ONLY date ←If you can ☺
  - If you need also time → next tips

Best practices

- Split fields in two (or more) fields
  - Do you remember the previous tip? ☺
    - Split a datetime field in two: Date AND Time

- Shrink fields to reduce cardinality
  - Changing the Precision of Numeric Columns

- Disable auto date/time functionality
  - Particularly when you have many dates
  - Cons: you need to manually build hierarchies!

Best practices – how to split

- Datetime
  - date → (cardinality = 365 * number of years)
  - time → (cardinality = 24h * 60min * 60sec = 86.400)
    - Ex: CAST(MyDateTime AS DATE) AS MyDate → 2019/02/23 ~~12:37:45~~
    - Ex: CAST(MyDateTime AS TIME(0)) AS MyTime → ~~2019/02/23~~ 12:37:45
- Big integer
  - High value (INT)
  - Low value (INT)
    - ex: CAST(MyBigValue / 1000000 AS INT) AS MyValue_High 1368405~~885980~~
    - ex: CAST(MyBigValue % 1000000 AS INT) AS MyValue_Low ~~1368405~~885980
    - To obtain the original value:
      - MyValue_High * 1000000 + MyValue_Low

Best practices – how to split

- Long strings
  - LEFT substrings (leftmost “n” chars)
  - RIGHT substring (rightmost “n” chars)
    - ex: LEFT(MyLongString, “n”) AS MyString_Left
    - ex: SUBSTRING(MyLongString, “n” + 1, LEN(MyLongString) – “n”) AS MyString_Right
    - To obtain the original value:
      - CONCAT(MyString_Left, MyString_Right)
- Decimal numbers
  - Integer part
  - Decimal part
    - ex: FLOOR(MyDecimalValue) AS MyValue_IntegerPart
    - ex: MyDecimalValue - FLOOR(MyDecimalValue) AS MyValue_DecimalPart

Best practices – shrink fields

- long strings used as ID’s
  - replace with integer ←If you can ☺
  - If used for a distinct count measure, replace with RANK()
- Decimal numbers
  - Reduce decimal precision ←If you can ☺
  - i.e. a temperature value with more than 2 decimal
- Calculations
  - Do not import columns containing result of calculation
  - Calculate the value in the model:
    - TotalSale = SUMX(Sales, Sales[Price] * Sales[Quantity])

Now the problem is …
“how to identify critical columns?”

Retrieve metadata info of Tabular model

- The “rough” way:
  - Connect to SSAS Tabular instance with SSMS
  - Execute a set of DMV using MDX/DMX queries
  - Store the results in a SQL database
  - Write some views to analyze data
- The “easy” way:
  - Connect to SSAS Tabular instance with Vertipaq Analyzer
  - Refresh the Power Pivot model

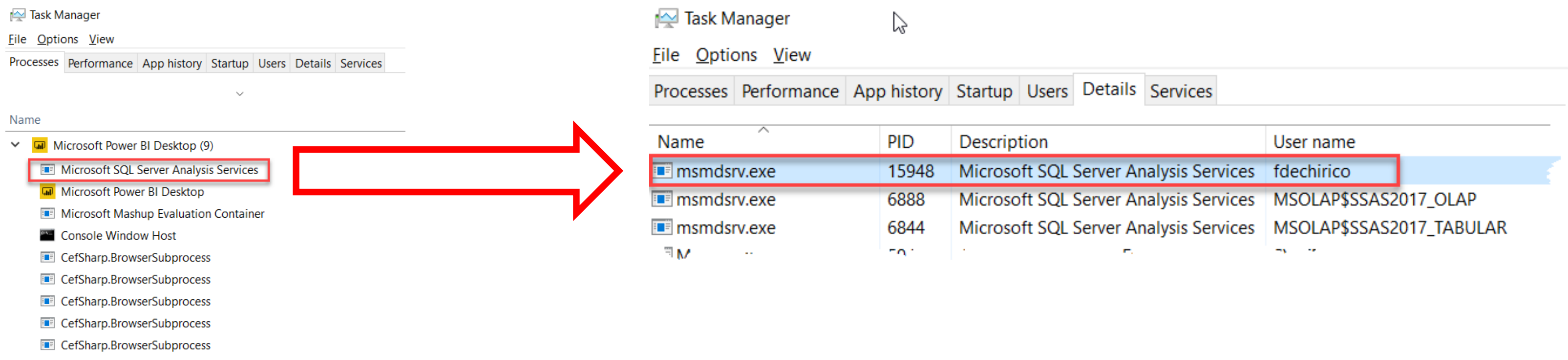
- Cons:
  - DMV’s results are not easy to analyze
  - Cannot join DMV’s
  - Many DMV’s to execute to retrieve useful data
- Pros:
  - Very simple to use
  - All calculations are provided in the Power Pivot model

DMV and VertiPaq Analyzer demo

Ok, but what about Power BI model?
Is it possible to connect to a .pbix data model?

Connect to a .pbix data model

- YES, it is possible!!! ☺
- Power BI Desktop runs a <u>local instance of SSAS Tabular</u> engine in the background and it assigns a <u>random port number</u> to that local instance:



Task Manager

File Options View

Processes | Performance | App history | Startup | Users | Details | Services

Name

- Microsoft Power BI Desktop (9)
  - Microsoft SQL Server Analysis Services
  - Microsoft Power BI Desktop
  - Microsoft Mashup Evaluation Container
  - Console Window Host
  - CefSharp.BrowserSubprocess
  - CefSharp.BrowserSubprocess
  - CefSharp.BrowserSubprocess
  - CefSharp.BrowserSubprocess
  - CefSharp.BrowserSubprocess

Task Manager

File Options View

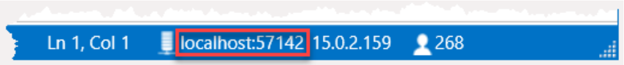
Processes | Performance | App history | Startup | Users | Details | Services

Name	PID	Description	User name
msmdsrv.exe	15948	Microsoft SQL Server Analysis Services	fdechirico
msmdsrv.exe	6888	Microsoft SQL Server Analysis Services	MSOLAP$SSAS2017_OLAP
msmdsrv.exe	6844	Microsoft SQL Server Analysis Services	MSOLAP$SSAS2017_TABULAR

- It is possible to connect to that instance and to the Power BI model **BUT** <u>you need to know that port number</u> ☹

How to get the port number?

- Command Line (CMD)
- DAX Studio
  - Connect to your .pbix
  - Local SSAS instance address on the right bottom of the DAX studio window



Ln 1, Col 1 localhost:57142 15.0.2.159 268

- Tabular Editor
- Power BI Desktop Temporary Location
  - Download Edition: *%LocalAppData%\Microsoft\Power BI Desktop\AnalysisServicesWorkspaces*
  - Store Edition: *%username%\Microsoft\Power BI Desktop Store App\AnalysisServicesWorkspaces*
  - Open “**msmdsrv.port.txt**”
- PowerShell (using the PowerShell module “PowerBIPS.Tools”)

Retrieve metadata info of Power BI desktop model

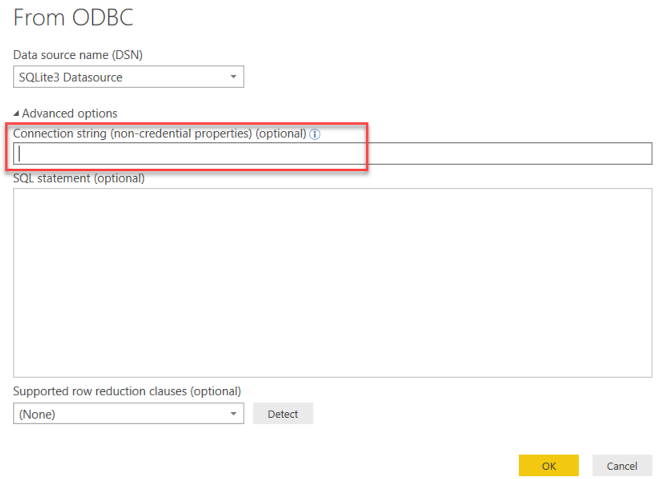
- The “rough” way:
  - Launch the Power BI desktop report you want to analyze
  - Found the port used by the local SSAS Tabular instance
  - Same steps used for Tabular model

- The “easy” way:
  - Launch the Power BI desktop report you want to analyze
  - Found the port used by the local SSAS Tabular instance
  - Same steps used for Tabular model

Retrieve metadata info of Power BI desktop model

- The “alternative” way:
  - Connect to the SQLite db used by the Power BI desktop model
    - Install SQLite ODBC driver (http://www.ch-werner.de/sqliteodbc) on your local machine
    - Open the .pbix model you want to analyze with Power Bi desktop
    - Go to “*%LocalAppData%\Microsoft\Power BI Desktop\AnalysisServicesWorkspaces*”
    - Identify the Workspace of your model → **can be tricky**
    - Copy the complete address of the **metadata.sqlitedb** (“*..\Data\<GUID>.<VERSION>.db*”)
    - Get Data → ODBC → SQLite 3 Datasource
    - In the “Advanced Options” connection string insert:

      “database=<your metadata.sqlitedb file address> ”



From ODBC

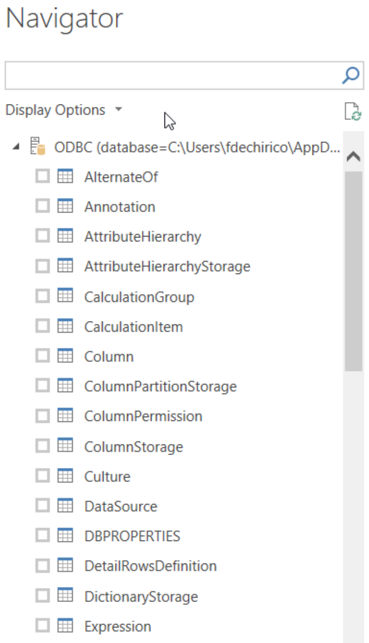
Data source name (DSN): SQLite3 Datasource

◢ Advanced options

Connection string (non-credential properties) (optional)

SQL statement (optional)

Supported row reduction clauses (optional): (None)



Navigator

ODBC (database=C:\Users\fdechirico\AppD...

- □ AlternateOf
- □ Annotation
- □ AttributeHierarchy
- □ AttributeHierarchyStorage
- □ CalculationGroup
- □ CalculationItem
- □ Column
- □ ColumnPartitionStorage
- □ ColumnPermission
- □ ColumnStorage
- □ Culture
- □ DataSource
- □ DBPROPERTIES
- □ DetailRowsDefinition
- □ DictionaryStorage
- □ Expression

Retrieve metadata info of Power BI desktop model

- The “ideal” (well, not exactly … my “desiderata” ☺) way:
  - Launch the Power BI desktop report you want to analyze
  - Launch a second Power BI desktop report specifically designed to retrieve model info
  - Digit the name of the first Power BI desktop report (the one you want to analyze)

- The question is …

IS IT POSSIBLE?

The answer is …

~~NO!~~ ~~YES!~~

The answer is … Power Vertipaq!!!!

Power VertiPaq demo