www.sqlbi.com

DirectQuery in Power BI and Analysis Services

Marco Russo
marco@sqlbi.com

We write
Books

We teach
Courses

DAX Workshop
DAX Workshop
DATA MODELING For Power BI
Power BI
SSAS TABULAR WORKSHOP

We provide
Consulting

- Remote Consulting
- Power BI/SSAS Optimization
- BI Architectural Review
- On-Site Consulting
- Custom Training & Mentoring

We are recognized
BI Experts

Microsoft Partner
Gold Business Intelligence
Gold Data Platform
Microsoft Certified Solutions Developer
Microsoft Certified Professional
Microsoft Certified Trainer
Microsoft Most Valuable Professional
SSAS MAESTRO

sqlbi.
www.sqlbi.com

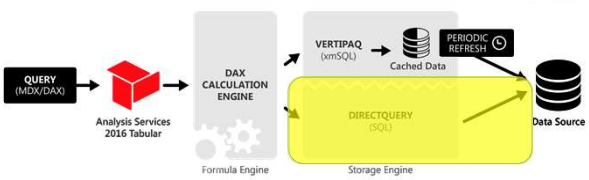
Agenda

- DirectQuery in Power BI and Analysis Services
  - What's changed since 2012/2014
  - How to create a data model in DirectQuery mode
- Querying data models in DirectQuery mode
  - What happens when you use DirectQuery
  - Differences between clients: Excel and Power BI
  - Security models
- Optimizing DirectQuery
  - Best practices for DAX, data modeling, and data sources
  - Leveraging columnstore indexes
- Final considerations
  - When to use DirectQuery, Q & A

Storage engines in Power BI & Analysis Services

- In-Memory (VertiPaq)
  - Create a copy of data in memory
  - Single access to data source
  - Queries run on a "snapshot" of data
- DirectQuery
  - Queries sent to data source, the Tabular engine does not "cache" data
  - No processing time
  - Real-time queries
  - Higher workload on data source

Tabular Query Architecture



DirectQuery Advantages

- No processing time
  - Queries executed when needed
  - Database contains only metadata
- Real-time queries
  - Always up-to-date information
- No bulk copy of data from the data source
  - Large datasets take time to move from the data source to be processed by the Tabular engine

DirectQuery Limitations

- Only one data source for a data model
- Limits on calculated columns and row-level security
  - Certain DAX functions cannot be used
  - E.g. CALCULATE, RELATEDTABLE, ...
- No calculated tables

Supported Data Sources for Power BI

- Amazon Redshift
- Azure HDInsight Spark (Beta)
- Azure SQL Database
- Azure SQL Data Warehouse
- Google BigQuery (Beta)
- IBM Netezza (Beta)
- Impala (version 2.x)
- Oracle Database (version 12 and above)
- SAP Business Warehouse (Beta)
- SAP HANA
- Snowflake
- Spark (Beta) (version 0.9 and above)
- SQL Server
- Teradata Database
- Vertica (Beta)

Supported Data Sources for Analysis Services

- Microsoft SQL Server (version 2008 or later)
- Microsoft SQL Azure Database
- Microsoft Azure SQL Data Warehouse
- Microsoft Analytics Platform System (APS)
- Oracle (version 9i or later)
- Teradata (V2R6, V12)

VertiPaq versus DirectQuery

- VertiPaq is a database
  - Data is compressed
  - Data need to be in-memory
  - Consolidation of different sources
  - Some ETL steps done in DAX
- DirectQuery is a query technology
  - Data is stored in SQL Server
  - No consolidation of different sources
  - Cannot perform ETL steps
  - No latency from updates (but slower queries)

DirectQuery Settings in Power BI

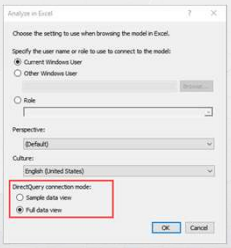
- Data Connectivity Mode
  - Import or DirectQuery
  - Decision made when model is created
  - It cannot be changed later
  - It always impersonates the current user

DirectQuery Settings in Analysis Services

- Model Setting
  - Prevents usage of unsupported features
- DirectQuery Mode
  - When activated, makes the model DirectQuery
- Impersonation Settings
  - Specific Windows account
  - Use the service account
  - Impersonate current user

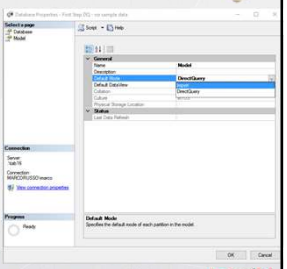
Sample Data Partitions

- Useful for debugging
- Not used in production
- Typically created with specific views for subset of data
- Sample Data is stored in VertiPaq mode and requires process



Setting DirectQuery with SSMS

- Deploy as VertiPaq
- Set DirectQuery later
- All sample data removed
- Model turns DirectQuery
- Can also use PowerShell, XMLA



Using DAX in DirectQuery

- Optimized functions
  - Functions that are translated straight in SQL
  - Example: DIVIDE, ISBLANK, ROUND…
- Optimized but cannot be used in calculated columns
  - Translated to SQL, but work on tables
  - Example: SUM, ALL, CALCULATE, FILTER
- Non-optimized functions
  - Computed by Formula Engine only
  - Based on raw data retrieved by DirectQuery
  - Example: TOPN, GENERATE

Calculated Columns

In calculated columns you can only use optimized functions working on a single row
Moreover, they are computed at every single table scan at query time

```
--
-- This calculated column works fine
--
Sales[Amount] = Sales[Quantity] * Sales[Unit Price]

--
-- This calculated column cannot be defined in DirectQuery
--
Products[TotalSales] =
SUMX (
    RELATEDTABLE ( Sales ),
    Sales[Quantity] * Sales[Unit Price]
)
```

Semantic Differences in DAX

- SQL semantic is different than DAX
- Many differences
  - 1/BLANK is fine in DAX, error in SQL
  - Statistical functions with one-row tables are different
  - Casting errors might appear
  - Numerical overflow
  - Supported date-time ranges
    - Since March 1st 1990 to December 31st 9999
  - Full list of differences: https://docs.microsoft.com/en-us/sql/analysis-services/tabular-models/dax-formula-compatibility-in-directquery-mode-ssas-2016

Row-Level Security

- Security expressions need to follow the same restrictions of calculated columns
- Only simple, optimized DAX functions
- WHERE and JOIN conditions injected in every SQL query to ensure security constraints

Query Limits

Large materializations are not supported; you can configure the *MaxIntermediateRowsetSize* setting (only SSAS, neither in Azure AS nor in Power BI), but you need to be careful when doing it

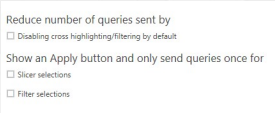
```
EVALUATE
ROW (
    "orders",
    COUNTROWS ( ALL ( Sales[Order Number], Sales[Order Line Number] ) )
)

Translates to this query and it is likely to fail on large datasets

SELECT TOP ( 1000001 )
        [t1].[Order Number],
        [t1].[Order Line Number]
FROM    ( SELECT * FROM Sales ) AS [t1]
GROUP BY [t1].[Order Number], [t1].[Order Line Number];
```

Monitoring DirectQuery

- Using SQL Server Profiler
  - Monitor SSAS for DirectQuery events
  - Monitor SQL for SQL queries
- Monitoring is very important
  - Performance of DirectQuery depend solely on SQL
  - Need to understand what kind of queries are executed
  - Optimize SQL model based on the queries

Query Reduction in Power BI

- Reduce the number of queries sent to the data source



Using MDX in DirectQuery

- MDX generated by Excel
  - Analyze in Excel with Power BI
- Limitations
  - **No used-defined hierarchies** (not displayed in Excel)
  - No relative object names
  - No session-scope MDX statements
  - No tuple with member from different levels in subselects
  - Detailed list of MDX limitations available at https://docs.microsoft.com/en-us/sql/analysis-services/tabular-models/directquery-mode-ssas-tabular

Real-time and Coherency of Values

Brand	CY 2007	CY 2008	CY 2009	Total
A. Datum	$1,181,110.71	$463,721.61	$451,352.33	$2,096,18
Adventure Wor	$2,249,988.11	$892,674.52	$868,449.65	$4,011,112.
Contoso	$2,729,818.54	$2,369,167.68	$2,253,412.80	$7,352,399.03
Fabrikam	$1,652,751.34	$1,993,123.48	$1,908,140.91	$5,554,015.73
Litware	$647,385.82	$1,487,846.74	$1,120,471.47	$3,255,704.03
Northwind Traders	$372,199.93	$469,827.70	$198,524.49	$1,040,552.13
Proseware	$880,095.80	$763,586.23	$902,462.12	$2,546,144.16
Southridge Video	$688,107.56	$294,635.04	$401,671.25	$1,384,413.85
Tailspin Toys	$74,603.14	$97,193.87	$153,245.41	$325,042.42
The Phone Company	$362,444.46	$355,629.36	$405,745.25	$1,123,819.07
Wide World Importers	$471,440.71	$740,176.76	$690,339.18	$1,901,956.66
Total	**$11,309,946.12**	**$9,927,582.99**	**$9,353,814.87**	**$30,591,343.**

4 queries, results might not match because of updates happening in the meantime

Real-time and DAX

Results might be confusing also for DAX code, many queries are executed for each DAX statement, no matter how simple it is

```
EVALUATE
SUMMARIZECOLUMNS (
    ROLLUPADDISSUBTOTAL ( Product[Color], "IsGrandTotalRowTotal" ),
    "Amt", [Sales Amount]
)
```

DirectQuery Scenarios

- Real-time Reporting
- No Processing Window
- Not enough memory for the model
- Create a semantic layer on top of SQL

Real-time Modeling

- Do you really need real-time on the entire data model?
  - Many limitations might create issues in the model
  - Large datasets are hard to optimize
  - Maybe real-time data is useful for the last few days, with simpler calculations
- Build two models
  - One VertiPaq model for historical data
  - One DirectQuery model for real-time data

DirectQuery: Conclusions

- DirectQuery is alternative to VertiPaq
- SSAS Tabular becomes just a semantic model
- The data source becomes the storage engine
- VertiPaq is faster, but requires copying data and RAM
- DirectQuery is slower, but it lowers update latency

Whitepaper

- DirectQuery in Analysis Services 2016
  - https://www.sqlbi.com/articles/directquery-in-analysis-services-2016/
    - It includes demos and further details about the content of this presentation
    - All the considerations are still valid in Analysis Services 2017 and Azure Analysis Services as of February 2018
    - New improvements for Power BI February 2018 not included in the whitepaper

Just like Jimi Hendrix ...

We love to get feedback

Please complete the session feedback forms

SQLBits - It's all about the community...

Please visit Community Corner, we are trying this year to get more people to learn about the SQL Community, equally if you would be happy to visit the community corner we'd really appreciate it.

sqlbi

Thank you!

Check our articles, whitepapers and courses on
www.sqlbi.com