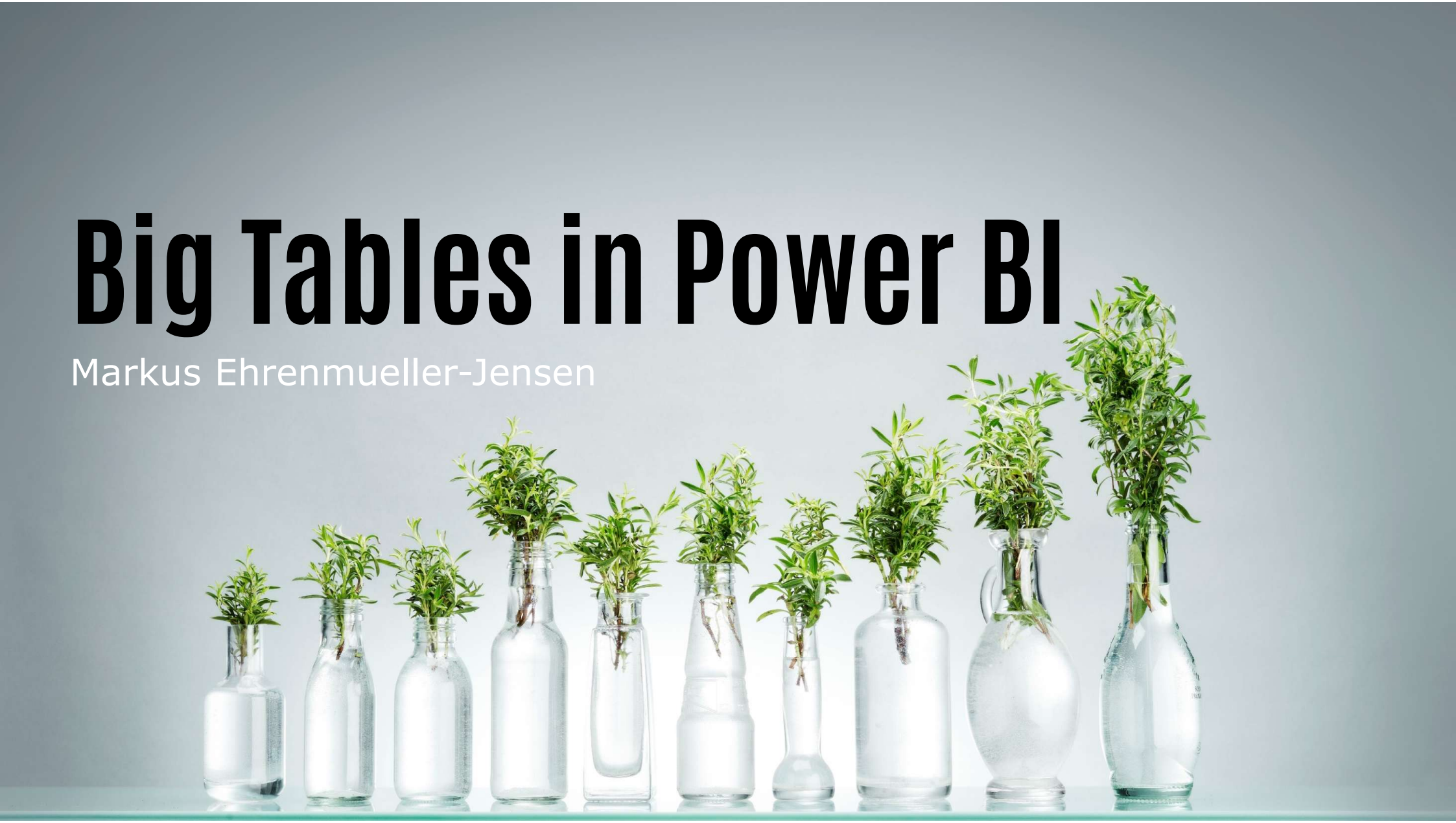
Big Tables in Power BI

Markus Ehrenmueller-Jensen



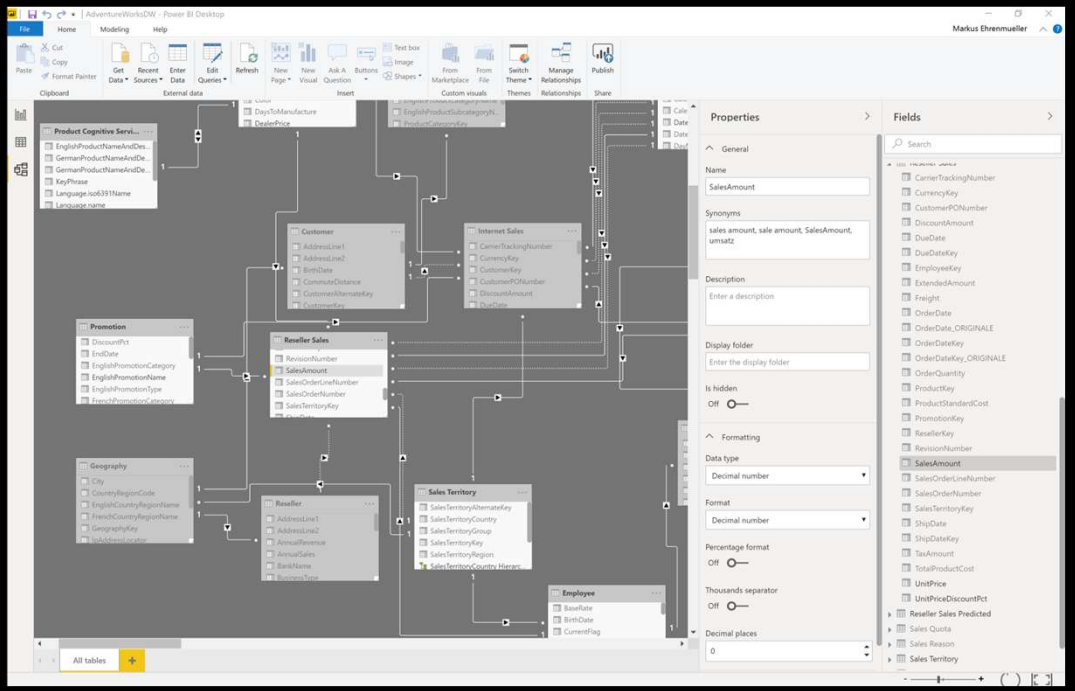
I work with models

Other people



Source: https://pxhere.com/en/photo/127022

Me



Big Data

Increasing volume
More velocity
Greater variety

Big Table

Increasing volume

More velocity

Greater variety

How to transform data

Idea A: Only storing query definitions
Idea B: Storing query results
Idea C: Storing query results and meta data
Idea D: Data in different granularities

Idea A: Only storing query definitions

Grocery list

Keeping data “unharmed” in the source system only

Queries (transformations) are executed at report time

No extra space needed
- Except for query text

Report data is always fresh
- “Real-time”

Reports might be slow
- DAX & Power Query M is automatically translated to SQL
- Complexity of query
- Source system is not built for such queries
- SELECT can block and be blocked

Data in source system must be available at query time
- During business time, archived data not available for reports

Some transformations might not be possible
- e. g. Slowly Changing Dimensions
- Limitations of query interface

e. g. a DirectQuery in Power BI, VIEW in Azure SQL DB



Demo

DirectQuery mode

Idea B: Storing query results

Grocery in food storage room

- Duplicating the data from the source system
- Scheduled queries (transformations) persist data
- Extra space needed
  - Compression can be applied
- Report data is always (a bit) stale
- Reports will be faster
  - Reports are satisfied from a “cache”
  - Report runs simple(r) queries
  - Complexity of queries is moved to the backend
  - Source system is not queried during report runtime
- Greatest choice of transformation tools
  - Spark, SQL, ADF, Fabric Dataflow Gen2, Power Query, etc.

Idea C: Storing query results and meta data

Pre-cooked meal in the refrigerator

Extra information about the persisted data is added
- Index (clustered vs. non-clustered)
- Semantic model (e. g. measures)

Scheduled queries (transformations) will take more time

Extra space needed for meta data
- Unless it helps with better compression

Reports will be faster
- Meta data speeds up query run-time

e. g. Import mode in Power BI, Index in Azure SQL DB



Demo

Import mode

Basic Storage Modes in Power BI

Direct Query

Data stays in remote database

Ad-hoc queries

DAX queries must be translated to SQL

Should be the exception

Idea A (grocery list)

Import

Data is compressed & in-memory

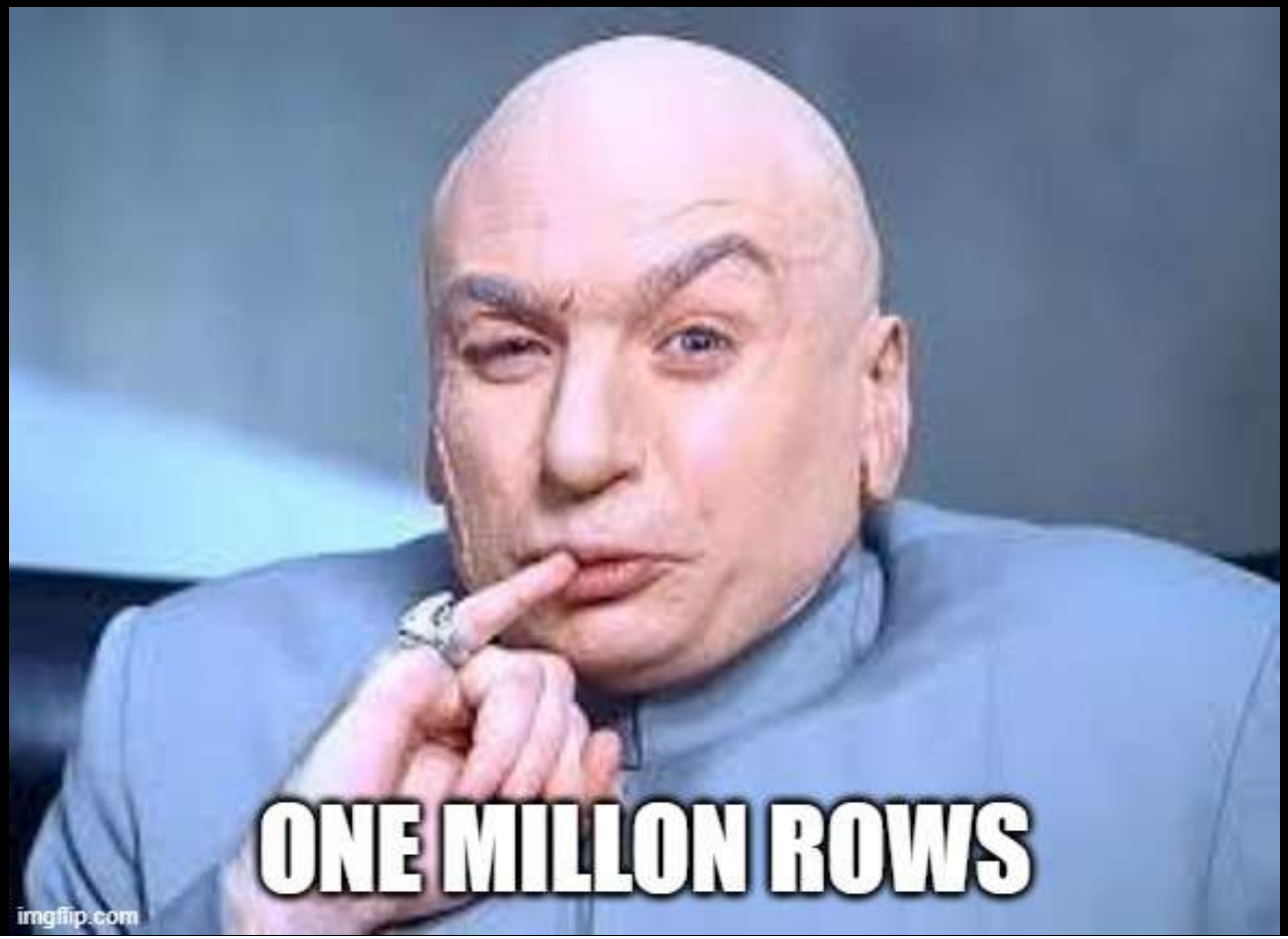
Regular refresh necessary

DAX queries run natively

Should be default

Idea C (pre-cooked meal)

~~Too many rows for~~ Import mode



Incremental refresh

Cook/eat in smaller portions

Power BI's default is full refresh

- All the model's/table's data is fully loaded and compressed
- Easy logic, can take long time

Incremental refresh only loads new data

- More complex logic, based on a timestamp in the source system
- Make sure to partition and index by timestamp in source system
- **Refresh will be faster**

Space used and report performance will not change

Built-in logic vs. manual partitioning



Demo

Incremental refresh

Composite model

Pre-cooked main (idea C) and picking up ice cream from around the corner for dessert (idea A)

- Not all tables are in the same storage mode
  - Some are in Import mode (typically small dimension tables)
  - Some are in DirectQuery mode (typically fact tables and large dimension tables)
- Only imported tables must be refreshed
  - Overall **refresh will be faster**
- Less spaced used
- Reports can be even slower
  - Applying filters can be complex/slow
  - SQL statements can be long and contain actual data



Demo

Composite model

Dual mode

Use grocery list when shopping is needed anyways (idea A) or eat pre-cooked meal, when nothing else needs to be shopped (idea C)

A table is simultaneously in both, Import and DirectQuery mode

If queried together with imported/dual tables, imported version will be used

If queried together with a table in DirectQuery mode, DirectQuery version will be used

- When both are from the same date source



Demo

Dual mode

Hybrid mode

Some rows of a table are in Import mode, others in DirectQuery mode

a) Cook fresh, but also use canned food
Most recent data is in DirectQuery mode to enable real-time
Older data is imported and refreshed when necessary
→ recommended if real-time is the goal (supported out-of-the-box)

b) Eat pre-cooked meal if fresh enough, use grocery list otherwise
Most recent data is in Import mode to enable fast reports
Older data is in DirectQuery mode
→ recommended if storage size is the issue (custom partitions)



Demo

Hybrid mode

Direct lake mode

Cooking fresh in the microwave oven

- Similar to DirectQuery mode
  - Data is not moved
  - Delta parquet files are directly queried
- Similar to Import mode
  - Parquet files are similar to VertiPaq storage
- Don’t forget to still model your data
- Only available in Fabric Semantic Model
  - Limitations apply

Demo

Direct Lake mode



Idea D: Data in different granularities

Eat pre-cooked meal if available, cook fresh otherwise

Several queries (*grocery list*) or tables (*pre-cooked meal*) for each level of granularity

Extra complexity at report query time

- To decide which query or table to use

Reports will be faster

- Reports are satisfied from query or table with lowest level of detail/granularity

e. g. one table as is, one table pre-aggregated by date and product, one table pre-aggregated by date only

Pre-aggregegregate data

Keep several versions of the same data in different granularities

Choose the table with the highest possible granularity for a query

- Make Power BI aware of the aggregation table or
- Implement your own logic

Aggregation table can be in any storage mode

- Import, DirectQuery, Dual, Hybrid

https://learn.microsoft.com/en-us/power-bi/transform-model/aggregations-advanced



Demo

Aggregation table

Storage Modes in Power BI

- Direct Query
  - *Idea A: Grocery list*
  - Source system is queried at report time
  - Variations: Live Connection, DirectLake
- Import
  - *Idea C: Pre-cooked meal*
  - Data is loaded into memory
- Composite model
  - *Pre-cooked main (idea C) and picking up ice cream from around the corner for dessert (idea A)*
  - Some tables in the model are in Import mode, some are in DirectQuery mode
- Dual
  - *Use grocery list when shopping is needed anyways (idea A) or eat pre-cooked meal, when nothing else needs to be shopped (idea C)*
  - One table is in both, Import and DirectQuery mode
- Hybrid
  - *Yesterday's pasta (idea C) with freshly fetched fish from the grocery store (idea A)*
  - Some partitions of a table are in Import mode, some are in DirectQuery mode

Summary

- Import mode is superior in terms of report performance
- Incremental refresh can speed up refresh time
- Avoid DirectQuery mode
  - Only if refresh takes too long, data is too big or row level security is too complex
  - Only for certain tables/partitions (consider hybrid approach)
  - Use Dual mode for the other tables (instead of Import mode) if they come from the same data source
- Add aggregation tables to speed up report performance

Questions?

Savory DATA

Markus Ehrenmueller-Jensen

Founder & BI Architect
markus@savorydata.com
@MEhrenmueller
/markus-ehrenmueller
www.savorydata.com

Microsoft CERTIFIED Solutions Expert
Business Intelligence
Data Platform
Microsoft CERTIFIED Trainer

Microsoft Data Platform

BI Developer
Database Developer
Database Admin



Microsoft MCT

MVP Microsoft® Most Valuable Professional

