After this session

Storage modes	Refresh operations	Partitioning	Orchestration
Understand the different storage modes in Power BI and how this relates to refreshes.	Understand which different options there are to refresh your Semantic Model manually as well as in an automated fashion.	Understand how different partitions in the data model are generated and can be triggered to refresh.	Understand how you can orchestrate your Power BI refresh and bind to other processes.

Data-Marc.com

Marc Lelijveld

Technical Evangelist | Solution Architect
Macaw Netherlands

🐦 @MarcLelijveld

in linkedin.com/in/MarcLelijveld

🌐 Data-Marc.com

DutchFabricUsergroup.com

FAVORITE STUFF:

What we cover today

- Refreshes?

- Incremental refresh

- High frequency refreshes & hybrid tables

- Refresh individual objects
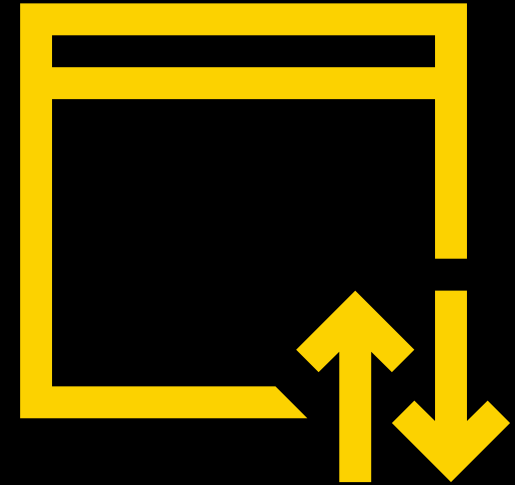
- Orchestration & automation

- Wrap-up

Refreshes

Understanding storage modes

Import

Dual

DirectQuery

Oh wait… we now also have DirectLake!

Data-Marc.com

Understanding Storage Modes

Three storage modes

- Import – data cached in the model

- DirectQuery – queries are submitted to the back-end data source

- Dual – can act in both above storage modes, depending on query context

Configuring storage modes

- Storage modes are set on table level

- Setting storage mode to Import is an irreversible operation

- Data in DirectQuery mode cannot be displayed in the data tab

Caches and DirectQuery

Risks of mixing storage modes

- Avoid mismatch in results when query bits DirectQuery compared to Import

- Data cached (import) could be behind compared to DirectQuery data

- Make sure cached data is kept in sync – regularly refresh!

Data-Marc.com

Benefits to choose one or the other

- Improve query performance – cache data for faster end-user performance

- Data refresh optimization – no need to refresh for non-cached data

- Near-real time requirements – reduce query latency when in DirectQuery mode

- Large Semantic Models – choose to not import <u>certain</u> data by leveraging aggregations for example

Data-Marc.com

Different refresh operation types

Type	What	Power BI Desktop	Power BI Service *(Interface)*
Manual	Individual Table	✔	✖
	Full model	✔	✔
Scheduled	Individual Table	✖	✖
	Full model	✖	✔

Data-Marc.com
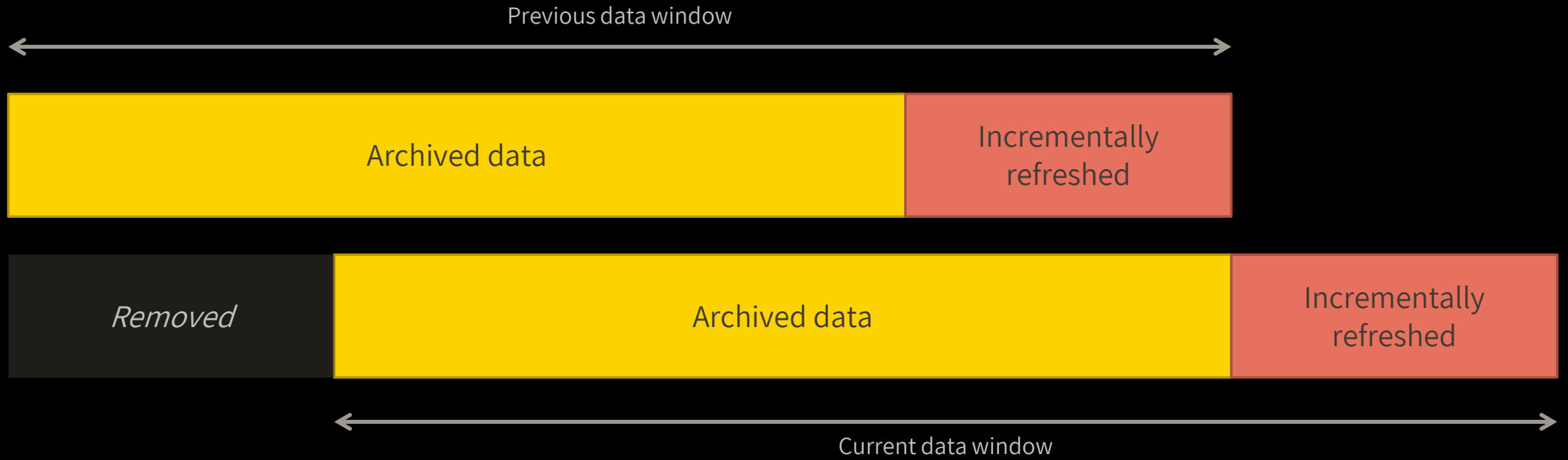
Incremental refresh

Considering incremental refresh

Why should you use it?

- Large data volumes

- Faster refreshes

- Resource consumption reduced
  (Both source & Power BI side)

Data-Marc.com
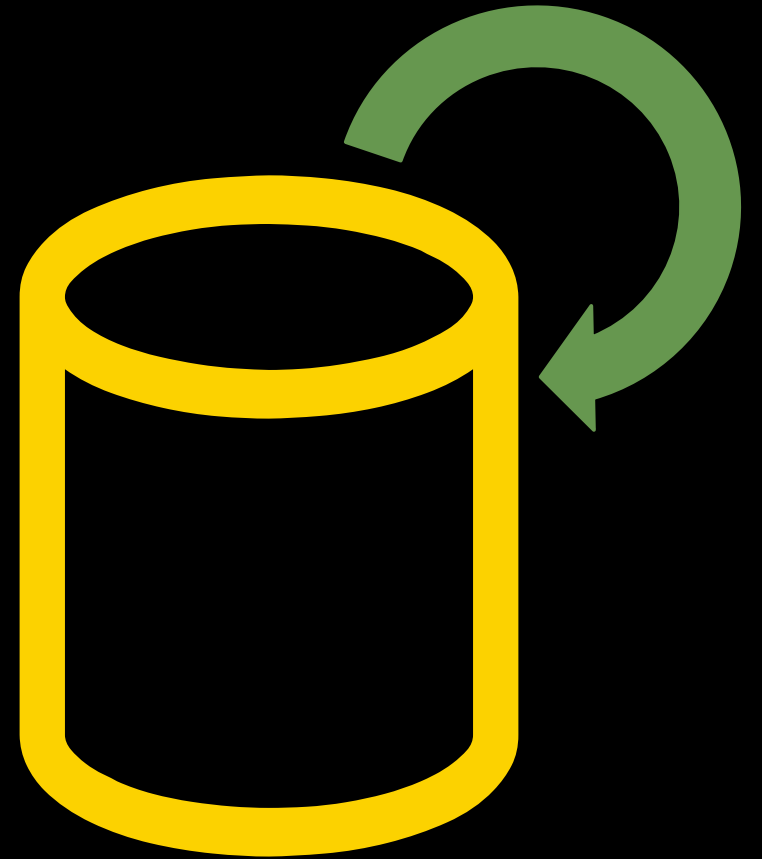
Incremental refresh

Incremental refresh is supported for Power BI Premium, Premium per user, **Power BI Pro**, and Power BI Embedded Semantic Models.

Getting the latest data in **real time** with DirectQuery is **only supported for Power BI Premium***, Premium per User, and Power BI Embedded Semantic Models.

* Premium capacities = Fabric capacities

Data-Marc.com

Supported data sources

Data sources must be

- Structured

- Relational

- Allow query folding

- Typically – SQL like sources

Data-Marc.com
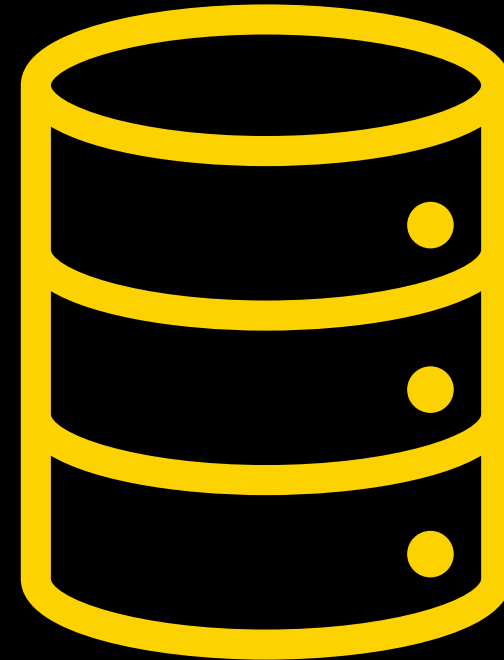
Breaking up your data in partitions

- Each partition represents a time frame

- New partitions get added automatically

- Old partitions (in archive) get merged


```
▲ ⬡ Model
   ■ Data Sources
   ■ Perspectives
   ■ Relationships
   ■ Roles
   ▶ ■ Shared Expressions
   ▲ ■ Tables
      ▲ ⊞ Timestamps
         ▲ ⬡ Partitions
            ⬡ 2022
            ⬡ 2023Q1
            ⬡ 2023Q204
            ⬡ 2023Q20501
            ⬡ 2023Q20502
            ⬡ 2023Q20503
            ⬡ 2023Q20504
            ⬡ 2023Q20505
            ⬡ 2023Q20506
            ⬡ 2023Q20507
            ⬡ 2023Q20508
            ⬡ 2023Q20509
         ▤ # Rows
         ▤ Max Timestamp
         ▤ Timestamp
   ▶ ■ Translations
```

Data-Marc.com
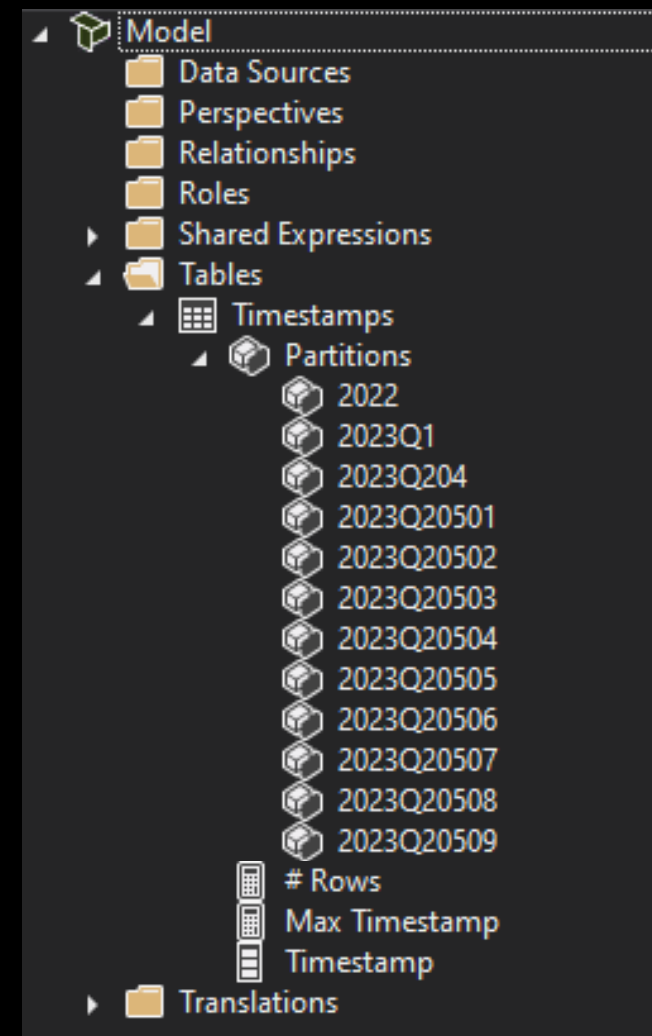
Configure Incremental Refresh

Manage Parameters

New

RangeStart ✕
RangeEnd

Name

RangeStart

Description

☑ Required

Type

Date/Time ▾

Suggested Values

Any value ▾

Current Value

1/1/2022 12:00:00 AM

OK   Cancel

Incremental refresh and real-time data

Refresh large tables faster with incremental refresh. Plus, get the latest data in real time with DirectQuery (Premium only). Learn more

ⓘ These settings will apply when you publish the dataset to the Power BI service. Once you do that, you won't be able to download it back to Power BI Desktop. Learn more

1. Select table

Orders ▾

2. Set import and refresh ranges

🔘 Incrementally refresh this table

Archive data starting 5 Years ▾ before refresh date

Data imported from 1/1/2017 to 11/17/2022 (inclusive)

Incrementally refresh data starting 3 Days ▾ before refresh date

Data will be incrementally refreshed from 11/18/2022 to 11/20/2022 (inclusive)

3. Choose optional settings

☐ Get the latest data in real time with DirectQuery (Premium only) Learn more

Selected table cannot be folded for DirectQuery.

☐ Only refresh complete days Learn more

☐ Detect data changes Learn more

4. Review and apply

Archived	Incremental Refresh
5 years before	3 days before Refresh date

Data-Marc.com
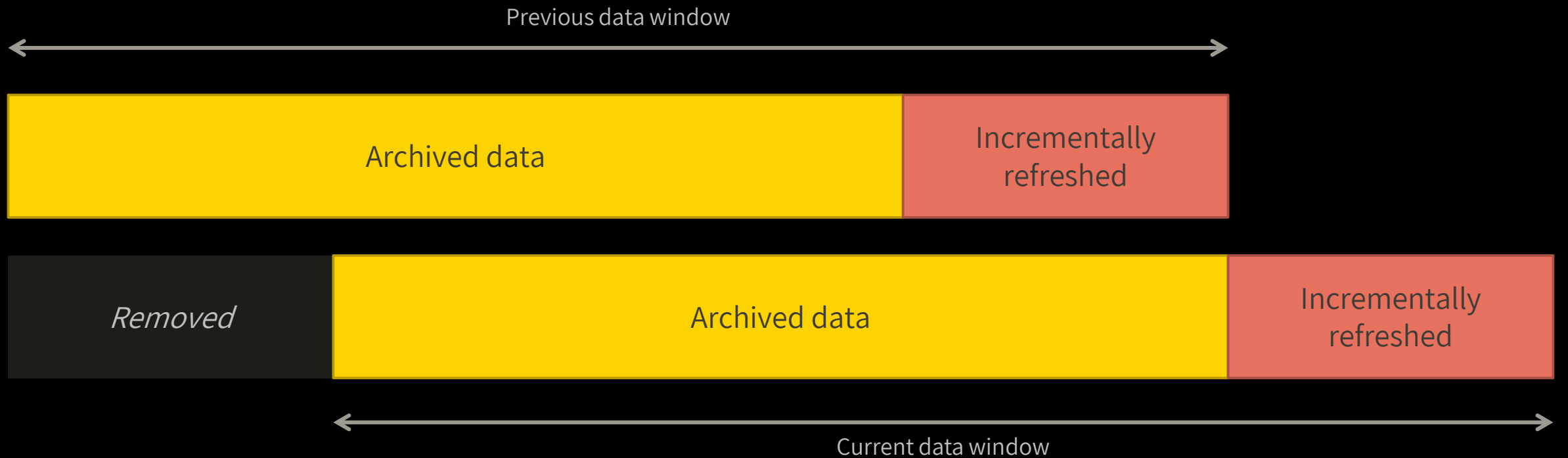
Detect data changes

- Even more selective refresh – only that what has been changed

- Based date/time column used to identify and refresh only those days where the data has changed

- Typically, system generated dates like Modified Date Time

- Should never be the same column as your incremental periods

- The maximum value is evaluated for each partition in the incremental range to detect whether the partition should be refreshed or not

Data-Marc.com

High frequency refreshes & hybrid tables

Data-Marc.com
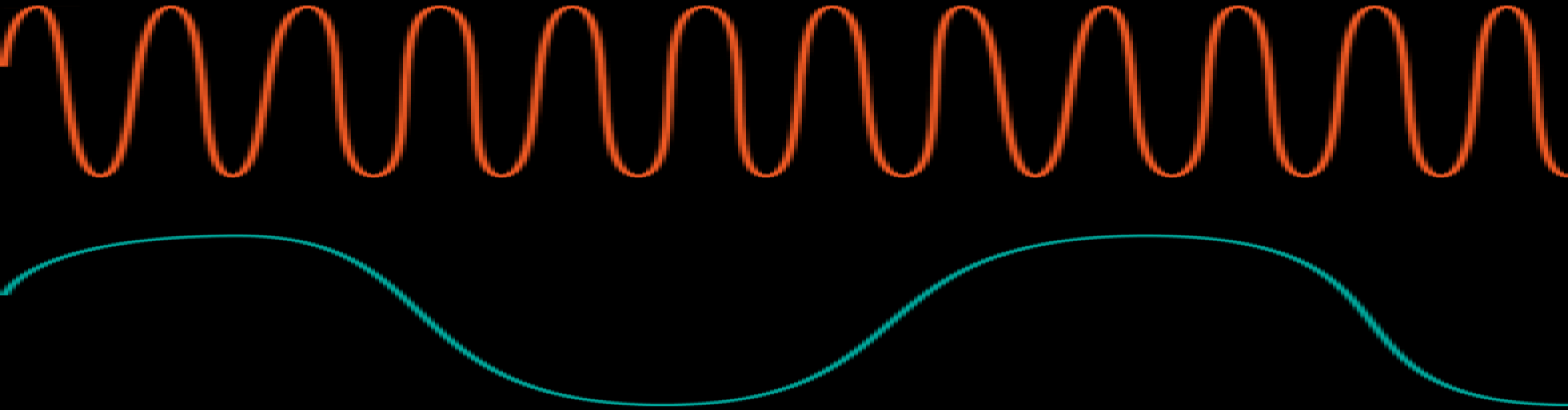
What is high frequency?

Let's say real time?

- Hourly?

- Minutes?

- Seconds?

Or…

- Just daily batches

Refresh limitations

Frequency	Manual	Scheduled (via the service)
Free	Up to 8 times per day	
Pro	Up to 8 times per day	
Premium per User	Unlimited	48 times per day
Premium / Fabric Capacity	Unlimited	48 times per day

Data-Marc.com

Hybrid tables

- Live / Realtime data in Power BI

- Combines different storage modes on partition level in a single table

- Goes hand-in-hand with Incremental Refresh

Granularity	Name	Row Count	
Year	2011	295,489,717	
Year	2012	297,678,498	
Year	2013	295,575,442	
Year	2014	292,477,875	
Year	2015	297,780,469	
Year	2016	294,060,081	Archived: **Import**
Year	2017	300,419,682	
Year	2018	296,541,108	
Year	2019	292,787,420	
Year	2020	299,273,979	
Quarter	2021Q1	74,135,277	
Month	2021Q104	24,939,498	
Day	2021Q10501	820,805	
Day	2021Q10502	826,885	Incremental refresh: **Import**
Day	2021Q10503	821,043	
Day-DirectQuery	2021Q10504-DQ	271,110	Real time: **DirectQuery**
Total		**3,063,898,887**	

Data-Marc.com

Hybrid tables

- Implementation with Incremental Refresh

- Customizable via 3rd party tooling like Tabular Editor *(unsupported)*

Limitation:

Only 1 DQ partition per table allowed.

Hybrid tables – what challenge does it solve?

- Realtime scenarios without full tables on DQ mode

- No complex refresh mechanisms needed with partition refresh and queries over XMLA

- No more multiple tables and complex DAX to combine to achieve the same goal

Data-Marc.com
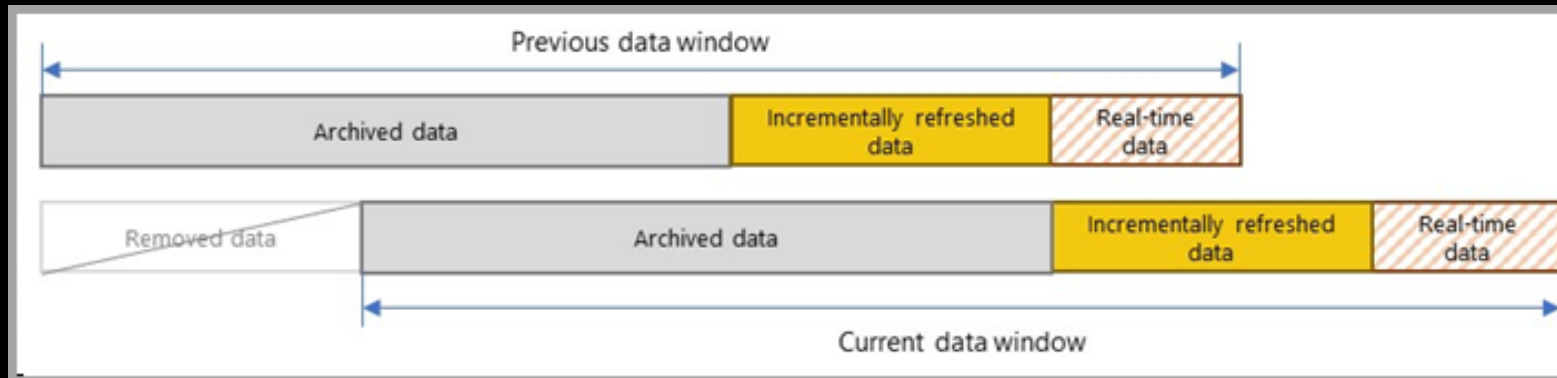
Hybrid tables – Keep in mind that...

- Premium feature

- DAX restrictions for DirectQuery apply

- Limited Power Query & DAX capabilities (due to DQ)

- Requires Large Semantic Model Format (storage) in workspace

- Performance hit on upstream data sources

Data-Marc.com

Demo - configure hybrid tables and streaming data

Data-Marc.com

Can I change partition storage modes?

You cannot adjust tables with incremental refresh policies applied. However, there are other options to consider.

Options	
Data View	Default
Mode	DirectQuery
Source Type	PolicyRange

Could not save metadata changes to database ✕

❌ Failed to save modifications to the server. Error returned: 'An incremental-refresh table can only have a single real-time partition. Please remove the additional real-time partitions from table '<pii>Sales Hybrid</pii>'. Learn more, see https://go.microsoft.com/fwlink/?linkid=2172127.

Technical Details:
RootActivityId: 169f2d83-4369-4392-9080-cb1b89d2063e
Date (UTC): 3/3/2022 7:04:14 PM
'.

Save the model as a file to make sure you do not lose any work. In case the database connection was lost, you can still apply your changes to the database by using the 'Model > Deploy' option.

OK

Data-Marc.com

Refresh individual objects

Data-Marc.com

Effective refreshing

Considerations

- Refreshing the entire model takes too long with high load on sources

- Can we only refresh certain tables?

- Can we only refresh certain partitions?

- Can we use DQ tables/partitions (Hybrid Tables)

What do we need?

- Enhanced refresh API

- XMLA Endpoints

Data-Marc.com

Smaller increments

Bind your refresh operations in Power BI to refresh operations in for example your data platform

Run smaller increments to speed up the end-to-end process

Trigger one (or multiple) selective table(s) or partition(s) at a time.

Data-Marc.com

Smaller increments

Smaller increments can only be triggered to refresh via:

- Manual processing via Management Studio (or other tools)

- Power BI Enhanced Refresh API - Previously known as Async refresh API

- XMLA endpoints

Data-Marc.com

Enhanced refresh API

More granular controls

- type - full / clearValues / calculate / dataonly / automatic / defragment

- Commitmode – default is transactional

- maxParallism – 10 max

- retryCount – number of retries before failing

- objects – which objects to refresh

- applyRefreshPolicy – true/false whether you want to apply potential incremental refresh if configured

- effectiveDate – in case of incremental refresh, this parameter overwrites the current date

Data-Marc.com

Enhanced refresh API

Specify the objects to refresh

POST
https://api.powerbi.com/v1.0/myorg/groups/f089354e-8366-4e18-aea3-4cb4a3a50b48/datasets/cfafbeb1-8037-4d0c-896e-a46fb27ff229/refreshes

```json
{
    "type": "Full",
    "commitMode": "transactional",
    "maxParallelism": 2,
    "retryCount": 2,
    "objects": [
        {
            "table": "DimCustomer",
            "partition": "DimCustomer"
        },
        {
            "table": "DimDate"
        }
    ]
}
```

Data-Marc.com

Demo – Refresh one table or partition

More than refreshes

Enhanced refresh API also allows you to

- **GET** refresh history

- **GET** individual refresh details like progress or failures

- **DELETE** a running refresh

JSON   ⧉ Copy

{
    "startTime": "2020-12-07T02:06:57.1838734Z",
    "endTime": "2020-12-07T02:07:00.4929675Z",
    "type": "Full",
    "status": "InProgress",
    "currentRefreshType": "Full",
    "objects": [
        {
            "table": "DimCustomer",
            "partition": "DimCustomer",
            "status": "InProgress"
        },
        {
            "table": "DimDate",
            "partition": "DimDate",
            "status": "InProgress"
        }
    ]
}

Data-Marc.com

Is the API the only option?

- Execute operations over XMLA endpoints

- Using Tabular Model Scripting Language

- More granular control using SQL Service Management Studio


Data-Marc.com
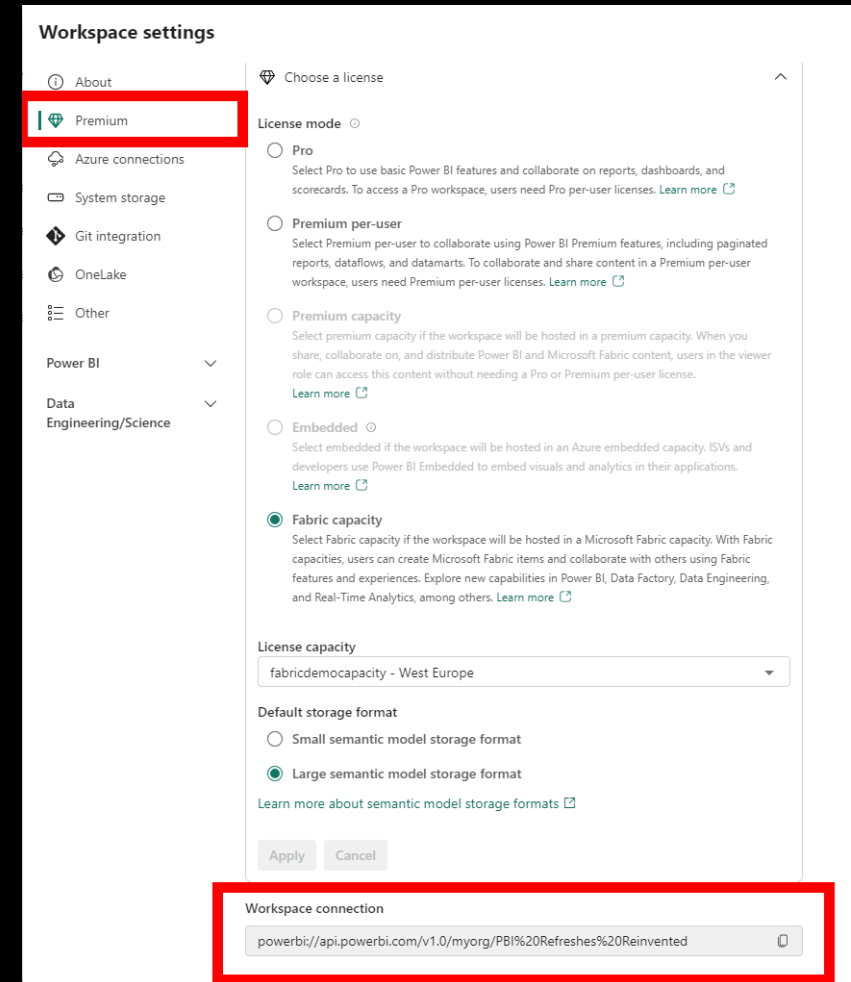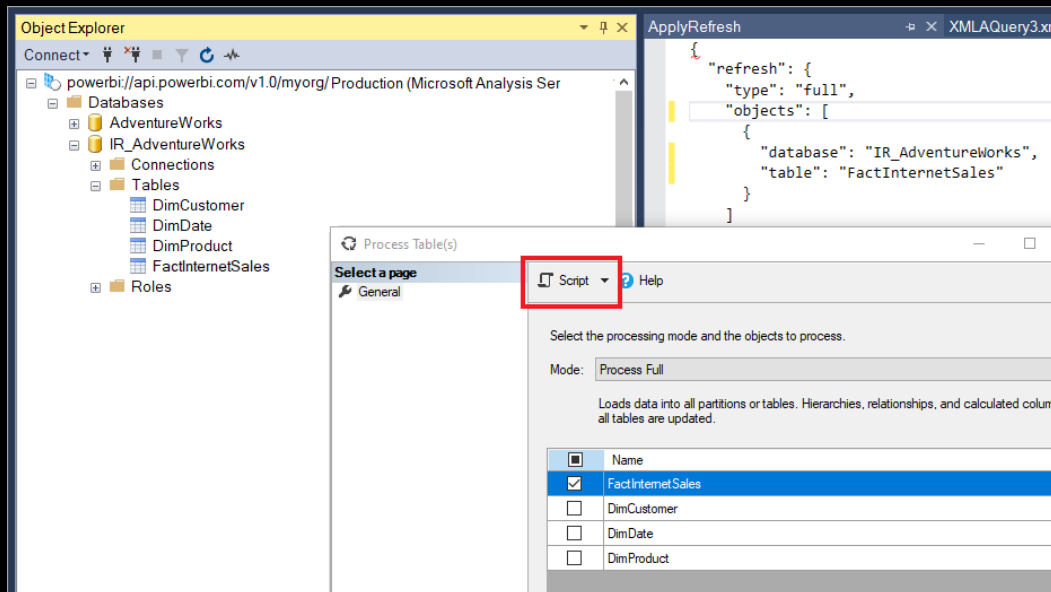
Orchestration & Automation

Binding refreshes to processes

Data-Marc.com

Central E2E orchestration

- Combine pipelines from Data Platform with Power BI

- Lowest latency between source and report

- Incremental loading where possible

- Consider including backup operations for Power BI

Data-Marc.com

# What to use to trigger and bind together?

Data-Marc.com
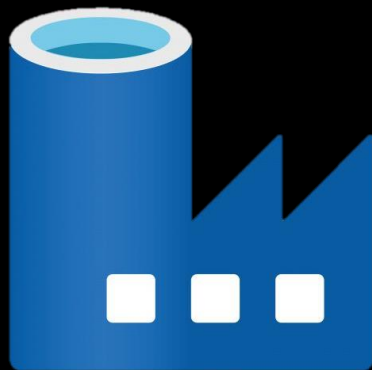
What is Semantic Link exactly?

Semantic Link is a feature in Microsoft Fabric that allows you to connect from ~~Synapse~~ Data Science Notebooks to Power BI Semantic Models.

This feature **only** exists and works in Microsoft Fabric.

Power BI

Semantic Link

Synapse Data Science

Apache Spark

Direct Lake

OneLake

Data-Marc.com
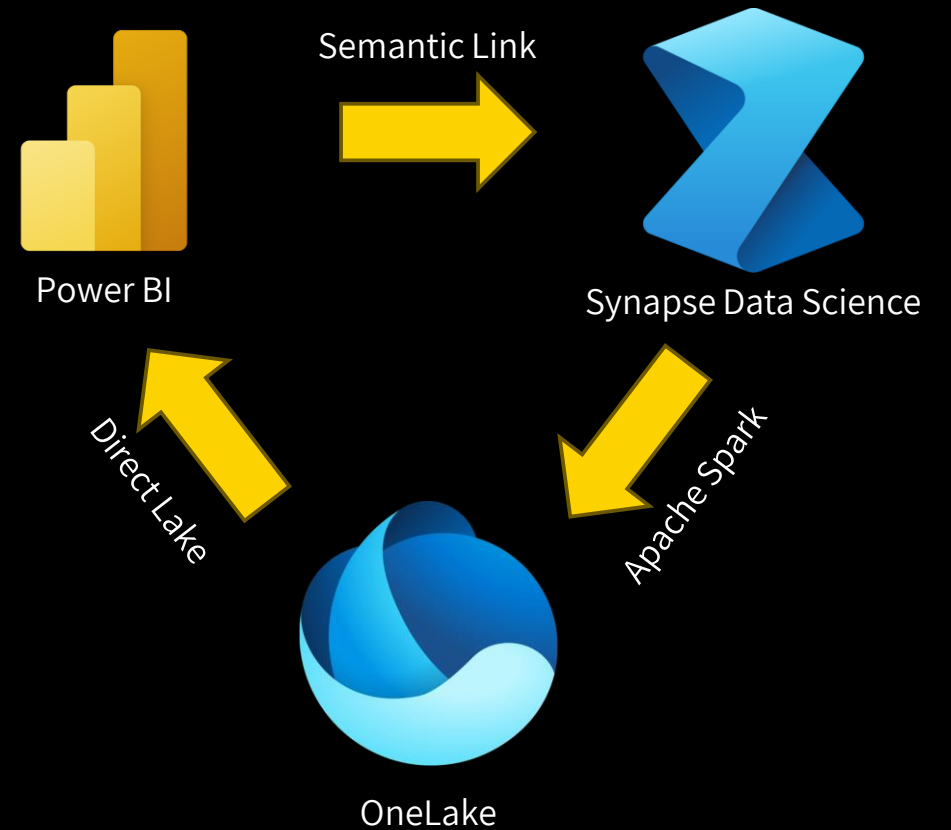
Semantic Link in Fabric notebooks!

Refresh your semantic models via a notebook and trigger dependent actions.

- Given Semantic Link uses the REST APIs, you can orchestrate not only your semantic model refresh, but also trigger upstream dataflows for example

- Refresh individual tables, partitions or reprocess partitions through enhanced refresh API

- Anything else what is possible with the REST API

```Python
Python                                                    Copy
PowerBIRestClient(token_provider: TokenProvider | None = None)
```

Data-Marc.com

Demo – Fabric Notebooks for automation

Wrap up

LET'S RECAP...

Refreshes can be manual or **automated**

Incremental refresh makes your refresh **more efficient**

With **detect data changes**, further optimization can be done

Partitioning in incremental refresh should be **based on functional dates**

Lower latency by linking Power BI refreshes to other processes

Use the enhanced refresh API to **trigger individual model objects**

Use the toolkit of your preference for **end-to-end orchestration & automation**

Data-Marc.com

Resources

Setting up scheduled refreshes
https://learn.microsoft.com/en-us/power-bi/connect-data/refresh-scheduled-refresh

Configure incremental refresh
https://learn.microsoft.com/en-us/power-bi/connect-data/incremental-refresh-configure

Enhanced refresh API
https://learn.microsoft.com/en-us/power-bi/connect-data/asynchronous-refresh

Semantic Model connectivity with XMLA endpoints
https://learn.microsoft.com/en-us/power-bi/enterprise/service-premium-connect-tools

Tabular Model Scripting Language (TMSL)
https://learn.microsoft.com/en-us/analysis-services/tmsl/tabular-model-scripting-language-tmsl-reference?view=asallproducts-allversions

These slides
https://github.com/marclelijveld/Slide-decks

Thanks for attending!

Marc Lelijveld
Technical Evangelist | Architect
Macaw Netherlands

Microsoft® Most Valuable Professional | Microsoft MCT | sessionize MOST ACTIVE SPEAKER 2022 | sessionize MOST ACTIVE SPEAKER 2023

@MarcLelijveld

linkedin.com/in/MarcLelijveld

Data-Marc.com

Feedback!

Data-Marc.com