**Incremental Refresh**

**Info source :** <https://learn.microsoft.com/en-us/power-bi/connect-data/incremental-refresh-configure>

Incremental refresh extends scheduled refresh operations by providing automated partition creation and management for dataset tables that frequently load new and updated data.

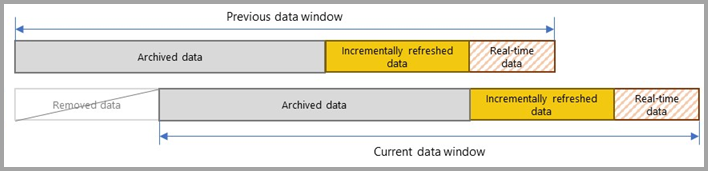
With incremental refresh and real-time data:

****Refreshes are faster****

****Resource consumption is reduced****

****Large datasets are enabled****

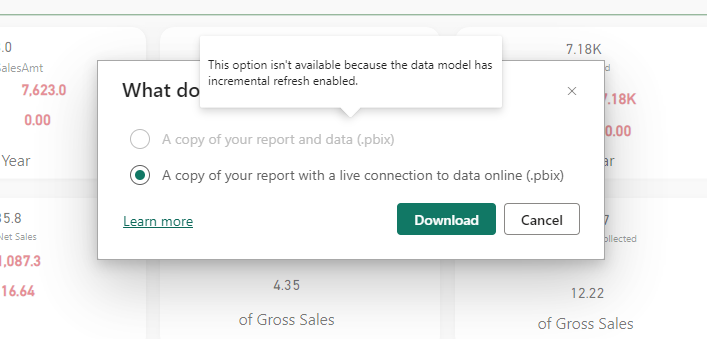
* When you configure incremental refresh in Power BI Desktop, parameters are used to filter that's loaded into the model.
* When report published to the Power BI service, with the first refresh operation the service creates incremental refresh and historical partitions
* The service then overrides the parameter values to filter and query data for each partition based on date/time values for each row.
* With each subsequent refresh, the query filters return only those rows within the refresh period dynamically defined by the parameters.
* Rows with a date/time no longer within the refresh period then become part of the historical period, which isn't refreshed.
* When a historical partition is no longer in the historical period defined by the policy, it's removed from the dataset entirely.



**Disadvantages :**

Incremental refresh files cannot be downloaded.

-> on click Download

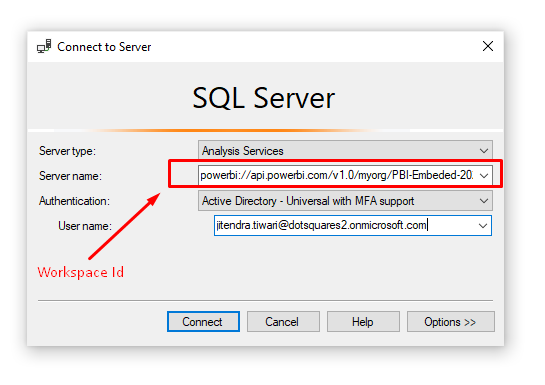


--> To verify your incremental refresh in SQL Server Management Studio , You must have an Premium Licence.

-->Incremental refresh can only be perform with import storage mode.

If you have a premium liscence

* You can connect your dataset of power bi services , to SQL Server management studio , with the help of workspace ID.



Steps to implement incremental refresh :

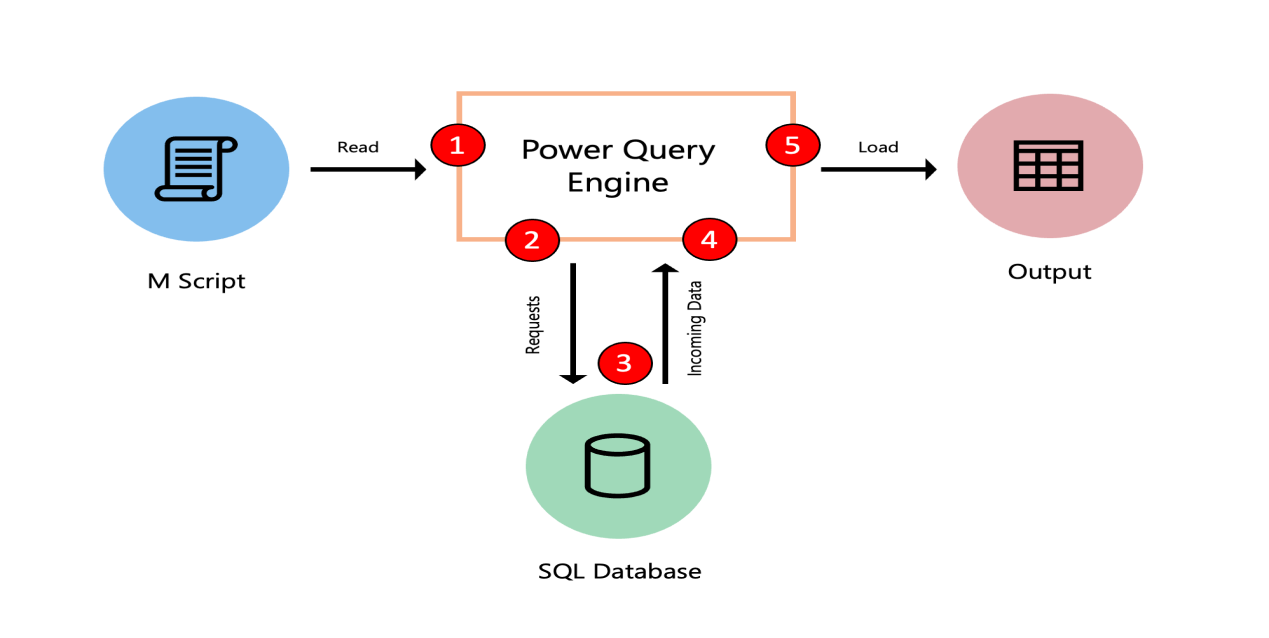
1. Create two parameters **RangeStart** and **RangeEnd**.
2. Apply filter based on parameter on your table’s date column.
3. Then , define incremental refresh policy on your table.
4. After this , publish your report to power bi service.
5. After the model is published to the service, RangeStart and RangeEnd are overridden automatically by the service to query data defined by the refresh period specified in the incremental refresh policy settings.
6. After publishing to the service, you perform an initial refresh operation on the dataset.
7. The initial refresh operation can take quite a while to complete.
8. The Power BI Service generates partitions over the table with the incremental refresh.
9. If in the future, we require to make some changes in the data model then we have to use some other tools than Power BI Desktop, such as **[Tabular Editor](https://github.com/TabularEditor/TabularEditor/releases/tag/2.16.5" \t "https://www.biinsight.com/implementing-incremental-refresh-in-power-bi-part-1/_blank)**, **[ALM Toolkit](http://alm-toolkit.com/" \t "https://www.biinsight.com/implementing-incremental-refresh-in-power-bi-part-1/_blank)** or **[SQL Server Management Studio (SSMS)](https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver15&WT.mc_id=DP-MVP-5003466" \t "https://www.biinsight.com/implementing-incremental-refresh-in-power-bi-part-1/_blank)** to deploy the changes to the existing dataset without overwriting the existing dataset.

**Query Folding**

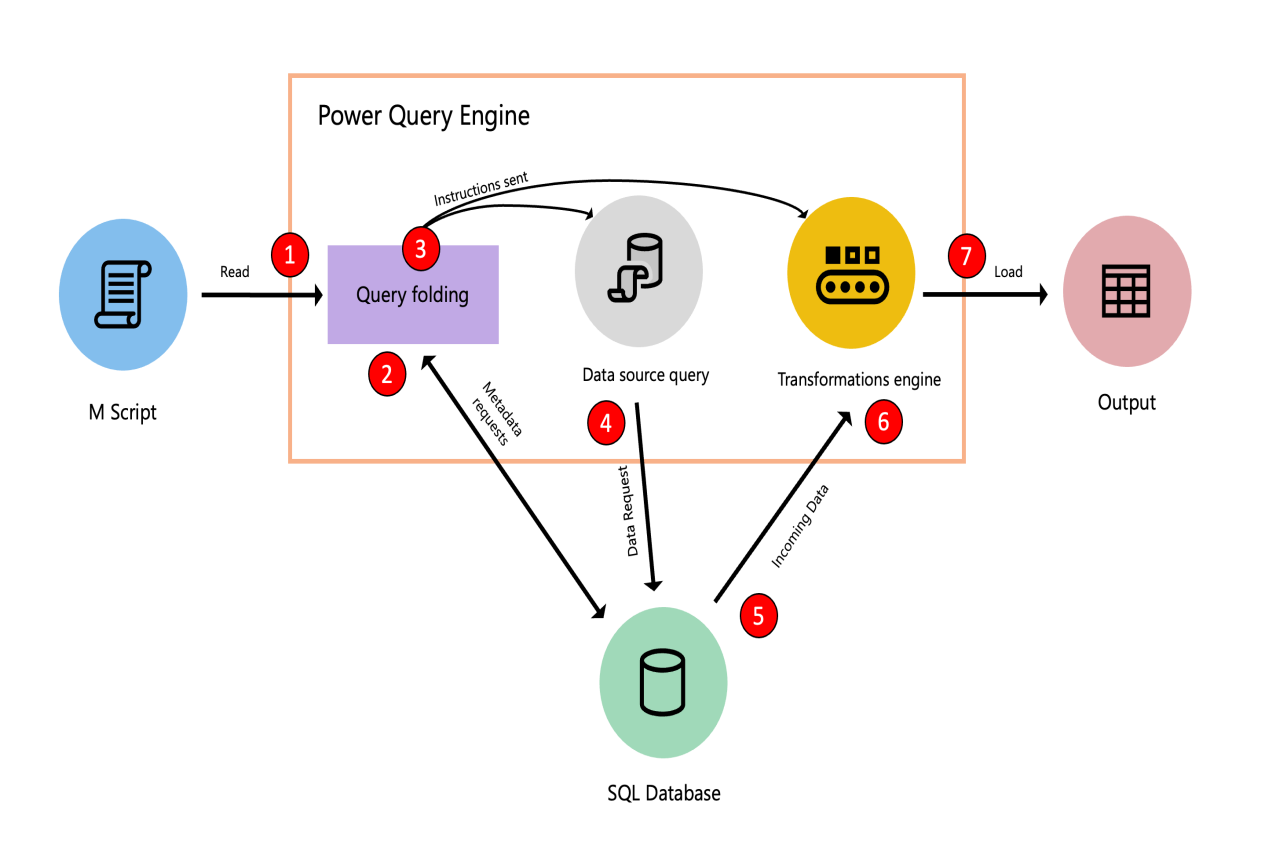
Query folding is the process of Power Query Editor, in which transformation done with data , converted to SQL.

Query folding is the ability for a Power Query query to generate a single query statement to retrieve and transform source data.

**Query evaluation in Power Query**



The following diagram demonstrates the steps that take place in this optimization process.



**Steps :**

1. M script, submitted to the Power Query engine.
2. The Query folding mechanism submits metadata requests to the data source to determine the capabilities of the data source, table schemas, relationships between different entities at the data source, and more.
3. Based on the metadata received, the query folding mechanism determines what information to extract from the data source and what set of transformations need to happen inside the Power Query engine.
4. Send instruction to two other components : data source query

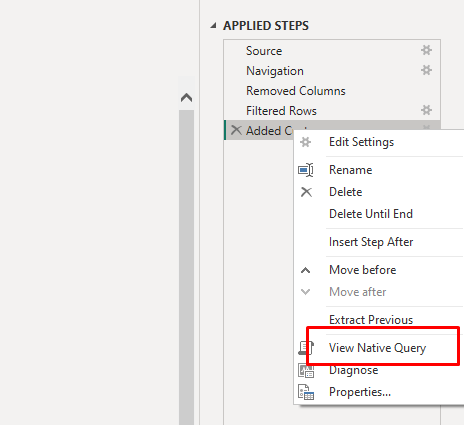
Transformation Engine

1. Send Request to data source.
2. Once the data is inside Power Query, the transformation engine does the transformations.
3. Results are loaded.

More info :

https://learn.microsoft.com/en-us/power-query/query-folding-basics

**-->How did you know that the transformation will be folded back or not?**



Go to > Power Query Editor > Query Setting > Right click on transformation

If the view native query option is enabled , it means the query will be folded back.