FABRIC HANDBOOK

Focused on certification exam

by Eugene Mazarakis

Table of Contents

01

About The Writer

"Anything worth having takes time."

Meet Eugene Mazarakis, a senior Data Engineer with a wealth of experience across diverse industries, including energy, banking and telecommunications. Over the course of numerous challenging projects, Eugene has honed his expertise in designing, implementing, and optimizing data solutions tailored to complex business needs.

He combines technical prowess with a deep understanding of industry demands, making him a trusted resource for transforming complex concepts into actionable solutions.

He holds a Master of Science degree in Theoretical Computer Science from the National and Kapodistrian University of Athens.

FABRIC EXAMINATION

02

You can take the examination online, which consists of approximately 60 questions, including multiple-choice questions and use cases.

You will have 1 hour and 40 minutes to complete it. If you are unsure about an answer, you can mark the question and review it later.

The passing score for the examination is 700.

03

FABRIC

01

Admin Portal

Some important settings that you need to know.

02

Datastores

What are the datastores, when do you need them, and what do they contain.

03

Deployment Solution

How can you deploy your solution from development to testing and then to the production system.

04

Tools Comparison

A comparison of the tools you need to use when working with the Fabric service.

Analytics' Types

Descriptive

The data you examine reveals **what happened**.

Diagnostic

The data you examine reveals **why did it happen** something.

Predictive

The data you examine reveals **what will happen in the future.**

Prescriptive

Examine **actions that need to happen.**

Fabric Admin Portal

Tenant Settings

- Preview features
- Security features
- Git integration

Capacity Settings

- Create, delete capacities
- Manage capacity permissions
- Changes the size of capacity (redirect to azure portal)

Fabric Data Stores

Lakehouse

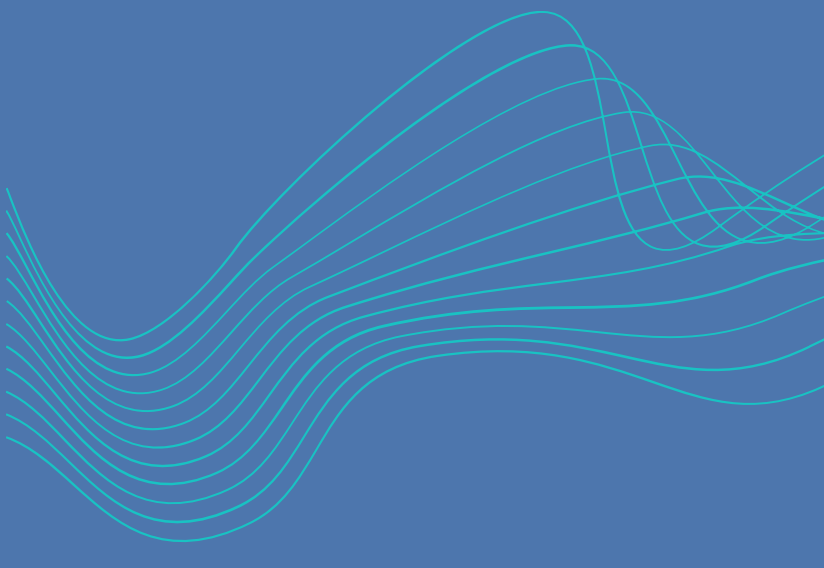
It's presented as a database above the Data Lake and uses Delta Format tables.

Warehouse

It's traditional version of a DWH, data is stored in a Delta format and can be queried through SQL.

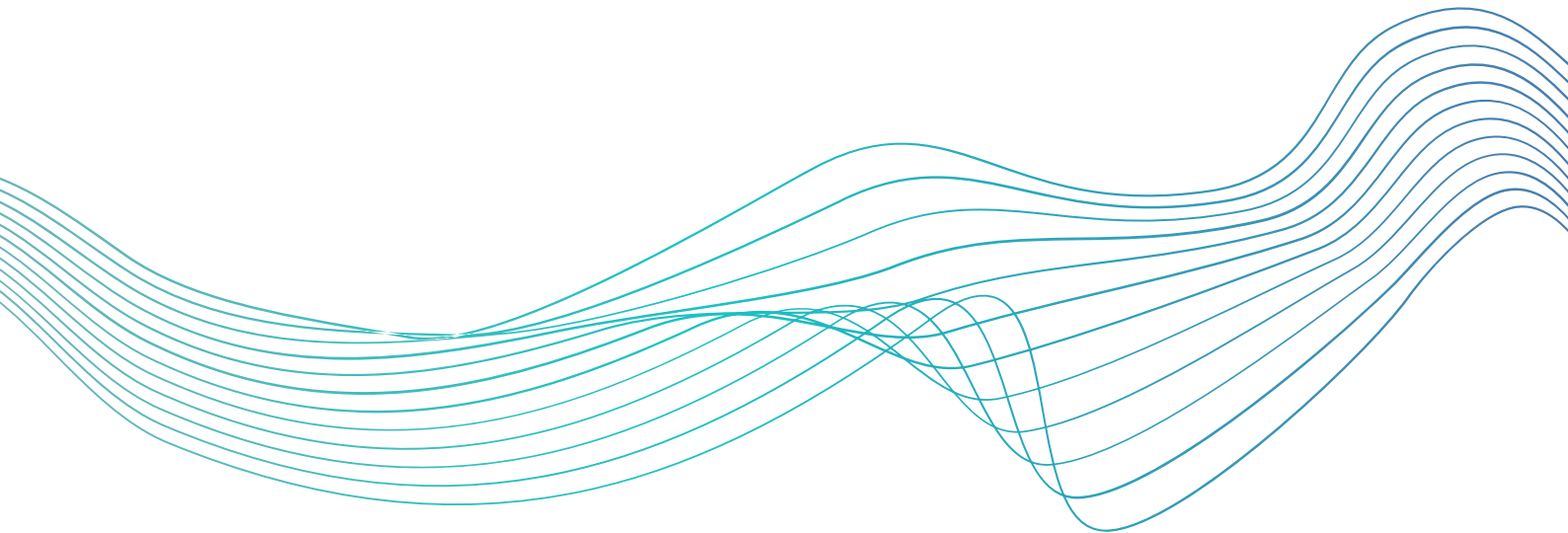
KQL database

It's used to analyze huge amount of data for Real-Time intelligence.

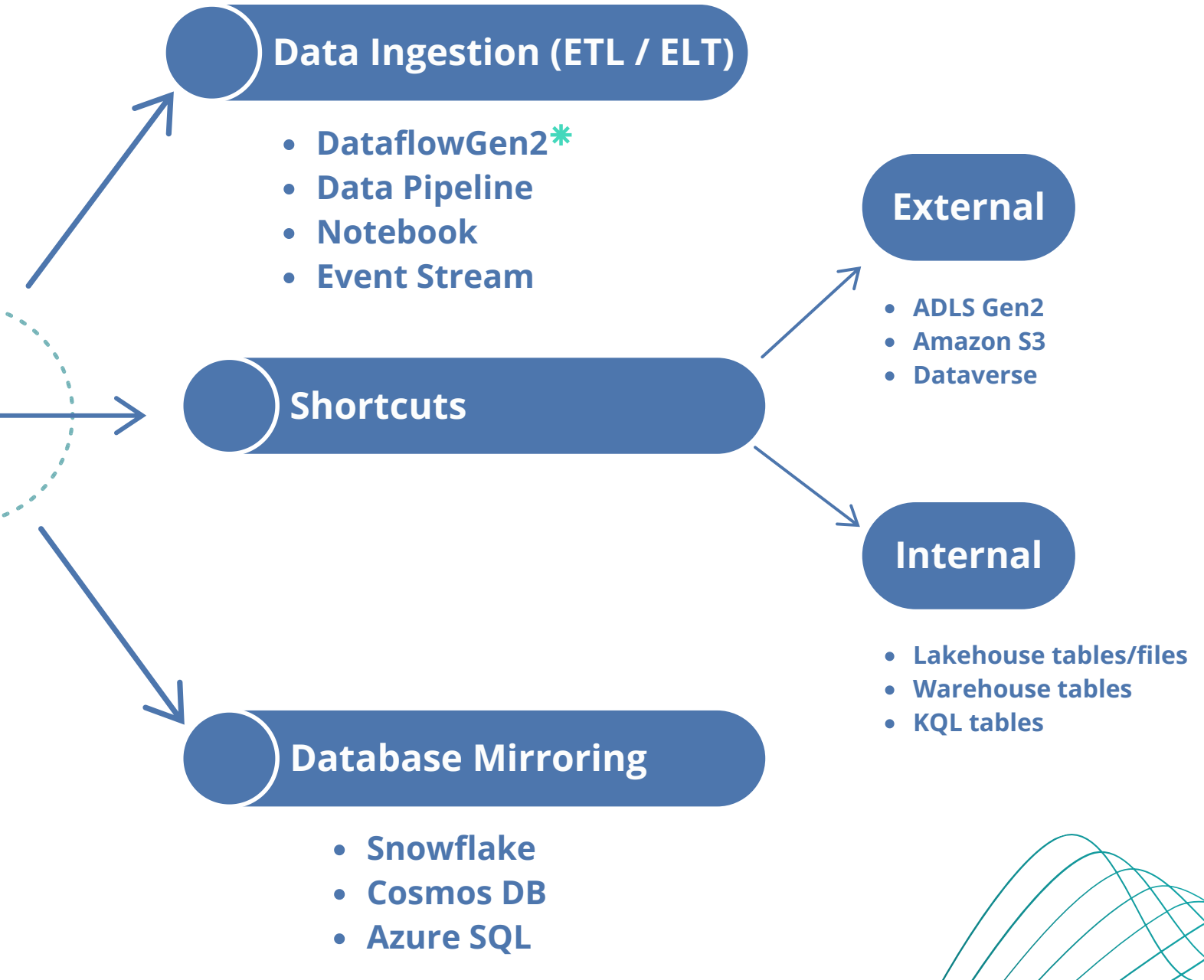


Data Types

	Structured	Semi-Structured	Unstructured	Real-Time
Lakehouse	✓	✓	✓	
DWH	✓			
KQL db				✓



Ingest Data Into Fabric



* Max number of refreshes per day are **48.**

Query Data by using SQL

SQL Analytics Endpoint

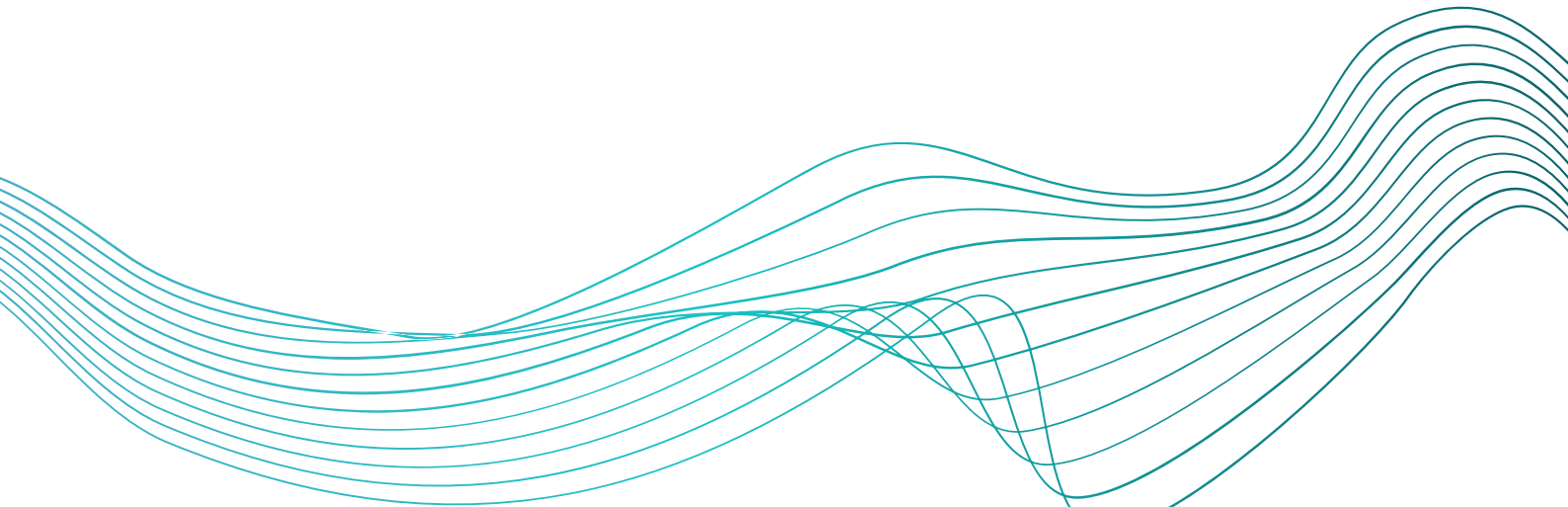
It applies only to read statements, specifically **SELECT** queries.

Fabric DWH

For **read** and **write** statements.

Query data through XMLA endpoint

Navigate to the workspace settings, obtain the link, and add it into **SSMS**. From there, you can access Lakehouse and DWH.



Lakehouse

Extra Permissions

- **Real all SQL endpoint data**: access to all data of the Lakehouse, through SQL Endpoint.
- **Read all Apache Spark:** access to underlying data files, through Apache Spark (notebook).
- **Build reports on the default semantic model**: use only the default model.

Delta Table Maintenance

- **Optimize command:** It combines many small files and makes them into a few large files.
- **Vacuum command:** Delete the files.

Warehouse

You can monitor the performance with the following 3 ways:

Capacity Metrics App

- It is a power bi app, which is installed in fabric and you can monitor performance in the warehouse.
- It shows resource intensive workloads.

Dynamic Management Views (DMVs)

They are systemic views that they give information about the warehouse.

- **sys.dm_exec_connections**: Info for the connection between warehouse and the engine.
- **sys.dm_exec_sessions**: Info for every session.
- **sys.dm_exec_requests**: Info for active request inside a session.

Query Insights

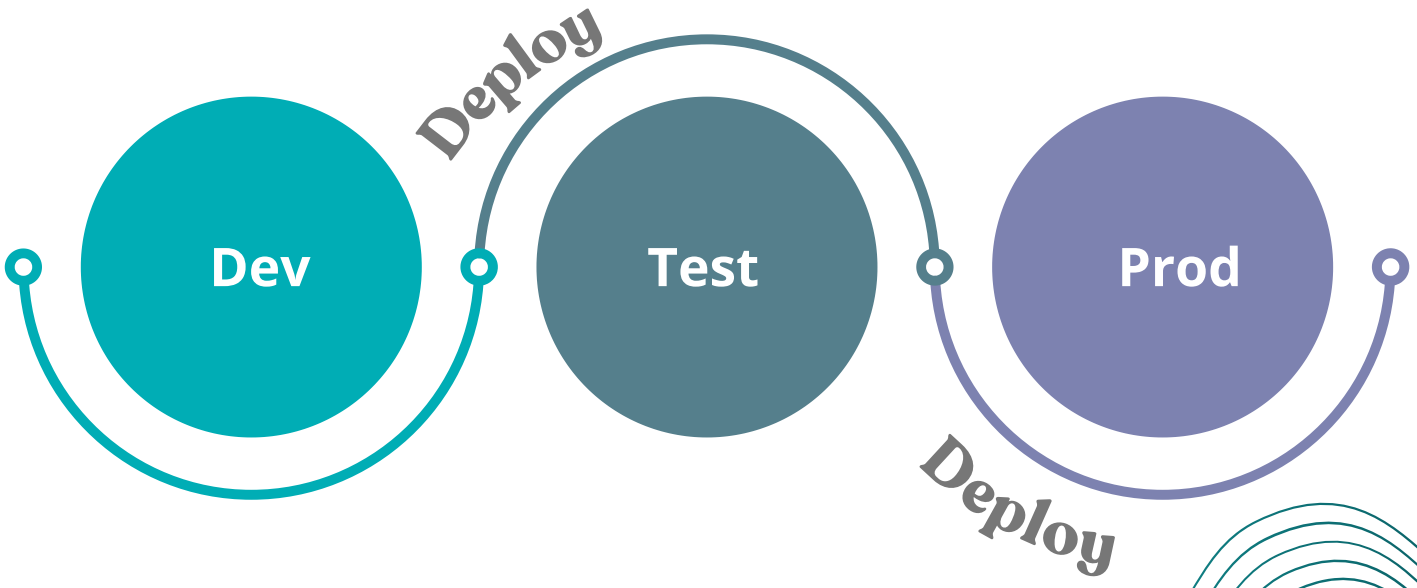
Some of the views that contains are:

- **exec_request_history:** Info for the completed SQL query.
- **frequently_run_queries**: Info for the frequently run queries.
- **long_running_queries**: Info for the execution time of the queries.

Deployment Solution

As you create it, you also define how many levels you want. Usually they are 3 (**Dev**, **Test**, **Prod**). Each of the 3 levels, assigned to a **Workspace**.

Dev: Here you make the changes.
Test: Sharing this so that tests can be done.
Prod: This is used by the customer.



Tools Comparison

DAX Studio	Tabular Editor
Write, execute, debug DAX queries.	Edit data models and create measures, calculation groups, perspectives.
Usage of Vertipaq Analyzer in order to see the size of the semantic model, the size of columns and tables.	Create iterative processes through scripting.
Analysis of data you export from the PBi desktop performance Analyzer.	Usage of the Best Practice Analyzer in order to locate common issues (i.e. Surrogate column with the 'Summarize by' property set to a value other than 'None').
Perform trace analysis.	Create Object Level Security.
	Disable implicit measures.
	Bootstrap the initial full load for the incremental refresh.

POWER BI

04

01

Workspace Level Sharing

Information about roles assigned to a workspace.

02

Types of Files & Visuals

Types of visuals and files and their respective use.

03

Power Query Data Profiling Tool

Give insights about the data in a column.

04

Incremental Refresh

Technique to load only the newly changed records of a table.

Workspace Level Sharing

Access can be granted to individuals or groups, through one of the following roles. The roles are applied to the workspace. A user, depending on the role, can also perform the corresponding actions.

Role	Actions
Admin	**Update, delete workspace. Only this role can define another admin.**
Member	**Add new user to a workspace, with <= permission.**
Contributor	**It can see content and can edit it.**
Viewer	**It can see content, but it cannot edit it.**

Types of Files

PBIP : power bi project file

If you save the file as .**pbip**, it will create **2** folders:

- One folder for the report metadata.
- One folder for the semantic model metadata.

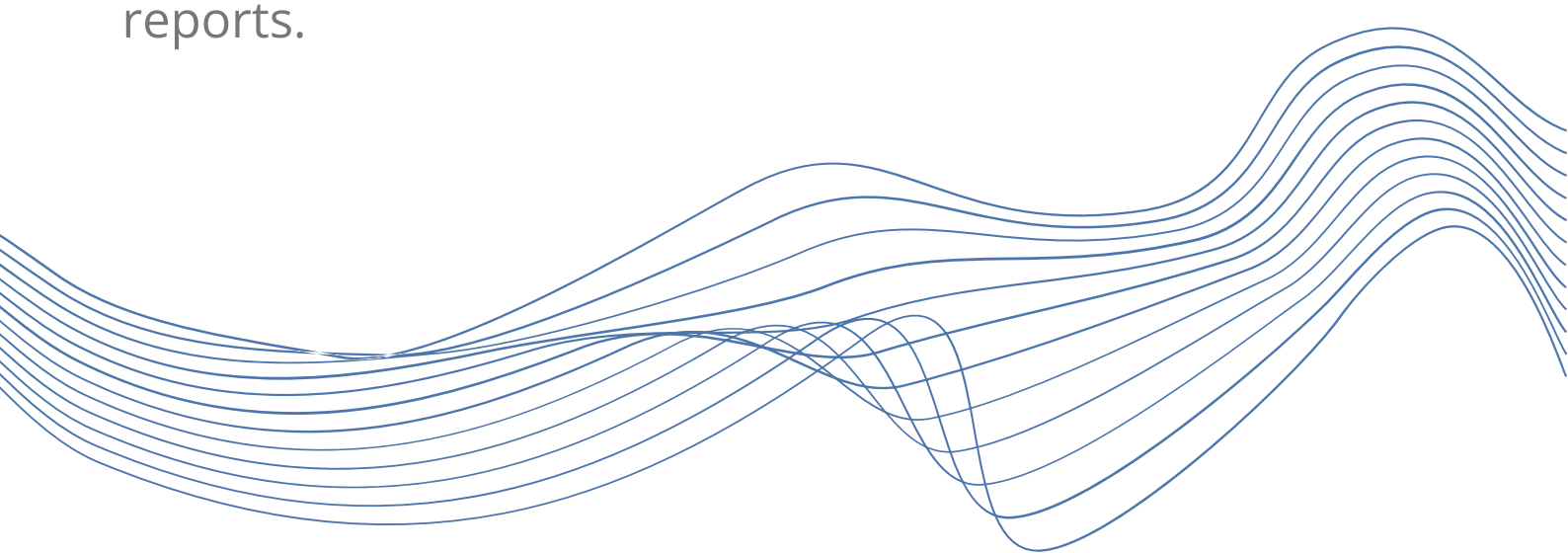
With this file type, you can use the GIT control in order to manage the report and the model.

PBIT: power bi template file

Re-usable asset for the creation of power bi reports.

PBIDS: power bi data source file

Re-usable asset to transfer data connections between reports.



Types of Visuals

Visual	Use Case
Table & Matrix	Present aggregate data.
Bar & Column	Present 1 categoric & 1 numeric variable.
Line & Area	Present time-series data.
Card	Present KPI metric.
Pie, Donut, TreeMap	Present ratios.
Combo Charts	Present more than one measure on Y-axis.
Funnel	Present movement along a linear process.
Gauge	To present progress towards a goal.
Waterfall	To present running total over a period.
Scatter	Present relationship between 2 numerical variables.
Q & A	Ask questions about the data.

Power Query Data Profiling Tool

Column Quality *

For each column in the query, it returns the following 3 percentages:

- **Valid**: % of valid values in the column.
- **Errors**: % of errors in the column.
- **Empty**: % of empty values in the column.

* By default, it occurs on the first 1,000 records, but it can also be applied to all data.

Column Value Distribution

For each column in the query, it returns the distribution of values within that column. It also returns:

- **Distinct**: The distinct values that appear in a column.
- **Unique**: The values that appear only once in a column.

Column Profile

For each column in the query, it returns statistical metrics and the value distribution. Some of the metrics are:

- **MIN**: the minimum value of the column.
- **MAX**: the maximum value of the column.
- **STD**: the standard deviation.

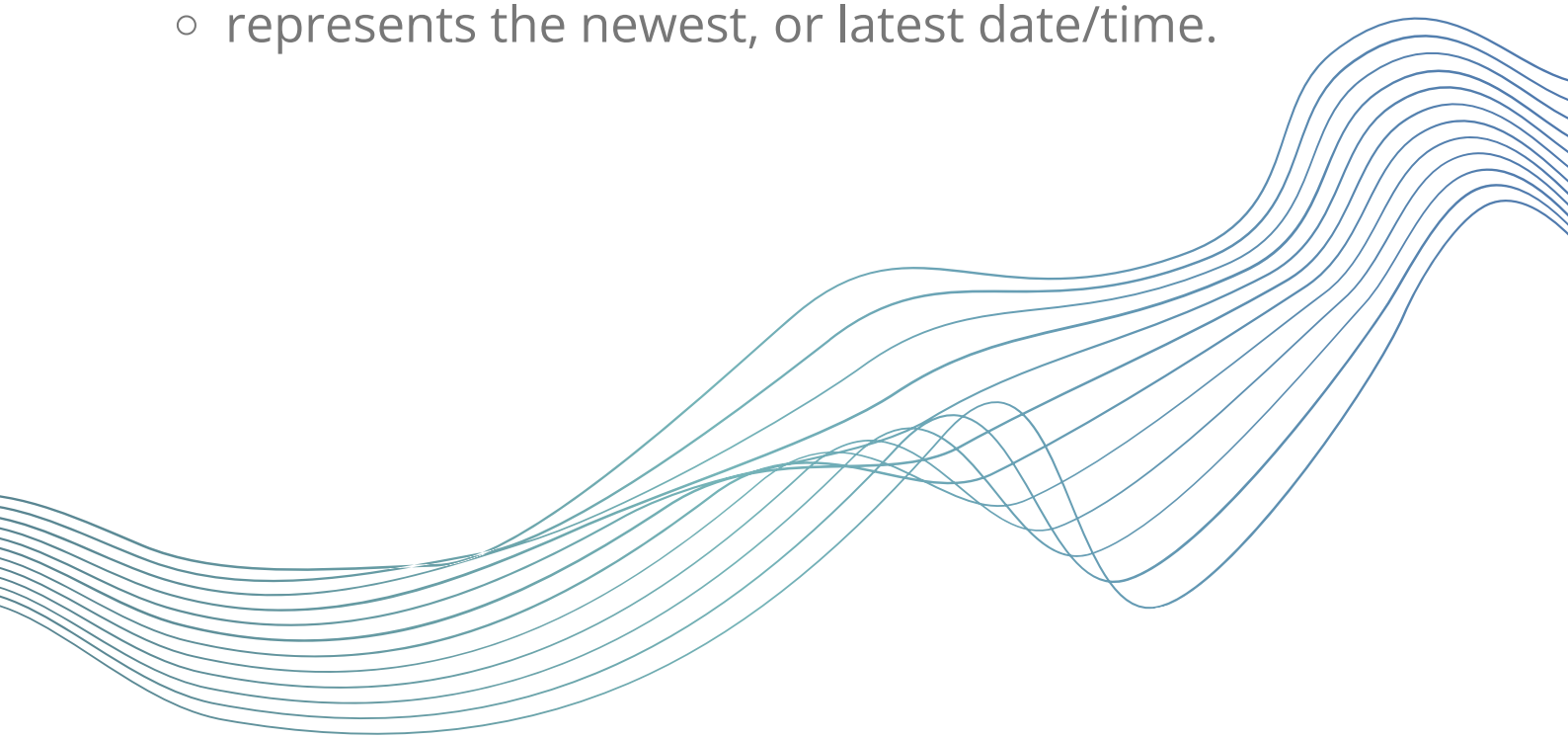
Incremental Refresh

Incremental refresh provide an efficient way to handle dynamic data and improve model refresh performance.

It reduces the amount of data that needs to be refreshed and allows for the inclusion of real-time data.

It is applicable to Fact tables in the semantic model, and it uses the following 2 parameters (*names are case sensitive*):

- **RangeStart:**
  - represents the oldest, or earliest date/time.
- **RangeEnd:**
  - represents the newest, or latest date/time.



05

SQL

01

SCD

Slowly Changing Dimensions

Framework for updating and maintaining data stored in dimension tables.

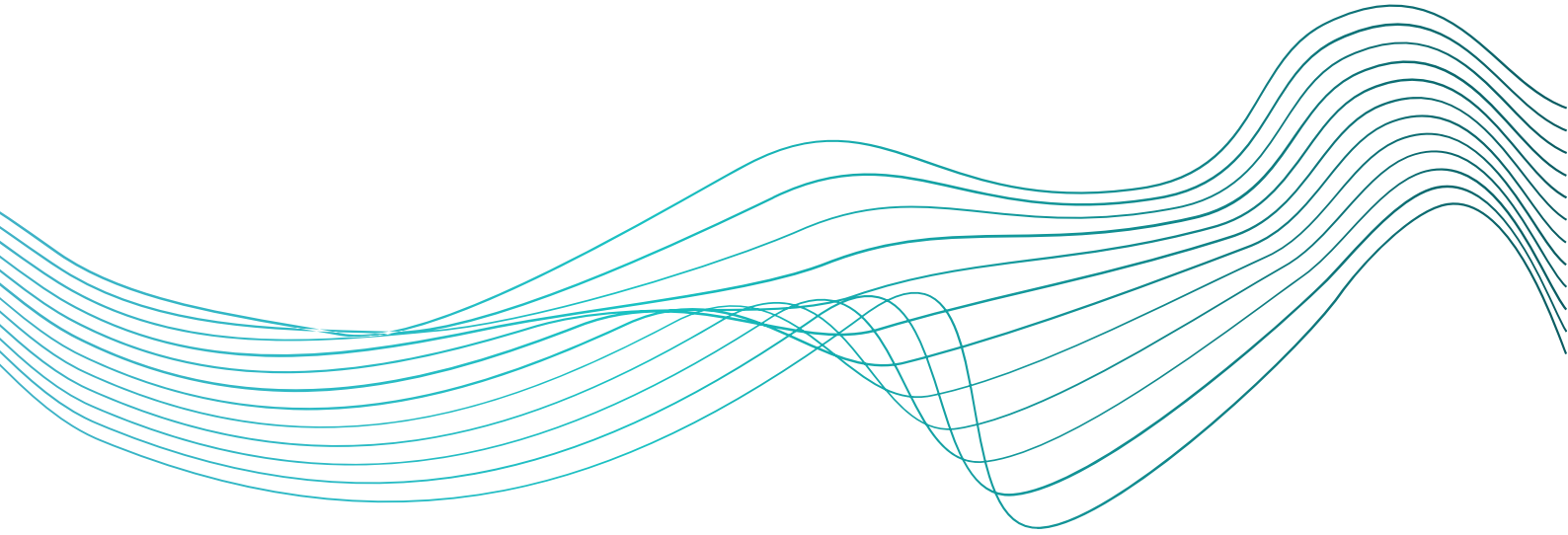
02

Functions

Some useful functions, that you need to know.

Slowly Changing Dimensions

Types	Notes
0	**Nothing changes**
1	**Overwrite data.** **No historicity on the table.**
2	**Maintains historicity. Adding a new record and also needs some columns to be defined, such as:** • **Valid from:** When does the new value start to apply? • **Valid to:** When does the new value stop to apply? • **IsCurrent:** Optional flag (0/1). Which row is the latest.



SQL Functions

LEAD

LEAD (column to see, offset >0)
It examines the next line and carries the value of the specified column to the previous line.

LAG

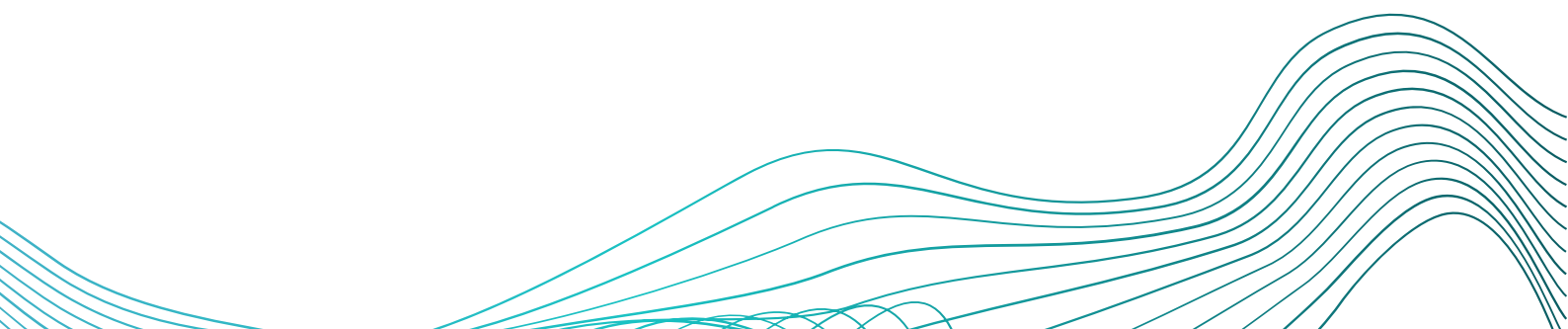
LAG (column to see, offset >0)
It examines the previous line and carries the value of the specified column to the next line.

LEAST

LEAST (column 1 [, ...column N])
It returns the minimum value from a list of one or more expressions.

GREATEST

GREATEST (column 1 [, ...column N])
It returns the maximum value from a list of one or more expressions.



STAY UPDATED

YouTube	The 1nf0rmant
Blog	emazarakis.github.io
Linked In	eugene-mazarakis