Data on Fire: A Hands-On Intro to Spark in Fabric

Ohio North Data User Group 2026

Jason
Romans

Cloud Data & Integration Developer

The DAX Shepherd

Microsoft® Most Valuable Professional (MVP)

@sql_jar

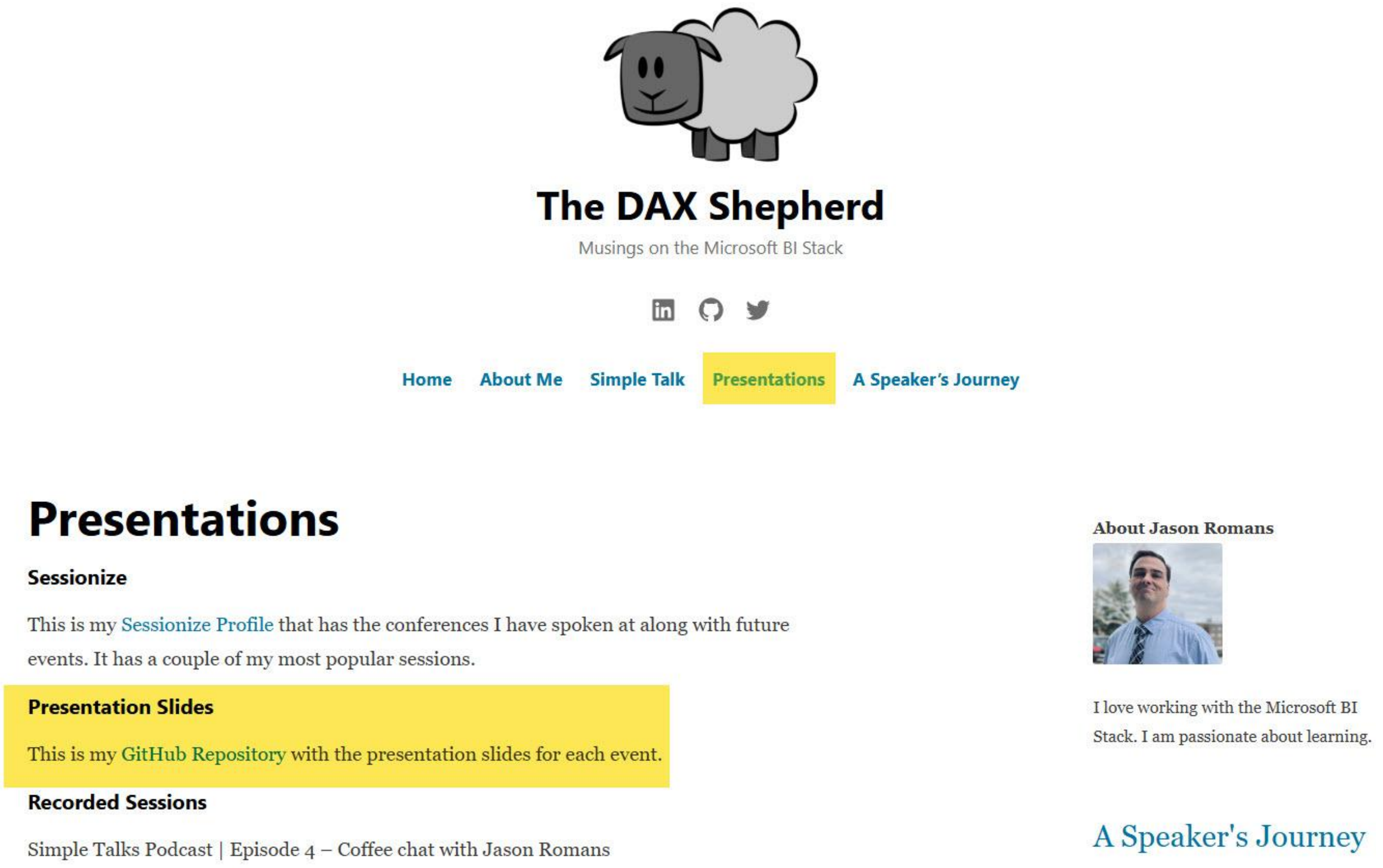
jason-r-sql-jar

https://thedaxshepherd.com/

- 📍 Nashville, TN, USA
- 🛠️ Began Career as a SQL Server DBA
- ➡️ Transitioned to Microsoft BI Stack
- 📦 Data Engineering to Data Modeling
- 🖊️ Infrequent Blogger
- 🧭 Fan of Dimensional Models & Doctor Who

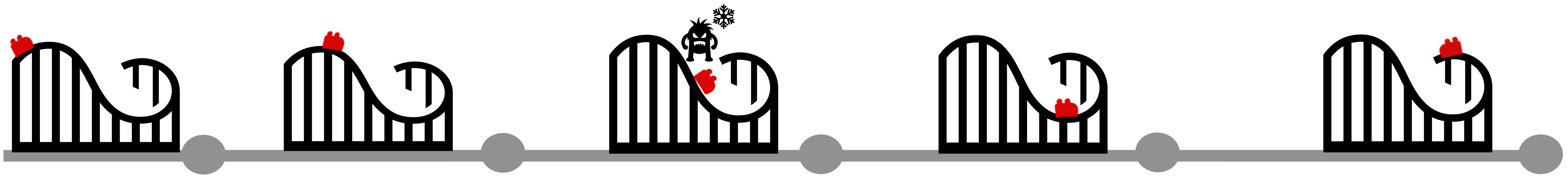
www.thedaxshepherd.com

Slides



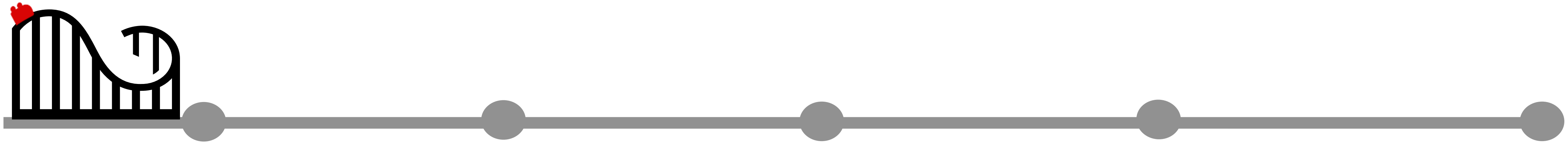
Shoulders of Giants

Our Journey



1. Intro
2. Python
3. PySpark
4. Uses
5. Conclusion

Our Journey



1. **Intro**
2. Python
3. PySpark
4. Uses
5. Conclusion

What lit the fire for Apache Spark

- The Netflix Prize
- Began Oct 2006
- Goal - improve Netflix's Cinematch algorithm by at least 10%
- Prize was 1 million dollars
- Took until 2009

Couldn't the Elephant* Help?

- Hadoop was not optimal for Machine Learning – multiple passes over disk
- Need for new tooling
- Shift to in-memory versus disk
- Like Analysis Services Multi-Dimensional to Tabular
- Contest must have led to work on Spark

* Hadoop's Mascot is an Elephant

Flashbacks of submitting homework digitally

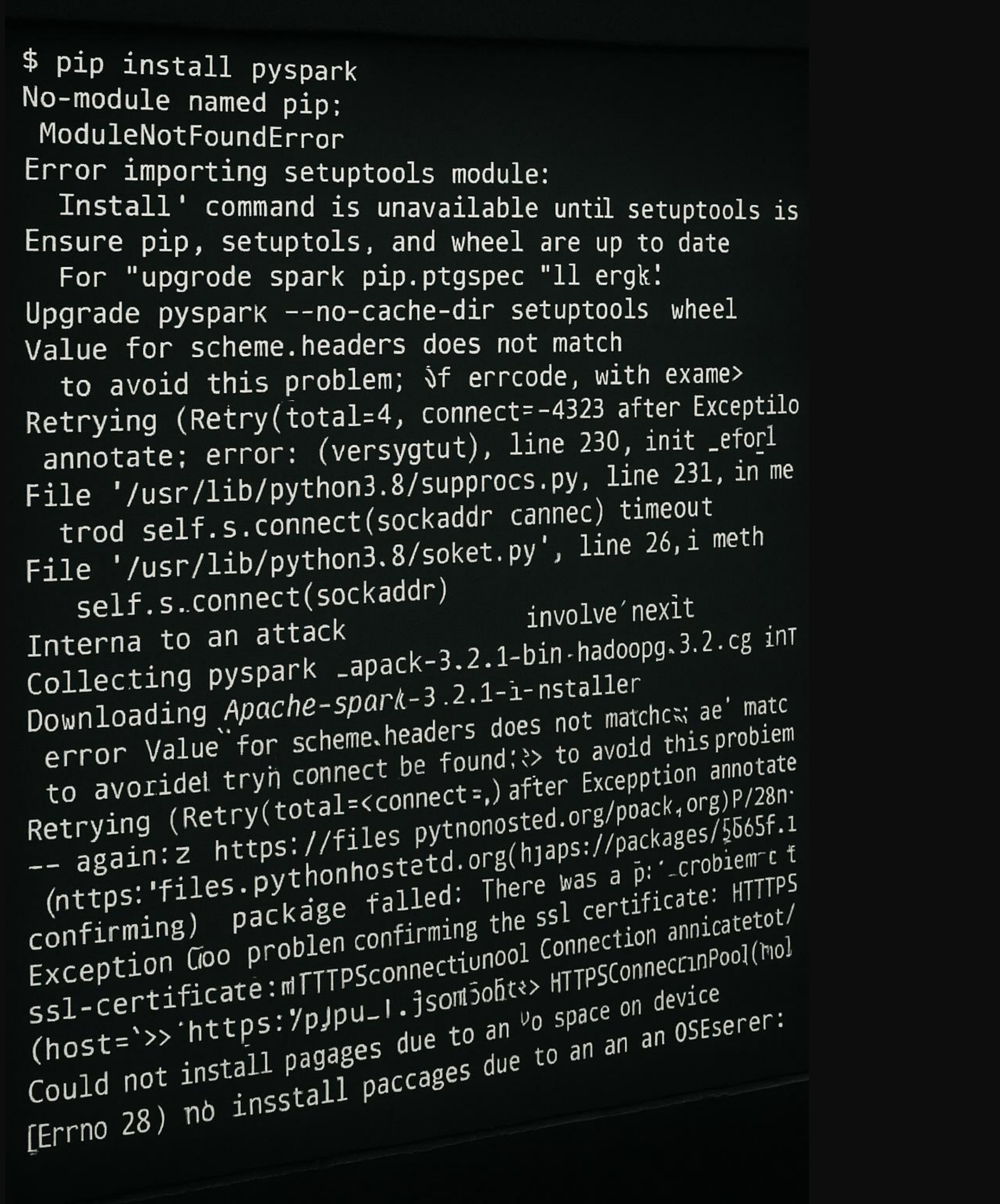
- Front runner BellKor’s Pragmatic Chaos
- Merger of teams from AT&A Labs and Commendo Research
- July 26, 2009 two teams met minimum requirements
- The Ensemble (Spark team) had a better improvement in score
- Lost by submitting 20 minutes later

What is Powered by Spark

- Apache Spark
- HDInsight
- Azure Synapse Analytics
- Databricks
- Microsoft Fabric

Now let us install Spark

Installing Spark
Step 1 of 42



Wait! Microsoft Fabric makes this easy

Notebooks in Microsoft Fabric

Can apply to other environments

Notebook Gallery

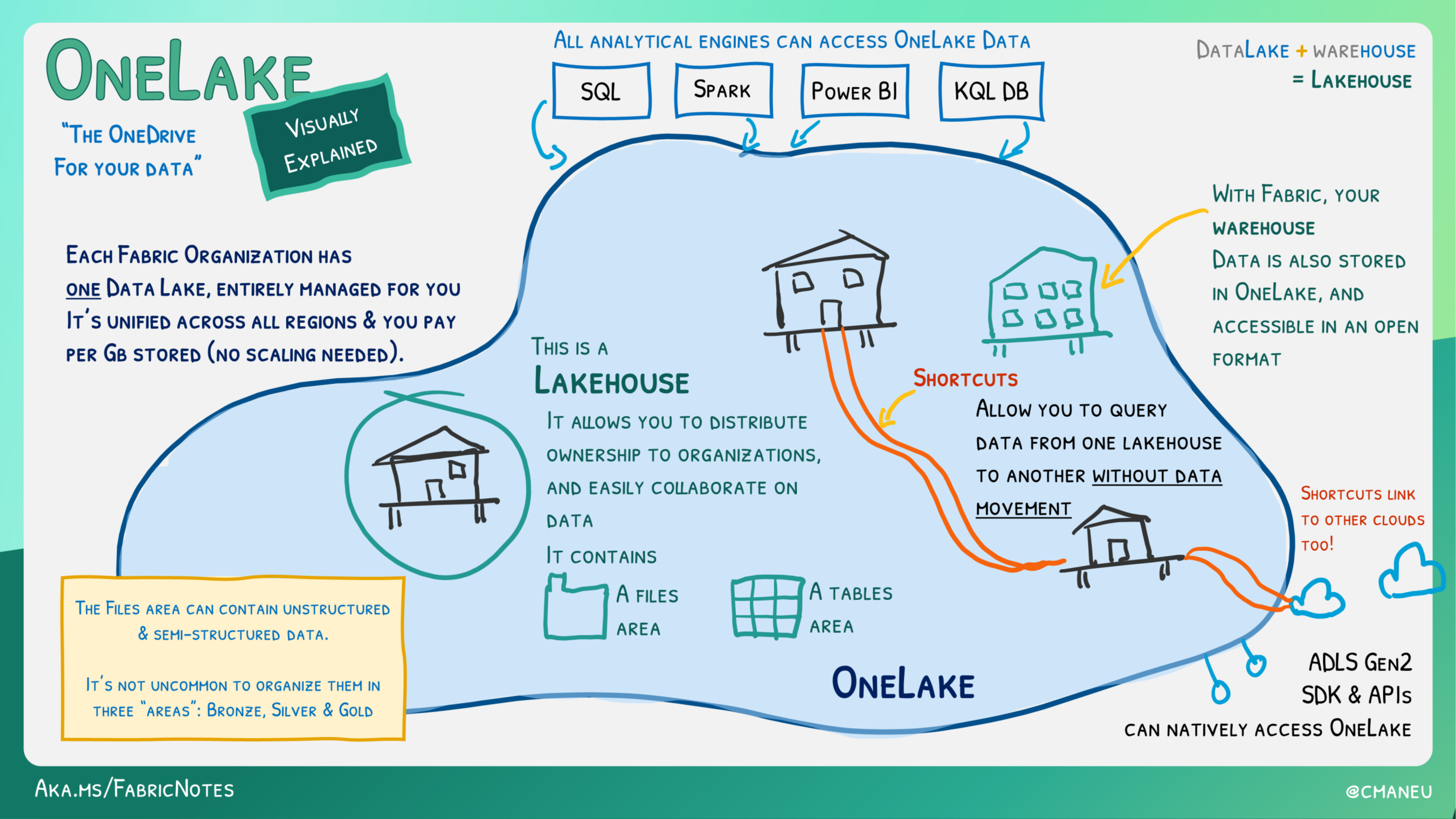
https://community.fabric.microsoft.com/t5/Notebook-Gallery/bd-p/pbi_notebookgallery

There was a notebook contest (it is closed now):

https://powerbi.microsoft.com/en-us/blog/introducing-the-first-ever-fabric-notebooks-competition-for-power-bi/

Microsoft Fabric Architecture

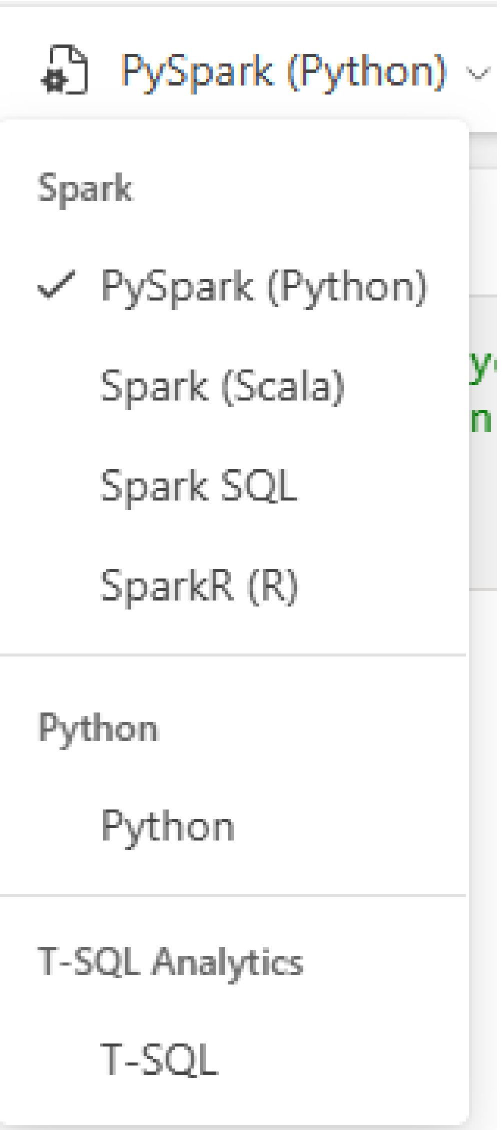
- Data is stored in OneLake


OneLake
"The OneDrive For your data"
Visually Explained
All analytical engines can access OneLake Data
SQL
Spark
Power BI
KQL DB
DataLake + warehouse = Lakehouse
Each Fabric Organization has one Data Lake, entirely managed for you
It's unified across all regions & you pay per Gb stored (no scaling needed).
This is a Lakehouse
It allows you to distribute ownership to organizations, and easily collaborate on data
It contains
A files area
A tables area
With Fabric, your warehouse Data is also stored in OneLake, and accessible in an open format
Shortcuts
Allow you to query data from one lakehouse to another without data movement
Shortcuts link to other clouds too!
The Files area can contain unstructured & semi-structured data.
It's not uncommon to organize them in three "areas": Bronze, Silver & Gold
OneLake
ADLS Gen2
SDK & APIs
can natively access OneLake
Aka.ms/FabricNotes
@CMANEU

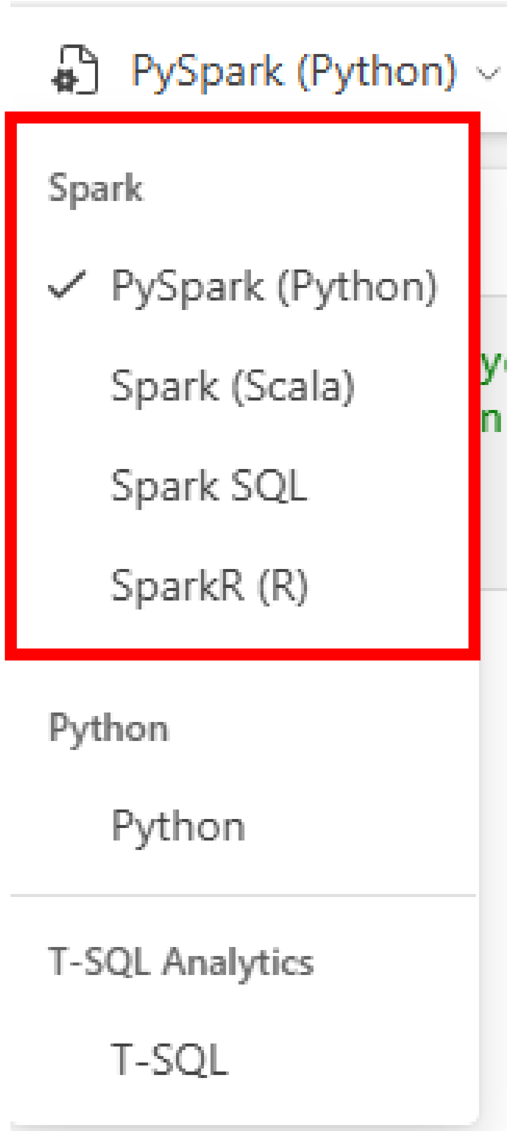
Microsoft Fabric Architecture

- Compute engines sitting on top of files
- Languages and compute
  - i.e. T-SQL with Warehouse

Compute & Language

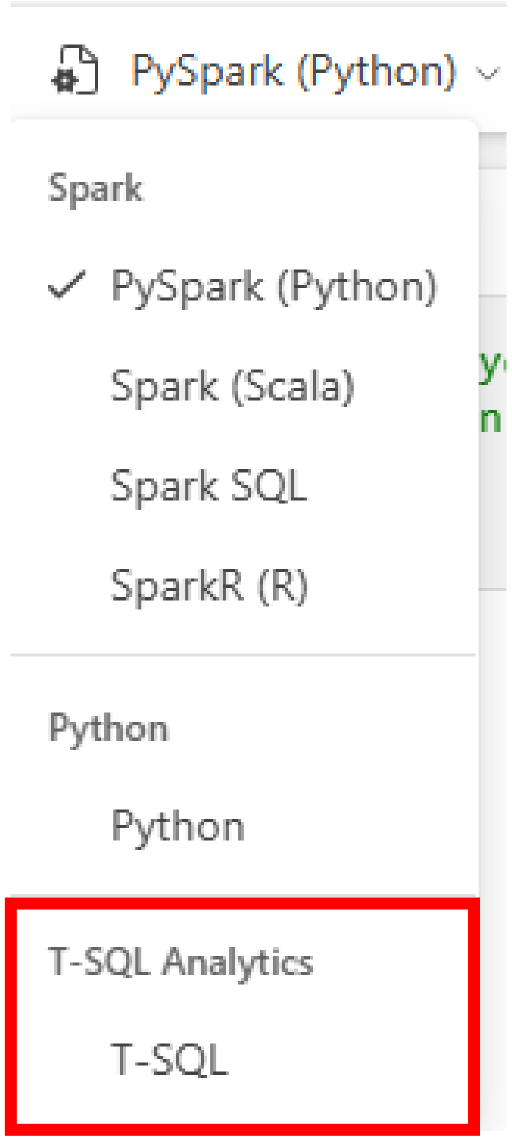


Spark
(Python, Scala, SQL, R)



Python (Python)

T-SQL Analytics (T-SQL)



Choosing PySpark or Python Compute (quick)

Scenario	Recommended Notebook
Includes pre-installed DuckDB and Polars libraries	Python Notebooks
Small to medium data (fits in memory)	Python Notebooks (or PySpark on single-node Spark cluster)
Rapid exploration & prototyping	Python Notebooks (or PySpark on single-node Spark cluster)
Large datasets (10GB+) exceeding memory	PySpark Notebooks
Complex data workflows or ETL pipelines	PySpark Notebooks
High-concurrency or parallel execution	PySpark Notebooks
Needs Spark-native APIs (MLlib, SQL, Streaming)	PySpark Notebooks

https://learn.microsoft.com/en-us/fabric/data-engineering/fabric-notebook-selection-guide

Choosing PySpark or Python Compute

Scenario	Python Notebooks (2-core VM)	PySpark Notebooks (Spark Compute)
Startup Time	The built-in starter pool initializes in approximately 5 seconds, while the on-demand pool takes around 3 minutes.	Start-up ranges from ~5 seconds (starter pool) to several minutes (on-demand Spark clusters).
Quick Transformations & API Calls	Ideal for small to medium sized datasets (up to 1GB)	Optimized for large datasets using vectorized execution.
Moderate Workloads	Not optimized for data sizes nearing memory saturation	Efficient at scaling via distributed compute.
Handling of Large Datasets	Limited by single-node memory. May struggle with scaling.	Distributed processing ensures scalable handling of multi-GB to TB workloads.
High-Concurrency Execution	Manual FIFO-style parallelism per notebook	System-managed concurrency with support for parallel execution.
Resource Customization & Scaling	Fixed compute (2-core VM); does not auto scale. Users can manually scale out using %%config within the notebook.	Flexible resource allocation; supports autoscaling and custom Spark configurations.

https://learn.microsoft.com/en-us/fabric/data-engineering/fabric-notebook-selection-guide

Type of Compute for Notebooks

- Spark Based
  - Cluster
- Single Node Python
  - 2 vCores, 16G RAM
- T-SQL Analytics
  - Warehouse

Python Notebook

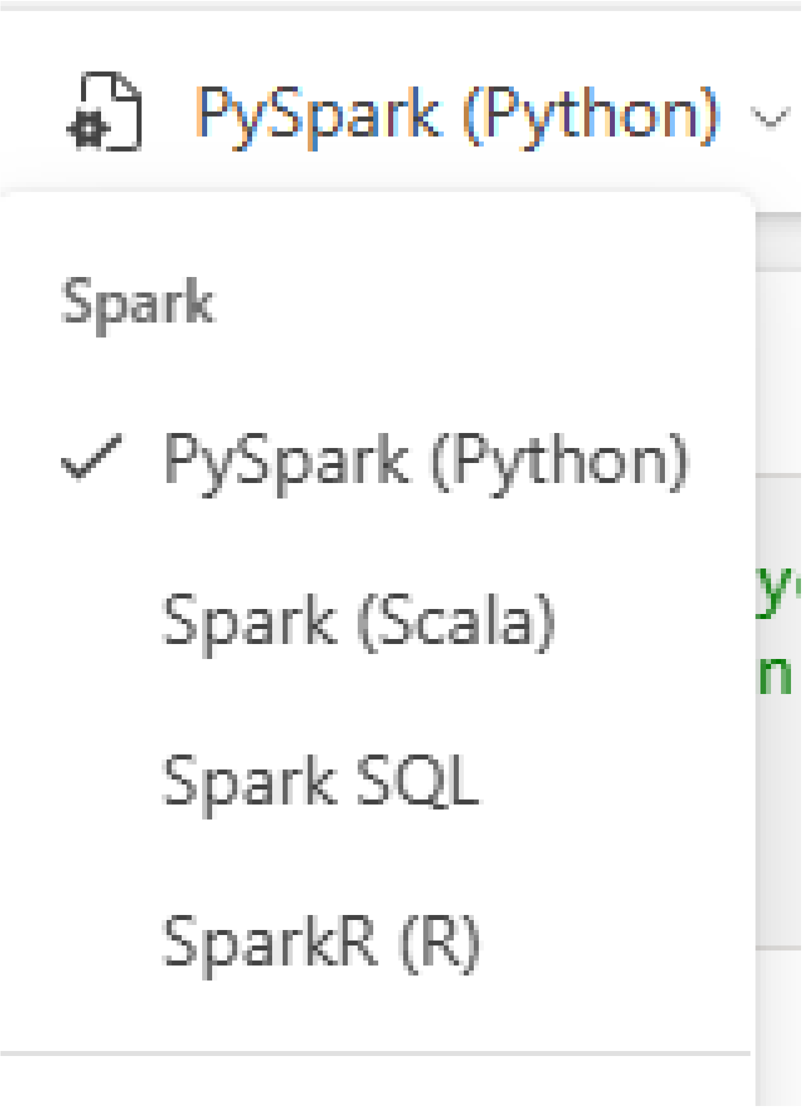
- Has libraries installed for dealing with “small-big” data
  - Less than 10 Gigabytes
  - Fits in memory
- Example Libraries installed
  - Polars
  - DuckDB

Languages for Spark

Different choices of languages

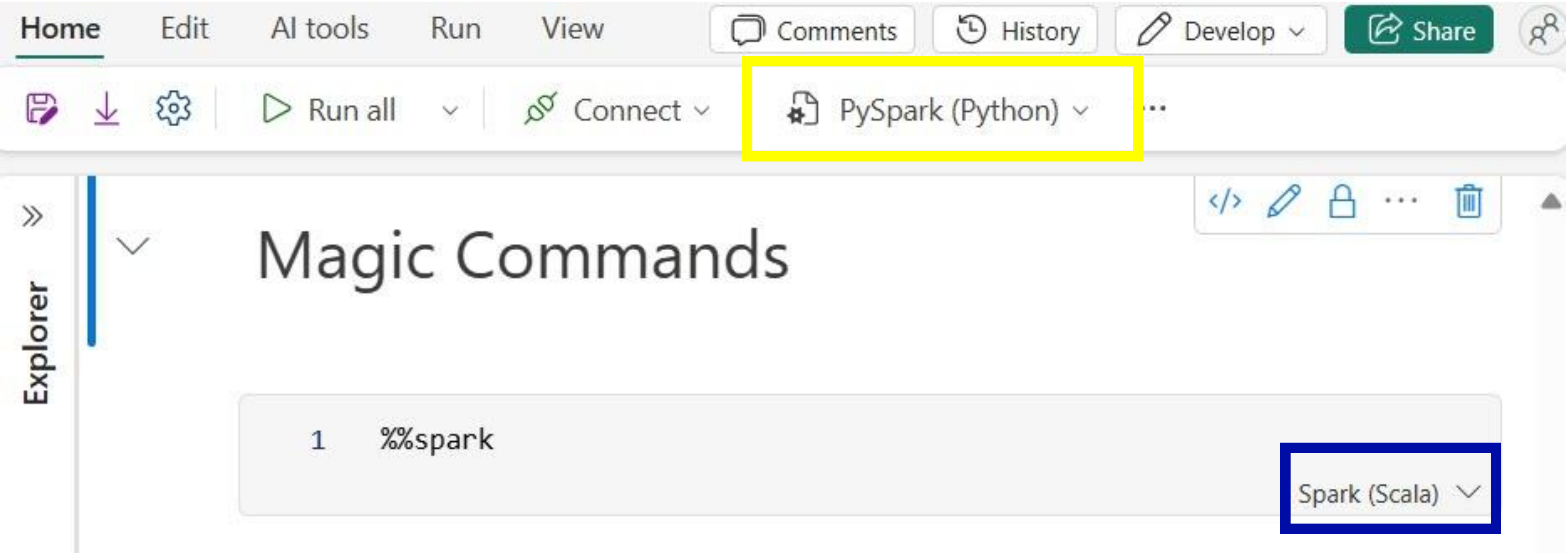
Built with Scala

- PySpark (Python)
- Spark (Scala)
- Spark SQL
- SparkR (R)



Magic Commands – set language by cell

Magic command	Language	Description
%%pyspark	Python	Execute a **Python** query against Apache Spark Context.
%%spark	Scala	Execute a **Scala** query against Apache Spark Context.
%%sql	SparkSQL	Execute a **SparkSQL** query against Apache Spark Context.
%%html	Html	Execute n **HTML** query against Apache Spark Context.
%%sparkr	R	Execute a **R** query against Apache Spark Context.


Home
Edit
AI tools
Run
View
Comments
History
Develop
Share
Run all
Connect
PySpark (Python)
Explorer
Magic Commands
1 %%spark
Spark (Scala)

Our Journey

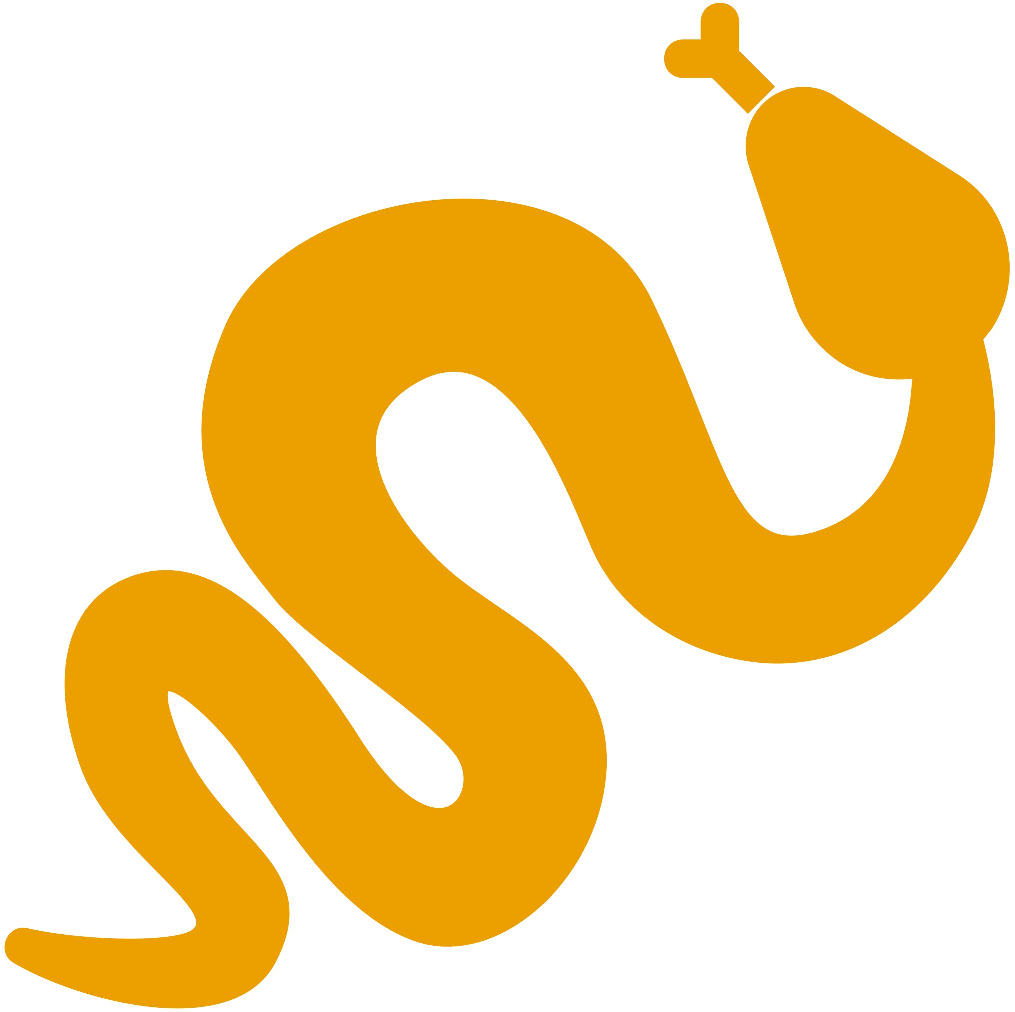
1. Intro
2. **Python**
3. PySpark
4. Uses
5. Conclusion

Python Language

HOW YOU INTERACT WITH SPARK

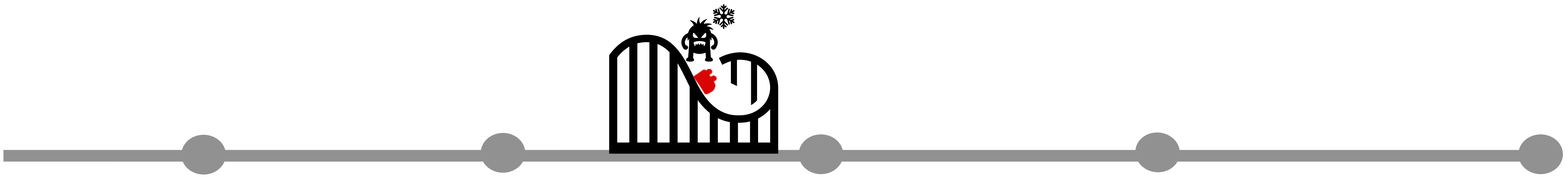
HOW YOU MANIPULATE THE DATA

Python Demo



***You can thank me for not having an actual picture of a snake here*

Our Journey



1. Intro
2. Python
3. **PySpark**
4. Other Uses
5. Conclusion

PySpark

Python API for Spark

Most operations on a DataFrame

Like Pandas but distributed

DataFrame

- Resilient Distributed Dataset (RDD)
- Conceptually the same as a table
  - Abstraction
  - Rows
  - Columns


Spark is
Lazy – in a
good way

Lazy Evaluation

- Waits until an action is requested
- Actions
  - Counting number of rows in a Spark DataFrame
  - Showing output
  - Writing data to a file or data source
  - Transferring data from a Spark DataFrame to a native object in Python

Benefits of Lazy Evaluation

- Saves resources
- Plan can be optimized

Pandas (non-spark, historic) is eager evaluation

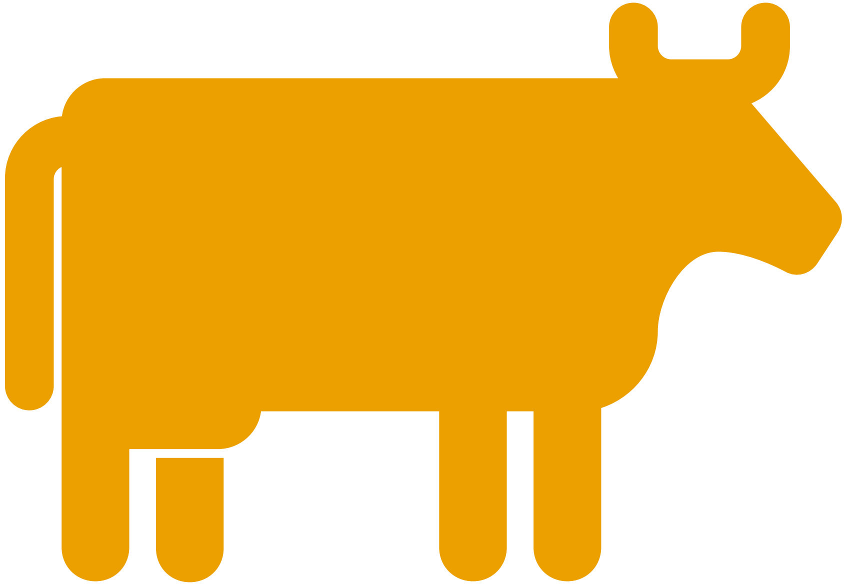
Fabric in Visual Studio Code

EDIT NOTEBOOKS

CONNECT TO COMPUTE IN MICROSOFT FABRIC

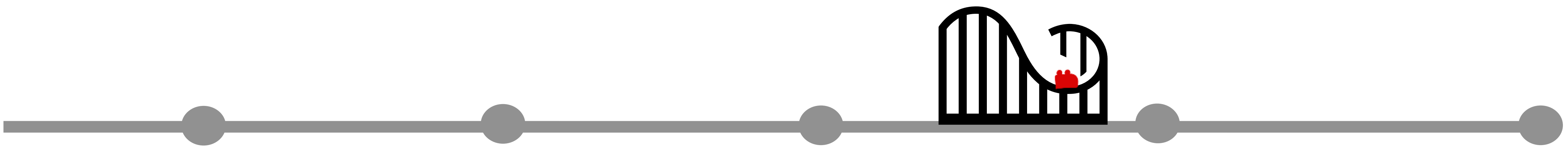
Data Wrangler

Like Power Query, but for PySpark and Python



PySpark Demo

Our Journey



1. Intro
2. Python
3. PySpark
4. **Other Uses**
5. Conclusion

Uses

- **Big Data Processing** - handling massive datasets
- **Machine Learning at Scale** - training ML models on large data
- **Real-Time Stream Processing** - processing live data streams
- **ETL & Data Pipelines** - building data workflows
- **Advanced Analytics** - complex calculations and BI
- **Data Exploration & Discovery** - interactive data analysis

Executing Notebook using a Pipeline

- Either
  1. Use Environment with Libraries
     - Microsoft recommended way
     - Only for Spark
  2. Python inline installation
     - Enable %pip install for pipeline, add "_inlineInstallationEnabled" as bool parameter equals True in the notebook activity parameters.

Reference:

- https://learn.microsoft.com/en-us/fabric/data-engineering/library-management

Semantic Link &

Semantic Link Labs

Semantic Link Labs

Microsoft Fabric

Semantic Link

List Tables

List Workspaces

List Models

List Reports

Semantic Link Labs

Migrate to Direct Lake

Best Practice Analyzer

Manage Reports

View Broken Reports

Rebind Reports

Semantic Link (Labs) Demo

Resources

Fabric Samples

- https://github.com/microsoft/fabric-samples

- Semantic Link
  - https://learn.microsoft.com/en-us/fabric/data-science/semantic-link-overview
- Semantic Link Labs
  - https://github.com/microsoft/semantic-link-labs

Resources

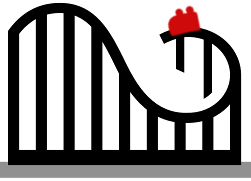
PySpark Book – Data Analysis with Python and PySpark

- https://www.oreilly.com/library/view/data-analysis-with/9781617297205/

PySpark Book – Intro to PySpark (Free HTML version)

- https://pedropark99.github.io/Introd-pyspark/

Our Journey



1. Intro
2. Python
3. PySpark
4. Other Uses
5. **Conclusion**

Conclusion

Powerful and Scalable Platform

Apache Spark in Microsoft Fabric Notebooks provides a robust and scalable solution for handling big data analytics tasks efficiently.

User-Friendly Tools

The platform offers intuitive and practical tools that simplify data analysis for professionals of all skill levels.

Unlocking Valuable Insights

Understanding core concepts and leveraging these tools enables data professionals to extract meaningful insights effectively.

Thank you

Jason Romans
thedaxshepherd@gmail.com
www.thedaxshepherd.com

The Dax Shepherd