Nawatech

Beyond Engineering

nawatech.co

KAI Workshop– Day 1

Prepared for: PT Kereta Api Indonesia

Copyright Nawatech 2024

Explore end-to-end analytics with Microsoft Fabric

Microsoft Fabric

Data Integration	Data Engineering	Data Warehouse	Data Science	Real Time Analytics	Business Intelligence	Observability
Data Factory	Synapse	Synapse	Synapse	Synapse	Power BI	Data Activator

- **Data Factory**: Data integration combining Power Query with the scale of Azure Data Factory to move and transform data.
- **Synapse Data Engineering**: Data engineering with a Spark platform for data transformation at scale.
- **Synapse Data Warehouse**: Data warehousing with industry-leading SQL performance and scale to support data use.
- **Synapse Data Science**: Data science with Azure Machine Learning and Spark for model training and execution tracking in a scalable environment.
- **Synapse Real-Time Analytics**: Real-time analytics to query and analyze large volumes of data in real-time.
- **Power BI**: Business intelligence for translating data to decisions.
- **Data Activator**: Real-time detection and monitoring of data that can trigger notifications and actions when it finds specified patterns in data.

Copyright Nawatech 2024

Data teams and Fabric

Fabric's unified management and governance make it easier for data professionals to work together.

Data Engineers	Analytics Engineers	Data Scientists	Data Analysts	Decision Makers
Data Factory Data Warehouse Data Engineering Real-time analytics	Data Factory Power BI Data Engineering Data Warehouse	Data Science Azure ML	Power BI	Power BI Microsoft 365

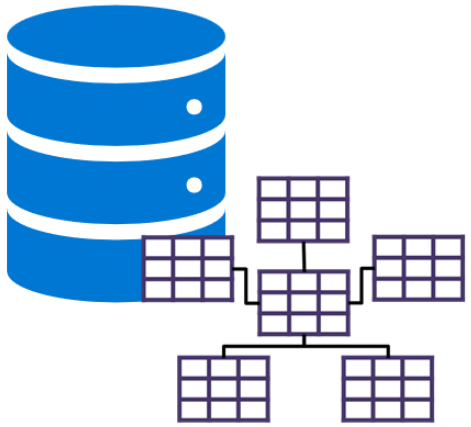
Data Stewards

Copyright Nawatech 2024

Get started with lakehouses in Microsoft Fabric

What is a lakehouse?

+



Data *Lake*

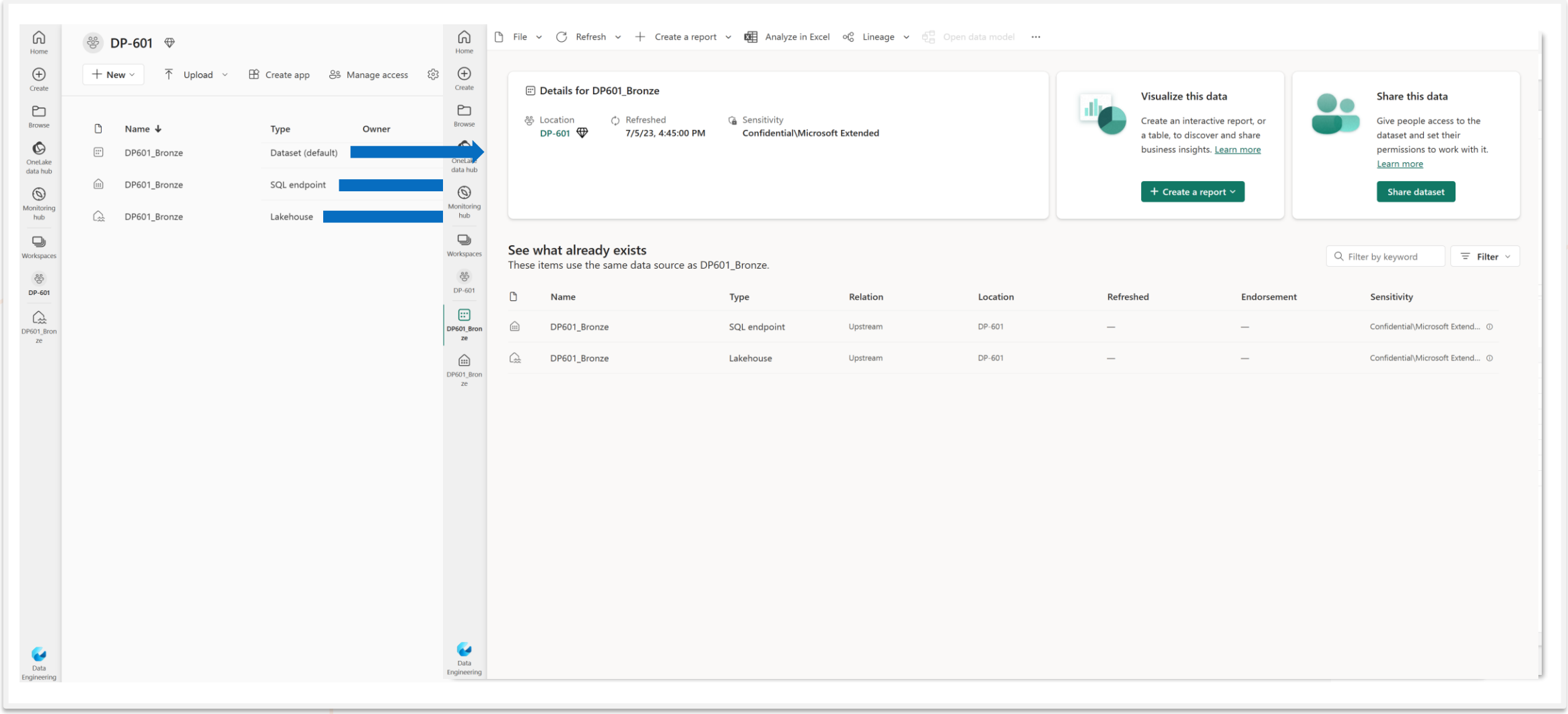
- Scalable, distributed file storage
- Flexible schema-on-read semantics
- Big data technology compatibility

Data Ware*house*

- Relational schema modeling
- SQL-based querying
- Proven basis for reporting and analytics

Copyright Nawatech 2024

Work with a Fabric lakehouse



Copyright Nawatech 2024

Explore, transform, and visualize data in the lakehouse

Tools and techniques to explore and transform data:

- Apache Spark
  - Notebooks
  - Spark Job Definitions
- SQL analytics endpoint
- Dataflows Gen2
- Data Pipelines

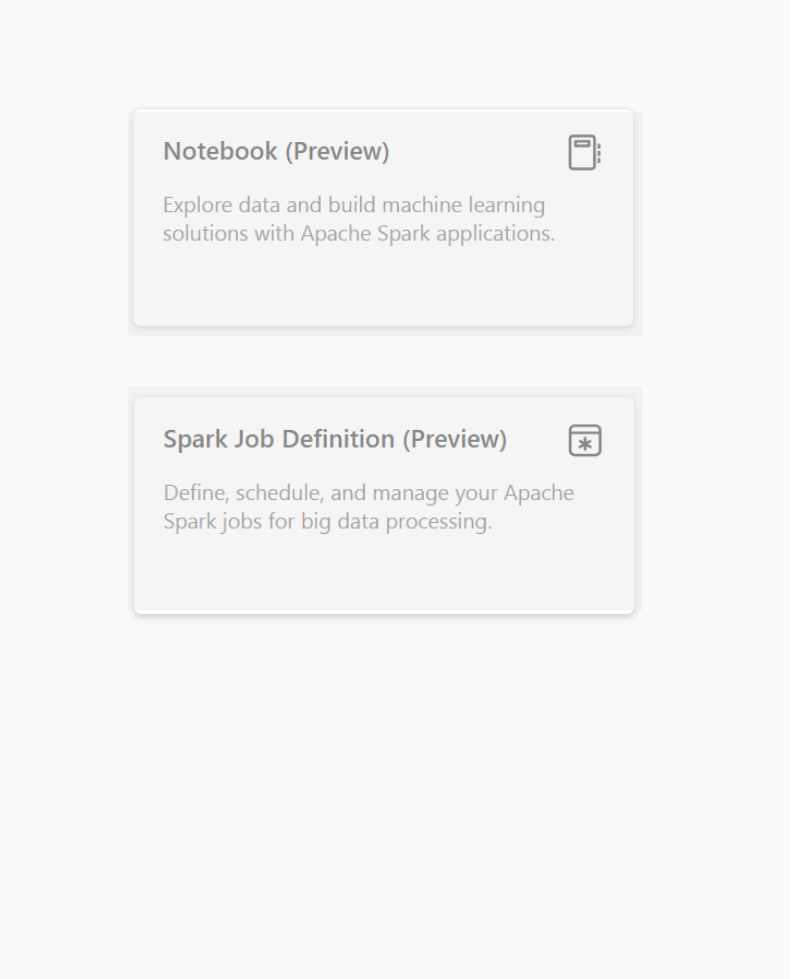
Visualize lakehouse data using Power BI

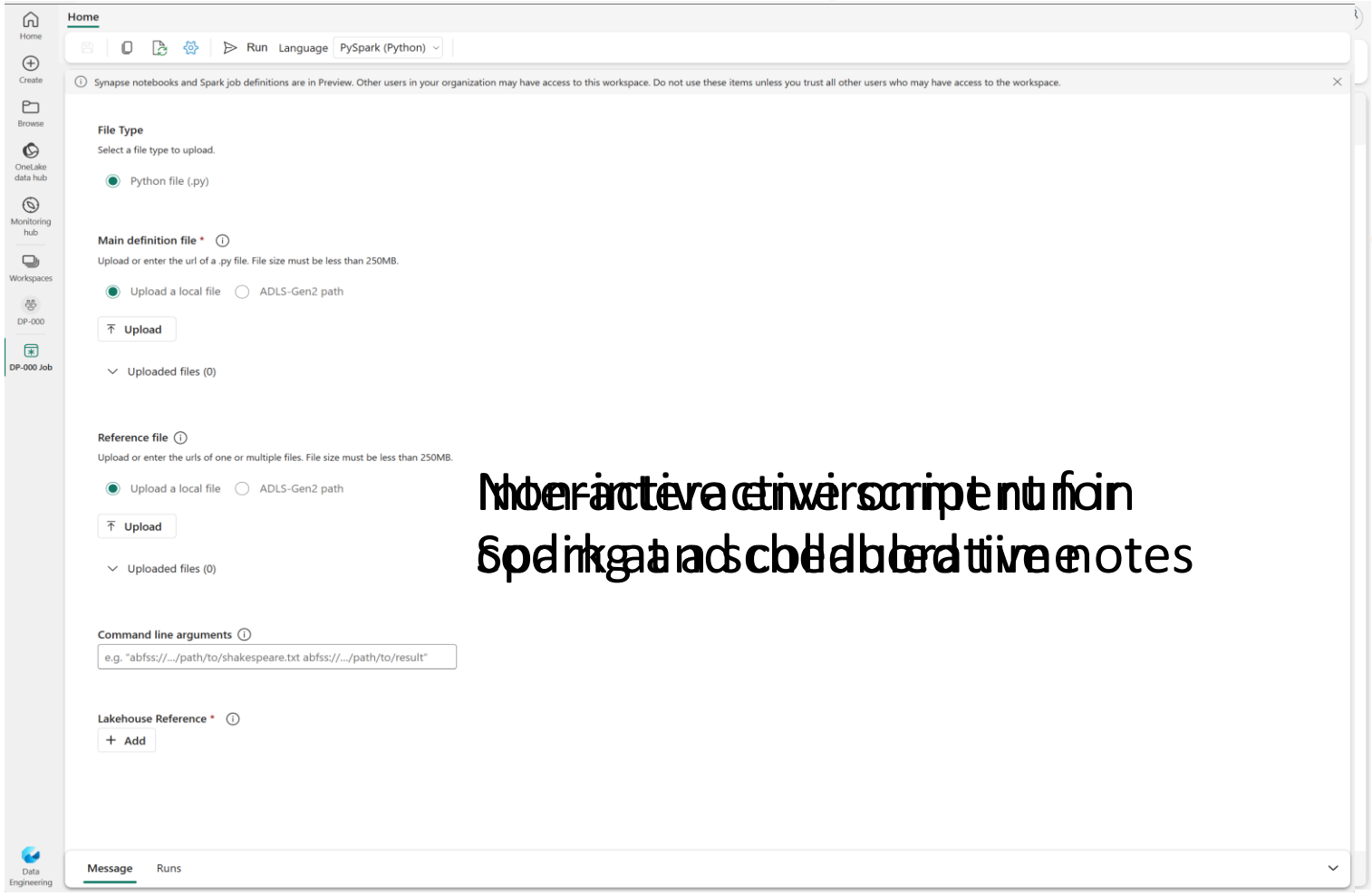
Copyright Nawatech 2024

Use Apache Spark in Microsoft Fabric

Run Spark in Fabric

- To edit and run Spark code in Fabric, use *notebooks* or create a *Spark Job Definition*





Interactive environment for Spark and collaborative notes

Run a script or job

Load data in a Spark Dataframe

- Schema can be either inferred or specified

Home Edit Run Data View

Run all | Stop session | Language | PySpark (Python) | Open in VS Code

Editing | Comments | Share

Lakehouse explorer

DP601_Bronze

Tables
- Re
- WWI_Products

Files
- Data

Files > Data

products.csv

Synapse notebooks and Spark job definitions are in Preview. Other users in your organization may have access to this workspace. Do not use these items unless you trust all other users who may have access to the workspace.

```python
from pyspark.sql.types import *
from pyspark.sql.functions import *

productSchema = StructType([
    StructField("ProductID", IntegerType()),
    StructField("ProductName", StringType()),
    StructField("Category", StringType()),
    StructField("ListPrice", FloatType())
    ])

df = spark.read.load('Files/data/product-data.csv',
    format='csv',
    schema=productSchema,
    header=False)
display(df.limit(3))
```

1 sec -Command executed in 950 ms by Shannon Lindsay (SHE/HER) on 1:22:24 PM, 7/06/23

Specify a schema

Spark jobs (1 of 1 succeeded) | Log

Table | Chart | Export results

Index	ProductID	ProductName	Category	ListPrice
1	771	Mountain-100 Silver, 38	Mountain Bikes	3399.99
2	772	Mountain-100 Silver, 42	Mountain Bikes	3399.99
3	773	Mountain-100 Silver, 44	Mountain Bikes	3399.99

Session ready | Save option: Automatic | Selected 1 of 4 cells

Copyright Nawatech 2024

Transform data in a Spark Dataframe

Home Edit Run Data View

Run all | Stop session | Language PySpark (Python) | Open in VS Code

Lakehouse explorer

Lakehouse000

Tables

salesorders

Files

orders.csv

Synapse notebooks and Spark job definitions are in Preview. Other users in your organization may have access to this space. Do not use these items unless you

```
1  bikes_df.write.mode("overwrite").parquet('Files/product_data/bikes.parquet')
```

[29] 2 sec -Command executed in 1 sec 754 ms by Shannon Lindsay (SHE/HER) on 12:16:38 AM, 7/11/23

Spark jobs (1 of 1 succeeded) Log

ID	Description	Status	Stages	Tasks	Duration
Job 30	parquet at NativeMethodAccessorImpl.java:0	Succeeded	1/1	1/1 succeeded	1 sec

Session ready | Save option: Automatic | 1 | Selected 1 of 2 cells

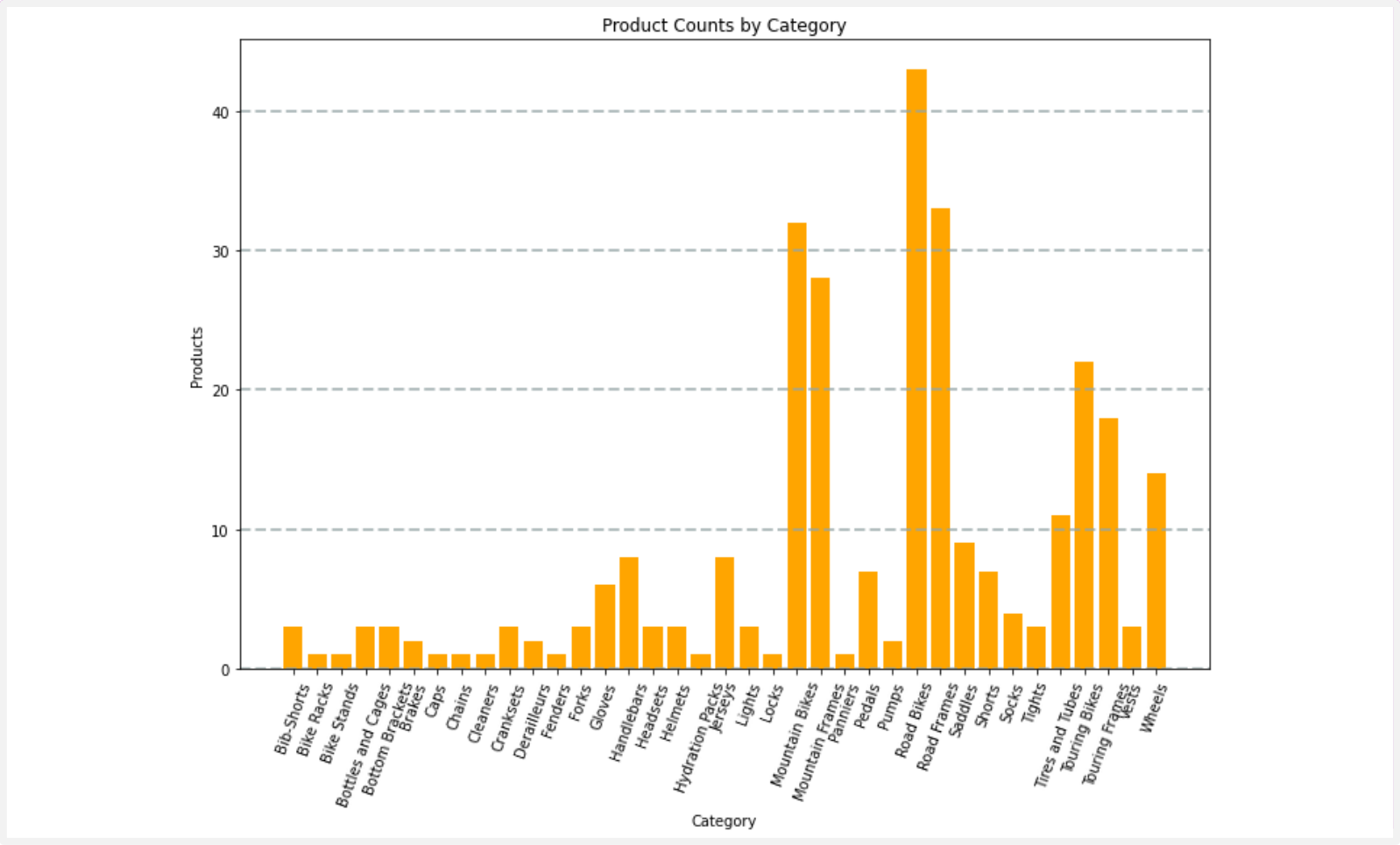
Save dataframe for further analysis

Copyright Nawatech 2024

Visualize data

- Two ways to visualize data in notebooks:

1. Use built-in notebook charts
2. Use graphics packages in code
   - This example uses Matplotlib



Copyright Nawatech 2024

Hands-on: Apache Spark

Access GitHub : https://s.id/qogLM

Run Spark

Load Data

Transform Data

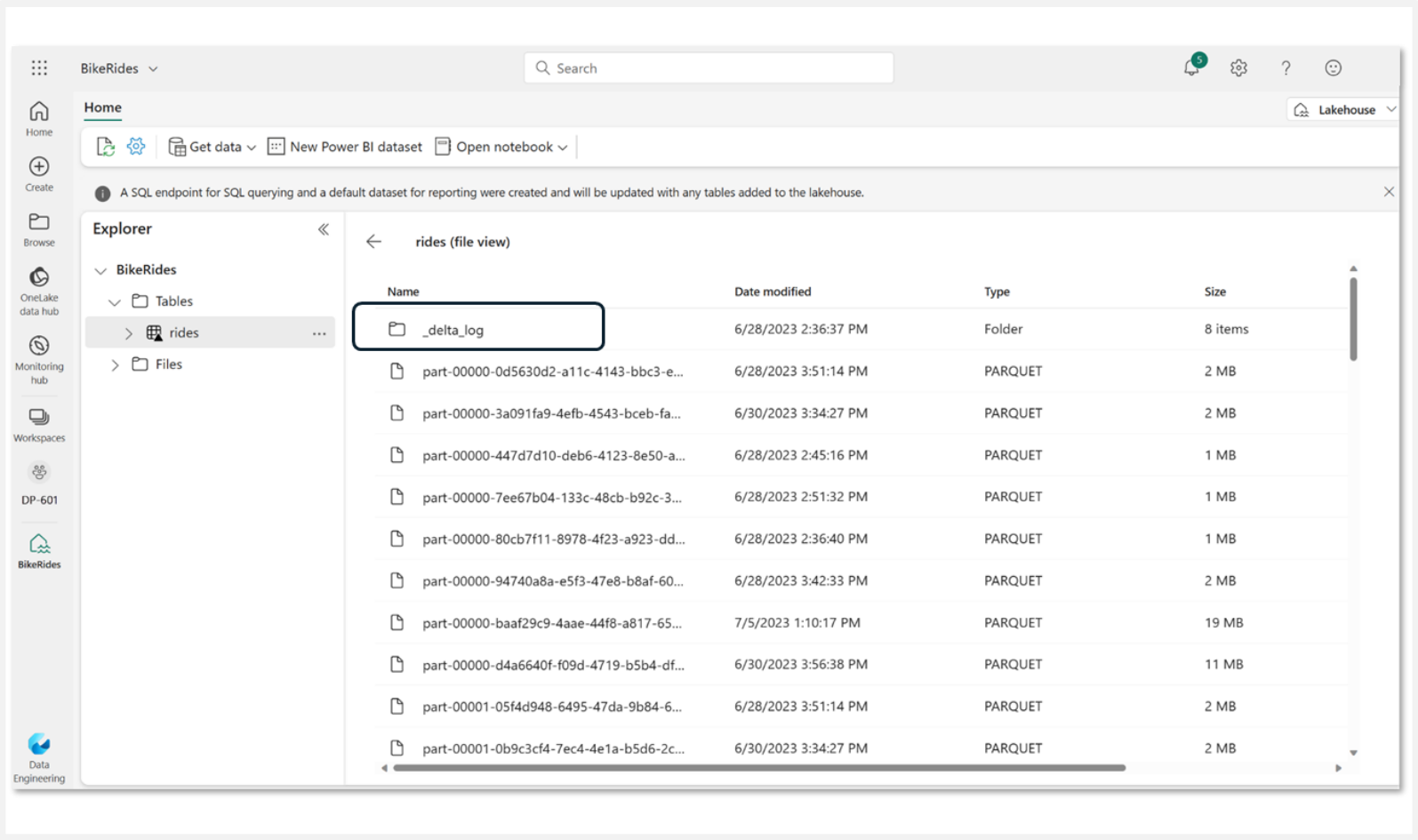
Visualize Data

Copyright Nawatech 2024

Work with Delta Lake tables in Microsoft Fabric

Understand Delta Lake

- Relational tables that support querying and data modification
- Support for ACID transactions
- Data versioning and time travel
- Standard formats and interoperability



Copyright Nawatech 2024

Create Delta tables using code in Spark

1. Save a dataframe as a managed table

```
# Load a file into a dataframe
df = spark.read.load('Files/mydata.csv', format='csv', header=True)

# Save the dataframe as a delta table
df.write.format("delta").saveAsTable("mytable")
```

2. Use Spark SQL

```
%%sql

CREATE TABLE salesorders
(
    Orderid INT NOT NULL,
    OrderDate TIMESTAMP NOT NULL,
    CustomerName STRING,
    SalesTotal FLOAT NOT NULL
)
USING DELTA
```

3. Save a dataframe in delta format in an explicit path

```
delta_path = "Files/mydatatable"
df.write.format("delta").save(delta_path)
```

Copyright Nawatech 2024

Managed vs external tables

Managed tables

- Defined without a specific location – Files are created in the default metastore folder (***Tables/…***)
- Dropping the table deletes the files

```
# Save a dataframe as a delta table
df.write.format("delta").saveAsTable("mytable")
```

External tables

- Defined with an explicit file location outside of the default metastore folder
- Dropping the table does not delete the files

```
df.write.format("delta").saveAsTable("myexternaltable",path="Files/myexternaltable")
```

Copyright Nawatech 2024

Work with Delta tables in Spark

1. Use Spark SQL to embed a SQL statement in PySpark
   - Embed SQL statements in other languages using the **spark.sql** library.

```
spark.sql("INSERT INTO products VALUES (1, 'Widget', 'Accessories', 2.99)")
```

2. Native Spark SQL using %%sql magic
   - Use %%sql magic in a notebook to run SQL statements.

```
%%sql

UPDATE products
SET Price = 2.49 WHERE ProductId = 1;
```

3. Use the Delta API
   - Create an instance of a DeltaTable from a folder location containing files in delta format, and then use the API to modify the data in the table.

```
from delta.tables import *
from pyspark.sql.functions import *

# Create a DeltaTable object
delta_path = "Files/mytable"
deltaTable = DeltaTable.forPath(spark, delta_path)

# Update the table (reduce price of accessories by 10%)
deltaTable.update(
    condition = "Category == 'Accessories'",
    set = { "Price": "Price * 0.9" })
```

Copyright Nawatech 2024

Hands-on: Delta Lake Tables

Access GitHub : https://s.id/qogLM

Create Delta Table in Spark

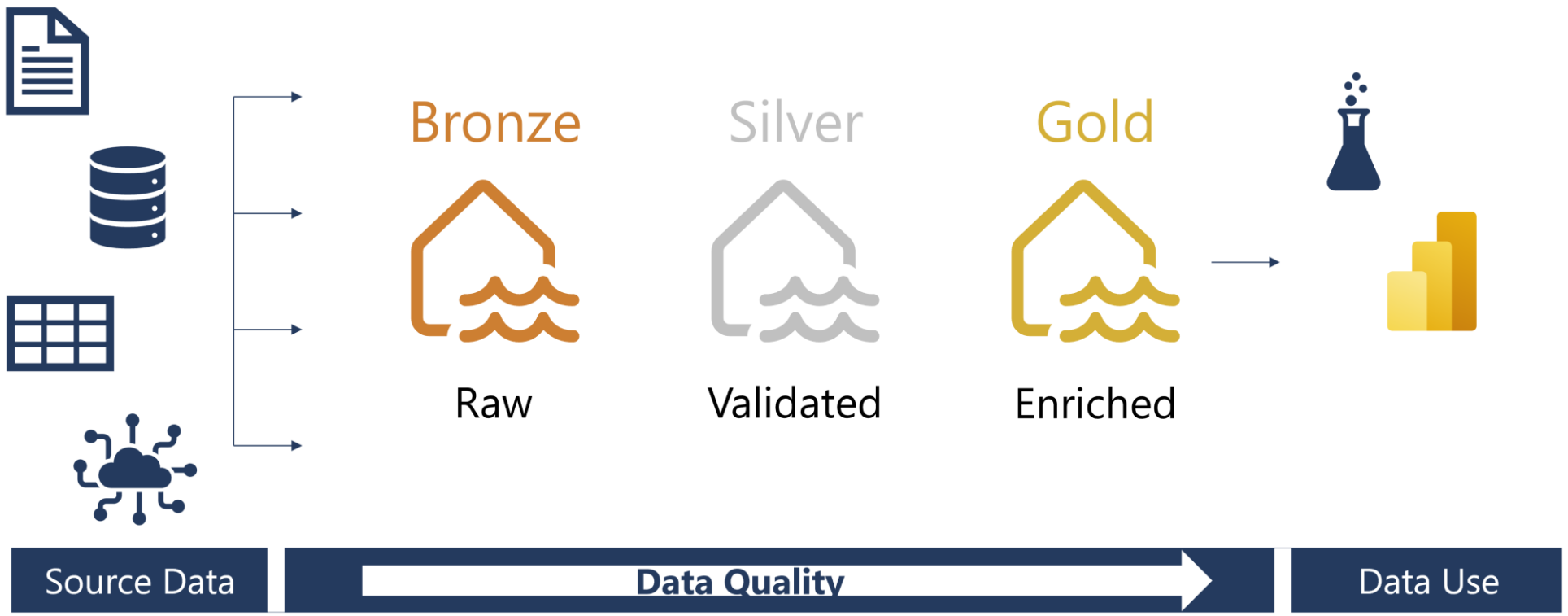
Save as Managed and External Tables

Work with Delta Table in Spark

Copyright Nawatech 2024

Organize a lakehouse with medallion architecture

Medallion architecture overview

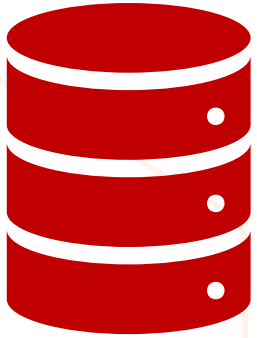


Copyright Nawatech 2024

Implement a medallion architecture

- Bronze

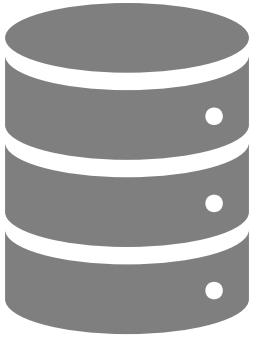
Ingest raw data



Pipelines, dataflows, notebooks

- Silver

Cleanse and validate data



Dataflows or notebooks

- Gold

Additional transformations and modeling

SQL analytics endpoint or semantic model

Copyright Nawatech 2024

Query and report on data in your Fabric lakehouse

Sales (1)

Search

Fabric Trial: 59 days left

Home

SQL endpoint

New SQL query | New visual query

A default dataset for faster reporting was created and will be automatically updated with any tables and views added to the lakehouse. Learn more

Manage default Power BI dataset

Explorer

Warehouses

- Sales
  - Schemas
    - dbo
      - Tables
        - dimcustomer...
        - dimdate_gold
        - dimproduct_...
        - factsales_gold
        - sales_silver
      - Views
      - Functions
      - Stored Procedur...
    - guest
    - INFORMATION_SCHE...
    - sys
  - Security
- Queries
  - My queries
    - SQL query 1
    - Visual query 1
  - Shared queries

Visual query 1 | SQL query 1

Run | Save as view

```
select CustomerName, SUM(Quantity) AS TotalQuantity
FROM [sales_silver]
GROUP BY CustomerName
ORDER BY TotalQuantity DESC
```

Messages | Results | Download Excel file | Visualize results

Limited to 10,000 rows

Search

	CustomerName	TotalQuantity
1	Samantha Jenkins	41
2	Henry Garcia	39
3	Charles Jackson	38
4	April Shan	35
5	Hailey Patterson	34
6	Ryan Thompson	34
7	Mason Roberts	33
8	Dalton Perez	32
9	Jason Griffin	31
10	Fernando Barnes	28
11	Jasmine Powell	27
12	Nicholas Brown	25
13	Ashley Henderson	25
14	Jennifer Simmons	24
15	Gina Martin	24
16	Ana Perry	24
17	Alexandra Jenkins	22

Succeeded (2 sec 671 ms)

Columns: 2 Rows: 10,000

Data | Query | Model

Copyright Nawatech 2024

Secure the medallion layers

- **Gold Layer Access Control**: Read-only.
- **Silver Layer Utilization**: Build optional.
- **Bronze Layer Access Control**: Read only.

Copyright Nawatech 2024

Nawatech

Thank you
Let’s discuss our collaborations

PT Nawa Darsana Teknologi
Gedung Office 8, 18th Floor, Jl. Jend Sudirman Kav 52-53
SCBD Lot 8, Senopati Jakarta Selatan 12190 DKI Jakarta, Indonesia
Website : www.nawatech.co
Phone : +62 21 29552754

Copyright Nawatech 2024

Contact
Innes Sabrina Husa
Senior Business Development Manager
Email : windy@nawatech.co
Phone : +62 812 8033 2457