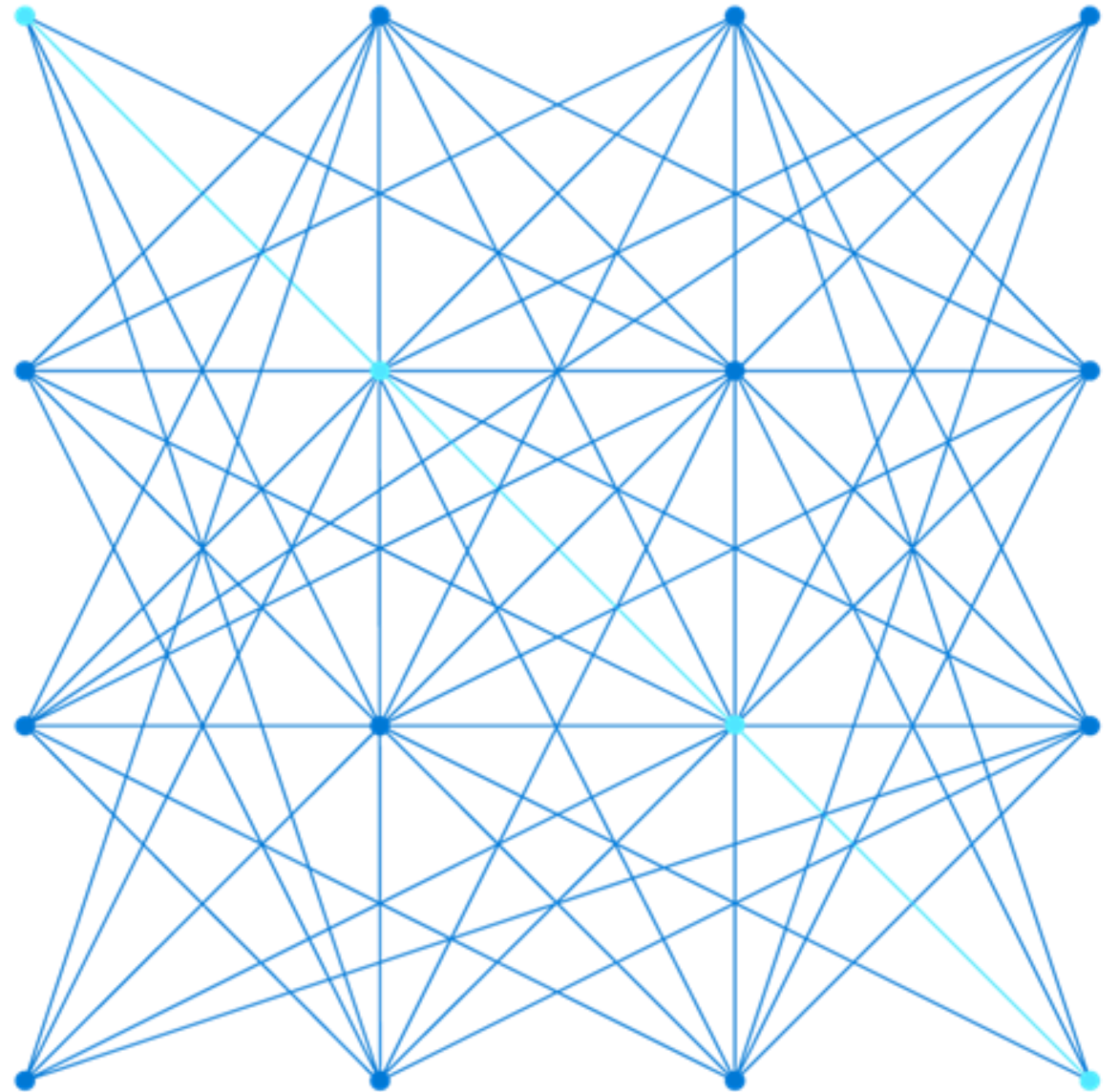


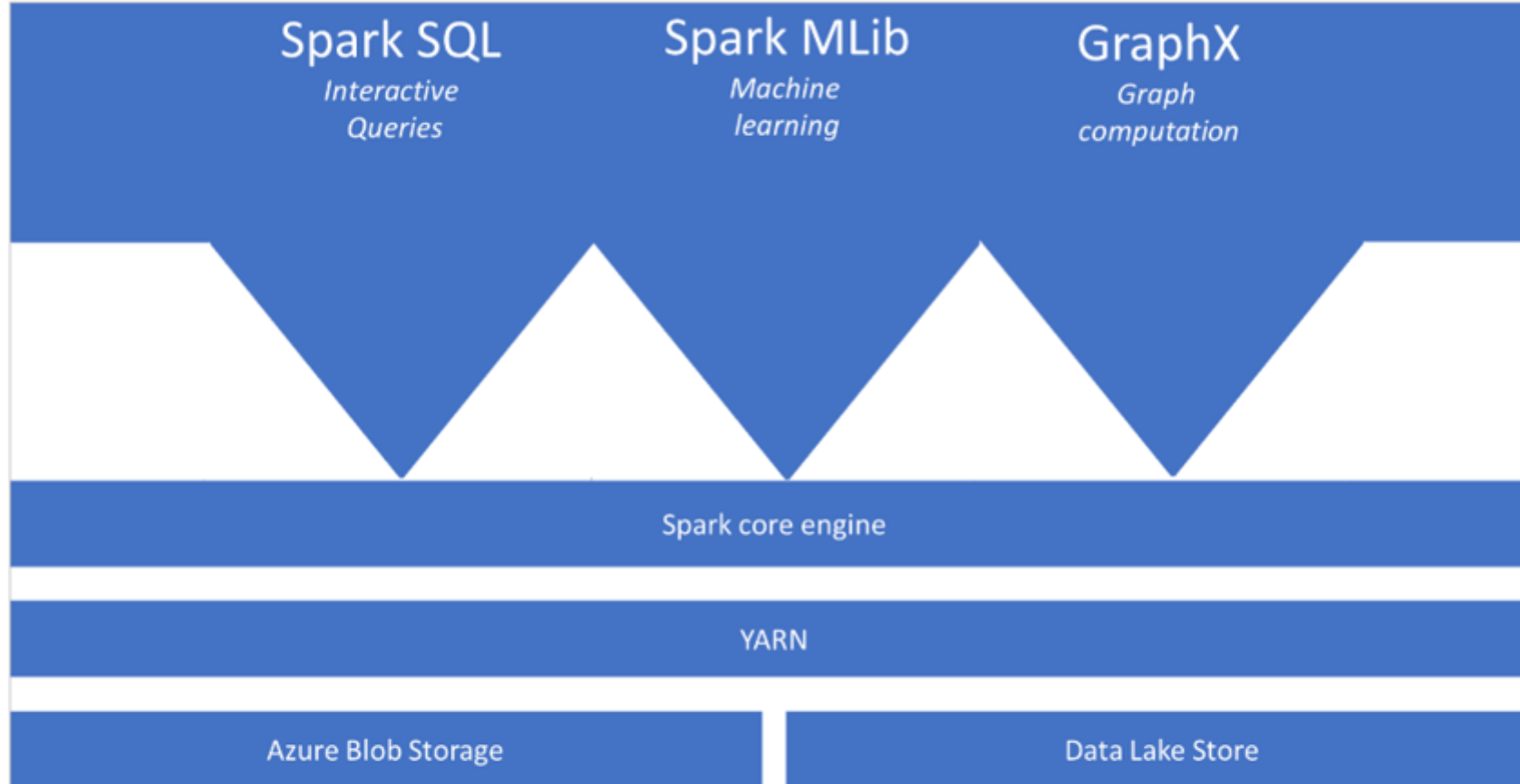
DP-203T00: Explore, transform, and load data into the Data Warehouse using Apache Spark



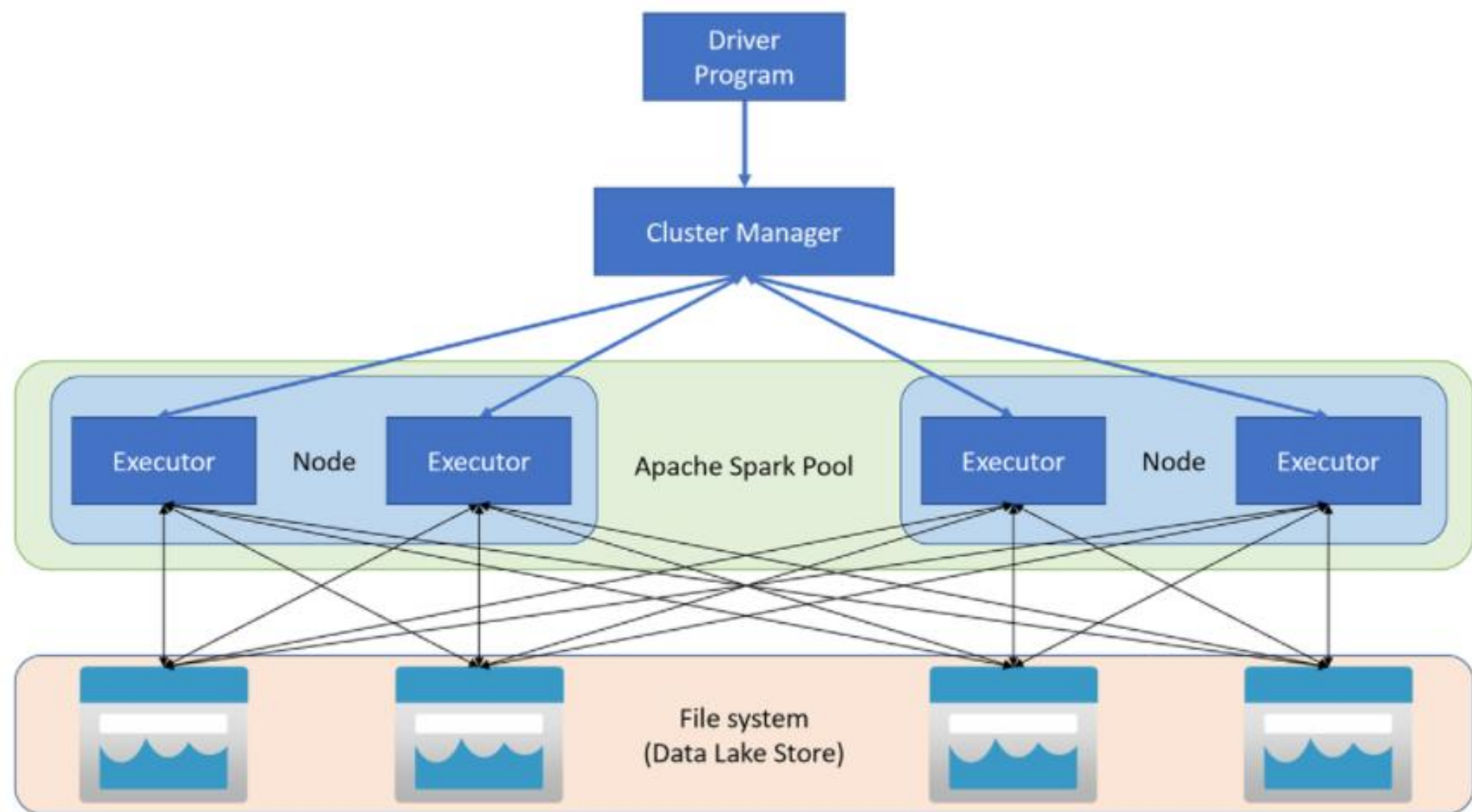
Lesson 01: Understand big data engineering with Apache Spark in Azure Synapse Analytics



Introduction to big data engineering with Apache Spark in Azure Synapse Analytics



How do Apache Spark pools work in Azure Synapse Analytics



How to create an Apache Spark pool in Azure Synapse Analytics

[Home](#) > [\[Placeholder\]](#) >

Create Apache Spark pool ...

* Basics

* Additional settings

Tags

Review + create

Create a Synapse Analytics Apache Spark pool with your preferred configurations. Complete the Basics tab then go to Review + create to provision with smart defaults, or visit each tab to customize.

Apache Spark pool details

Name your Apache Spark pool and choose its initial settings.

Apache Spark pool name *

sprkpl01 ✓

Node size family

MemoryOptimized

Node size *

Small (4 vCores / 32 GB) ✓

Autoscale * ⓘ

☒ Enabled ☐ Disabled

Number of nodes *

3

27

Estimated price ⓘ

Est. cost per hour

[Placeholder]

[View pricing details](#)

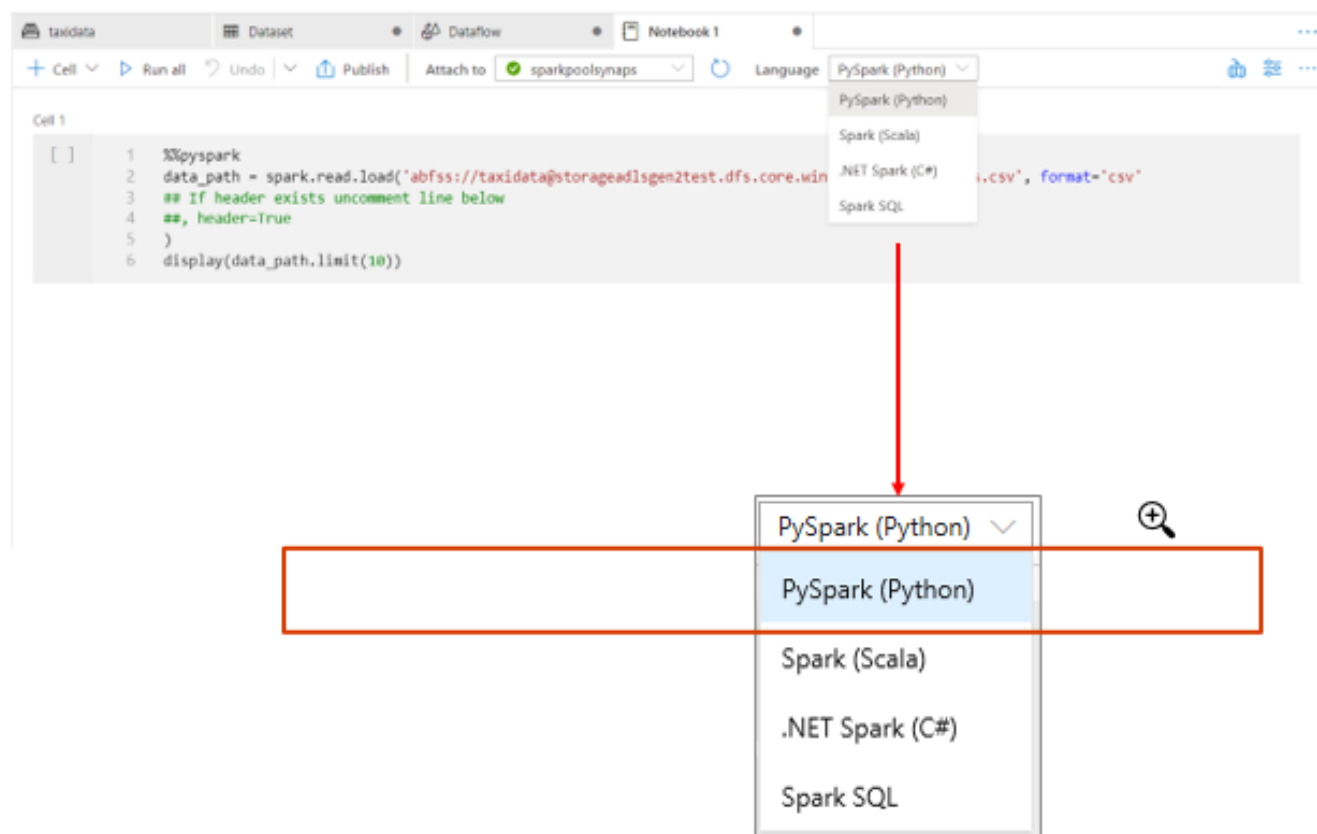
Lesson 02: Ingest data with Apache Spark notebooks in Azure Synapse Analytics



Apache Spark notebooks features in Azure Synapse Analytics

Notebooks

- Access through Synapse Studio
- Examples Available through Knowledge Center
- Allows to write multiple languages in one notebook by using %%<Name of language>
- Support for Language Syntax highlight, syntax error, syntax code completion
- Offers temporary tables across languages
- Export results



Creating a notebook in Azure Synapse Analytics

✓ Validate all

↑ Publish all

1

🗑 Discard all

Develop

+

≡

⏪

🔍 Filter resources by name

📁 Notebooks

1

• 📄 Notebook 1

Notebook 1

+

Cell

⏮ Run all

📄 Publish

⚠ Please select a Spark pool to attach before running cell

Attach to

Select Spark pool

📌 sparkpoolmod

🗑 Manage pools

🔄

Language

PySpark (Python)

🔍

NextGen Notebooks (Preview)

⚙

⋮

⋮

▶

1

🗑

+

📄 Properties

General

📘

Choose a name for your Notebook.
This name can be updated at any time until it is published.

Name

Notebook 1

Description

Type

.ipynb notebook

Size

191 bytes

Notebook settings

✓

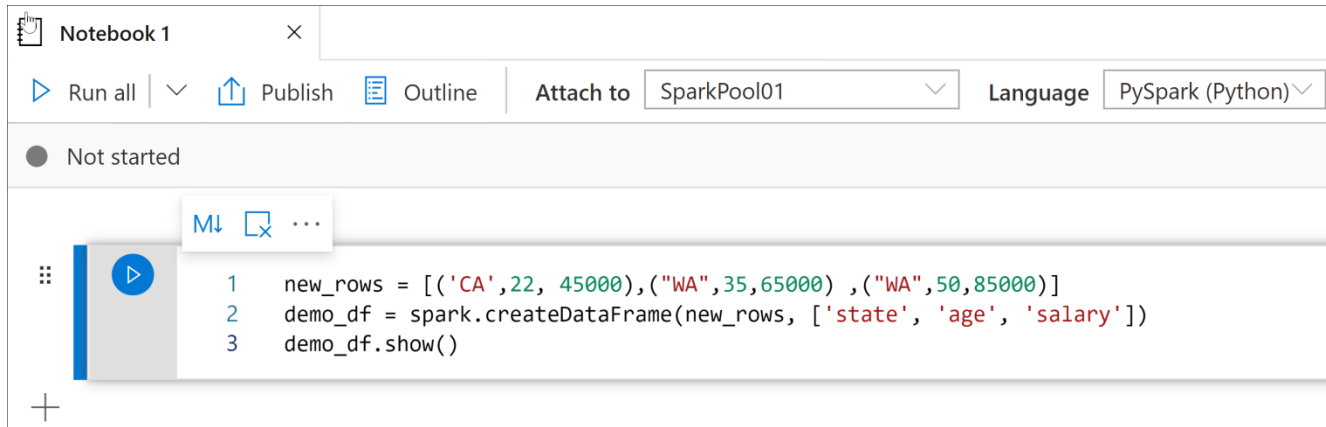
Include cell output when saving

Session

[Configure session](#)

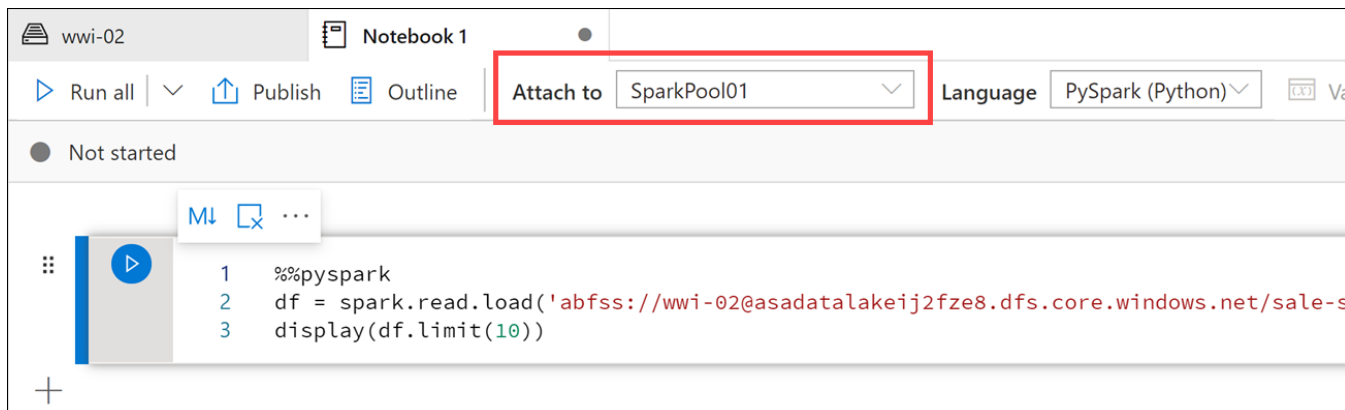
Ingest data with Apache Spark notebooks in Azure Synapse Analytics

> Generating data while executing the command



```
1 new_rows = [('CA', 22, 45000), ('WA', 35, 65000), ('WA', 50, 85000)]
2 demo_df = spark.createDataFrame(new_rows, ['state', 'age', 'salary'])
3 demo_df.show()
```

> Loading data in a single command from a data file

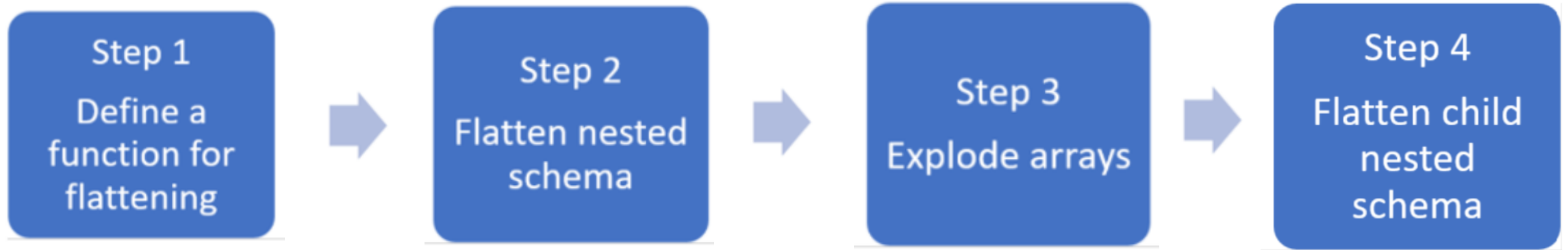


```
1 %%pyspark
2 df = spark.read.load('abfss://wwi-02@asadatalakeij2fze8.dfs.core.windows.net/sale-s
3 display(df.limit(10))
```

Lesson 03: Transform data with DataFrames in Apache Spark Pools in Azure Synapse Analytics



Transform data with DataFrames in Apache Spark Pools in Azure Synapse Analytics

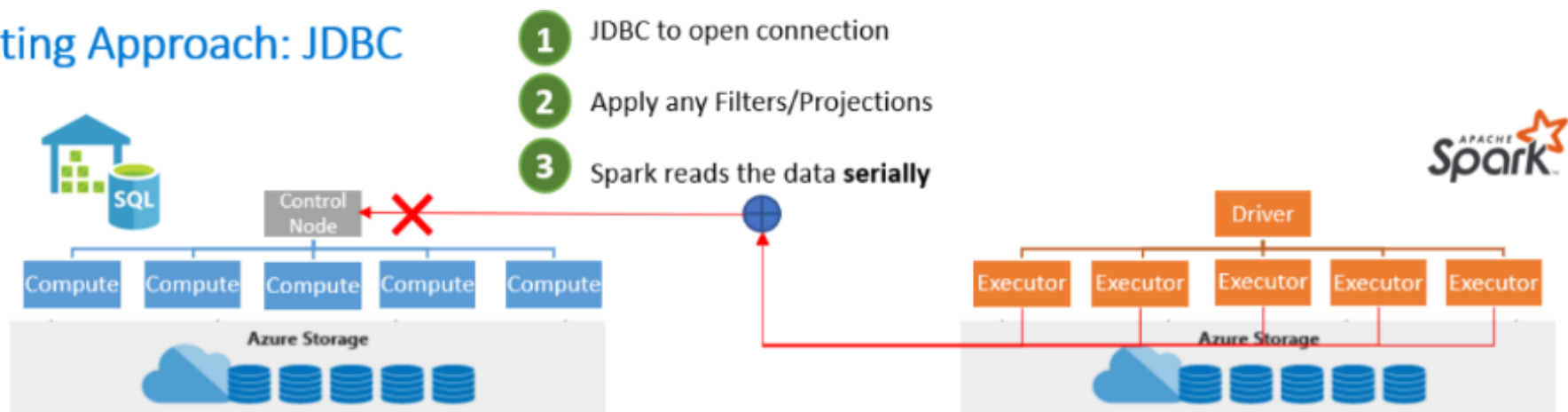


Lesson 04: Integrate SQL and Apache Spark pools in Azure Synapse Analytics

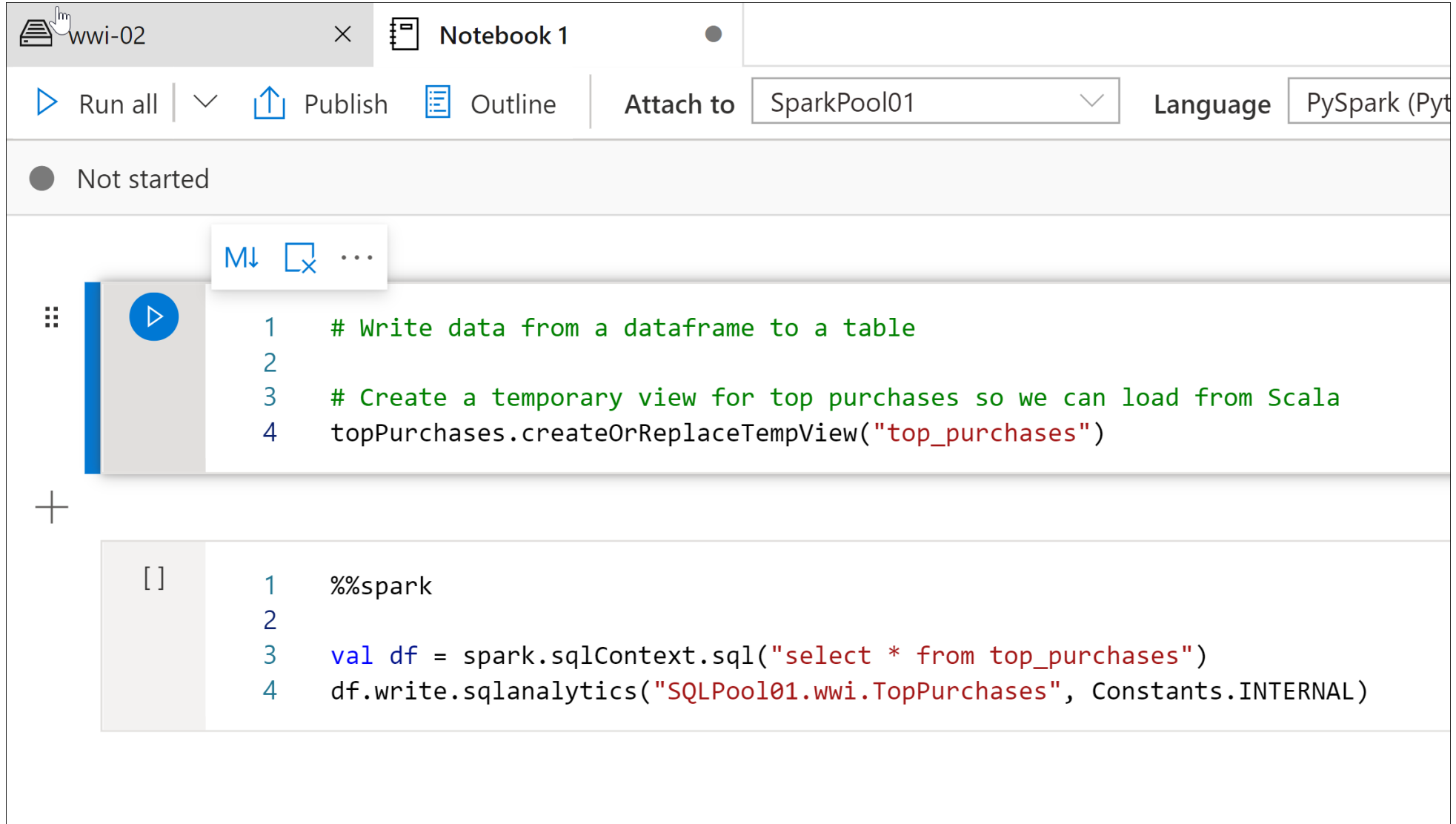


Integrate SQL and Apache Spark pools in Azure Synapse Analytics

Existing Approach: JDBC



Write data from Apache Spark pools to a dedicated SQL pool



The screenshot displays a Databricks notebook titled "wwi-02" and "Notebook 1". The interface includes a toolbar with "Run all", "Publish", "Outline", and "Attach to" (set to "SparkPool01"). The language is set to "PySpark (Py)".

The notebook content consists of two code blocks:

```
1 # Write data from a dataframe to a table
2
3 # Create a temporary view for top purchases so we can load from Scala
4 topPurchases.createOrReplaceTempView("top_purchases")
```

```
1 %%spark
2
3 val df = spark.sqlContext.sql("select * from top_purchases")
4 df.write.sqlanalytics("SQLPool01.wwi.TopPurchases", Constants.INTERNAL)
```

Write data from a dedicated SQL pool to Apache Spark pools

The screenshot shows a Databricks notebook interface. At the top, the workspace is labeled 'wwi-02' and the notebook is titled 'Notebook 1'. The toolbar includes buttons for 'Run all', 'Publish', 'Outline', and 'Attach to', which is set to 'SparkPool01'. A status bar indicates 'Not started'. The notebook content consists of two cells. The first cell is a comment: '# Write data from a table to a view in Spark'. The second cell is a code block starting with a magic command '%spark', followed by two lines of Scala code: 'val df2 = spark.read.sqlanalytics("SQLPool01.wwi.TopPurchases")' and 'df2.createTempView("top_purchases_sql")'.

wwi-02 Notebook 1

Run all Publish Outline Attach to SparkPool01 Language

Not started

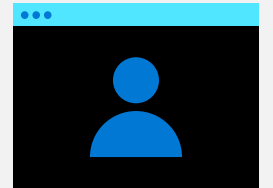
M↓ □x ...

1 # Write data from a table to a view in Spark

+

```
[ ] 1 %%spark
    2 val df2 = spark.read.sqlanalytics("SQLPool01.wwi.TopPurchases")
    3 df2.createTempView("top_purchases_sql")
```

Lab: Explore, transform, and load data into the Data Warehouse using Apache Spark



Lab overview

This lab teaches you how to explore data stored in a data lake, transform the data, and load data into a relational data store. You will explore Parquet and JSON files and use techniques to query and transform JSON files with hierarchical structures. Then you will use Apache Spark to load data into the data warehouse and join Parquet data in the data lake with data in the dedicated SQL pool.

Lab objectives

After completing this lab, you will be able to:

Perform Data Exploration in Synapse Studio

Ingest data with Spark notebooks in Azure Synapse Analytics

Transform data with DataFrames in Spark pools in Azure Synapse Analytics

Integrate SQL and Spark pools in Azure Synapse Analytics