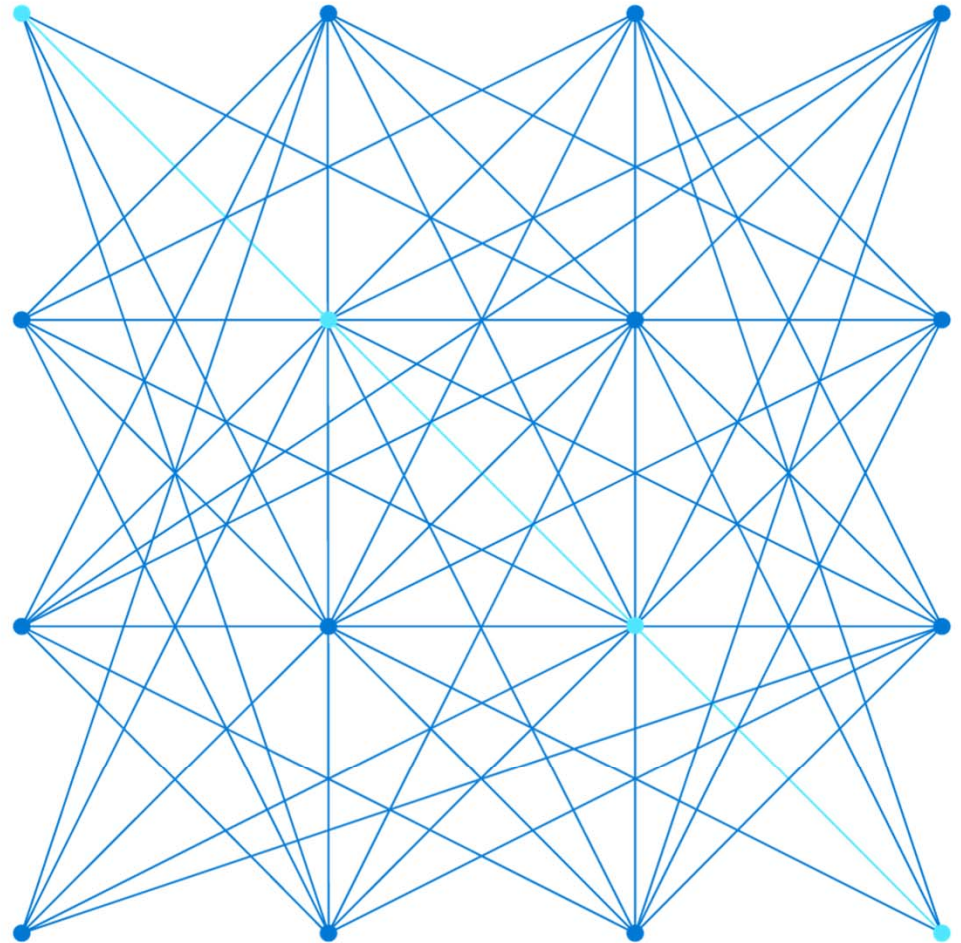


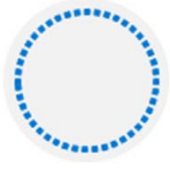
DP-203T00: Explore, transform, and load data into the Data Warehouse using Apache Spark



Agenda



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



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



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

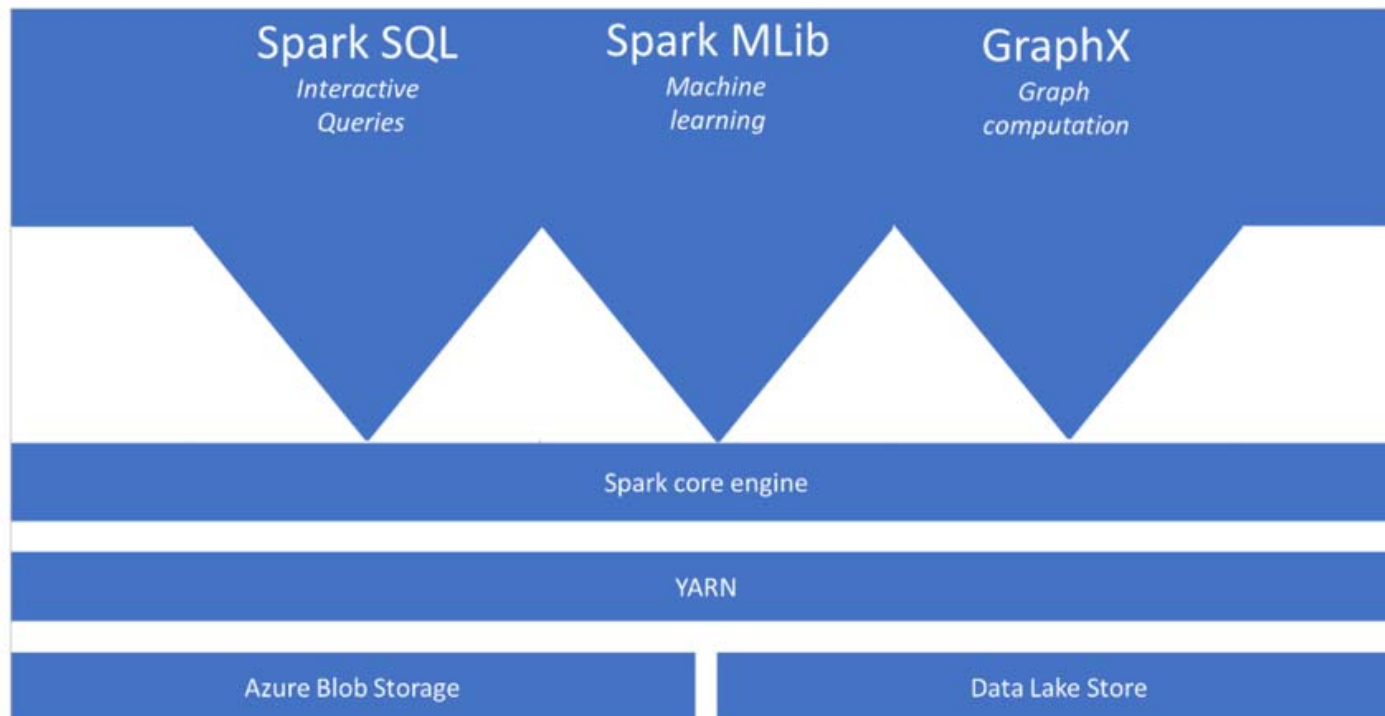


Lesson 04 – Integrate SQL and Apache Spark pools in Azure Synapse Analytics

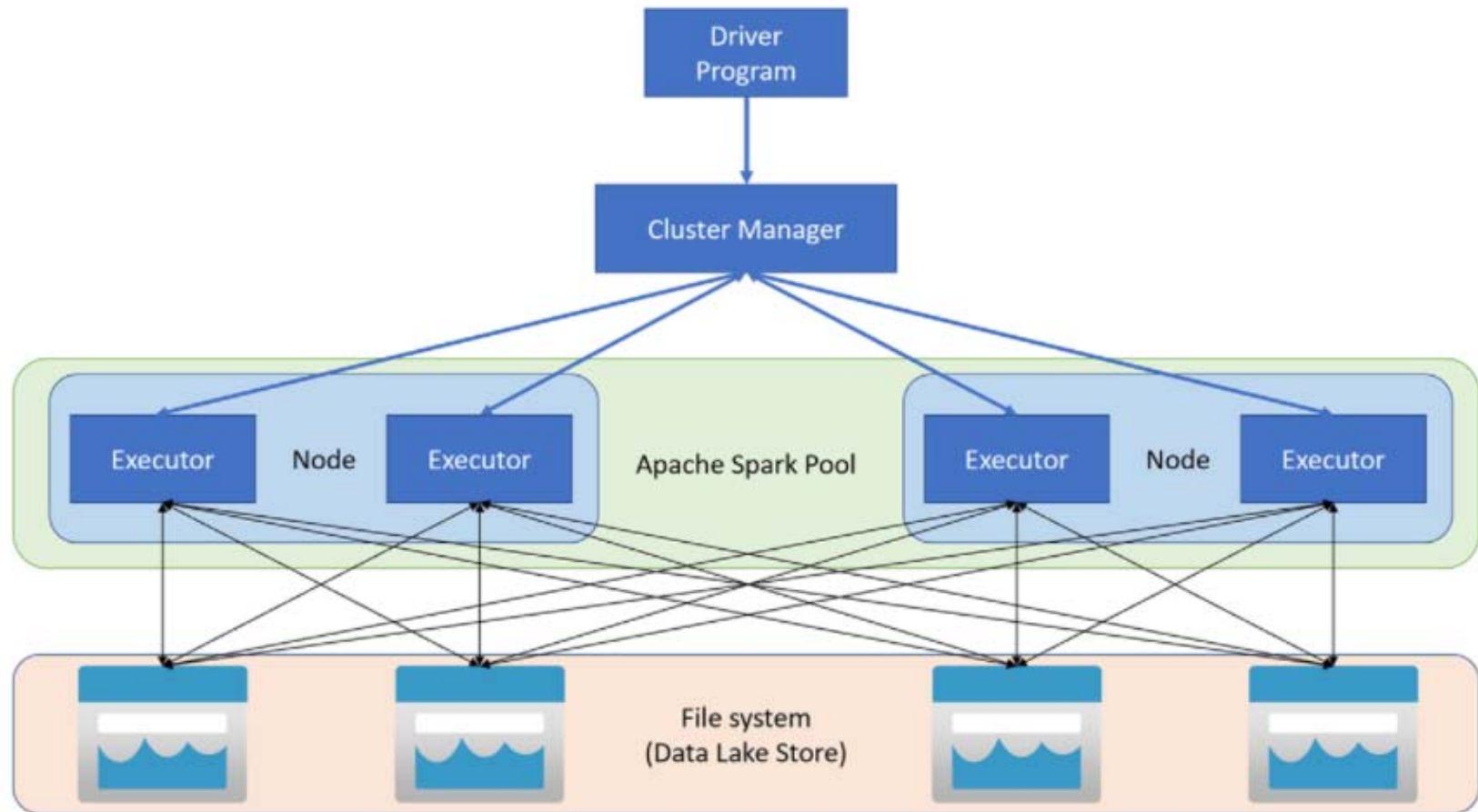
Lesson 01: Understand big data engineering with Apache Spark in Azure Synapse Analytics



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 > >

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 *	<input type="text" value="sprkpl01"/> ✓
Node size family	MemoryOptimized
Node size *	<input type="text" value="Small (4 vCores / 32 GB)"/> ▼
Autoscale * ⓘ	<input checked="" type="radio"/> Enabled <input type="radio"/> Disabled
Number of nodes *	<input type="text" value="3"/>  <input type="text" value="27"/>
Estimated price ⓘ	<div>Est. cost per hour</div> <div></div> <div>View pricing details</div>

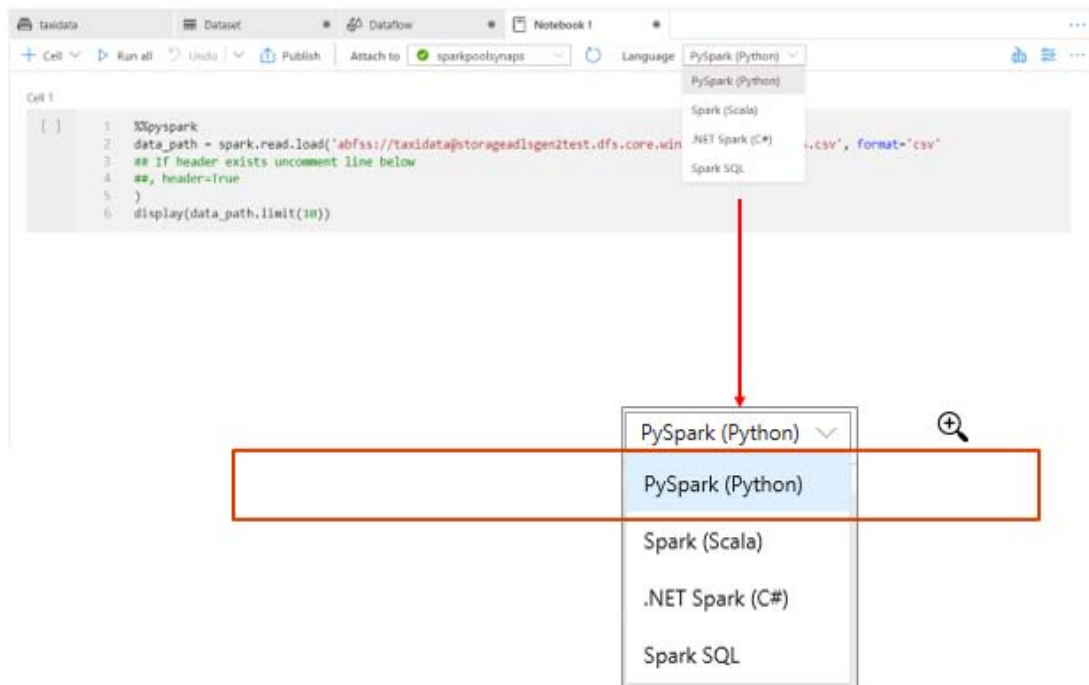
Lesson 02: Ingest data with Apache Spark notebooks in Azure Synapse Analytics



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

The screenshot displays the Azure Synapse Analytics 'Develop' environment. At the top, there are buttons for 'Validate all', 'Publish all' (with a yellow notification icon), and 'Discard all'. Below this, the 'Develop' sidebar on the left shows a search bar and a list of notebooks, with 'Notebook 1' selected. The main workspace is titled 'Notebook 1' and contains a toolbar with '+ Cell', 'Run all', and 'Publish' buttons. A red box highlights the 'Attach to' dropdown menu, which is set to 'Select Spark pool'. Another red box highlights the 'Language' dropdown menu, which is set to 'PySpark (Python)'. A third red box highlights the 'sparkpoolmod' option in the 'Attach to' dropdown. A warning message states: 'Please select a Spark pool to attach before running cell'. The right sidebar, titled 'Properties', shows the 'General' tab with a message: 'Choose a name for your Notebook. This name can be updated at any time until it is published.' Below this, the 'Name' field is set to 'Notebook 1'. The 'Description' field is empty. The 'Type' is '.ipynb notebook' and the 'Size' is '191 bytes'. Under 'Notebook settings', the checkbox 'Include cell output when saving' is checked. At the bottom, there is a 'Session' section with a link to 'Configure session'.

Develop

Filter resources by name

Notebooks 1

Notebook 1

Notebook 1

+ Cell Run all Publish

Attach to Select Spark pool

Language PySpark (Python)

Please select a Spark pool to attach before running cell

sparkpoolmod

Manage pools

NextGen Notebooks (Preview)

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

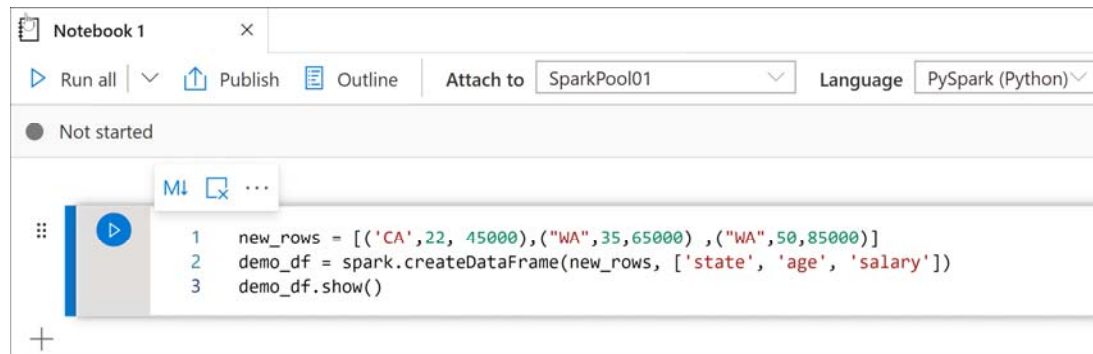
☒ Include cell output when saving

Session

[Configure session](#)

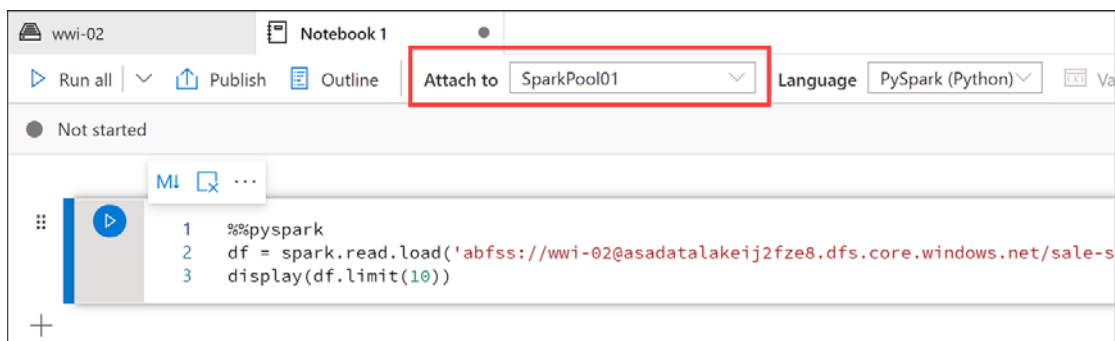
Ingest data with Apache Spark notebooks in Azure Synapse Analytics

- 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

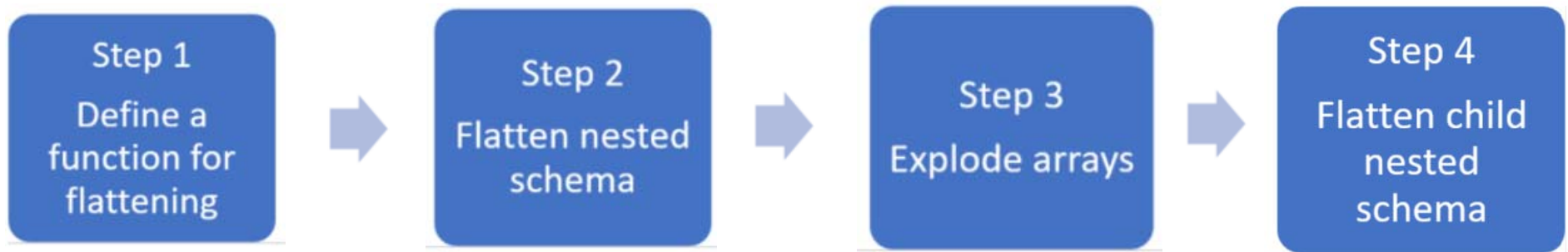


```
1 %%pyspark
2 df = spark.read.load('abfss://wwi-02@asadatalakeij2fze8.dfs.core.windows.net/sale-s
3 display(df.limit(10))
```

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

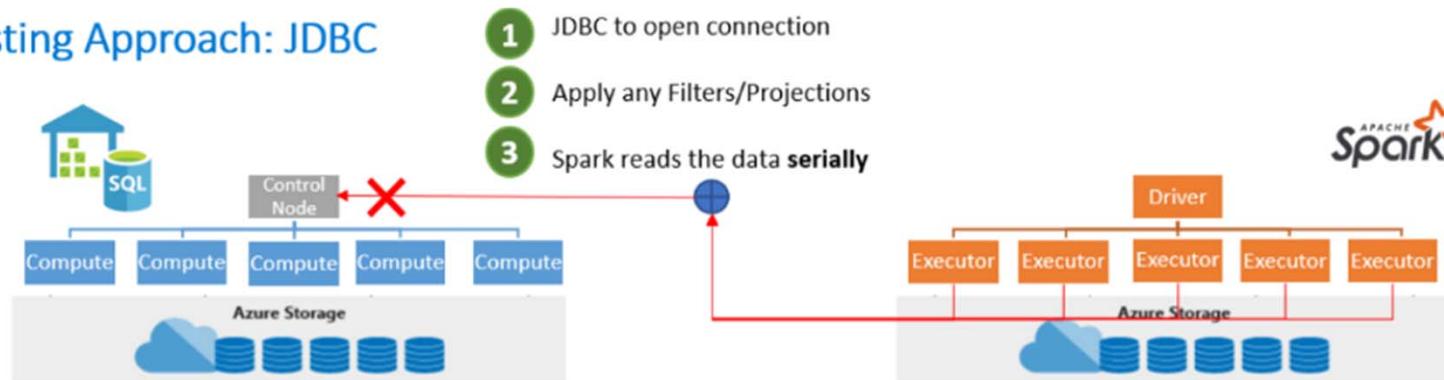


Lesson 04: Integrate SQL and Apache Spark pools in Azure Synapse Analytics

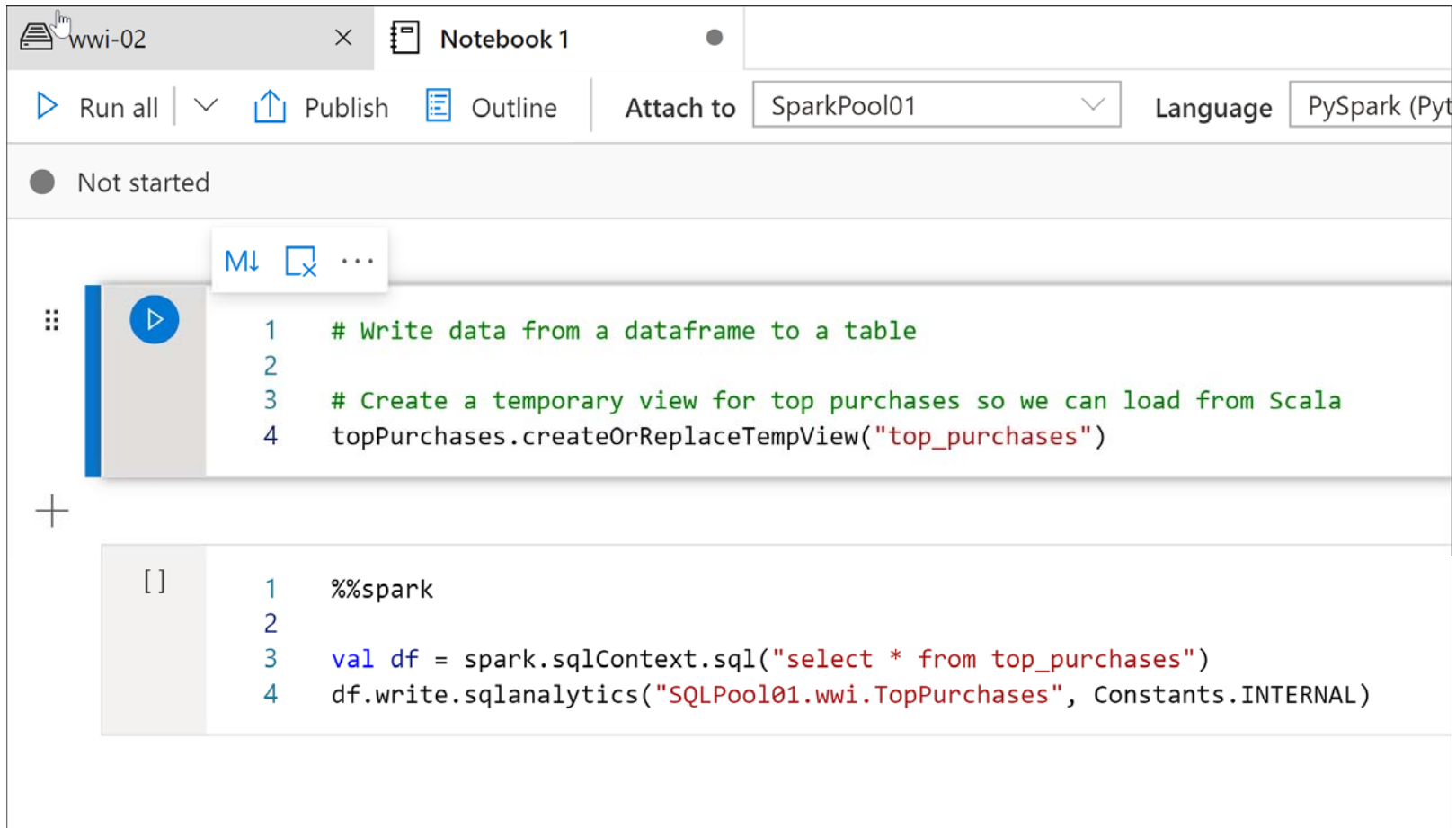


Integrate SQL and Apache Spark pools in Azure Synapse Analytics

Existing Approach: JDBC



Write data from Apache Spark pools to a dedicated SQL pool



The screenshot shows a Databricks notebook interface. At the top, there's a header bar with a tab labeled 'wwi-02', a close button, and 'Notebook 1'. Below this is a toolbar with buttons for 'Run all', 'Publish', 'Outline', 'Attach to' (set to 'SparkPool01'), and 'Language' (set to 'PySpark (Py)'). A status bar indicates 'Not started'. The notebook content consists of two code blocks. The first block is a Scala code cell with a blue play button icon and a menu showing 'M↓', a copy icon, and '...'. It contains four lines of Scala code. The second block is a magic command cell, indicated by a '+' icon and a light blue background, containing two lines of Scala code. The code in the first block creates a temporary view from a DataFrame, and the code in the second block writes the data from that view to a dedicated SQL pool.

```
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 notebook is named "wwi-02" and "Notebook 1". The toolbar includes buttons for "Run all", "Publish", "Outline", and "Attach to" (set to "SparkPool01"). Below the toolbar, a status bar indicates "Not started". The main area contains a code cell with a blue play button icon and a menu with "M↓", "□x", and "...". The code cell contains a single line of green text: "# Write data from a table to a view in Spark". Below this, there is a plus sign icon and a code editor with a light gray background. The code editor contains three lines of code: 1. "%spark", 2. "val df2 = spark.read.sqlanalytics("SQLPool01.wwi.TopPurchases")", and 3. "df2.createTempView("top_purchases_sql")".

wwi-02 Notebook 1

Run all | Publish | Outline | Attach to SparkPool01 | Language

Not started

M↓ □x ...

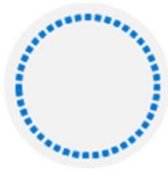
1 # Write data from a table to a view in Spark

+

[]

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


Review questions



Q01 – What is an element of a Spark Pool in Azure Synapse Analytics?

A01 – Spark Instance



Q02 – How can all Apache Spark notebooks in Synapse Studio be saved?

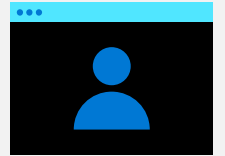
A02 – Select the Publish all button on the workspace command bar.



Q03 – When is it unnecessary to use import statements for transferring data between a dedicated SQL and Spark pool?

A03 – Use the integrated notebook experience from Azure Synapse Studio.

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:

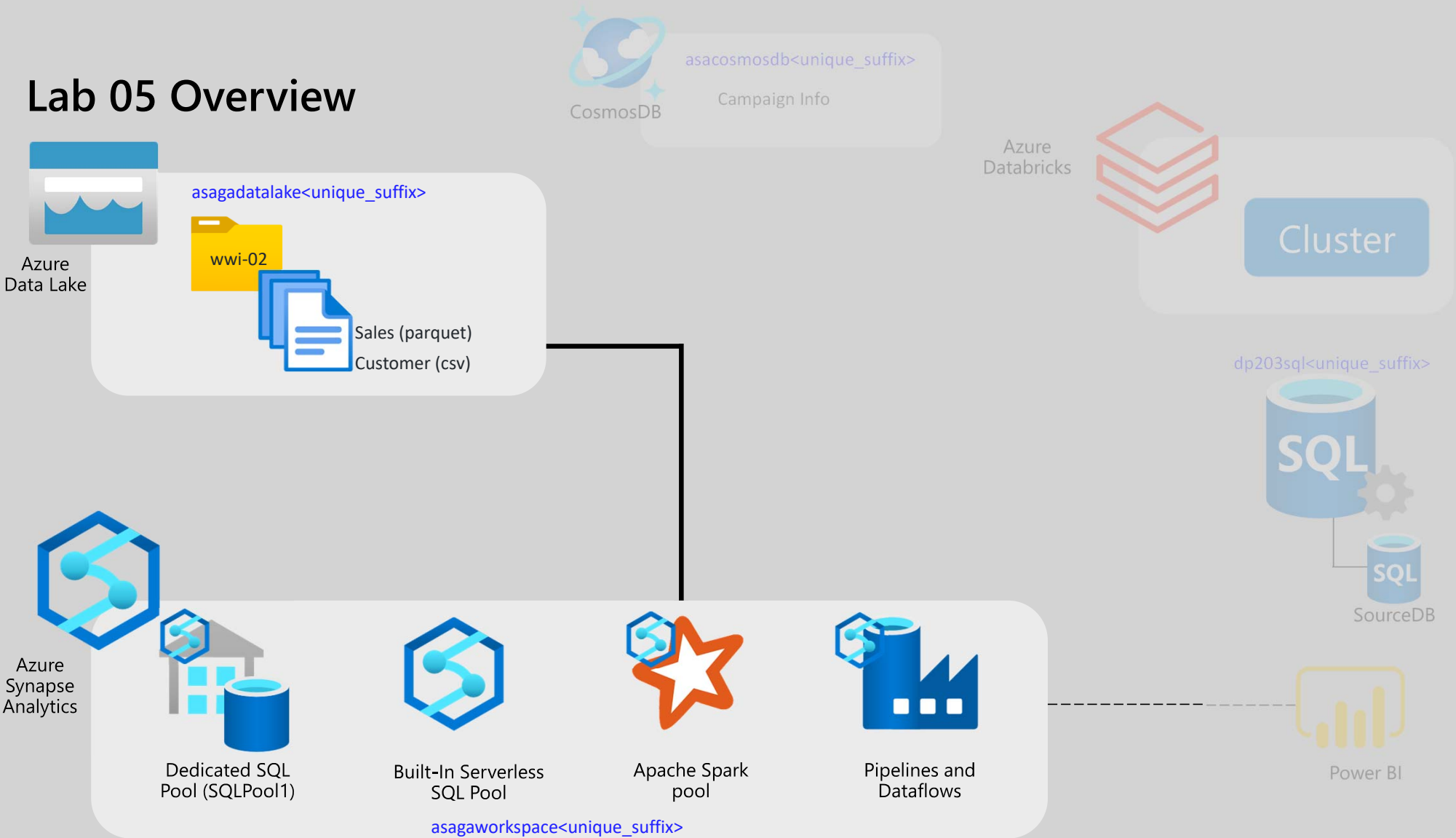
Perform Data Exploration in Synapse Studio

Ingest data with Spark notebooks in Azure Synapse Analytics

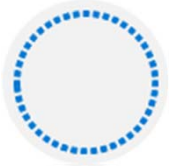
Transform data with DataFrames in Spark pools in Azure Synapse Analytics

Integrate SQL and Spark pools in Azure Synapse Analytics

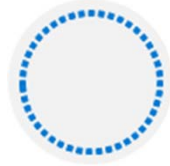
Lab 05 Overview



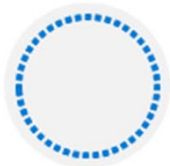
Lab review



Q01 – Which command is used to analyze parquet files and infer schema's using the Spark Engine?



Q02 – What is an option to you query JSON files using the SQL syntax in an Apache Spark Notebook connected to a Spark Pool in Azure Synapse Analytics?



Q03 – How do you set the language of a cell in an Apache Spark Notebook?

Module summary

In this module, you have learned about:

Azure Synapse Analytics

Apache Spark Notebooks

Integration of SQL and Spark

DataFrames

Apache Spark Architecture

Next steps

After the course, consider visiting [[Azure Apache Spark for Azure Synapse Analytics](#)]. The Apache Spark in Azure Synapse Analytics provides an overview of how Apache Spark is integrated with Azure Synapse Analytics.

