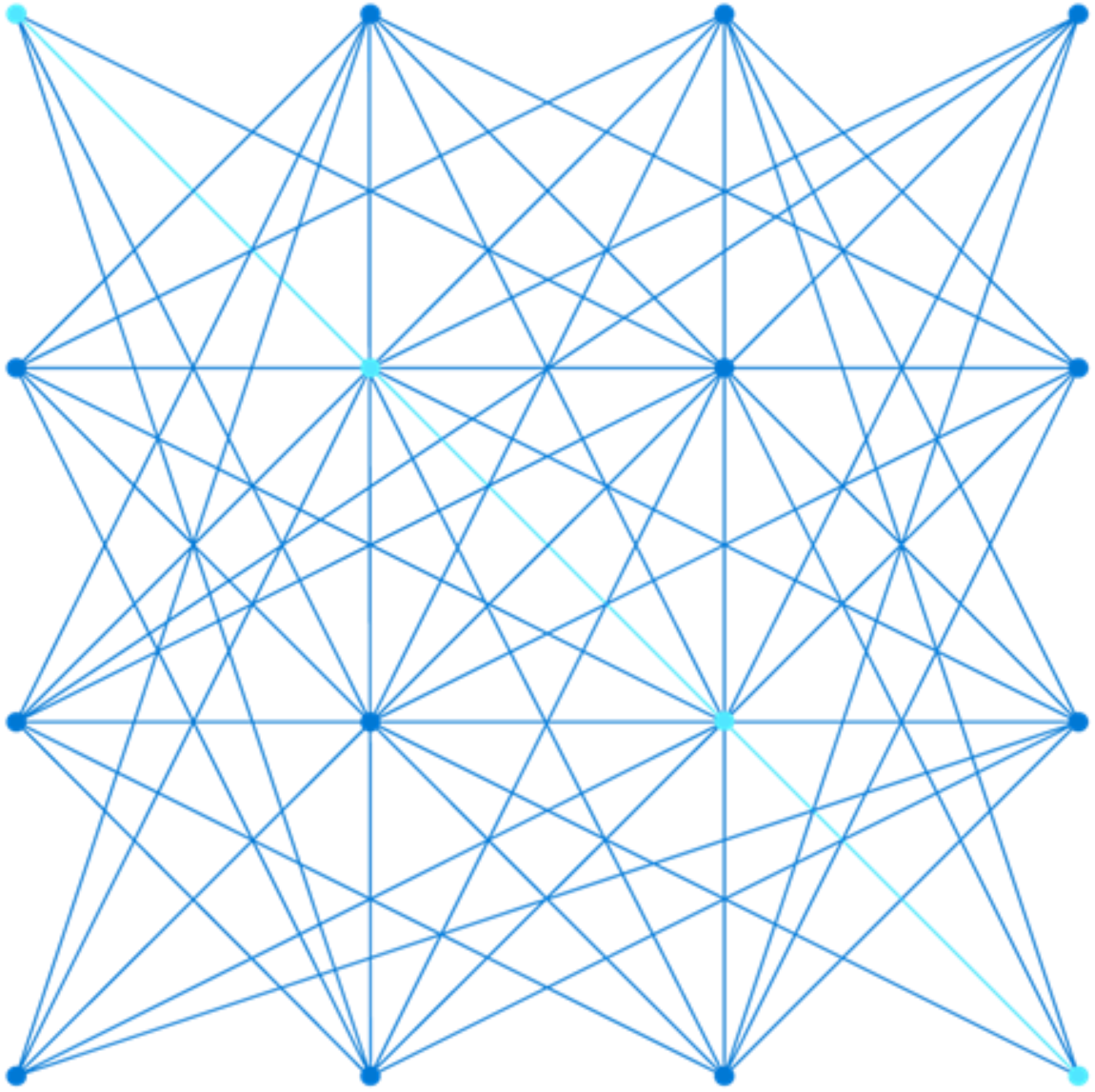
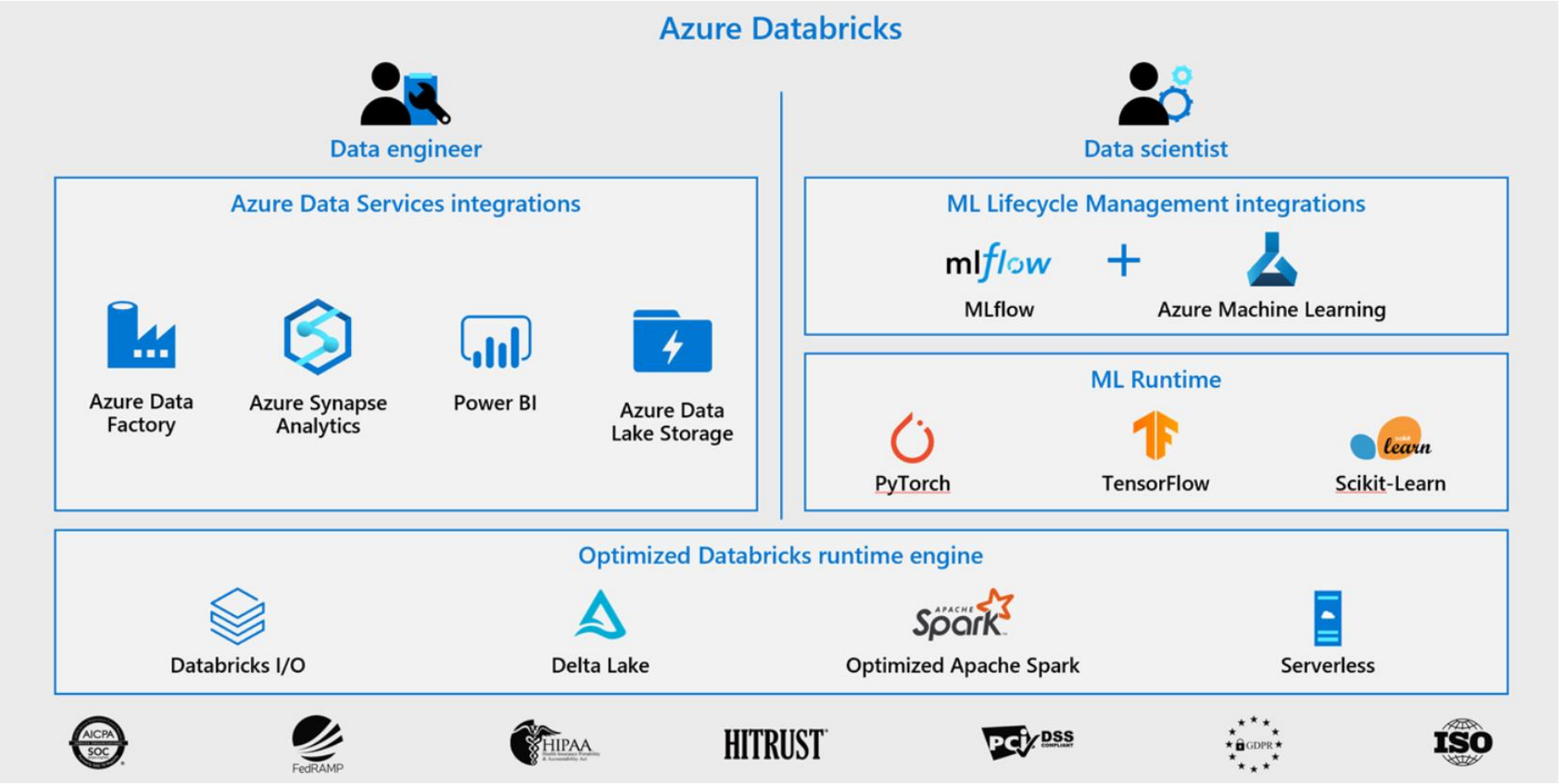
Microsoft Azure

DP-203T00: Data Exploration and Transformation in Azure Databricks



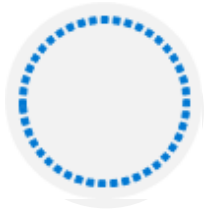
Lesson 01: Understand Azure Databricks

Understand Azure Databricks



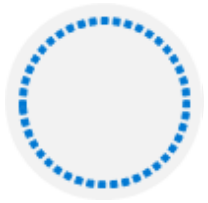
Lesson 02: Read and write data in Azure Databricks

Read and write data in Azure Databricks



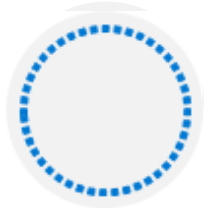
Multiple format support

Reading data from CSV, PARQUET, JSON and many others



Integrated with several Azure Data Services

Reading and writing from and to Azure Data Lake Storage, Azure Synapse Analytics, etc.



Notebook experience

Reading and writing by simply writing code in a shared notebook experience

Read data in Azure Databricks

work with dataframes (Scala)

Test cluster | File | Edit | View: Standard | Permissions | Run All | Clear

Cmd 1

```
1  spark.conf.set("fs.azure.account.auth.type", "OAuth")
2  spark.conf.set("fs.azure.account.oauth.provider.type", "org.apache.hadoop.fs.azurebfs.oauth2.ClientCredsTokenProvider")
3  spark.conf.set("fs.azure.account.oauth2.client.id.a                .dfs.core.windows.net", "                                47")
4  spark.conf.set("fs.azure.account.oauth2.client.secret.as           2.dfs.core.windows.net", "                                ")
5  spark.conf.set("fs.azure.account.oauth2.client.endpoint.as         2.dfs.core.windows.net", "https://login.microsoftonline.com/72f                                /oauth2/token")
```

Command took 0.52 seconds -- by [redacted] at 7/6/2021, 3:21:39 PM on Test cluster

Cmd 2

```
1  val df = spark.read.option("header",true).csv("abfss://wwi-02[redacted]dfs.core.windows.net/sale-poc/sale-20170501.csv")
```

▸ (1) Spark Jobs

▸ df: org.apache.spark.sql.DataFrame = [TransactionId: string, CustomerId: string ... 9 more fields]

```
df: org.apache.spark.sql.DataFrame = [TransactionId: string, CustomerId: string ... 9 more fields]
```

Command took 1.53 seconds -- by [redacted] at 7/6/2021, 3:26:36 PM on Test cluster

Cmd 3

```
1  display(df.limit(10))
```

▸ (1) Spark Jobs

	TransactionId	CustomerId	ProductId	Quantity	Price	TotalAmount	TransactionDate	ProfitAmount	Hour	Minute	StoreId
1	e067fc11-e07d-4517-bc93-f7dc4b44f35e	3	4581	4	20.84	91.696	20170501	26.048	2	30	7922
2	e067fc11-e07d-4517-bc93-f7dc4b44f35e	3	1365	4	26.52	116.688	20170501	29.436	2	30	7922
3	e067fc11-e07d-4517-bc93-f7dc4b44f35e	3	2641	4	29.71	130.724	20170501	37.4	2	30	7922
4	e067fc11-e07d-4517-bc93-f7dc4b44f35e	3	220	2	27.6	60.72	20170501	15.356	2	30	7922
5	e067fc11-e07d-4517-bc93-f7dc4b44f35e	3	110	3	28.41	93.753	20170501	33	2	30	7922
6	e067fc11-e07d-4517-bc93-f7dc4b44f35e	3	2	1	39.78	43.758	20170501	11.528	2	30	7922
7	cdd2ed88-8aae-4295-884a-ac4d40c3c33c	11	3323	1	30.52	33.572	20170501	10.252	20	43	3573

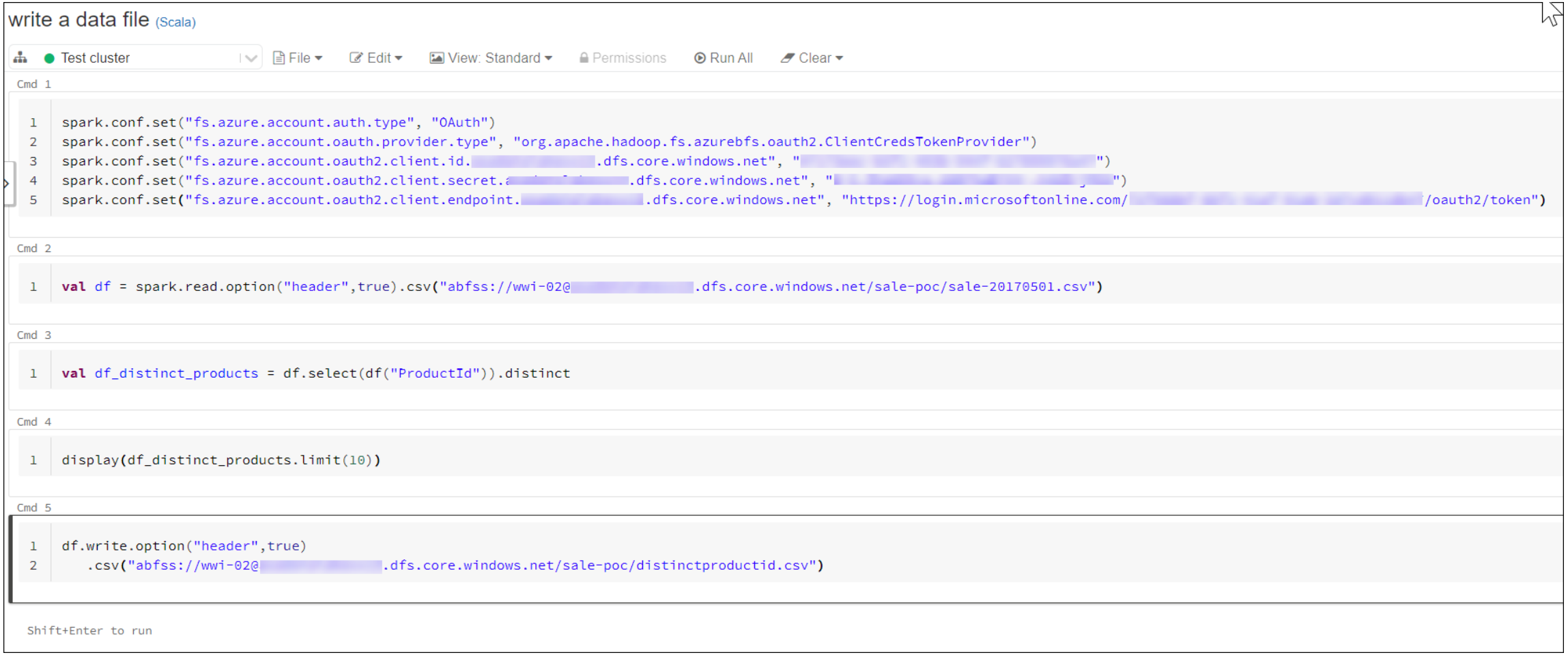
Showing all 10 rows.

Command took 0.79 seconds -- by [redacted] at 7/6/2021, 3:28:04 PM on Test cluster

Working with Select in Azure Databricks

SQL	DataFrame (Python)
SELECT col_1 FROM myTable	df.select(col("col_1"))
DESCRIBE myTable	df.printSchema()
SELECT * FROM myTable WHERE col_1 > 0	df.filter(col("col_1") > 0)
..GROUP BY col_2	..groupBy(col("col_2"))
..ORDER BY col_2	..orderBy(col("col_2"))
..WHERE year(col_3) > 1990	..filter(year(col("col_3")) > 1990)
SELECT * FROM myTable LIMIT 10	df.limit(10)
display(myTable) (text format)	df.show()
display(myTable) (html format)	display(df)

Write data in Azure Databricks



Lesson 03: Work with DataFrames in Azure Databricks

Work with DataFrames in Azure Databricks

- Apache Spark DataFrame API reading data in a single command

```
parquetDir = source + "/wikipedia/pagecounts/staging_parquet_en_only_clean/"

pagecountsEnAllDF = (spark  # Our SparkSession & Entry Point
  .read                     # Our DataFrameReader
  .parquet(parquetDir)      # Returns an instance of DataFrame
)
print(pagecountsEnAllDF)    # Python hack to see the data type
```

Working with transformations in Azure Databricks

Transformations	Description
Select(…)	The select(…) command enables you to specify the columns to include in a query
drop(…)	The drop(…) command enables you to specify the columns you don't want
distinct(…)	The distinct(…) command returns a distinct set of values in a DataFrame
dropDuplicates(…)	The dropDuplicates(…) command is an alias of the distinct(…) command.
show(…)	The show(..) command is part of the core Spark API and simply prints the results to the console
display(…)	The display(…) command provides more flexibility than show(…) such as downloading results against csv, rendering charts and showing up to 100 rows
limit(…)	The limit(…) command can be used to control the number of records that are returned to a DataFrame

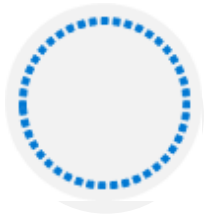
Optimize DataFrames in Azure Databricks

- Mix DataFrame operations

```
(pagecountsEnAllDF
  .cache()          # Mark the DataFrame as cached
  .count()          # Materialize the cache
)
```

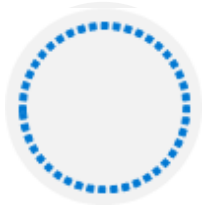
Lesson 04: Work with DataFrames advanced methods in Azure Databricks

Work with DataFrames advanced methods in Azure Databricks



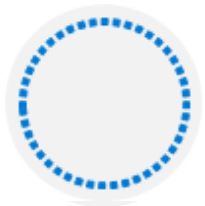
DateTime manipulation

Enabling different DateTime techniques to use across DataFrames



Aggregate Functions

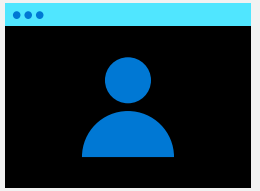
groupBy() function, sum(), count(), avg(), min(), max() functions



Deduplication of Data

Removing duplicates, by ensuring you only keep 1 record

Lab: Data Exploration and Transformation in Azure Databricks



Lab overview

This lab teaches you how to use various Apache Spark DataFrame methods to explore and transform data in Azure Databricks. You will learn how to perform standard DataFrame methods to explore and transform data. You will also learn how to perform more advanced tasks, such as removing duplicate data, manipulate date/time values, rename columns, and aggregate data.

Lab objectives

After completing this lab, you will be able to:

- Use DataFrames in Azure Databricks to explore and filter data
- Cache a DataFrame for faster subsequent queries
- Remove duplicate data
- Manipulate date/time values
- Remove and rename DataFrame columns
- Aggregate data stored in a DataFrame