**Title: Introduction to PySpark in Fabric Notebook**

**Step 1: Upload Raw CSV Files**

from pyspark.sql import SparkSession

# Initialize Spark session

spark = SparkSession.builder.appName("Example").getOrCreate()

# Load CSV file into the DataFrame

df1 = spark.read.csv("Files/sales\_data01.csv", header=True, inferSchema=True)

df2 = spark.read.csv("Files/sales\_data02.csv", header=True, inferSchema=True)

# Show the content of the dataframes

df1.show()

df2.show()

**Step 2: Transform the Data**

# concatenate the dataframes

df = df1.union(df2)

# remove rows with missing values:

df.na.drop()

# Convert thne sales\_amount column to numeric

from pyspark.sql.functions import col

df=df.withColumn("sales\_amount", col("sales\_amount").cast("double"))

#Show the transformed dataframe

df.show()

**Step 3: Aggregate the Data**

aggregated\_df = df.groupBy("product\_id").sum("sales\_amount")

# Rename the aggregated column

aggregated\_df = aggregated\_df.withColumnRenamed("sum(sales\_amount)", "total\_sales")

#show the aggregated dataframe

aggregated\_df.show()

**Step 4: Visualize the Data in Fabric Lakehouse**

#Visualize - Convert Spark dataframe to pandas'

pands\_df=aggregated\_df.toPandas()

# Plot the data using pands

import matplotlib.pyplot as plt

#create a line chart

plt.figure(figsize=(10,6))

plt.plot(pands\_df["product\_id"], pands\_df["total\_sales"], marker=0)

plt.xlabel("Product Id")

plt.ylabel("Total sales")

plt.title("Total Sales by Product")

plt.grid(True)

plt.plot()

plt.show()

**Save to Delta Lake**

# Save the data as Delta Table

aggregated\_df.write.format("delta").save("Tables/sales\_data")

**Reading the Delta Table and Running SQL Command**:

# Reading the delta table

sales\_data\_df=spark.read.format("delta").load("Tables/sales\_data")

#Show the data

sales\_data\_df.show()

# Create a temporary view to run SQL queries

sales\_data\_df.createOrReplaceTempView("sales\_data\_view")

result\_df = spark.sql("select \* from sales\_data\_view")

result\_df.show()

**Save DataFrame as a Parquet File or Table:**

# Path to your CSV file in the Lakehouse

csv\_file\_path = "Files/sales\_data03.csv"

# Load the CSV file into a DataFrame

df = spark.read.csv(csv\_file\_path, header=True, inferSchema=True)

# Path to save the Parquet file in the Lakehouse

parquet\_file\_path = "Tables/sales\_data\_parquet"

# Save the DataFrame as a Parquet file

df.write.format("parquet").mode("overwrite").save(parquet\_file\_path)

# Save the DataFrame as a Parquet table

df.write.format("parquet").mode("overwrite").saveAsTable("sales\_data\_parquet\_table")