Ministry of Higher Education and Scientific Research
University of Abdelhafid Boussouf - Mila
Institute of Mathematics and Computer Science
Department of Computer Science
Master 2 I2A – Big Data
2025/2026

Directed Works TD 5 – Working with DataFrames and Spark SQL

Exercise 1: Creating DataFrames

Context: We want to analyze sales data stored in a CSV file called "**sales.csv**" with the columns: *Product, Category, Price, Quantity, Region*

Question 1: Complete the missing parts of the code below to create a Spark DataFrame:

```
from pyspark.sql import SparkSession
spark = SparkSession.builder.appName("SalesAnalysis").getOrCreate()
# TODO: Load the CSV file into a DataFrame
df = spark.read.option("header", True).____________("sales.csv")
# Display the first few rows
df.________()
```

Question 2: After loading, what will be the schema of this DataFrame?

Write the column names and their data types (assume Price and Quantity are numeric).

Exercise 2: Basic Transformations

Context: We now want to find the total revenue per product (Revenue = Price × Quantity).

Question 1: Fill in the blanks to create a new column called *"Revenue"*:

```
from pyspark.sql.functions import col
df2 = df.____(col("Price") * col("Quantity")).alias("Revenue"))
```

Question 2: Write the DataFrame code to select only the columns: *Product, Category, and Revenue*, and show the first 10 rows.

Question 3: If the DataFrame has 100,000 rows, and we call df2.show(10),

- Is this a transformation or an action?
- Briefly explain why.

Exercise 3: Aggregations

Context: We want to compute the total revenue per category.

Question 1: Complete this code fragment:

```
from pyspark.sql.functions import sum
category_revenue = df2.groupBy("Category").____(sum("Revenue").alias("TotalRevenue"))
```

Question 2: If Spark executes this, will it trigger a shuffle operation between nodes? Explain your reasoning in one or two sentences.

Exercise 4: Using Spark SQL

Context: The same DataFrame df2 is registered as a temporary SQL table.

```
df2.createOrReplaceTempView("sales")
```

Question 1: Write the SQL query to find the top 3 categories by total revenue.

Question 2: Translate your SQL query into equivalent DataFrame API code.

Exercise 5: Logical Reasoning

For each statement below, mark whether it is **TRUE** or **FALSE** and justify your answer:

a) filter() and where() perform the same operation in Spark.
b) The collect() action is safe to use on very large datasets.
c) Transformations in Spark are executed immediately when written.
d) The persist() and cache() methods both store intermediate results in memory.
e) Spark SQL queries are converted internally into DataFrame transformations.

Exercise 6: Mini Case Study

Context: A university is collecting academic performance data from multiple departments and campuses. The dataset is very large and distributed, containing millions of rows with the following structure:

Name, Subject, Score, City, Department, Year

Each record represents a student's grade in a specific subject.

You are asked to perform several analyses using Spark SQL and DataFrame operations to support decision-making about academic quality and regional performance.

Question 1: Write a Spark SQL query that calculates the average score per city and sorts the results from the highest to the lowest average.

Question 2: Now, the university wants to know which department in each city has the highest average score.

Write a SQL query (or describe the logic in pseudocode if needed) that provides:

City | Department | AverageScore

Question 3: If this dataset is accessed multiple times during different analyses (for example, one query per department), what Spark feature could you use to avoid re-reading the same data from disk each time? (Explain your answer briefly.)

Question 4: Suppose one executor fails while Spark is computing averages.

Explain how Spark's fault tolerance mechanism ensures the computation still completes successfully.

Question 5 (Open question):

If this analysis were done using traditional MapReduce, what main disadvantages would appear compared to Spark's DataFrame and SQL APIs?