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 6 – Data Cleaning, Joins, Window Functions & Intro to MLlib

## Exercise 1: Data Cleaning & Schema Reasoning

You are given a DataFrame sales with columns:
Product, Category, Price, Quantity, Region
(loaded from CSV → all strings)

**Question 1:** Infer the schema after cleaning

```
sales2 = sales.withColumn("Price", col("Price").cast("double")) \
              .withColumn("Quantity", col("Quantity").cast("int")) \
              .withColumn("Region", upper(trim(col("Region"))))
```

Write the resulting schema:

```
Column        Type
Product       ?
Category      ?
Price         ?
Quantity      ?
Region        ?
```

**Question 2:** Missing data count (complete the code)

```
from pyspark.sql.functions import sum, col

missing = sales.select(
    sum(col("Price").isNull().cast("int")).alias("MissingPrice"),
    sum(col("Quantity").isNull().cast("int")).alias("MissingQuantity"),
)
missing._______()
```

**Question** 3: Mean vs median imputation (short reasoning)
If the distribution of Quantity is highly skewed, explain why median imputation is better than mean, in 2–3 lines.

**Question** 4: Write a filter expression:
We want to keep only rows where:
Price > 0 AND Price ≤ 10,000
Complete:

```
df.filter( ____________________________________ )
```

### Exercise 2: Joins

You have:
• sales_clean
• products(*Product, ProductID, Supplier*)

**Question** 1: Choose the correct join
Which join keeps all rows from sales_clean and matches product info when possible?

- ☐ inner
- ☐ left
- ☐ right
- ☐ full

Explain briefly.

**Question** 2: Complete the join code

```
result = sales_clean.join(products, on="Product", how="________")
```

**Question** 3: Interpretation
What does this return?

```
products.join(sales_clean, "Product", "left_anti")
```

**Question** 4: Broadcast?
Products has 200 rows, sales_clean has 3 million.
Would you broadcast products for the join? YES or NO?
Explain briefly.

## Exercise 3: Window Functions

Given DataFrame df(*Product, Category, Region, Revenue*).

**Question** 1: Window specification

```
w = Window.partitionBy("Category").orderBy( ___________________ )
```

**Question** 2: Add a rank column

```
ranked = df.withColumn("Rank", ____________________________________ )
```

**Question** 3: Running sum window

```
w2 = Window.partitionBy("Region") \
           .orderBy("Product") \
           .rowsBetween( ____________________________ )
```

**Question** 4: Explanation
Why don't window functions reduce the number of rows?
Answer in one sentence.

## Exercise 4 – MLlib Introduction

**Dataset:**
*Age (int), Salary (double), Purchased (0/1)*

_____

**Question** 1: VectorAssembler inputs

```
assembler = VectorAssembler(
    inputCols=[ ________, ________ ],
    outputCol="features"
)
```

---

**Question** 2: Write the 3 MLlib steps (in order)

---

**Question** 3: Probability vs prediction
Explain the meaning of the two columns in 1–2 lines.

_____

**Question** 4: Two advantages of DataFrame-based MLlib
List 2 improvements over RDD-based MLlib.

_____

_____

## Exercise 5 – True / False

Indicate **TRUE** or **FALSE** and justify:
a) Window functions require a shuffle.
b) show() is an action.
c) broadcast() reduces shuffles during joins.
d) VectorAssembler always produces sparse vectors.
e) SQL queries in Spark are converted into physical plans by Catalyst.