LAB 1

SHORT NOTES

Full Name: Moukhlim Chaimaa
Siky Ibtissam
Nasserallah Salma
Aouidate Hassnaa

Module: Apache Spark – Data Processing

1. WHAT DOES THE DAG REPRESENT?

The DAG (Directed Acyclic Graph) represents the logical execution plan of a Spark job. It visualizes how data flows through a series of transformations, where each node corresponds to an operation and each edge represents a dependency. The DAG enables Spark to optimize execution by identifying which operations can be pipelined together and where data shuffles are required.

---

2. HOW MANY STAGES RAN DURING THE SCRIPT?

A total of **12 stages** were created during script execution:

- 10 completed stages
- 2 skipped stages (reused from cache)

Stages involving shuffle operations (such as aggregations) displayed non-zero Shuffle Write values, indicating data redistribution across partitions.

---

3. HOW MANY TASKS PER STAGE?

The number of tasks per stage ranged from **1 to 8**, depending on the number of data partitions:

Stage ID	Tasks Completed
0	*1*
1	*4*
2	*3*
3	*8*
6	*1*
7	*4*
8	*3*
9	*8*
11	*1*

In `local[*]` mode, Spark utilizes all available CPU cores (8 in this case), enabling parallel task execution.

---

4. WHAT DID YOU NOTICE IN THE EXECUTORS TAB?

The Executors tab displayed a single executor entry labeled "driver", which is expected in local mode where the driver process handles both coordination and task execution.

Key observations:

- **Cores:** 8
- **Storage Memory:** 139.3 KiB / 434.4 MiB
- **Total Tasks:** 34 completed
- **Shuffle Read/Write:** 937 B each
- **Status:** Active

---

5. WHAT PATTERN DID YOU OBSERVE IN JOB TRIGGERING?

Each action triggered a separate Spark job, while transformations remained lazy until an action was invoked. The script executed 4 actions, resulting in 4 distinct jobs:

Action	Job Triggered
`df.show()`	*Job 0*
`df.agg(avg()).show()`	*Job 1*
`df.filter().show()`	*Job 2*
`df.count()`	*Job 3*

This confirms Spark's lazy evaluation model: transformations are only executed when an action forces computation.

SUMMARY

Aspect	Observation
DAG	*Logical execution plan showing operations and dependencies*
Stages	*12 total (10 completed, 2 skipped)*
Tasks per Stage	*1 to 8 tasks based on partition count*
Executors	*1 executor (driver) with 8 cores*
Job Triggering	*Each action triggers a new job; transformations are lazy*