Lab 4: HBase

1. **Chargement de fichiers**
   - **Enregistrement du fichier purchases_2.txt**

```
root@hadoop-master:~# hadoop fs -put /shared_volume/purchases_2.txt input
root@hadoop-master:~# hadoop fs -ls input
Found 2 items
-rw-r--r--   2 root supergroup       2549 2025-11-13 10:36 input/purchases2.txt
-rw-r--r--   2 root supergroup  243309611 2025-11-15 11:24 input/purchases_2.txt
root@hadoop-master:~# hbase shell
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hbase/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.cla
]
SLF4J: Found binding in [jar:file:/usr/local/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/S
ticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
HBase Shell
Use "help" to get list of supported commands.
Use "exit" to quit this interactive shell.
Version 1.4.12, r6ae4a77408ad35d6a7a4e5cebfd401fc4b72b5ec, Sun Nov 24 13:25:41 CST 2019

hbase(main):001:0>
```

   - **hbase org.apache.hadoop.hbase.mapreduce.ImportTsv **
     **-libjars /usr/local/hbase/lib/commons-lang-2.6.jar **
     **-Dimporttsv.separator=',' **
     -
     Dimporttsv.columns=HBASE_ROW_KEY,cf:date,cf:time,cf:town,cf:pro
     **duct,cf:price,cf:payment **
     products /user/root/input/purchases_2.txt

```
2025-11-15 11:41:40,383 INFO  [main] mapreduce.Job:  map 73% reduce 0%
2025-11-15 11:41:46,466 INFO  [main] mapreduce.Job:  map 79% reduce 0%
2025-11-15 11:41:52,564 INFO  [main] mapreduce.Job:  map 84% reduce 0%
2025-11-15 11:41:58,683 INFO  [main] mapreduce.Job:  map 89% reduce 0%
2025-11-15 11:42:01,804 INFO  [main] mapreduce.Job:  map 91% reduce 0%
2025-11-15 11:42:04,835 INFO  [main] mapreduce.Job:  map 94% reduce 0%
2025-11-15 11:42:10,925 INFO  [main] mapreduce.Job:  map 97% reduce 0%
2025-11-15 11:42:15,997 INFO  [main] mapreduce.Job:  map 100% reduce 0%
2025-11-15 11:42:16,026 INFO  [main] mapreduce.Job: Job job_1763152930623_0002 completed successfully
2025-11-15 11:42:16,313 WARN  [main] counters.FileSystemCounterGroup: HDFS_BYTES_READ_EC is not a recognized counter.
2025-11-15 11:42:16,366 WARN  [main] counters.FrameworkCounterGroup: MAP_PHYSICAL_MEMORY_BYTES_MAX is not a recognized counter.
2025-11-15 11:42:16,366 WARN  [main] counters.FrameworkCounterGroup: MAP_VIRTUAL_MEMORY_BYTES_MAX is not a recognized counter.
2025-11-15 11:42:16,404 INFO  [main] mapreduce.Job:  read=243313951
                HDFS: Number of bytes written=0
                HDFS: Number of read operations=4
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=0
        Job Counters
                Launched map tasks=2
                Data-local map tasks=2
                Total time spent by all maps in occupied slots (ms)=236058
                Total time spent by all reduces in occupied slots (ms)=0
                Total time spent by all map tasks (ms)=236058
                Total vcore-milliseconds taken by all map tasks=236058
                Total megabyte-milliseconds taken by all map tasks=241723392
        Map-Reduce Framework
                Map input records=4138476
                Map output records=4138476
                Input split bytes=244
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=5668
                CPU time spent (ms)=109590
                Physical memory (bytes) snapshot=500940800
                Virtual memory (bytes) snapshot=3958550528
                Total committed heap usage (bytes)=321388544
        ImportTsv
                Bad Lines=0
        File Input Format Counters
                Bytes Read=243313707
```

     Vérification :

```
hbase(main):001:0> get 'products','8',{COLUMN => 'cf:town'}
COLUMN                          CELL
 cf:town                        timestamp=1763206795282, value=New York
1 row(s) in 0.3010 seconds
```

2. **Traitement de données avec Spark :**

```
java.lang.IllegalStateException: Received event is not valid: Closed
        at org.apache.hadoop.hbase.zookeeper.ZooKeeperWatcher.connectionEvent(ZooKeeperWatcher.java:702)
        at org.apache.hadoop.hbase.zookeeper.ZooKeeperWatcher.process(ZooKeeperWatcher.java:624)
        at org.apache.hadoop.hbase.zookeeper.PendingWatcher.process(PendingWatcher.java:40)
        at org.apache.zookeeper.ClientCnxn$EventThread.processEvent(ClientCnxn.java:587)
        at org.apache.zookeeper.ClientCnxn$EventThread.run(ClientCnxn.java:562)
2025-11-15 12:09:24,362 INFO zookeeper.ZooKeeper: Session: 0x19a841fd8dc001c closed
2025-11-15 12:09:24,362 INFO zookeeper.ClientCnxn: EventThread shut down for session: 0x19a841fd8dc001c
2025-11-15 12:09:24,409 INFO executor.Executor: Finished task 0.0 in stage 0.0 (TID 0). 1004 bytes result sent to driver
2025-11-15 12:09:24,439 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 0.0 (TID 0) in 17674 ms on hadoop-master (executor driver) (1/2)
2025-11-15 12:09:29,615 INFO client.ConnectionManager$HConnectionImplementation: Closing zookeeper sessionid=0x19a841fd8dc001b
2025-11-15 12:09:29,731 ERROR zookeeper.ClientCnxn: Error while calling watcher
java.lang.IllegalStateException: Received event is not valid: Closed
        at org.apache.hadoop.hbase.zookeeper.ZooKeeperWatcher.connectionEvent(ZooKeeperWatcher.java:702)
        at org.apache.hadoop.hbase.zookeeper.ZooKeeperWatcher.process(ZooKeeperWatcher.java:624)
        at org.apache.hadoop.hbase.zookeeper.PendingWatcher.process(PendingWatcher.java:40)
        at org.apache.zookeeper.ClientCnxn$EventThread.processEvent(ClientCnxn.java:587)
        at org.apache.zookeeper.ClientCnxn$EventThread.run(ClientCnxn.java:562)
2025-11-15 12:09:29,731 INFO zookeeper.ZooKeeper: Session: 0x19a841fd8dc001b closed
2025-11-15 12:09:29,732 INFO zookeeper.ClientCnxn: EventThread shut down for session: 0x19a841fd8dc001b
2025-11-15 12:09:29,750 INFO executor.Executor: Finished task 1.0 in stage 0.0 (TID 1). 961 bytes result sent to driver
2025-11-15 12:09:29,757 INFO scheduler.TaskSetManager: Finished task 1.0 in stage 0.0 (TID 1) in 22933 ms on hadoop-master (executor driver) (2/2)
2025-11-15 12:09:29,765 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 0.0, whose tasks have all completed, from pool
2025-11-15 12:09:29,771 INFO scheduler.DAGScheduler: ResultStage 0 (count at HbaseSparkProcess.java:36) finished in 23.387 s
2025-11-15 12:09:29,788 INFO scheduler.DAGScheduler: Job 0 is finished. Cancelling potential speculative or zombie tasks for this job
2025-11-15 12:09:29,790 INFO scheduler.TaskSchedulerImpl: Killing all running tasks in stage 0: Stage finished
2025-11-15 12:09:29,800 INFO scheduler.DAGScheduler: Job 0 finished: count at HbaseSparkProcess.java:36, took 23.740180 s
Nombre d'enregistrements dans 'products' : 4138476
2025-11-15 12:09:29,837 INFO server.AbstractConnector: Stopped Spark@3e1162e7{HTTP/1.1, (http/1.1)}{0.0.0.0:4040}
2025-11-15 12:09:29,847 INFO ui.SparkUI: Stopped Spark web UI at http://hadoop-master:4040
2025-11-15 12:09:29,899 INFO spark.MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
2025-11-15 12:09:29,937 INFO memory.MemoryStore: MemoryStore cleared
2025-11-15 12:09:29,938 INFO storage.BlockManager: BlockManager stopped
2025-11-15 12:09:29,951 INFO storage.BlockManagerMaster: BlockManagerMaster stopped
2025-11-15 12:09:29,956 INFO scheduler.OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped!
2025-11-15 12:09:29,979 INFO spark.SparkContext: Successfully stopped SparkContext
2025-11-15 12:09:29,985 INFO util.ShutdownHookManager: Shutdown hook called
2025-11-15 12:09:29,986 INFO util.ShutdownHookManager: Deleting directory /tmp/spark-ddba36b0-cdf3-4271-a501-2f85e5f27a49
2025-11-15 12:09:29,996 INFO util.ShutdownHookManager: Deleting directory /tmp/spark-63f7478b-b17c-4847-803b-5e06941f7eed
```

3. cat > HbaseSparkSum.java << 'EOF'

```
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.hbase.HBaseConfiguration;
import org.apache.hadoop.hbase.client.Result;
import org.apache.hadoop.hbase.io.ImmutableBytesWritable;
import org.apache.hadoop.hbase.mapreduce.TableInputFormat;
import org.apache.hadoop.hbase.util.Bytes;
import org.apache.spark.SparkConf;
import org.apache.spark.api.java.JavaPairRDD;
import org.apache.spark.api.java.JavaRDD;
import org.apache.spark.api.java.JavaSparkContext;

import java.math.BigDecimal;
import java.util.Optional;

public class HbaseSparkSum {

  // Fonction de parsing de prix
  private static Optional<BigDecimal> parsePrice(Result r) {
    try {
      byte[] rawVal = r.getValue(Bytes.toBytes("cf"), Bytes.toBytes("price"));
      if (rawVal == null) return Optional.empty();

      String priceStr = Bytes.toString(rawVal).trim();

      if (priceStr.isEmpty()) return Optional.empty();
```

```
      // Nettoyage : enlever $, espaces, virgules de milliers
      priceStr = priceStr.replaceAll("[^0-9.,-]", "");
      priceStr = priceStr.replace(",", "");

      if (priceStr.isEmpty()) return Optional.empty();

      return Optional.of(new BigDecimal(priceStr));

    } catch (Exception ex) {
      return Optional.empty();
    }
  }

  public void runJob() {
    // Config HBase
    Configuration config = HBaseConfiguration.create();

    // Config Spark
    SparkConf sconf = new SparkConf()
        .setAppName("SparkHBaseSum")
        .setMaster("local[4]");

    JavaSparkContext jsc = new JavaSparkContext(sconf);

    // Lire la table HBase
    config.set(TableInputFormat.INPUT_TABLE, "products");

    JavaPairRDD<ImmutableBytesWritable, Result> hBaseRDD =
        jsc.newAPIHadoopRDD(
            config,
            TableInputFormat.class,
            ImmutableBytesWritable.class,
            Result.class
        );

    // Extraire les prix
    JavaRDD<BigDecimal> prices = hBaseRDD
        .values()
        .map(result -> parsePrice(result).orElse(BigDecimal.ZERO));

    // Faire la somme
```

```
    BigDecimal total = prices.reduce(BigDecimal::add);

    System.out.println("====================================");
    System.out.println("  SOMME TOTALE DES VENTES : " + total);
    System.out.println("====================================");

    jsc.close();
  }

  public static void main(String[] args) {
    new HbaseSparkSum().runJob();
  }
}
EOF
```