# TP HBASE

# HABACH Abdelghafour

## Partie 1 :

```
ROW                    COLUMN+CELL
 101                   column=customer:city, timestamp=1763025688721, value=Los An
                       geles, CA
 101                   column=customer:name, timestamp=1763025688617, value=John W
                       hite
 101                   column=sales:amount, timestamp=1763025691959, value=$400.00
 101                   column=sales:product, timestamp=1763025688785, value=Chairs
 102                   column=customer:city, timestamp=1763025702326, value=Atlant
                       a, GA
 102                   column=customer:name, timestamp=1763025702229, value=Jane B
                       rown
 102                   column=sales:amount, timestamp=1763025708816, value=$200.00
 102                   column=sales:product, timestamp=1763025702384, value=Lamps
 103                   column=customer:city, timestamp=1763025711623, value=Pittsb
                       urgh, PA
 103                   column=customer:name, timestamp=1763025711554, value=Bill G
                       reen
```

```
hbase(main):001:0> scan 'sales_ledger'
ROW                    COLUMN+CELL
 101                      column=customer:city, timestamp=1763025688721, value=Los Angeles, CA
 101                      column=customer:name, timestamp=1763025688617, value=John White
 101                      column=sales:amount, timestamp=1763025691959, value=$400.00
 101                      column=sales:product, timestamp=1763025688785, value=Chairs
 102                      column=customer:city, timestamp=1763025702326, value=Atlanta, GA
 102                      column=customer:name, timestamp=1763025702229, value=Jane Brown
 102                      column=sales:amount, timestamp=1763025708816, value=$200.00
 102                      column=sales:product, timestamp=1763025702384, value=Lamps
 103                      column=customer:city, timestamp=1763025711623, value=Pittsburgh, PA
 103                      column=customer:name, timestamp=1763025711554, value=Bill Green
 103                      column=sales:amount, timestamp=1763025711818, value=$500.00
 103                      column=sales:product, timestamp=1763025711665, value=Desk
 104                      column=customer:city, timestamp=1763025721164, value=St. Louis, MO
 104                      column=customer:name, timestamp=1763025721116, value=Jack Black
 104                      column=sales:amount, timestamp=1763025722234, value=$1,600.00
```

```
hbase> t.get 'r1'
hbase> t.get 'r1', {TIMERANGE => [ts1, ts2]}
hbase> t.get 'r1', {COLUMN => 'c1'}
hbase> t.get 'r1', {COLUMN => ['c1', 'c2', 'c3']}
hbase> t.get 'r1', {COLUMN => 'c1', TIMESTAMP => ts1}
hbase> t.get 'r1', {COLUMN => 'c1', TIMERANGE => [ts1, ts2], VERSIONS => 4}
hbase> t.get 'r1', {COLUMN => 'c1', TIMESTAMP => ts1, VERSIONS => 4}
hbase> t.get 'r1', {FILTER => "ValueFilter(=, 'binary:abc')"}
hbase> t.get 'r1', 'c1'
hbase> t.get 'r1', 'c1', 'c2'
hbase> t.get 'r1', ['c1', 'c2']
hbase> t.get 'r1', {CONSISTENCY => 'TIMELINE'}
hbase> t.get 'r1', {CONSISTENCY => 'TIMELINE', REGION_REPLICA_ID => 1}
```

**.HBase API**

**J'ai Travaille avec purchases.text , type : TSV**

**(50 enregistrements)**

**La commande est la suivante :**

hbase org.apache.hadoop.hbase.mapreduce.ImportTsv \

-libjars /usr/local/hbase/lib/commons-lang-2.6.jar \

-Dimporttsv.columns=HBASE_ROW_KEY,cf:date,cf:time,cf:town,cf:product,cf:price,cf:payment \

products /user/root/input/purchases.txt

```
hbase(main):001:0> get 'products' , '2012-01-01'
COLUMN                    CELL
 cf:date                  timestamp=1763030455612, value=09:02
 cf:price                 timestamp=1763030455612, value=Discover
 cf:product               timestamp=1763030455612, value=344.09
 cf:time                  timestamp=1763030455612, value=Chandler
 cf:town                  timestamp=1763030455612, value=Books
```

```
hbase(main):002:0> get 'products', '2012-01-01', {COLUMN => 'cf:town'}
COLUMN                    CELL
 cf:town                  timestamp=1763030455612, value=Books
1 row(s) in 0.0270 seconds
```

**Traitement de données avec Spark**

```
2025-11-13 11:19:35,771 INFO client.TransportClientFactory: Successfully created connection to hado
op-master/172.19.0.2:45429 after 27 ms (0 ms spent in bootstraps)
2025-11-13 11:19:35,777 INFO util.Utils: Fetching spark://hadoop-master:45429/jars/processing-hbase
.jar to /tmp/spark-40bcbca0-6687-4d6a-9463-c3eac0bf8755/userFiles-f2eafd6a-a0a4-4d79-bf57-417577c75
d94/fetchFileTemp1192361426579634617.tmp
2025-11-13 11:19:35,807 INFO executor.Executor: Adding file:/tmp/spark-40bcbca0-6687-4d6a-9463-c3ea
c0bf8755/userFiles-f2eafd6a-a0a4-4d79-bf57-417577c75d94/processing-hbase.jar to class loader
2025-11-13 11:19:35,820 INFO util.Utils: Successfully started service 'org.apache.spark.network.net
ty.NettyBlockTransferService' on port 45757.
2025-11-13 11:19:35,820 INFO netty.NettyBlockTransferService: Server created on hadoop-master:45757
2025-11-13 11:19:35,823 INFO storage.BlockManager: Using org.apache.spark.storage.RandomBlockReplic
ationPolicy for block replication policy
2025-11-13 11:19:35,833 INFO storage.BlockManagerMaster: Registering BlockManager BlockManagerId(dr
iver, hadoop-master, 45757, None)
```

```
2025-11-13 11:19:34,494 INFO resource.ResourceUtils: =================================================
=================
2025-11-13 11:19:34,495 INFO spark.SparkContext: Submitted application: SparkHBaseTest
2025-11-13 11:19:34,529 INFO resource.ResourceProfile: Default ResourceProfile created, executor re
sources: Map(cores -> name: cores, amount: 1, script: , vendor: , memory -> name: memory, amount: 2
048, script: , vendor: , offHeap -> name: offHeap, amount: 0, script: , vendor: ), task resources:
Map(cpus -> name: cpus, amount: 1.0)
2025-11-13 11:19:34,549 INFO resource.ResourceProfile: Limiting resource is cpus at 1 tasks per exe
cutor
```