Custom Learnings

Day 18

Pyspark

Use of persist method to serialize the data.

To manually unpersist, unpersist() method is used.

In the latest Spark versions, the unpersist happens automatically.

Least recently used algorithm.

CPU time is higher as to read files the data has to be deserialized.

Write Operations in Spark:

If write operation happens to a table where the db is mentioned, then it gets saved in the default database.

test= spark.sql("describe extended emp_tbl")

test.show()

the default datatype of this table is managed table.

Managed by databricks itself.

Managed table stores metadata and the actual data. Both are present in the Spark environment itself.

In external table, custom paths are given to store the actual data.

In [23]:
```
import findspark
findspark.init()
from pyspark.sql import SparkSession
from pyspark.sql.functions import *
#Initilize Sparksession
spark = SparkSession.builder.getOrCreate()
sc=spark.sparkContext
test_df = [("James","Sales","NY",90000,34,10000),
    ("Michael","Sales","NY",86000,56,20000),
    ("Robert","Sales","CA",81000,30,23000),
    ("Maria","Finance","CA",90000,24,23000),
    ("Raman","Finance","CA",99000,40,24000),
    ("Scott","Finance","NY",83000,36,19000),
    ("Jen","Finance","NY",79000,53,15000),
    ("Jeff","Marketing","CA",80000,25,18000),
    ("Kumar","Marketing","NY",91000,50,21000)
  ]
```

In [24]:
```
ud_scehma = ["employee_name","department","state","salary","age","bonus"]

df = spark.createDataFrame(data=test_df,schema = ud_scehma)
```

In [25]:
```
df.cache().count()
```

Out[25]:
```
9
```

In [26]:
```
df.show()
```

```
+-------------+----------+-----+------+---+-----+
|employee_name|department|state|salary|age|bonus|
+-------------+----------+-----+------+---+-----+
|        James|     Sales|   NY| 90000| 34|10000|
|      Michael|     Sales|   NY| 86000| 56|20000|
|       Robert|     Sales|   CA| 81000| 30|23000|
|        Maria|   Finance|   CA| 90000| 24|23000|
|        Raman|   Finance|   CA| 99000| 40|24000|
|        Scott|   Finance|   NY| 83000| 36|19000|
|          Jen|   Finance|   NY| 79000| 53|15000|
|         Jeff| Marketing|   CA| 80000| 25|18000|
|        Kumar| Marketing|   NY| 91000| 50|21000|
+-------------+----------+-----+------+---+-----+
```

In [27]:
```
from pyspark import StorageLevel
```

In [28]:
```
test = StorageLevel(useDisk=False, useMemory=True,useOffHeap=False, deserialized=False, replication=2)
```

In [29]:
```
df.persist(storageLevel=test)
```

```
23/09/25 04:45:36 WARN CacheManager: Asked to cache already cached data.
```

Out[29]:
```
DataFrame[employee_name: string, department: string, state: string, salary: bigint, age: bigint, bonus: bigint]
```

In [30]:
```
df.unpersist()
```

Out[30]:
```
DataFrame[employee_name: string, department: string, state: string, salary: bigint, age: bigint, bonus: bigint]
```

In [32]:
```
df.groupBy("state").max("salary").alias("max_sal_by_state").show()
```

```
[Stage 10:>                                                         (0 + 2) / 2]
```

```
+-----+-----------+
|state|max(salary)|
+-----+-----------+
|   CA|      99000|
|   NY|      91000|
+-----+-----------+
```

In [33]:
```
df.groupBy("department").max("salary").alias("max_sal_by_state").show()
```

```
+----------+-----------+
|department|max(salary)|
+----------+-----------+
|     Sales|      90000|
|   Finance|      99000|
| Marketing|      91000|
+----------+-----------+
In [39]: df.groupBy("department","state").agg(sum("salary").alias("sum_salary"),
                                      avg("salary").alias("avg_salary")) \
         .where(col("avg_salary")>80000).show()

         +----------+-----+----------+----------+
         |department|state|sum_salary|avg_salary|
         +----------+-----+----------+----------+
         |     Sales|   CA|     81000|   81000.0|
         |   Finance|   CA|    189000|   94500.0|
         |     Sales|   NY|    176000|   88000.0|
         |   Finance|   NY|    162000|   81000.0|
         | Marketing|   NY|     91000|   91000.0|
         +----------+-----+----------+----------+
```

```
In [37]: df.createOrReplaceTempView("ida")
```

```
In [40]: spark.sql("select * from ida")

Out[40]: DataFrame[employee_name: string, department: string, state: string, salary: bigint, age: bigint, bonus: bigint]
```

```
In [41]: df.write.saveAsTable("emp_tbl")
```

```
In [42]: test = spark.sql("DESCRIBE emp_tbl")
```

```
In [43]: test.show()

         +-------------+---------+-------+
         |     col_name|data_type|comment|
         +-------------+---------+-------+
         |employee_name|   string|   null|
         |   department|   string|   null|
         |        state|   string|   null|
         |       salary|   bigint|   null|
         |          age|   bigint|   null|
         |        bonus|   bigint|   null|
         +-------------+---------+-------+
```

```
In [44]: test01=spark.sql("DESCRIBE EXTENDED emp_tbl")
         test01.show()

         +--------------------+--------------------+-------+
         |            col_name|           data_type|comment|
         +--------------------+--------------------+-------+
         |       employee_name|              string|   null|
         |          department|              string|   null|
         |               state|              string|   null|
         |              salary|              bigint|   null|
         |                 age|              bigint|   null|
         |               bonus|              bigint|   null|
         |                    |                    |       |
         |# Detailed Table ...|                    |       |
         |             Catalog|       spark_catalog|       |
         |            Database|             default|       |
         |               Table|             emp_tbl|       |
         |        Created Time|Mon Sep 25 05:14:...|       |
         |         Last Access|             UNKNOWN|       |
         |          Created By|         Spark 3.4.1|       |
         |                Type|             MANAGED|       |
         |            Provider|             parquet|       |
         |            Location|file:/home/labuse...|       |
         +--------------------+--------------------+-------+
```

```
In [46]: spark.sql("CREATE DATABASE idashell")

Out[46]: DataFrame[]
```

```
In [47]: spark.sql("use idashell")

Out[47]: DataFrame[]
```

```
In [48]: df.write.saveAsTable("test_tbl")
In [49]: test01=spark.sql("DESCRIBE EXTENDED test_tbl")
         test01.show()
```

```
+--------------------+--------------------+-------+
|            col_name|           data_type|comment|
+--------------------+--------------------+-------+
|       employee_name|              string|   null|
|          department|              string|   null|
|               state|              string|   null|
|              salary|              bigint|   null|
|                 age|              bigint|   null|
|               bonus|              bigint|   null|
|                    |                    |       |
|# Detailed Table ...|                    |       |
|             Catalog|       spark_catalog|       |
|            Database|            idashell|       |
|               Table|            test_tbl|       |
|        Created Time|Mon Sep 25 05:22:...|       |
|         Last Access|             UNKNOWN|       |
|          Created By|         Spark 3.4.1|       |
|                Type|             MANAGED|       |
|            Provider|             parquet|       |
|            Location|file:/home/labuse...|       |
+--------------------+--------------------+-------+
```

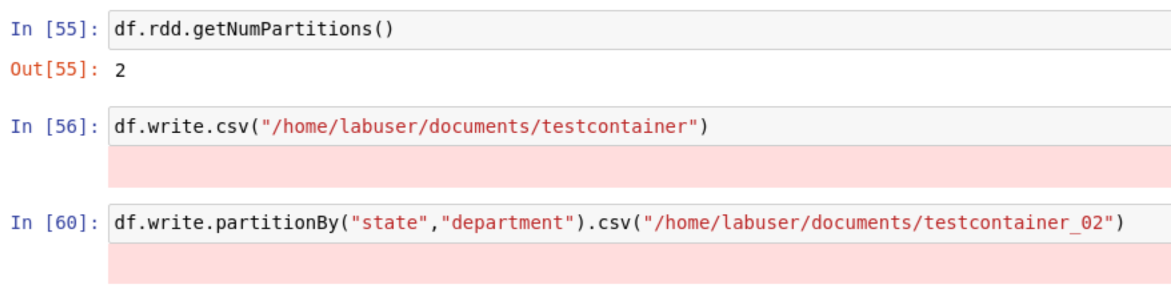
```
In [51]: df.write.option("path", "/home/labuser/Documents/my_tbl_data").saveAsTable("e_emptbl")
```

```
In [52]: test01=spark.sql("DESCRIBE EXTENDED e_emptbl")
         test01.show()
```

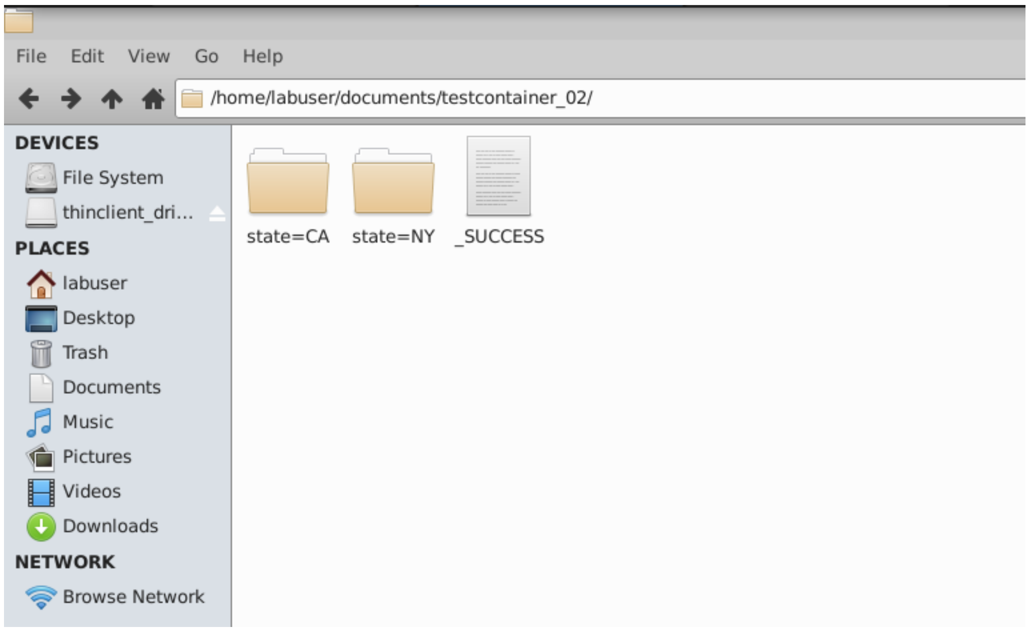
```
+--------------------+--------------------+-------+
|            col_name|           data_type|comment|
+--------------------+--------------------+-------+
|       employee_name|              string|   null|
|          department|              string|   null|
|               state|              string|   null|
|              salary|              bigint|   null|
|                 age|              bigint|   null|
|               bonus|              bigint|   null|
|                    |                    |       |
|# Detailed Table ...|                    |       |
|             Catalog|       spark_catalog|       |
|            Database|            idashell|       |
|               Table|            e_emptbl|       |
|        Created Time|Mon Sep 25 05:34:...|       |
|         Last Access|             UNKNOWN|       |
|          Created By|         Spark 3.4.1|       |
|                Type|            EXTERNAL|       |
|            Provider|             parquet|       |
|            Location|file:///home/labu...|       |
+--------------------+--------------------+-------+
```

```
In [53]: spark.sql("drop table default.emp_tbl")
Out[53]: DataFrame[]
```

```
In [54]: spark.sql("drop table idashell.e_emptbl")
Out[54]: DataFrame[]
```

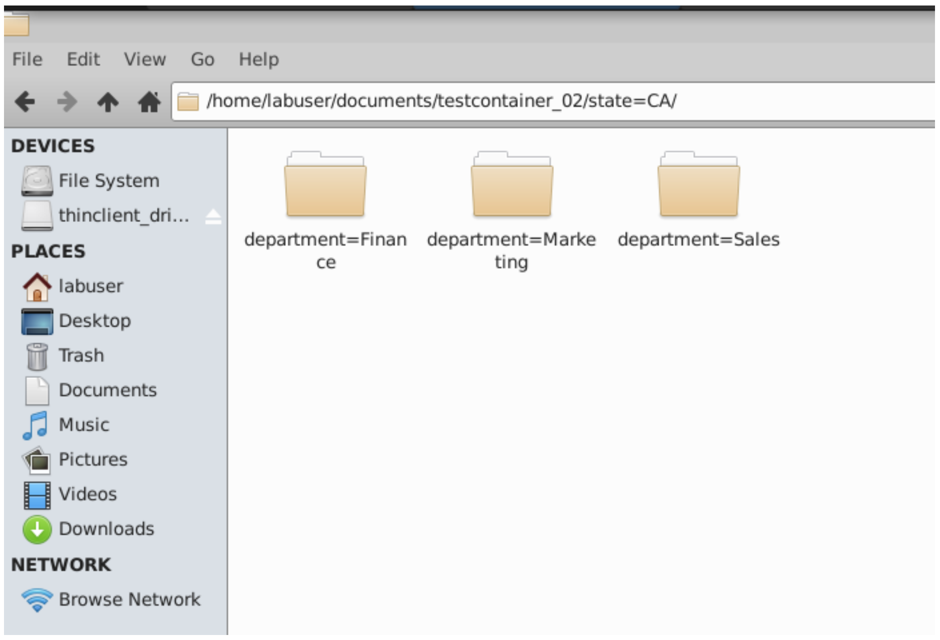


```
In [55]: df.rdd.getNumPartitions()
Out[55]: 2

In [56]: df.write.csv("/home/labuser/documents/testcontainer")

In [60]: df.write.partitionBy("state","department").csv("/home/labuser/documents/testcontainer_02")
```



/home/labuser/documents/testcontainer_02/

state=CA  state=NY  _SUCCESS



/home/labuser/documents/testcontainer_02/state=CA/

department=Finance  department=Marketing  department=Sales

In [45]:
```
json_data = [
    '{"name": "Alice", "age": 25}',
    '{"name": "Bob", "age": 30}',
    '{"name": "Charlie", "age": 35}'
]

from pyspark.sql.functions import *
from pyspark.sql.types import *
schema = StructType([
    StructField("name", StringType(), True),
    StructField("age", IntegerType(), True)
])

df = spark.read.schema(schema).json(spark.sparkContext.parallelize(json_data))
df.show()
```

```
+-------+---+
|   name|age|
+-------+---+
|  Alice| 25|
|    Bob| 30|
|Charlie| 35|
+-------+---+
```

In [46]:
```
json_data = [
    '{"name": "Alice", "age": 25, "address": {"city": "New York", "state": "NY"}}',
    '{"name": "Bob", "age": 30, "address": {"city": "San Francisco", "state": "CA"}}',
    '{"name": "Charlie", "age": 35, "address": {"city": "Los Angeles", "state": "CA"}}'
]

schema = StructType([
    StructField("name", StringType(), True),
    StructField("age", IntegerType(), True),
    StructField("address", StructType([
        StructField("city", StringType(), True),
        StructField("state", StringType(), True)
    ]), True)
])


df = spark.read.schema(schema).json(spark.sparkContext.parallelize(json_data))
df.show()
```

```
+-------+---+-------------------+
|   name|age|            address|
+-------+---+-------------------+
|  Alice| 25|     {New York, NY}|
|    Bob| 30|{San Francisco, CA}|
|Charlie| 35|  {Los Angeles, CA}|
+-------+---+-------------------+
```

In [ ]: