databricks practica_a_solucion

(https://databricks.com)

Semana 6 - Práctica

2

```
import org.apache.spark.sql.types.{StructType, StructField}
import org.apache.spark.sql.types.{StringType, IntegerType, DoubleType}
import org.apache.spark.sql.types._
import org.apache.spark.sql.{functions => f}
import org.apache.spark.sql.functions.udf
import org.apache.spark.sql.SparkSession
import org.apache.spark.sql.DataFrame

//Reserva de recursos
var spark = SparkSession.builder.
appName("Mi Aplicacion").
config("spark.driver.memory", "1000g"). // memoria maxima del cluster
config("spark.dynamicAllocation.maxExecutors", "100"). //numero maximo de ejecutores dinamicos
config("spark.executor.cores", "4"). //cantidad de nucleos
config("spark.executor.memory", "5g"). //memoria asignada a cada ejecutor
config("spark.executor.memoryOverhead", "100g"). //memoria de reserva
config("spark.default.parallelism", "100"). //trabajos en paralelo que se pueden realizar
getOrCreate()
```

```
import org.apache.spark.sql.types.{StructType, StructField}
import org.apache.spark.sql.types.{StringType, IntegerType, DoubleType}
import org.apache.spark.sql.types._
import org.apache.spark.sql.{functions=>f}
import org.apache.spark.sql.functions.udf
import org.apache.spark.sql.SparkSession
import org.apache.spark.sql.DataFrame
spark: org.apache.spark.sql.SparkSession = org.apache.spark.sql.SparkSession@219d353b
```

Parte 1

4

```
//Leemos el archivo de persona
var dfRiesgo = spark.read.format("csv").option("header", "true").option("delimiter", ";").schema(
    StructType(
        Array(
            StructField("ID_CLIENTE", StringType, true),
            StructField("RIESGO_CENTRAL_1", DoubleType, true),
            StructField("RIESGO_CENTRAL_2", DoubleType, true),
            StructField("RIESGO_CENTRAL_3", DoubleType, true)
        )
    )
).load("/FileStore/tables/RIESGO_CREDITICIO.csv")
```

▸ dfRiesgo: org.apache.spark.sql.DataFrame = [ID_CLIENTE: string, RIESGO_CENTRAL_1: double ... 2 more fields]

```
dfRiesgo: org.apache.spark.sql.DataFrame = [ID_CLIENTE: string, RIESGO_CENTRAL_1: double ... 2 more fields]
```

5

```
val jsonDF: DataFrame = spark.read
  .option("inferSchema", "true") // Inferir el esquema automáticamente
  .json("/FileStore/tables/transacciones_bancarias.json")

jsonDF.printSchema()
```

▸ jsonDF: org.apache.spark.sql.DataFrame = [EMPRESA: struct, PERSONA: struct ... 1 more field]

```
root
 |-- EMPRESA: struct (nullable = true)
```

```
 |    |-- ID_EMPRESA: string (nullable = true)
 |    |-- NOMBRE_EMPRESA: string (nullable = true)
 |-- PERSONA: struct (nullable = true)
 |    |-- EDAD: long (nullable = true)
 |    |-- ID_PERSONA: string (nullable = true)
 |    |-- NOMBRE_PERSONA: string (nullable = true)
 |    |-- SALARIO: double (nullable = true)
 |-- TRANSACCION: struct (nullable = true)
 |    |-- FECHA: string (nullable = true)
 |    |-- MONTO: double (nullable = true)

jsonDF: org.apache.spark.sql.DataFrame = [EMPRESA: struct<ID_EMPRESA: string, NOMBRE_EMPRESA: string>, PERSONA: struc
t<EDAD: bigint, ID_PERSONA: string ... 2 more fields> ... 1 more field]
```

Parte 2

7

```
import org.apache.spark.sql.functions.col

val dfTransaccion =
jsonDF.select(
  col("EMPRESA.ID_EMPRESA").alias("ID_EMPRESA"),
  col("PERSONA.ID_PERSONA").alias("ID_PERSONA"),
  col("TRANSACCION.MONTO").alias("MONTO"),
  col("TRANSACCION.FECHA").alias("FECHA")
)

val dfEmpresa =
jsonDF.select(
  col("EMPRESA.ID_EMPRESA").alias("ID_EMPRESA"),
  col("EMPRESA.NOMBRE_EMPRESA").alias("NOMBRE_EMPRESA")
)

val dfPersona =
jsonDF.select(
  col("PERSONA.ID_PERSONA").alias("ID_PERSONA"),
  col("PERSONA.NOMBRE_PERSONA").alias("NOMBRE_PERSONA"),
  col("PERSONA.EDAD").alias("EDAD"),
  col("PERSONA.SALARIO").alias("SALARIO"),
)
```

- dfEmpresa: org.apache.spark.sql.DataFrame = [ID_EMPRESA: string, NOMBRE_EMPRESA: string]
- dfPersona: org.apache.spark.sql.DataFrame = [ID_PERSONA: string, NOMBRE_PERSONA: string ... 2 more fields]
- dfTransaccion: org.apache.spark.sql.DataFrame = [ID_EMPRESA: string, ID_PERSONA: string ... 2 more fields]

```
import org.apache.spark.sql.functions.col
dfTransaccion: org.apache.spark.sql.DataFrame = [ID_EMPRESA: string, ID_PERSONA: string ... 2 more fields]
dfEmpresa: org.apache.spark.sql.DataFrame = [ID_EMPRESA: string, NOMBRE_EMPRESA: string]
dfPersona: org.apache.spark.sql.DataFrame = [ID_PERSONA: string, NOMBRE_PERSONA: string ... 2 more fields]
```

Parte 3

9

```
//Limpieza e la tabla persona
val filteredDfPersona = dfPersona.filter($"EDAD".between(0, 60) && $"SALARIO".between(0, 1000000) &&
$"ID_PERSONA".isNotNull)
filteredDfPersona.show()

//Limpieza e la tabla empresa
val filteredDfEmpresa = dfEmpresa.filter($"ID_EMPRESA".isNotNull)
filteredDfEmpresa.show()

//Limpieza e la tabla transaccion
val filteredDfTransaccion = dfTransaccion.filter($"ID_PERSONA".isNotNull && $"MONTO".between(0, 1000000) &&
$"ID_PERSONA".isNotNull)
filteredDfTransaccion.show()

//Limpieza e la tabla riesgo
val filteredDfRiesgo = dfRiesgo.filter($"ID_CLIENTE".isNotNull && $"RIESGO_CENTRAL_1".between(0,1) &&
$"RIESGO_CENTRAL_2".between(0,1) && $"RIESGO_CENTRAL_3".between(0,1))
filteredDfRiesgo.show()
```

▸ filteredDfEmpresa: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [ID_EMPRESA: string, NOMBRE_EMPRESA: string]
▸ filteredDfPersona: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [ID_PERSONA: string, NOMBRE_PERSONA: string ... 2 more fields]
▸ filteredDfRiesgo: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [ID_CLIENTE: string, RIESGO_CENTRAL_1: double ... 2 more fields]
▸ filteredDfTransaccion: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [ID_EMPRESA: string, ID_PERSONA: string ... 2 more fields]

```
+----------+--------------+----+-------+
|ID_PERSONA|NOMBRE_PERSONA|EDAD|SALARIO|
+----------+--------------+----+-------+
|        24|         Amaya|  24| 1801.0|
|         1|          Carl|  32|20095.0|
|        65|         Nehru|  34|12423.0|
|        71|         Doris|  23|11538.0|
|        83|       Giselle|  45| 2503.0|
|        96|          Amos|  42|15855.0|
|       100|       Cynthia|  57| 8682.0|
|        22|          Kibo|  22| 7449.0|
|         8|         Jonah|  23|17040.0|
|        73|         Fiona|  42| 9960.0|
|        76|          Omar|  34|12163.0|
|        80|         Ebony|  59| 3600.0|
|        80|         Ebony|  59| 3600.0|
|        84|         Keith|  33|13348.0|
|        35|        Aurora|  54| 4588.0|
|        60|       Bernard|  27|10825.0|
|        72|      Tallulah|  46| 9867.0|
|        42|         Wanda|  42| 5419.0|
```

Parte 4

11

```
val calcularRiesgoPonderado = udf((riesgo1: Double, riesgo2: Double, riesgo3: Double) => {
  (2 * riesgo1 + 3 * riesgo2 + 3 * riesgo3) / 7
})

val dfRiesgoPonderado = dfRiesgo.withColumn("riesgo_ponderado", calcularRiesgoPonderado(
  col("RIESGO_CENTRAL_1"),
  col("RIESGO_CENTRAL_2"),
  col("RIESGO_CENTRAL_3")
))

dfRiesgoPonderado.show()
```

▸ dfRiesgoPonderado: org.apache.spark.sql.DataFrame = [ID_CLIENTE: string, RIESGO_CENTRAL_1: double ... 3 more fields]

```
+----------+----------------+----------------+----------------+-------------------+
|ID_CLIENTE|RIESGO_CENTRAL_1|RIESGO_CENTRAL_2|RIESGO_CENTRAL_3|   riesgo_ponderado|
+----------+----------------+----------------+----------------+-------------------+
|         1|             0.1|             0.6|             0.4| 0.4571428571428572|
|         2|             0.2|             0.8|             0.3| 0.5285714285714286|
|         3|             0.9|             0.8|             0.8|  0.942857142857143|
```

```
|         4|            0.5|            0.4|            0.6| 0.5714285714285714|
|         5|            0.6|            0.6|            0.6| 0.6857142857142857|
|         6|            0.3|            0.4|            0.6| 0.5142857142857143|
|         7|            0.2|            0.4|            0.3|0.35714285714285715|
|         8|            0.7|            0.9|            0.7| 0.8857142857142856|
|         9|            0.3|            1.0|            1.0| 0.9428571428571428|
|        10|            0.9|            1.0|            0.4| 0.8571428571428571|
|        11|            0.6|            0.1|            0.3|0.34285714285714286|
|        12|            0.7|            0.6|            0.7|  0.757142857142857|
|        13|            0.7|            0.1|            0.5| 0.4571428571428572|
|        14|            0.1|            0.2|            0.0| 0.1142857142857143|
|        15|            0.9|            0.7|            0.6| 0.8142857142857142|
|        16|            1.0|            0.4|            0.7| 0.7571428571428571|
|        17|            0.0|            0.2|            0.5|                0.3|
```

Parte 5

13

```
//Union transacciones y persona
val filteredDfTransaccion2 = filteredDfTransaccion
  .withColumnRenamed("ID_PERSONA", "ID_PERSONAx")

val df1 = filteredDfTransaccion2.join(filteredDfPersona, filteredDfTransaccion2("ID_PERSONAx") ===
filteredDfPersona("ID_PERSONA"), "inner").dropDuplicates

//Union df1 y empresa
val filteredDfEmpresa2 = filteredDfEmpresa
  .withColumnRenamed("ID_EMPRESA", "ID_EMPRESAx")

val dfTablon = df1.join(filteredDfEmpresa2, df1("ID_EMPRESA") === filteredDfEmpresa2("ID_EMPRESAx"),
"inner").dropDuplicates

//Eliminar columnas innecesarias
val columnasNecesarias =
Seq("ID_EMPRESA","ID_PERSONA","MONTO","FECHA","NOMBRE_PERSONA","EDAD","SALARIO","NOMBRE_EMPRESA")

val dfTablon2 = dfTablon.select(columnasNecesarias.head, columnasNecesarias.tail: _*)

//Union dfTablon y dfRiesgos
val df2 = dfTablon2.join(dfRiesgoPonderado, dfTablon2("ID_PERSONA") === dfRiesgoPonderado("ID_CLIENTE"),
"inner").dropDuplicates

//Visualizar
df1.show()
dfTablon.show()
df2.show()
```

- df1: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [ID_EMPRESA: string, ID_PERSONAx: string ... 6 more fields]
- df2: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [ID_EMPRESA: string, ID_PERSONA: string ... 11 more fields]
- dfTablon: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [ID_EMPRESA: string, ID_PERSONAx: string ... 8 more fields]
- dfTablon2: org.apache.spark.sql.DataFrame = [ID_EMPRESA: string, ID_PERSONA: string ... 6 more fields]
- filteredDfEmpresa2: org.apache.spark.sql.DataFrame = [ID_EMPRESAx: string, NOMBRE_EMPRESA: string]
- filteredDfTransaccion2: org.apache.spark.sql.DataFrame = [ID_EMPRESA: string, ID_PERSONAx: string ... 2 more fields]

```
+----------+-----------+------+----------+----------+--------------+----+-------+
|ID_EMPRESA|ID_PERSONAx| MONTO|     FECHA|ID_PERSONA|NOMBRE_PERSONA|EDAD|SALARIO|
+----------+-----------+------+----------+----------+--------------+----+-------+
|         7|         80|4250.0|2018-11-28|        80|         Ebony|  59| 3600.0|
|         2|         48|2334.0|2018-12-05|        48|       Illiana|  18| 1454.0|
|        10|         22|2583.0|2018-11-28|        22|          Kibo|  22| 7449.0|
|         4|         96|2887.0|2018-04-19|        96|          Amos|  42|15855.0|
|         2|         22|1398.0|2018-12-05|        22|          Kibo|  22| 7449.0|
|         6|         42| 995.0|2018-04-19|        42|         Wanda|  42| 5419.0|
|         7|          8|2538.0|2018-12-05|         8|         Jonah|  23|17040.0|
|         3|         31|3107.0|2018-04-19|        31|         Rylee|  47|21591.0|
|        10|          1| 238.0|2018-04-19|         1|          Carl|  32|20095.0|
|         1|         84| 769.0|2018-12-05|        84|         Keith|  33|13348.0|
|         1|         72|2668.0|2018-04-19|        72|      Tallulah|  46| 9867.0|
|         5|         71|1548.0|2018-12-05|        71|         Doris|  23|11538.0|
|         4|         73|3878.0|2018-11-28|        73|         Fiona|  42| 9960.0|
|         6|         24|1745.0|2018-12-05|        24|         Amaya|  24| 1801.0|
```

```
|         3|         35|3546.0|2018-04-19|        35|        Aurora|  54| 4588.0|
|        10|         60|3399.0|2018-11-28|        60|       Bernard|  27|10825.0|
|         5|         83|2233.0|2018-04-19|        83|       Giselle|  45| 2503.0|
|         2|         65|4007.0|2018-04-19|        65|         Nehru|  34|12423.0|
```

Parte 6

15

```
//PRE PROCESAMIENTO

val dfTablon1 = df2.filter($"MONTO" > 500 && $"NOMBRE_EMPRESA" === "Amazon")
dfTablon1.show()
```

▸ dfTablon1: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [ID_EMPRESA: string, ID_PERSONA: string ... 11 more fields]

```
+----------+----------+------+----------+--------------+----+-------+--------------+----------+----------------+-----
-----------+----------------+--------------------+
|ID_EMPRESA|ID_PERSONA| MONTO|     FECHA|NOMBRE_PERSONA|EDAD|SALARIO|NOMBRE_EMPRESA|ID_CLIENTE|RIESGO_CENTRAL_1|RIESG
O_CENTRAL_2|RIESGO_CENTRAL_3|   riesgo_ponderado|
+----------+----------+------+----------+--------------+----+-------+--------------+----------+----------------+-----
-----------+----------------+--------------------+
|         5|        54|2401.0|2018-11-28|          Lars|  25|20573.0|        Amazon|        54|             0.9|
0.0|             0.1|                0.3|
|         5|        60|1994.0|2018-11-28|       Bernard|  27|10825.0|        Amazon|        60|             0.6|
0.2|             0.3|0.38571428571428573|
|         5|        68| 781.0|2018-12-05|         Hayes|  31| 7523.0|        Amazon|        68|             0.2|
0.9|             0.8| 0.7857142857142857|
|         5|        29|1422.0|2018-12-05|          Jana|  39| 6483.0|        Amazon|        29|             0.9|
0.0|             0.2| 0.3428571428571429|
|         5|        83|4017.0|2018-12-05|       Giselle|  45| 2503.0|        Amazon|        83|             0.0|
1.0|             0.4|                0.6|
|         5|        92| 606.0|2018-04-19|        Lesley|  19|23547.0|        Amazon|        92|             0.6|
0.2|             0.3|0.38571428571428573|
|         5|        36|3567.0|2018-11-28|         Keely|  41|10373.0|        Amazon|        36|             0.0|
0.0|             0.3|0.12857142857142856|
|         5|        61|1322.0|2018-11-28|          Abel|  33|15070.0|        Amazon|        61|             0.8|
```

Parte 7

17

```
//PROCESAMIENTO

val dfReporte1 = dfTablon1.filter($"EDAD".between(30,39) && $"SALARIO".between(1000,5000))

val dfReporte2 = dfTablon1.filter($"EDAD".between(40,49) && $"SALARIO".between(2500,7000))

val dfReporte3 = dfTablon1.filter($"EDAD".between(50,60) && $"SALARIO".between(3500,10000))
```

▸ dfReporte1: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [ID_EMPRESA: string, ID_PERSONA: string ... 11 more fields]
▸ dfReporte2: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [ID_EMPRESA: string, ID_PERSONA: string ... 11 more fields]
▸ dfReporte3: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [ID_EMPRESA: string, ID_PERSONA: string ... 11 more fields]

```
dfReporte1: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [ID_EMPRESA: string, ID_PERSONA: string ... 11 m
ore fields]
dfReporte2: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [ID_EMPRESA: string, ID_PERSONA: string ... 11 m
ore fields]
dfReporte3: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [ID_EMPRESA: string, ID_PERSONA: string ... 11 m
ore fields]
```

Parte 8

19