PráCtica sesión 3 Spark -

proyecto 3

El objetivo del siguiente proyecto, es evaluar lo visto en la sesión y en los ejercicios.

**Sumario**

[Ejercicio final Repaso (Proyecto 3) 2](#__RefHeading___Toc631_2103982880)

## Ejercicio final Repaso (Proyecto 3)

Trabajamos en una empresa con las siguientes caracteristicas

**Empleados**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| emp\_id | nombre | superior\_emp\_id | fecha\_inicio | emp\_dept\_id | genero | sueldo |
| 1 | Jaime | -1 | 2018 | 10 | M | 1300 |
| 2 | Maria | 1 | 2010 | 20 | F | 1400 |
| 3 | Rosa | 1 | 2010 | 10 | F | 1000 |
| 4 | Vanesa | 2 | null | 10 | F | 1800 |
| 5 | Leonardo | 2 | 2010 | 40 | M | 1300 |
| 6 | Paula | 2 | 2013 | 20 | F | 1600 |
| 7 | Jose Luis | 2 | 2014 | 30 | M | 1200 |

**Departamentos**

|  |  |
| --- | --- |
| dept\_nombre | dept\_id |
| Financiero | 10 |
| Marketing | 20 |
| Ventas | 30 |
| Desarrollo | 40 |

Crea los dataframes necesarios con la información anterior

Adjunto las listas para facilitar la creación:

*empleados = [(1,"Jaime",-1,"2018","10","M",1300), \*

*(2,"Maria",1,"2010","20","F",1400), \*

*(3,"Rosa",1,"2010","10","F",1000), \*

*(4,"Vanesa",2,None,"10","F",1800), \*

*(5,"Leonardo",2,"2010","40","M",1300), \*

*(6,"Paula",2,"2013","20","F",1600), \*

*(7,"Jose Luis",2,"2014","30","F",1200) \*

*]*

*departamento = [("Financiero",10),("Marketing",20),("Ventas",30),("Desarrollo",40)]*

**from pyspark.sql.types import StructField,StructType,IntegerType,StringType,FloatType**

**est\_emp = StructType([StructField('emp\_id',IntegerType(),False),**

**StructField('nom',StringType(),True),**

**StructField('superior\_emp\_id',IntegerType(),True),**

**StructField('data\_i',StringType(),True),**

**StructField('dept\_id',StringType(),True),**

**StructField('genere',StringType(),True),**

**StructField('sou',IntegerType(),True)**

**])**

**est\_dep = StructType([StructField('dept\_nom',StringType(),False),**

**StructField('dept\_id',IntegerType(),False)**

**])**

**df\_emp = spark.createDataFrame(empleados,schema=est\_emp)**

**df\_dep = spark.createDataFrame(departamento,schema=est\_dep)**

Necesitamos lo siguiente:

1. Saber ¿Cuál es el sueldo medio de cada departamento?

**df\_emp.select("dept\_id","sou").groupBy("dept\_id").avg().orderBy("dept\_id").show()**

1. Eliminar las filas que tengan el campo fecha\_inicio a nulo

**df\_emp.na.drop(subset= "data\_i").show()**

1. A los empleados que tienen más de 10 años de antigüedad se les va a subir el sueldo al doble calcula con una función UDF el nuevo sueldo de los empleados y añádelo a una nueva columna Sueldo\_nuevo

**from pyspark.sql.functions import udf**

**def doble(s):**

**return 2 \* s**

**doble\_udf = udf(lambda x: doble(x),IntegerType())**

**df\_emp.where(df\_emp["data\_i"]<="2011").withColumn('sou\_nou',doble\_udf(df\_emp['sou'])).show()**

1. De los trabajadores cuya responsable es Maria ¿Quien es la persona que más cobra?

**df\_emp.where(df\_emp["superior\_emp\_id"]==(df\_emp.where(df\_emp['nom']=='Maria').select('emp\_id').collect()[0][0])).orderBy(df\_emp['sou'].desc()).limit(1).show()**

Ho he intentat també amb joins, que crec que quedaria més fàcil, però em donava problemes d’àlies (a pesar que els posava)

També ho podem passar tot a SQL utilitzant vistes temporals (per a mi més còmode)

**df\_emp.createTempView("emp")**

**df\_dep.createTempView("dep")**

**spark.sql("SELECT e1.emp\_id, e1.nom, e1.sou FROM emp e1 INNER JOIN emp e2 ON e1.superior\_emp\_id=e2.emp\_id WHERE e2.nom='Maria' ORDER BY e1.sou DESC LIMIT 1").show()**

1. Crea una vista que se pueda acceder a ella desde otra sesión que muestre el total que se gasta en sueldos por departamentos

**df\_emp.join(df\_dep,df\_emp.dept\_id==df\_dep.dept\_id,'INNER').select('dept\_nom','sou').groupBy('dept\_nom').sum().createGlobalTempView("Departaments")**

**spark.sql("Select \* from global\_temp.Departaments").show()**

1. Se acaba de contratar una jefa de ventas, llamada Elena Martínez, va a cobrar 1700, necesitamos añadirla a nuestra base de datos

**df\_emp\_nova = spark.createDataFrame([(8,"Elena Martínez",-1,"2021","30","F",1700)],schema=est\_emp)**

**df\_emp = df\_emp.union(df\_emp\_nova)**