CONFIGURACIÓN DE AMBIENTE APACHE SPARK

Se usa apache spark a falta de una plataforma con bases de datos SQL para pruebas:

1. En el IDE favorito realizar la siguiente instalación copiando y pegando el siguiente código y crear los dataframe con los que se trabajaran las sentencias SQL

*#------------- instalación de libreríasy conexión a spark-------------*

*#Realizar las instalaciones de las librerias findspark y pyspark con los siguietnes comandos:*

*pip3 install findspark*

*pip3 install pyspark*

*#findspark permite usar pyspark (interfaz de python a Spark) desde cualquier IDE*

*# Importar librerias*

*import* findspark

findspark.init()

*from* pyspark.sql *import* SparkSession

*# Inicialización de variables obligatorias*

*from* pyspark *import* SparkConf, SparkContext

*# SparkContext configuración de spark*

*# SparkSession representa la conexión para trabajar con SQL*

sparkConf = SparkConf().setAppName("My SparkQL Application")

sc = SparkContext(*conf*=sparkConf)

spark = SparkSession(sc)

*#----- Crear dos dataframe vuelos y pilotos para resolver el test-----*

vuelos\_ = spark.read.load(

"/Users/yohanalfonsohernandez/Documents/Pruebas/prueba\_sql\_y python/vuelos.csv", *format*="csv", *sep*=",", *inferSchema*="true", *header*="true",

)

*#vuelos.show()*

pilotos = spark.read.load(

"/Users/yohanalfonsohernandez/Documents/Pruebas/prueba\_sql\_y python/pilotos.csv", *format*="csv", *sep*=",", *inferSchema*="true", *header*="true",

)

*#pilotos.show()*

vuelos=vuelos\_.withColumnRenamed("Aerolínea","Aerolinea") *##agregar esta linea al test*

*## Se crean vistas temporales para trabajar con sentencias SQL*

vuelos.createOrReplaceTempView("vw\_vuelos") *# este es el nombre de la tabla*

pilotos.createOrReplaceTempView("vw\_pilotos") *# este es el nombre de la tabla*

*# Se realiza la consulta usando directamente SQL como prueba*

spark.sql("SELECT \* FROM vw\_vuelos")*.show()*

spark.sql("SELECT \* FROM vw\_pilotos")*.show()*

S