package org.itc.com  
import org.apache.spark.SparkConf  
import org.apache.spark.sql.SparkSession  
import org.apache.spark.sql.functions.\_  
import org.apache.spark.sql.types.\_  
import org.apache.spark.sql.expressions.Window  
  
object jsondftask17apr extends App{  
 val *spark* = SparkSession.*builder*()  
 .appName("Jsonreadfile").master("local[1]")  
 .getOrCreate()  
 import *spark*.implicits.\_  
 // Define schema for the JSON  
 val *schema* = StructType(Seq(  
 StructField("filename", StringType),  
 StructField("datasets", ArrayType(StructType(Seq(  
 StructField("orderId", StringType),  
 StructField("customerId", StringType),  
 StructField("orderDate", StringType),  
 StructField("shipmentDetails", StructType(Seq(  
 StructField("street", StringType),  
 StructField("city", StringType),  
 StructField("state", StringType),  
 StructField("postalCode", StringType),  
 StructField("country", StringType)  
 ))),  
 StructField("orderDetails", ArrayType(StructType(Seq(  
 StructField("productId", StringType),  
 StructField("quantity", IntegerType),  
 StructField("sequence", IntegerType),  
 StructField("totalPrice", StructType(Seq(  
 StructField("gross", IntegerType),  
 StructField("net", IntegerType),  
 StructField("tax", IntegerType)  
 )))  
 ))))  
 ))))  
 ))  
// val df = spark.read.option("multiline", "true").json("D:\\spark\_code\\sparkdemo\\input\\jsondf.json")  
val *df* = *spark*.read.schema(*schema*)  
 .option("multiLine", true)  
 .option("mode", "PERMISSIVE").json("D:\\spark\_code\\sparkdemo\\input\\jsondf.json")  
  
*df*.show()  
 *df*.printSchema()  
 val *flattenedDf* = *df*.select(  
 $"filename",  
 *explode*($"datasets").as("ds"),  
 $"ds.orderId",  
 $"ds.customerId",  
 $"ds.orderDate",  
 $"ds.shipmentDetails.street",  
 $"ds.shipmentDetails.city",  
 $"ds.shipmentDetails.state",  
 $"ds.shipmentDetails.postalcode",  
 $"ds.shipmentDetails.country",  
 $"ds.orderDetails.productId",  
 $"ds.orderDetails.quantity",  
 $"ds.orderDetails.sequence",  
 $"ds.orderDetails.totalPrice.gross",  
 $"ds.orderDetails.totalPrice.net",  
 $"ds.orderDetails.totalPrice.tax"  
 )  
  
 *flattenedDf*.filter($"ds.orderId" === "ord1001").show()  
  
}







