Apply schema on a delimited text file

1. **Case class:**

   case class Employee(firstName: String, lastName:String, deptId: Int)

   val people = sc.textFile("/input/employee.txt").map(_.split(",")).map(e => Employee(e(0), e(1), e(2).trim.toInt)).toDF()

   Note:

   a. make sure to
      import sqlContext.implicits._
   b. case class can have only 22 fields at the max

2. **Schema structure:**

   // Generate schema

   val schema = StructType(Seq(StructField("fname", StringType, true), StructField("lname", StringType, true), StructField("deptId", IntegerType, true)))

   // Convert records of RDD to Rows.

   val rowRDD = people.map(_.split(",")).map(e =>Row(e(0), e(1), e(2).trim.toInt))

   // Apply the schema to the RDD.

   val peopleDataFrame = sqlContext.createDataFrame(rowRDD, schema)

   Note: Make sure to import following packages
   import org.apache.spark.sql.Row
   import org.apache.spark.sql.types.{StructType, StructField, StringType, IntegerType};

Read json file

val emp = sqlContext.read.json("/input/employee.json")

Register a dataframe as a table

myDataframe.registerTempTable("employee")

Query your registered table

val deptOne = sqlContext.sql("SELECT firstName FROM employee WHERE deptId = 1")

Reference:

https://spark.apache.org/docs/1.6.3/sql-programming-guide.html