1. Load and show the attached datafile in AWS notebook

In [1]:

```
import org.apache.spark.sql.{DataFrame, Dataset, SparkSession}
import org.apache.spark.sql.catalyst.ScalaReflection
import org.apache.spark.sql.types.StructType
import scala.reflect.runtime.universe._

object Source {
    def readCSV(path: String, header:String = "true", delimiter:String = ",", er
        = spark.read.option("header",header).option("delimiter", delimiter).opti

    def read[T <: Product](path: String, header:String = "true", delimiter:Strir
    import spark.implicits._

    spark
      .read
      .option("header", header)
      .option("delimiter", delimiter)
      .option("encoding", encoding)
      .option("inferSchema", "false")
      .schema(ScalaReflection.schemaFor[T].dataType.asInstanceOf[StructType])
      .csv(path)
      .as[T]
  }
}

val path = "s3://jyin-axa-test/source/position-des-bus-en-circulation-sur-le-res

val df = Source.readCSV(path=path, delimiter=";")

df.printSchema
df.show
```

```
Starting Spark application
```

ID	YARN Application ID	Kind	State	Spark UI	Driver log	Current session?
0	application_1627586830606_0001	spark	idle	Link	Link	✔

```
SparkSession available as 'spark'.

import org.apache.spark.sql.{DataFrame, Dataset, SparkSession}
import org.apache.spark.sql.catalyst.ScalaReflection
import org.apache.spark.sql.types.StructType
import scala.reflect.runtime.universe._
defined object Source
path: String = s3://jyin-axa-test/source/position-des-bus-en-circulation-sur-le-
reseau-star-en-temps-reel.csv
df: org.apache.spark.sql.DataFrame = [Bus (ID): string, Bus (num?ro): string ...
7 more fields]
root
 |-- Bus (ID): string (nullable = true)
 |-- Bus (num?ro): string (nullable = true)
 |-- Etat: string (nullable = true)
 |-- Ligne (ID): string (nullable = true)
 |-- Ligne (nom court): string (nullable = true)
 |-- Code du sens: string (nullable = true)
 |-- Destination: string (nullable = true)
 |-- Coordonn?es: string (nullable = true)
 |-- Avance / Retard: string (nullable = true)
```

```
+----------+-----------+-----------+---------+----------------+-----------+
--------------------+------------------+--------------+
|  Bus (ID)|Bus (num?ro)|        Etat|Ligne (ID)|Ligne (nom court)|Code du sens|
Destination|        Coordonn?es|Avance / Retard|
+----------+-----------+-----------+---------+----------------+-----------+
--------------------+------------------+--------------+
| 149722672|   149722672|    En ligne|      0005|              C5|           1|
Patton|48.126706,-1.665295|           1218|
| 146720212|   146720212|Hors-service|      null|            null|        null|
null|48.110314,-1.642594|           null|
|1205787936|  1205787936|Hors-service|      null|            null|        null|
null|48.110638,-1.642596|           null|
| 100682492|   100682492|    En ligne|      0034|              34|           0|
Chantepie|48.099402,-1.620724|            -54|
|1203586132|  1203586132|Hors-service|      null|            null|        null|
null|48.146719,-1.706516|           null|
| 166536448|   166536448|     Inconnu|      0057|              57|           1|
Rennes|  48.10432,-1.67107|           null|
| 144718572|   144718572|     Inconnu|      0004|              C4|           1|
ZA Saint-Sulpice|48.119439,-1.688266|           null|
| 122100040|   122100040|    En ligne|      0153|            153ex|           0|
L'Hermitage| 48.12315,-1.805114|             49|
| 180347764|   180347764|    En ligne|      0072|              72|           1|
Rennes|47.999996,-1.696828|              0|
| 144918736|   144918736|    En ligne|      0001|              C1|           1|
Champs Blancs|48.104342,-1.670076|           1655|
| 122300204|   122300204|    En ligne|      0053|              53|           0|
La Chapelle-Thoua...|48.116172,-1.709381|           1202|
| 180547928|   180547928|Hors-service|      null|            null|        null|
null|48.099681,-1.632671|           null|
|1203786296|  1203786296|Hors-service|      null|            null|        null|
null|48.146956,-1.706672|           null|
| 122500368|   122500368|    En ligne|      0059|              59|           1|
Rennes|48.023382,-1.736128|             39|
| 180748092|   180748092|Hors-service|      null|            null|        null|
null| 48.10992,-1.641147|           null|
| 122700532|   122700532|    En ligne|      0157|            157ex|           0|
Bruz Centre|48.030553,-1.759458|           -116|
| 122900696|   122900696|    En ligne|      0159|            159ex|           0|
Bruz|48.020942,-1.751458|             45|
| 181548748|   181548748|     Inconnu|      0040|              40|           0|
R?publique|48.109833,-1.665716|           null|
|  65053300|    65053300|    En ligne|      0220|             220|           1|
Acign? |  Timoni?re|48.119032,-1.598197|              0|
| 140114800|   140114800|    En ligne|      0001|              C1|           1|
Champs Blancs|48.135121,-1.620806|           3158|
+----------+-----------+-----------+---------+----------------+-----------+
--------------------+------------------+--------------+
only showing top 20 rows
```

1. Clean the dataset if necessary, we need to export a CSV file comma ',' separated file, theheaders must be translated into english (use google translate, optional)

In [6]:

```
import org.apache.spark.sql.{ Dataset, Encoder, SaveMode, SparkSession, DataFram
import org.apache.spark.sql.functions.{ col, udf }
import scala.util.{ Failure, Success, Try }

object FakeGoogleAPIHelper {

    val nameMap = Map("Ligne (ID)" -> "Line (ID)", "Avance / Retard" -> "Advance
```

```
    def translate(queryText: String, source: String = "fr", target: String = "er
}

val VALID_STATE = "En ligne"



def cleanDF(df: DataFrame): DataFrame = {
    val isNumeric = udf((value: String) => Try(value.toInt).isSuccess)

    val isValidLatLon = udf((value: String) => {
        val arr = value.split(",")
        arr.size == 2 && arr.forall(a => Try(a.toDouble).isSuccess)
    })

    df.na.drop
      .filter(col("Etat") === VALID_STATE)
      .filter(isNumeric(col("Avance / Retard")))
      .filter(isValidLatLon(col("Coordonnées")))
}

def renameDF(df: DataFrame, nameMap: Map[String, String]): DataFrame = {
    val oldColumnNames: Seq[String] = df.columns.toSeq
    val renamedColumns: Seq[Column] = oldColumnNames.map(name =>{
      val newName = nameMap.get(name).getOrElse(name)
      col(name).as(newName)
    })
    df.select(renamedColumns : _*)
}

val nameMap = df.columns.toSeq.flatMap(name => FakeGoogleAPIHelper.translate(nam
val renamedDF = renameDF(cleanDF(df), nameMap)
val output = "s3://jyin-axa-test/output"

renamedDF.write.mode(SaveMode.Overwrite).option("delimiter", ",").option("header

df.printSchema
renamedDF.show
```

```
import org.apache.spark.sql.{Dataset, Encoder, SaveMode, SparkSession, DataFram
e, Column}
import org.apache.spark.sql.functions.{col, udf}
import scala.util.{Failure, Success, Try}
defined object FakeGoogleAPIHelper
VALID_STATE: String = En ligne
cleanDF: (df: org.apache.spark.sql.DataFrame)org.apache.spark.sql.DataFrame
renameDF: (df: org.apache.spark.sql.DataFrame, nameMap: Map[String,String])org.a
pache.spark.sql.DataFrame
nameMap: scala.collection.immutable.Map[String,String] = Map(Ligne (ID) -> Line
(ID), Avance / Retard -> Advance / Delay, Bus (ID) -> Bus (ID), Ligne (nom cour
t) -> Line (short name), Destination -> Destination, Code du sens -> Code of mea
ning, Etat -> state)
renamedDF: org.apache.spark.sql.DataFrame = [Bus (ID): string, Bus (num?ro): str
ing ... 7 more fields]
output: String = s3://jyin-axa-test/output
root
 |-- Bus (ID): string (nullable = true)
 |-- Bus (num?ro): string (nullable = true)
 |-- Etat: string (nullable = true)
```

```
 |-- Ligne (ID): string (nullable = true)
 |-- Ligne (nom court): string (nullable = true)
 |-- Code du sens: string (nullable = true)
 |-- Destination: string (nullable = true)
 |-- Coordonn?es: string (nullable = true)
 |-- Avance / Retard: string (nullable = true)

+---------+-----------+--------+---------+----------------+--------------+---
----------------+------------------+--------------+
| Bus (ID)|Bus (num?ro)|   state|Line (ID)|Line (short name)|Code of meaning|
Destination|        Coordonn?es|Advance / Delay|
+---------+-----------+--------+---------+----------------+--------------+---
----------------+------------------+--------------+
|149722672|   149722672|En ligne|     0005|              C5|             1|
Patton|48.126706,-1.665295|           1218|
|100682492|   100682492|En ligne|     0034|              34|             0|
Chantepie|48.099402,-1.620724|            -54|
|122100040|   122100040|En ligne|     0153|            153ex|             0|
L'Hermitage| 48.12315,-1.805114|             49|
|180347764|   180347764|En ligne|     0072|              72|             1|
Rennes|47.999996,-1.696828|              0|
|144918736|   144918736|En ligne|     0001|              C1|             1|
Champs Blancs|48.104342,-1.670076|           1655|
|122300204|   122300204|En ligne|     0053|              53|             0|La
Chapelle-Thoua...|48.116172,-1.709381|           1202|
|122500368|   122500368|En ligne|     0059|              59|             1|
Rennes|48.023382,-1.736128|             39|
|122700532|   122700532|En ligne|     0157|            157ex|             0|
Bruz Centre|48.030553,-1.759458|           -116|
|122900696|   122900696|En ligne|     0159|            159ex|             0|
Bruz|48.020942,-1.751458|             45|
| 65053300|    65053300|En ligne|     0220|             220|             1| Ac
ign? |  Timoni?re|48.119032,-1.598197|              0|
|140114800|   140114800|En ligne|     0001|              C1|             1|
Champs Blancs|48.135121,-1.620806|           3158|
|182349404|   182349404|En ligne|     0004|              C4|             0|
Grand Quartier|48.128898,-1.634642|           1141|
|145919556|   145919556|En ligne|     0003|              C3|             0|
Henri Fr?ville|48.087754,-1.674665|              0|
|140314964|   140314964|En ligne|     0004|              C4|             0|
Grand Quartier|48.108262,-1.694368|           2238|
|184150880|   184150880|En ligne|     0006|              C6|             1|
Cesson-S?vign?|48.114474,-1.593277|            847|
|146119720|   146119720|En ligne|     0001|              C1|             1|
Champs Blancs| 48.12553,-1.642128|           1812|
|140515128|   140515128|En ligne|     0002|              C2|             0|
Haut Sanc?| 48.13319,-1.685219|            735|
|184551208|   184551208|En ligne|     0001|              C1|             0|
Chantepie|48.113659,-1.693299|           1139|
|121699712|   121699712|En ligne|     0050|              50|             1|
Thorign?|48.110206,-1.653999|           1200|
|140715292|   140715292|En ligne|     0003|              C3|             0|
Henri Fr?ville|48.096798,-1.680484|           1489|
+---------+-----------+--------+---------+----------------+--------------+---
----------------+------------------+--------------+
only showing top 20 rows
```

1. which line have the most bus travelling

In [7]:

```
case class BusPosition(busId: String, busNo: String, state:String, lineId: Strir

val output = "s3://jyin-axa-test/output"
```

```
val ds = Source.read[BusPosition](path = output, delimiter=",")
ds.show
```

```
defined class BusPosition
output: String = s3://jyin-axa-test/output
ds: org.apache.spark.sql.Dataset[BusPosition] = [busId: string, busNo: string
... 7 more fields]
+---------+---------+--------+------+--------+------------+--------------------
+--------------------+--------+
|    busId|    busNo|   state|lineId|lineName|codeDirection|                dest
|           latlon|timeDiff|
+---------+---------+--------+------+--------+------------+--------------------
+--------------------+--------+
|149722672|149722672|En ligne|  0005|      C5|           1|              Patton
|48.126706,-1.665295|    1218|
|100682492|100682492|En ligne|  0034|      34|           0|           Chantepie
|48.099402,-1.620724|     -54|
|122100040|122100040|En ligne|  0153|   153ex|           0|         L'Hermitage
| 48.12315,-1.805114|      49|
|180347764|180347764|En ligne|  0072|      72|           1|              Rennes
|47.999996,-1.696828|       0|
|144918736|144918736|En ligne|  0001|      C1|           1|       Champs Blancs
|48.104342,-1.670076|    1655|
|122300204|122300204|En ligne|  0053|      53|           0|La Chapelle-Thoua...
|48.116172,-1.709381|    1202|
|122500368|122500368|En ligne|  0059|      59|           1|              Rennes
|48.023382,-1.736128|      39|
|122700532|122700532|En ligne|  0157|   157ex|           0|         Bruz Centre
|48.030553,-1.759458|    -116|
|122900696|122900696|En ligne|  0159|   159ex|           0|                Bruz
|48.020942,-1.751458|      45|
| 65053300| 65053300|En ligne|  0220|     220|           1| Acign? |  Timoni?re
|48.119032,-1.598197|       0|
|140114800|140114800|En ligne|  0001|      C1|           1|       Champs Blancs
|48.135121,-1.620806|    3158|
|182349404|182349404|En ligne|  0004|      C4|           0|      Grand Quartier
|48.128898,-1.634642|    1141|
|145919556|145919556|En ligne|  0003|      C3|           0|      Henri Fr?ville
|48.087754,-1.674665|       0|
|140314964|140314964|En ligne|  0004|      C4|           0|      Grand Quartier
|48.108262,-1.694368|    2238|
|184150880|184150880|En ligne|  0006|      C6|           1|      Cesson-S?vign?
|48.114474,-1.593277|     847|
|146119720|146119720|En ligne|  0001|      C1|           1|       Champs Blancs
| 48.12553,-1.642128|    1812|
|140515128|140515128|En ligne|  0002|      C2|           0|          Haut Sanc?
| 48.13319,-1.685219|     735|
|184551208|184551208|En ligne|  0001|      C1|           0|           Chantepie
|48.113659,-1.693299|    1139|
|121699712|121699712|En ligne|  0050|      50|           1|            Thorign?
|48.110206,-1.653999|    1200|
|140715292|140715292|En ligne|  0003|      C3|           0|      Henri Fr?ville
|48.096798,-1.680484|    1489|
+---------+---------+--------+------+--------+------------+--------------------
+--------------------+--------+
only showing top 20 rows
```

In [8]:

```
def findMostBusLine(dataset: Dataset[BusPosition]): (String, Int) = {
    import spark.implicits._

    dataset
```

```
      .groupByKey(_.lineId)
      .mapGroups{
        case (lineId, grps) => (lineId, grps.toSeq.map(_.busId).distinct.size)
      }.reduce( (line1, line2) => if(line1._2 > line2._2) line1 else line2)
  }

findMostBusLine(ds)
```

```
findMostBusLine: (dataset: org.apache.spark.sql.Dataset[BusPosition])(String, In
t)
res42: (String, Int) = (0001,15)
```

1. which destinations have "Saint" in the name?

In [9]:

```
def findDestinationByName(dataset: Dataset[BusPosition], name: String): Seq[Stri
    import spark.implicits._

    dataset.map(_.dest).distinct.filter(_.toLowerCase.contains(name.toLowerCase)
  }

findDestinationByName(ds, "Saint")
```

```
findDestinationByName: (dataset: org.apache.spark.sql.Dataset[BusPosition], nam
e: String)Seq[String]
res44: Seq[String] = WrappedArray(Saint-Laurent, Pac? | Saint-Gilles, Noyal-Ch?t
illon | Saint-Erblon, ZA Saint-Sulpice, Saint-Jacques, Saint-Sulpice-la-For?t, N
oyal |  Saint-Erblon, Saint-Gr?goire | Betton, Saint-Gr?goire)
```

1. which is the nearest bus from the eiffel tower based on provided coordinates ?

In [11]:

```
object DistanceHelper {
  val PI = 3.1415926
  val R: Double  = 6370.99681

  def getDistance(lat1: Double, lon1: Double, lat2: Double, lon2: Double): Doubl
    val a1 = lat1 * PI /180.0
    val a2 = lon1 * PI /180.0
    val b1 = lat2 * PI /180.0
    val b2 = lon2 * PI /180.0
    var t1: Double = Math.cos(a1) * Math.cos(a2) * Math.cos(b1)* Math.cos(b2)
    var t2: Double = Math.cos(a1) * Math.sin(a2) * Math.cos(b1)* Math.sin(b2)
    var t3: Double = Math.sin(a1) * Math.sin(b1)
    val distance = Math.acos(t1 + t2 + t3) * R
    distance
  }
}

def findNearestBusByLatLon(dataset: Dataset[BusPosition], latlon: (Double, Doubl
    import spark.implicits._
    dataset.map(b => {
      val gps = b.latlon.split(",")
      val (lat2, lon2) = (gps(0).toDouble, gps(1).toDouble)
      (b, DistanceHelper.getDistance(lat1 = latlon._1, lon1 = latlon._2, lat2 =
    }).reduce((pair1, pair2) => if(pair1._2 < pair2._2) pair1 else pair2)._1
  }
```

```
val latlonTourEiffel = (48.858370, -2.294481)
findNearestBusByLatLon(ds, latlonTourEiffel)
```

```
defined object DistanceHelper
findNearestBusByLatLon: (dataset: org.apache.spark.sql.Dataset[BusPosition], lat
lon: (Double, Double))BusPosition
latlonTourEiffel: (Double, Double) = (48.85837,-2.294481)
res48: BusPosition = BusPosition(1205387608,1205387608,En ligne,0082,82,1,Rennes
| Villejean-Universit?,48.218336,-1.887687,19)
```

In [ ]: