ex52

August 18, 2022

[ ]:
```
from pyspark import SparkContext, SparkConf
from pyspark.sql import SparkSession
from graphframes import GraphFrame

conf = SparkConf().setAppName("ex52")
sc = SparkContext(conf = conf)
ssql = SparkSession.builder.getOrCreate()
```

[2]:
```
edgesPath = "data/Ex52/data/edges.csv"
vertexesPath = "data/Ex52/data/vertexes.csv"
outputPath = "out52/"
```

[3]:
```
v = ssql.read.load(
    vertexesPath,
    format="csv",
    header=True,
    inferSchema=True
)

e = ssql.read.load(
    edgesPath,
    format="csv",
    header=True,
    inferSchema=True
)
```

[4]:
```
v.show(), v.printSchema()
e.show(), e.printSchema()
```

```
+---+-----+---+
| id| name|age|
+---+-----+---+
| u1|Alice| 34|
| u2|  Bob| 36|
| u3| John| 30|
| u4|David| 29|
| u5| Paul| 32|
| u6| Adel| 36|
```

```
| u7| Eddy| 60|
+---+-----+---+

root
 |-- id: string (nullable = true)
 |-- name: string (nullable = true)
 |-- age: integer (nullable = true)

+---+---+--------+
|src|dst|linktype|
+---+---+--------+
| u1| u2|  friend|
| u1| u4|  friend|
| u1| u5|  friend|
| u2| u1|  friend|
| u2| u3|  follow|
| u3| u2|  follow|
| u4| u1|  friend|
| u4| u5|  friend|
| u5| u1|  friend|
| u5| u4|  friend|
| u5| u6|  follow|
| u6| u3|  follow|
| u7| u6|  follow|
+---+---+--------+

root
 |-- src: string (nullable = true)
 |-- dst: string (nullable = true)
 |-- linktype: string (nullable = true)
```

[4]:
```
(None, None)
```

[5]:
```
filteredEdges = e.filter("linktype='follow'")
```

[ ]:
```
g = GraphFrame(v, filteredEdges)
```

[7]:
```
finalDF = g.inDegrees.withColumnRenamed("inDegree", "NFollowers")
```

```
/home/webbelle/univenv/lib/python3.10/site-
packages/pyspark/sql/dataframe.py:127: UserWarning: DataFrame constructor is
internal. Do not directly use it.
  warnings.warn("DataFrame constructor is internal. Do not directly use it.")
```

[8]:
```
finalDF.show(), finalDF.printSchema()
```

```
+---+----------+
```

```
| id|NFollowers|
+---+----------+
| u3|         2|
| u6|         2|
| u2|         1|
+---+----------+

root
 |-- id: string (nullable = true)
 |-- NFollowers: integer (nullable = false)
```

[8]:
```
(None, None)
```

[9]:
```
finalDF.write.csv(outputPath, header=True)
```

ex53

August 18, 2022

[ ]:
```
from pyspark import SparkContext, SparkConf
from pyspark.sql import SparkSession
from graphframes import GraphFrame

conf = SparkConf().setAppName("ex53")
sc = SparkContext(conf = conf)
ssql = SparkSession.builder.getOrCreate()
```

[19]:
```
edgesPath = "data/Ex53/data/edges.csv"
vertexesPath = "data/Ex53/data/vertexes.csv"
outputPath = "out53/"
```

[4]:
```
eDF = ssql.read.load(
    edgesPath,
    format="csv",
    header=True,
    inferSchema=True
)

vDF = ssql.read.load(
    vertexesPath,
    format="csv",
    header=True,
    inferSchema=True
)
```

[5]:
```
eDF.show(), eDF.printSchema()
vDF.show(), vDF.printSchema()
```

```
+---+---+--------+
|src|dst|linktype|
+---+---+--------+
| u1| u2|  friend|
| u1| u4|  friend|
| u1| u5|  friend|
| u2| u1|  friend|
| u2| u3|  follow|
| u3| u2|  follow|
```

```
| u4| u1|  friend|
| u4| u5|  friend|
| u5| u1|  friend|
| u5| u4|  friend|
| u5| u6|  follow|
| u6| u3|  follow|
| u7| u6|  follow|
+---+---+--------+

root
 |-- src: string (nullable = true)
 |-- dst: string (nullable = true)
 |-- linktype: string (nullable = true)

+---+-----+---+
| id| name|age|
+---+-----+---+
| u1|Alice| 34|
| u2|  Bob| 36|
| u3| John| 30|
| u4|David| 29|
| u5| Paul| 32|
| u6| Adel| 36|
| u7| Eddy| 60|
+---+-----+---+

root
 |-- id: string (nullable = true)
 |-- name: string (nullable = true)
 |-- age: integer (nullable = true)
```

[5]:
```
(None, None)
```

[6]:
```
filteredEDF = eDF.filter("linktype='follow'")
```

[ ]:
```
g = GraphFrame(vDF, filteredEDF)
```

[9]:
```
nFollowerDF = g.inDegrees.withColumnRenamed("inDegree", "NFollowers")
nFollowerDF.printSchema(), nFollowerDF.show()
```

```
/home/webbelle/univenv/lib/python3.10/site-
packages/pyspark/sql/dataframe.py:127: UserWarning: DataFrame constructor is
internal. Do not directly use it.
  warnings.warn("DataFrame constructor is internal. Do not directly use it.")

root
 |-- id: string (nullable = true)
```

```
 |-- NFollowers: integer (nullable = false)
```

```
+---+----------+
| id|NFollowers|
+---+----------+
| u3|         2|
| u6|         2|
| u2|         1|
+---+----------+
```

[9]:
```
(None, None)
```

[10]:
```
maxFollowersDF = nFollowerDF\
    .agg({"NFollowers":"max"})\
        .withColumnRenamed("max(NFollowers)", "MaxNFollowers")
```

[15]:
```
#se non selezionassi la colonna del DF, maxNFollowers conterrebbe SOLO un␣
 ↪oggetto Row con quel valore
maxNFollowers = maxFollowersDF.first().MaxNFollowers
maxNFollowers
```

[15]:
```
2
```

[16]:
```
nFollowerDFfinal = nFollowerDF.filter(nFollowerDF.NFollowers==maxNFollowers)
```

[17]:
```
nFollowerDFfinal.show()
```

```
+---+----------+
| id|NFollowers|
+---+----------+
| u3|         2|
| u6|         2|
+---+----------+
```

[20]:
```
nFollowerDFfinal.write.csv(outputPath, header=True)
```

ex54

August 18, 2022

[ ]:
```
from pyspark import SparkContext, SparkConf
from pyspark.sql import SparkSession
from graphframes import GraphFrame

conf = SparkConf().setAppName("ex54")
sc = SparkContext(conf=conf)
ssql = SparkSession.builder.getOrCreate()
```

[2]:
```
edgesPath = "data/Ex54/data/edges.csv"
vertexesPath = "data/Ex54/data/vertexes.csv"
outputPath = "out54/"
```

[4]:
```
eDF = ssql.read.load(
    edgesPath,
    format="csv",
    header=True,
    inferSchema=True
)

vDF = ssql.read.load(
    vertexesPath,
    format="csv",
    header=True,
    inferSchema=True
)
```

[5]:
```
filteredEDF = eDF.filter("linktype='friend'")
```

[ ]:
```
g = GraphFrame(vDF, filteredEDF)
```

[10]:
```
filteredG = g.find("(userX)-[]->(userY);!(userY)-[]->(userX)")
```

[11]:
```
filteredG.show(), filteredG.printSchema()
```

```
+--------------+--------------+
|         userX|         userY|
+--------------+--------------+
|{u4, David, 29}|{u1, Alice, 34}|
```

```
|{u1, Alice, 34}|  {u2, Bob, 36}|
+---------------+---------------+

root
 |-- userX: struct (nullable = false)
 |    |-- id: string (nullable = true)
 |    |-- name: string (nullable = true)
 |    |-- age: integer (nullable = true)
 |-- userY: struct (nullable = false)
 |    |-- id: string (nullable = true)
 |    |-- name: string (nullable = true)
 |    |-- age: integer (nullable = true)
```

[11]:
```
(None, None)
```

[12]:
```
formatteDFG = filteredG.selectExpr("userX.id as IDFriend", "userY.id as␣
↪IDNotFriend")
```

[13]:
```
formatteDFG.write.csv(outputPath, header=True)
```

ex55

August 18, 2022

[ ]:
```
from pyspark import SparkContext, SparkConf
from pyspark.sql import SparkSession
from graphframes import GraphFrame

conf = SparkConf().setAppName("ex55")
sc = SparkContext(conf=conf)
ssql = SparkSession.builder.getOrCreate()
```

[2]:
```
edgesPath = "data/Ex55/data/edges.csv"
vertexesPath = "data/Ex55/data/vertexes.csv"
outputPath = "out55/"
```

[3]:
```
eDF = ssql.read.load(
    edgesPath,
    format="csv",
    header=True,
    inferSchema=True
)

vDF = ssql.read.load(
    vertexesPath,
    format="csv",
    header=True,
    inferSchema=True
)
```

[4]:
```
eDF.printSchema(), vDF.printSchema()
```

```
root
 |-- src: string (nullable = true)
 |-- dst: string (nullable = true)
 |-- linktype: string (nullable = true)

root
 |-- id: string (nullable = true)
 |-- entityName: string (nullable = true)
 |-- name: string (nullable = true)
```

[4]: (None, None)

[5]:
```
eDF.show(), vDF.show()
```

```
+---+---+----------+
|src|dst|  linktype|
+---+---+----------+
| V1| V2|      like|
| V1| V3|    follow|
| V1| V4|    follow|
| V3| V2|    follow|
| V3| V4|    follow|
| V5| V2|  expertOf|
| V2| V4|correlated|
| V4| V2|correlated|
+---+---+----------+

+---+----------+--------+
| id|entityName|    name|
+---+----------+--------+
| V1|      user|   Paolo|
| V2|     topic|     SQL|
| V3|      user|   David|
| V4|     topic|Big Data|
| V5|      user|    John|
+---+----------+--------+
```

[5]: (None, None)

[6]:
```
filteredEDF = eDF.filter("linktype='follow'")
```

[ ]:
```
g = GraphFrame(vDF, filteredEDF)
```

[ ]:
```
resultDF = g.find("(userID)-[follow]->(topicID)")
```

[11]:
```
resultDF.show(), resultDF.printSchema()
```

```
+----------------+---------------+--------------------+
|          userID|         follow|             topicID|
+----------------+---------------+--------------------+
|{V1, user, Paolo}|{V1, V3, follow}|   {V3, user, David}|
|{V1, user, Paolo}|{V1, V4, follow}|{V4, topic, Big D…|
|{V3, user, David}|{V3, V2, follow}|    {V2, topic, SQL}|
|{V3, user, David}|{V3, V4, follow}|{V4, topic, Big D…|
+----------------+---------------+--------------------+

root
```

```
 |-- userID: struct (nullable = false)
 |    |-- id: string (nullable = true)
 |    |-- entityName: string (nullable = true)
 |    |-- name: string (nullable = true)
 |-- follow: struct (nullable = false)
 |    |-- src: string (nullable = true)
 |    |-- dst: string (nullable = true)
 |    |-- linktype: string (nullable = true)
 |-- topicID: struct (nullable = false)
 |    |-- id: string (nullable = true)
 |    |-- entityName: string (nullable = true)
 |    |-- name: string (nullable = true)
```

[11]: (None, None)

[12]:
```
#qui faccio un filter perchè un utente può anche followare un altro utente␣
 ↪oltre che un topic
topicsDF = resultDF.filter("userID.entityName='user' AND topicID.
 ↪entityName='topic'")
```

[13]:
```
topicsDF.show(), topicsDF.printSchema()
```

```
+----------------+---------------+--------------------+
|          userID|         follow|             topicID|
+----------------+---------------+--------------------+
|{V1, user, Paolo}|{V1, V4, follow}|{V4, topic, Big D…|
|{V3, user, David}|{V3, V2, follow}|    {V2, topic, SQL}|
|{V3, user, David}|{V3, V4, follow}|{V4, topic, Big D…|
+----------------+---------------+--------------------+

root
 |-- userID: struct (nullable = false)
 |    |-- id: string (nullable = true)
 |    |-- entityName: string (nullable = true)
 |    |-- name: string (nullable = true)
 |-- follow: struct (nullable = false)
 |    |-- src: string (nullable = true)
 |    |-- dst: string (nullable = true)
 |    |-- linktype: string (nullable = true)
 |-- topicID: struct (nullable = false)
 |    |-- id: string (nullable = true)
 |    |-- entityName: string (nullable = true)
 |    |-- name: string (nullable = true)
```

[13]: (None, None)

[14]:
```
finalDF = topicsDF.selectExpr("userID.name AS userName", "topicID.name AS␣
↪topicName")
```

[15]:
```
finalDF.show()
```

```
+--------+---------+
|userName|topicName|
+--------+---------+
|   Paolo| Big Data|
|   David|      SQL|
|   David| Big Data|
+--------+---------+
```

[16]:
```
finalDF.write.csv(outputPath, header=True)
```

ex56

August 18, 2022

[ ]:
```
from pyspark import SparkConf, SparkContext
from pyspark.sql import SparkSession
from graphframes import GraphFrame

conf = SparkConf().setAppName("ex56")
sc = SparkContext(conf=conf)
ssqdl = SparkSession.builder.getOrCreate()
```

[2]:
```
edgesPath = "data/Ex56/data/edges.csv"
vertexesPath = "data/Ex56/data/vertexes.csv"
outputPath = "out56/"
```

[4]:
```
eDF = ssqdl.read.load(
    edgesPath,
    format="csv",
    header=True,
    inferSchema=True
)

vDF = ssqdl.read.load(
    vertexesPath,
    format="csv",
    header=True,
    inferSchema=True
)
```

[5]:
```
eDF.show(), eDF.printSchema()
vDF.show(), vDF.printSchema()
```

```
+---+---+----------+
|src|dst|  linktype|
+---+---+----------+
| V1| V2|      like|
| V1| V3|    follow|
| V1| V4|    follow|
| V3| V2|    follow|
| V3| V4|    follow|
| V5| V2|  expertOf|
```

```
| V2| V4|correlated|
| V4| V2|correlated|
+---+---+----------+

root
 |-- src: string (nullable = true)
 |-- dst: string (nullable = true)
 |-- linktype: string (nullable = true)

+---+----------+--------+
| id|entityName|    name|
+---+----------+--------+
| V1|      user|   Paolo|
| V2|     topic|     SQL|
| V3|      user|   David|
| V4|     topic|Big Data|
| V5|      user|    John|
+---+----------+--------+

root
 |-- id: string (nullable = true)
 |-- entityName: string (nullable = true)
 |-- name: string (nullable = true)
```

[5]:
```
(None, None)
```

[6]:
```
filteredEDF = eDF.filter("linktype='follow' OR linktype='correlated'")
```

[ ]:
```
g = GraphFrame(vDF, filteredEDF)
```

[9]:
```
pathsDF = g.find("(v1)-[e1]->(v2);(v2)-[e2]->(v3)")
pathsDF.show(), pathsDF.printSchema()
```

```
/home/webbelle/univenv/lib/python3.10/site-
packages/pyspark/sql/dataframe.py:127: UserWarning: DataFrame constructor is
internal. Do not directly use it.
  warnings.warn("DataFrame constructor is internal. Do not directly use it.")
+--------------------+--------------------+--------------------+---------------
----+--------------------+
|                  v1|                  e1|                  v2|
e2|                  v3|
+--------------------+--------------------+--------------------+---------------
----+--------------------+
|   {V1, user, Paolo}|    {V1, V3, follow}|   {V3, user, David}|    {V3, V4,
follow}|{V4, topic, Big D…|
|   {V1, user, Paolo}|    {V1, V3, follow}|   {V3, user, David}|    {V3, V2,
```

```
follow}|    {V2, topic, SQL}|
|   {V1, user, Paolo}|    {V1, V4, follow}|{V4, topic, Big D…|{V4, V2,
correlated}|    {V2, topic, SQL}|
|   {V3, user, David}|    {V3, V2, follow}|    {V2, topic, SQL}|{V2, V4,
correlated}|{V4, topic, Big D…|
|   {V3, user, David}|    {V3, V4, follow}|{V4, topic, Big D…|{V4, V2,
correlated}|    {V2, topic, SQL}|
|    {V2, topic, SQL}|{V2, V4, correlated}|{V4, topic, Big D…|{V4, V2,
correlated}|    {V2, topic, SQL}|
|{V4, topic, Big D…|{V4, V2, correlated}|    {V2, topic, SQL}|{V2, V4,
correlated}|{V4, topic, Big D…|
+-------------------+-------------------+-------------------+----------------
----+-------------------+

root
 |-- v1: struct (nullable = false)
 |    |-- id: string (nullable = true)
 |    |-- entityName: string (nullable = true)
 |    |-- name: string (nullable = true)
 |-- e1: struct (nullable = false)
 |    |-- src: string (nullable = true)
 |    |-- dst: string (nullable = true)
 |    |-- linktype: string (nullable = true)
 |-- v2: struct (nullable = false)
 |    |-- id: string (nullable = true)
 |    |-- entityName: string (nullable = true)
 |    |-- name: string (nullable = true)
 |-- e2: struct (nullable = false)
 |    |-- src: string (nullable = true)
 |    |-- dst: string (nullable = true)
 |    |-- linktype: string (nullable = true)
 |-- v3: struct (nullable = false)
 |    |-- id: string (nullable = true)
 |    |-- entityName: string (nullable = true)
 |    |-- name: string (nullable = true)
```

[9]:
```
(None, None)
```

[10]:
```
finalDF = pathsDF.filter("""
    v1.entityName='user'
    AND e1.linktype='follow'
    AND v2.entityName='topic'
    AND e2.linktype='correlated'
    AND v3.entityName='topic'
    AND v3.name='Big Data'
""")
```

```
finalDF.show(), finalDF.printSchema()
```

```
+----------------+---------------+---------------+-------------------+------
--------------+
|              v1|             e1|             v2|                 e2|
v3|
+----------------+---------------+---------------+-------------------+------
--------------+
|{V3, user, David}|{V3, V2, follow}|{V2, topic, SQL}|{V2, V4, correlated}|{V4,
topic, Big D…|
+----------------+---------------+---------------+-------------------+------
--------------+

root
 |-- v1: struct (nullable = false)
 |    |-- id: string (nullable = true)
 |    |-- entityName: string (nullable = true)
 |    |-- name: string (nullable = true)
 |-- e1: struct (nullable = false)
 |    |-- src: string (nullable = true)
 |    |-- dst: string (nullable = true)
 |    |-- linktype: string (nullable = true)
 |-- v2: struct (nullable = false)
 |    |-- id: string (nullable = true)
 |    |-- entityName: string (nullable = true)
 |    |-- name: string (nullable = true)
 |-- e2: struct (nullable = false)
 |    |-- src: string (nullable = true)
 |    |-- dst: string (nullable = true)
 |    |-- linktype: string (nullable = true)
 |-- v3: struct (nullable = false)
 |    |-- id: string (nullable = true)
 |    |-- entityName: string (nullable = true)
 |    |-- name: string (nullable = true)
```

[10]:
```
(None, None)
```

[13]:
```
finalResult = finalDF.selectExpr("v1.name AS USERNAME")
finalResult.write.csv(outputPath, header=True)
```

ex57b

August 18, 2022

[1]:
```
from pyspark import SparkConf, SparkContext
from pyspark.sql import SparkSession
from graphframes import GraphFrame

conf = SparkConf().setAppName("ex57")
sc = SparkContext(conf=conf)
ssql = SparkSession.builder.getOrCreate()
```

```
22/08/18 18:46:17 WARN Utils: Your hostname, webbelle-XPS-15-7590 resolves to a
loopback address: 127.0.1.1; using 192.168.1.62 instead (on interface wlp58s0)
22/08/18 18:46:17 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another
address
```

```
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use
setLogLevel(newLevel).
```

```
22/08/18 18:46:17 WARN NativeCodeLoader: Unable to load native-hadoop library
for your platform… using builtin-java classes where applicable
22/08/18 18:46:18 WARN Utils: Service 'SparkUI' could not bind on port 4040.
Attempting port 4041.
22/08/18 18:46:18 WARN Utils: Service 'SparkUI' could not bind on port 4041.
Attempting port 4042.
22/08/18 18:46:18 WARN Utils: Service 'SparkUI' could not bind on port 4042.
Attempting port 4043.
22/08/18 18:46:18 WARN Utils: Service 'SparkUI' could not bind on port 4043.
Attempting port 4044.
22/08/18 18:46:18 WARN Utils: Service 'SparkUI' could not bind on port 4044.
Attempting port 4045.
22/08/18 18:46:18 WARN Utils: Service 'SparkUI' could not bind on port 4045.
Attempting port 4046.
```

[2]:
```
from pyspark.sql import types
from graphframes.lib import AggregateMessages
from pyspark.sql import functions as F
```

[3]:
```
edgesPath = "data/Ex57b/data/edges.csv"
vertexesPath = "data/Ex57b/data/vertexes.csv"
outputPath = "out57b/"
```

[4]:
```
eDF = ssql.read.load(
    edgesPath,
    format="csv",
    header=True,
    inferSchema=True
)

vDF = ssql.read.load(
    vertexesPath,
    format="csv",
    header=True,
    inferSchema=True
)
```

[5]:
```
eDF.show(), eDF.printSchema()
vDF.show(), vDF.printSchema()
```

```
+---+---+--------+
|src|dst|linktype|
+---+---+--------+
| u1| u2|  friend|
| u1| u4|  friend|
| u1| u5|  friend|
| u2| u1|  friend|
| u2| u3|  follow|
| u3| u2|  follow|
| u4| u1|  friend|
| u4| u5|  friend|
| u5| u1|  friend|
| u5| u4|  friend|
| u5| u6|  follow|
| u6| u3|  follow|
+---+---+--------+

root
 |-- src: string (nullable = true)
 |-- dst: string (nullable = true)
 |-- linktype: string (nullable = true)

+---+-----+---+
| id| name|age|
+---+-----+---+
| u1|Alice| 34|
| u2|  Bob| 36|
| u3| John| 30|
| u4|David| 29|
| u5| Paul| 32|
```

```
| u6| Adel| 36|
| u7| Eddy| 60|
+---+-----+---+

root
 |-- id: string (nullable = true)
 |-- name: string (nullable = true)
 |-- age: integer (nullable = true)
```

[5]:
```
(None, None)
```

[6]:
```
#ritornare un int piuttosto che un boolean ha molto più senso perchè posso␣
 ↪usare una sum come aggregazione dopo!
def checkAge(age):
    if age<35:
        return 1
    else:
        return 0

ssql.udf.register("ageCheck", checkAge, types.IntegerType())
```

[6]:
```
<function __main__.checkAge(age)>
```

[7]:
```
filteredVDF = vDF.selectExpr("*", "ageCheck(age) AS AgeLess35")
```

[8]:
```
filteredVDF.show()
```

```
+---+-----+---+---------+
| id| name|age|AgeLess35|
+---+-----+---+---------+
| u1|Alice| 34|        1|
| u2|  Bob| 36|        0|
| u3| John| 30|        1|
| u4|David| 29|        1|
| u5| Paul| 32|        1|
| u6| Adel| 36|        0|
| u7| Eddy| 60|        0|
+---+-----+---+---------+
```

[9]:
```
g = GraphFrame(filteredVDF, eDF)
```

```
/home/webbelle/univenv/lib/python3.10/site-
packages/pyspark/sql/dataframe.py:148: UserWarning: DataFrame.sql_ctx is an
internal property, and will be removed in future releases. Use
```

```
DataFrame.sparkSession instead.
  warnings.warn(
```

[10]:
```
msgToDst = AggregateMessages.src["AgeLess35"]
```

[ ]:
```
#senza l'import e l'uso di F questo bit non funziona perchè va in conflitto con␣
 ↪python nativo!!!
ageLess35 = g.aggregateMessages(F.sum(AggregateMessages.msg),
    sendToSrc=None,
    sendToDst=msgToDst
).withColumnRenamed("sum(MSG)", "nn")
```

[12]:
```
ageLess35.show()
```

```
+---+---+
| id| nn|
+---+---+
| u3|  0|
| u4|  2|
| u5|  2|
| u1|  2|
| u6|  1|
| u2|  2|
+---+---+
```

[13]:
```
ageLess35.filter("nn>=2").write.csv(outputPath, header=True)
```

ex57

August 18, 2022

[ ]:
```
from pyspark import SparkConf, SparkContext
from pyspark.sql import SparkSession
from graphframes import GraphFrame

conf = SparkConf().setAppName("ex57")
sc = SparkContext(conf=conf)
ssql = SparkSession.builder.getOrCreate()
```

[2]:
```
edgesPath = "data/Ex57/data/edges.csv"
vertexesPath = "data/Ex57/data/vertexes.csv"
outputPath = "out57/"
```

[3]:
```
eDF = ssql.read.load(
    edgesPath,
    format="csv",
    header=True,
    inferSchema=True
)

vDF = ssql.read.load(
    vertexesPath,
    format="csv",
    header=True,
    inferSchema=True
)
```

[4]:
```
eDF.show(), eDF.printSchema()
vDF.show(), vDF.printSchema()
```

```
+---+---+--------+
|src|dst|linktype|
+---+---+--------+
| u1| u2|  friend|
| u1| u4|  friend|
| u1| u5|  friend|
| u2| u1|  friend|
| u2| u3|  follow|
| u3| u2|  follow|
```

```
| u4| u1|  friend|
| u4| u5|  friend|
| u5| u1|  friend|
| u5| u4|  friend|
| u5| u6|  follow|
| u6| u3|  follow|
+---+---+--------+

root
 |-- src: string (nullable = true)
 |-- dst: string (nullable = true)
 |-- linktype: string (nullable = true)

+---+-----+---+
| id| name|age|
+---+-----+---+
| u1|Alice| 34|
| u2|  Bob| 36|
| u3| John| 30|
| u4|David| 29|
| u5| Paul| 32|
| u6| Adel| 36|
| u7| Eddy| 60|
+---+-----+---+

root
 |-- id: string (nullable = true)
 |-- name: string (nullable = true)
 |-- age: integer (nullable = true)
```

[4]: (None, None)

[5]:
```
g = GraphFrame(vDF, eDF)
```

```
/home/webbelle/univenv/lib/python3.10/site-
packages/pyspark/sql/dataframe.py:148: UserWarning: DataFrame.sql_ctx is an
internal property, and will be removed in future releases. Use
DataFrame.sparkSession instead.
  warnings.warn(
```

[6]:
```
shortestPathsDF = g.shortestPaths(["u1"])
```

```
/home/webbelle/univenv/lib/python3.10/site-
packages/pyspark/sql/dataframe.py:127: UserWarning: DataFrame constructor is
internal. Do not directly use it.
  warnings.warn("DataFrame constructor is internal. Do not directly use it.")
```

[7]:
```
shortestPathsDF.show(), shortestPathsDF.printSchema()
```

```
+---+-----+---+---------+
| id| name|age|distances|
+---+-----+---+---------+
| u6| Adel| 36|{u1 -> 3}|
| u3| John| 30|{u1 -> 2}|
| u2|  Bob| 36|{u1 -> 1}|
| u4|David| 29|{u1 -> 1}|
| u5| Paul| 32|{u1 -> 1}|
| u1|Alice| 34|{u1 -> 0}|
| u7| Eddy| 60|       {}|
+---+-----+---+---------+

root
 |-- id: string (nullable = true)
 |-- name: string (nullable = true)
 |-- age: integer (nullable = true)
 |-- distances: map (nullable = true)
 |    |-- key: string
 |    |-- value: integer (valueContainsNull = false)
```

[7]:
```
(None, None)
```

[8]:
```
finalDF = shortestPathsDF.filter("distances['u1']<3 AND id<>'u1'")
finalDF.show()
```

```
+---+-----+---+---------+
| id| name|age|distances|
+---+-----+---+---------+
| u3| John| 30|{u1 -> 2}|
| u2|  Bob| 36|{u1 -> 1}|
| u4|David| 29|{u1 -> 1}|
| u5| Paul| 32|{u1 -> 1}|
+---+-----+---+---------+
```

[10]:
```
resultDF = finalDF.selectExpr("name", "distances['u1'] AS NumHops")
resultDF.show()
```

```
+-----+-------+
| name|NumHops|
+-----+-------+
| John|      2|
|  Bob|      1|
|David|      1|
| Paul|      1|
```

```
+-----+-------+
```

[11]:
```
resultDF.write.csv(outputPath, header=True)
```