Page Rank
GraphFrames API

Fixed number of iterations

```
PR = [1.0 / n]*n
oldPR =[1.0 / n]*n

for iter in xrange(maxIter);
     swap(oldPR, PR)
     for i in xrange(n):
          PR[i]=(1-alpha)/n+alpha * sum(map(lambda j:oldPR[j]/outDeg[j], inNBRS[i]))
```

Fixed number of iterations

```
PR = [1.0 / n]*n
oldPR =[1.0 / n]*n

for iter in xrange(maxIter);
     swap(oldPR, PR)
     for i in xrange(n):
          PR[i]=(1-alpha)/n+alpha * sum(map(lambda j:oldPR[j]/outDeg[j], inNBRS[i]))
```

Fixed number of iterations

```
PR = [1.0 / n]*n
oldPR =[1.0 / n]*n

for iter in xrange(maxIter);
     swap(oldPR, PR)
     for i in xrange(n):
          PR[i]=(1-alpha)/n+alpha * sum(map(lambda j:oldPR[j]/outDeg[j], inNBRS[i]))
```

Fixed number of iterations

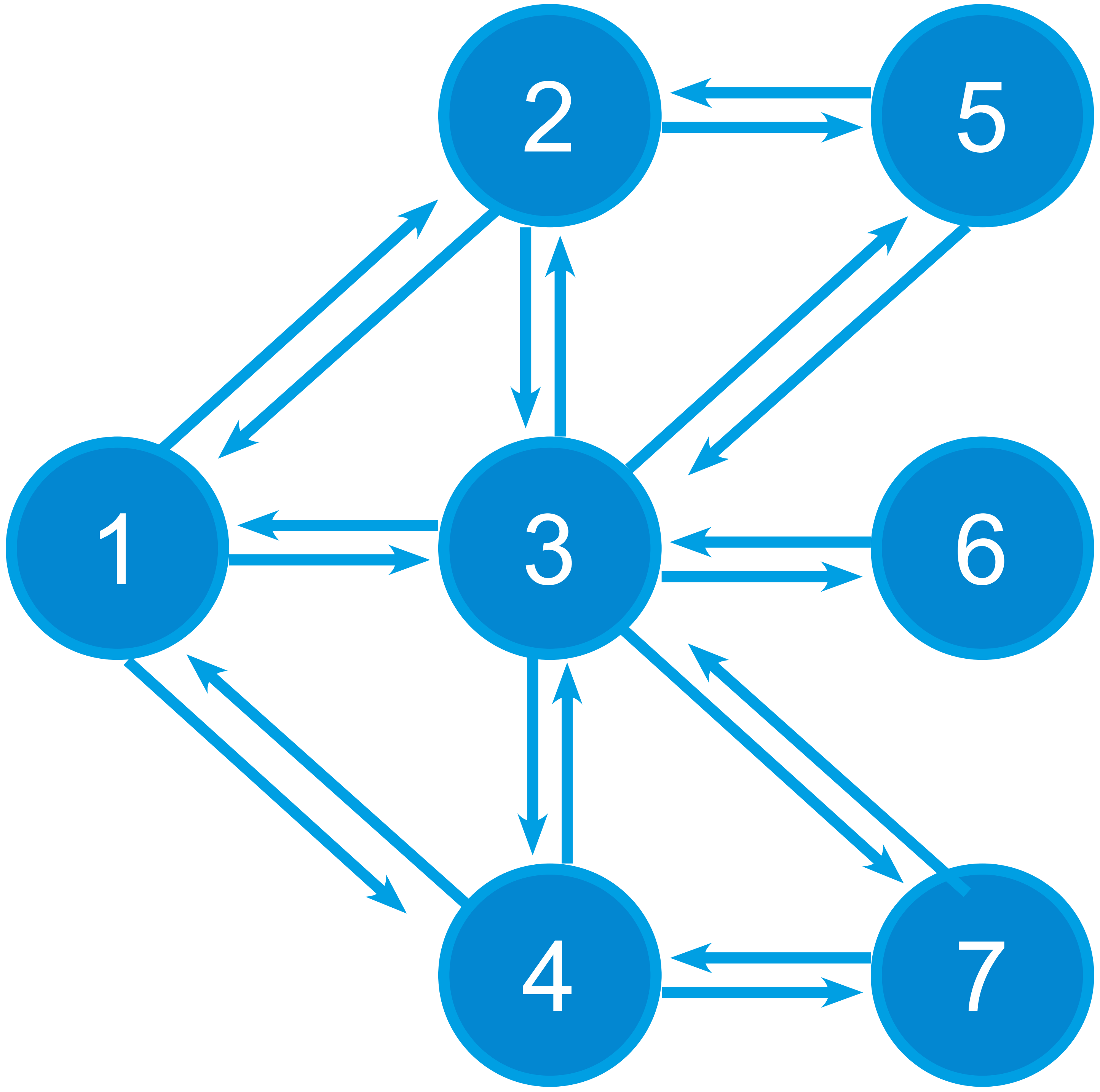
```
PR = [1.0 / n]*n
oldPR =[1.0 / n]*n

for iter in xrange(maxIter);
     swap(oldPR, PR)
     for i in xrange(n):
          PR[i]=(1-alpha)/n+alpha * sum(map(lambda j:oldPR[j]/outDeg[j], inNBRS[i]))
```

Fixed number of iterations

```
PR = [1.0 / n]*n
oldPR =[1.0 / n]*n

for iter in xrange(maxIter);
     swap(oldPR, PR)
     for i in xrange(n):
          PR[i]=(1-alpha)/n+alpha * sum(map(lambda j:oldPR[j]/outDeg[j], inNBRS[i]))
```

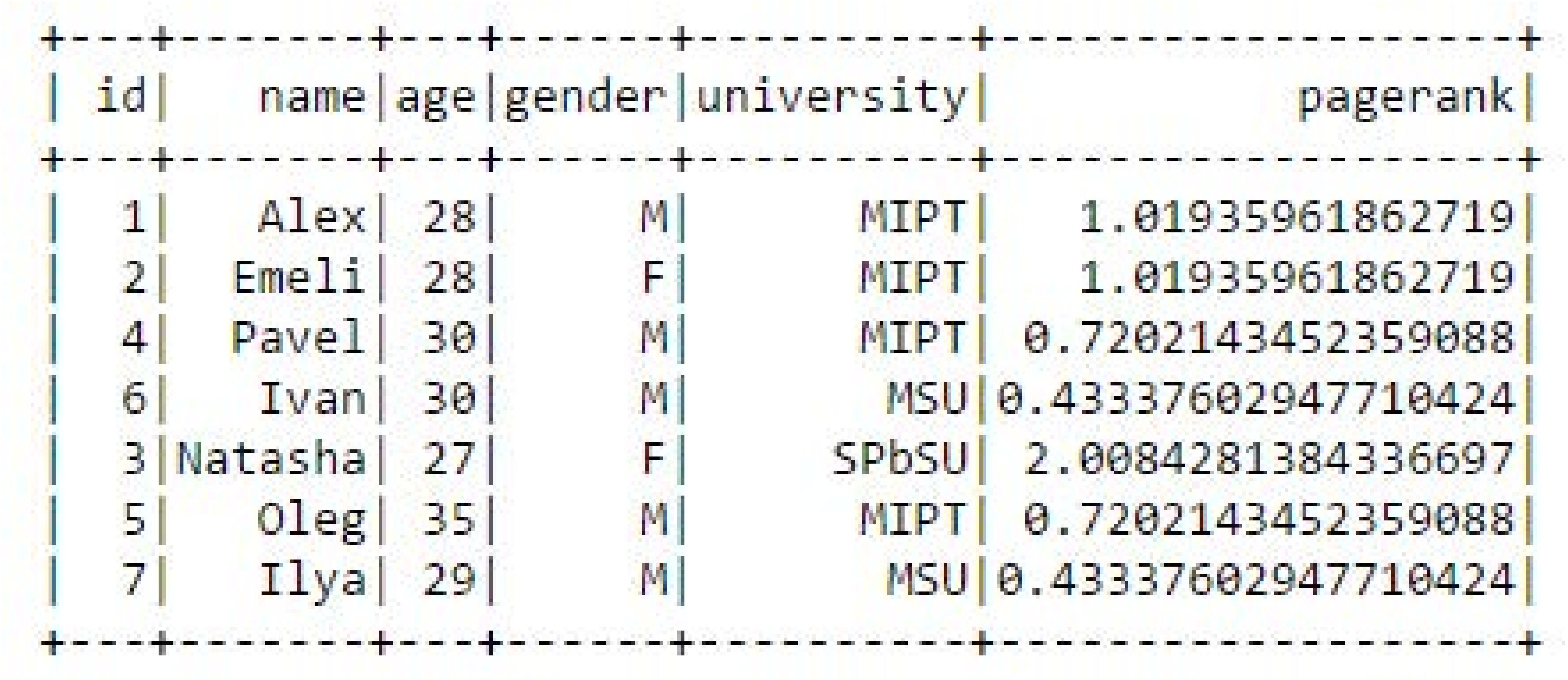
Until Convergence

```
PR = [1.0 / n]*n
oldPR =[1.0 / n]*n

while abs(PR-oldPR) >tol;
   swap(oldPR,PR)
   for i in xrange(n):
       PR[i]=(1-alpha)/n+alpha * sum(map(lambda j:oldPR[j]/outDeg[j], inNBRS[i]))
```

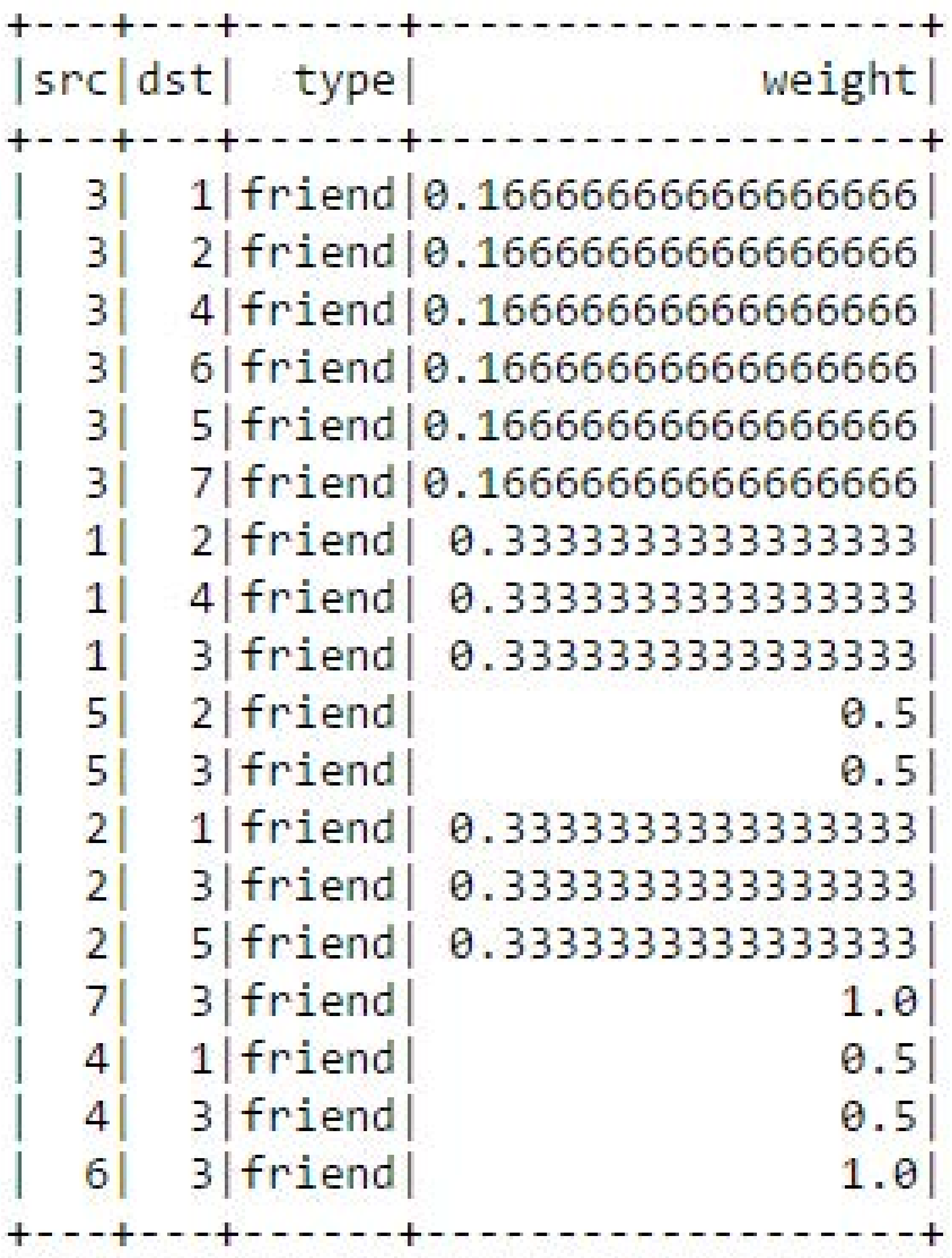

2
5
1
3
6
4
7

```
from graphframes.examples import Graphs

vertices = sparkSession.createDataFrame([
   ("1"."Alex", 28, "M","MIPT"),
   ("2","Emeli", 28, "F","MIPT"),
   ("3","Natasha", 27, "F","SPbSU"),
   ("4","Pavel", 30, "M","MIPT"),
   ("5","Oleg", 35, "M","MIPT"),
   ("6","Ivan", 30, "M","MSU"),
   ("7","Ilya", 29, "M","MSU")], ["id","name","age","gender","university"])

edges = sparkSession.createDataFrame([
   ("1","2","friend"),("2","1","friend"),
   ("1","3","friend"),("3","1","friend"),
   ("1","4","friend"),("4","1","friend"),
   ("2","3","friend"),("3","2","friend"),
   ("2","5","friend"),("5","2","friend"),
   ("3","4","friend"),("4","3","friend"),
   ("3","5","friend"),("5","3","friend"),
   ("3","6","friend"),("6","3","friend"),
   ("3","7","friend"),("7","3","friend"),
], ["src", "dst" , "type"])
g = GraphFrame(vertices, edges)
```

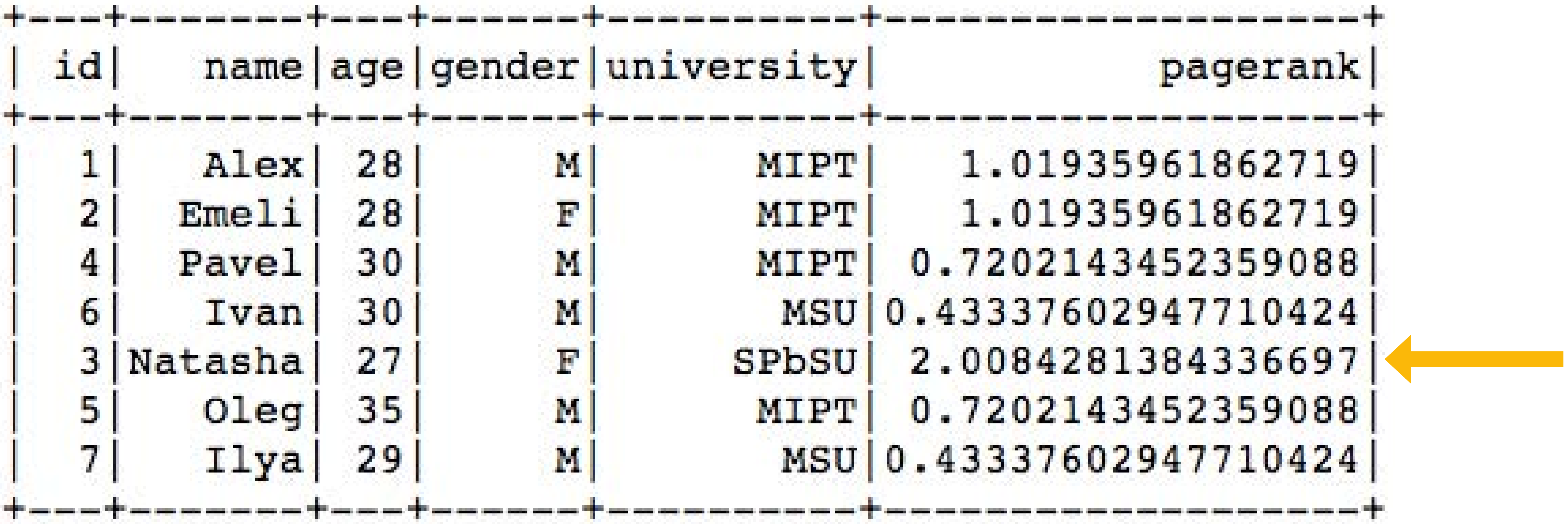
```
# Run PageRank until convergence to tolerance "tol".
results = g.pageRank(resetProbability=0.15, tol=0.01)
results.vertices.show()
```



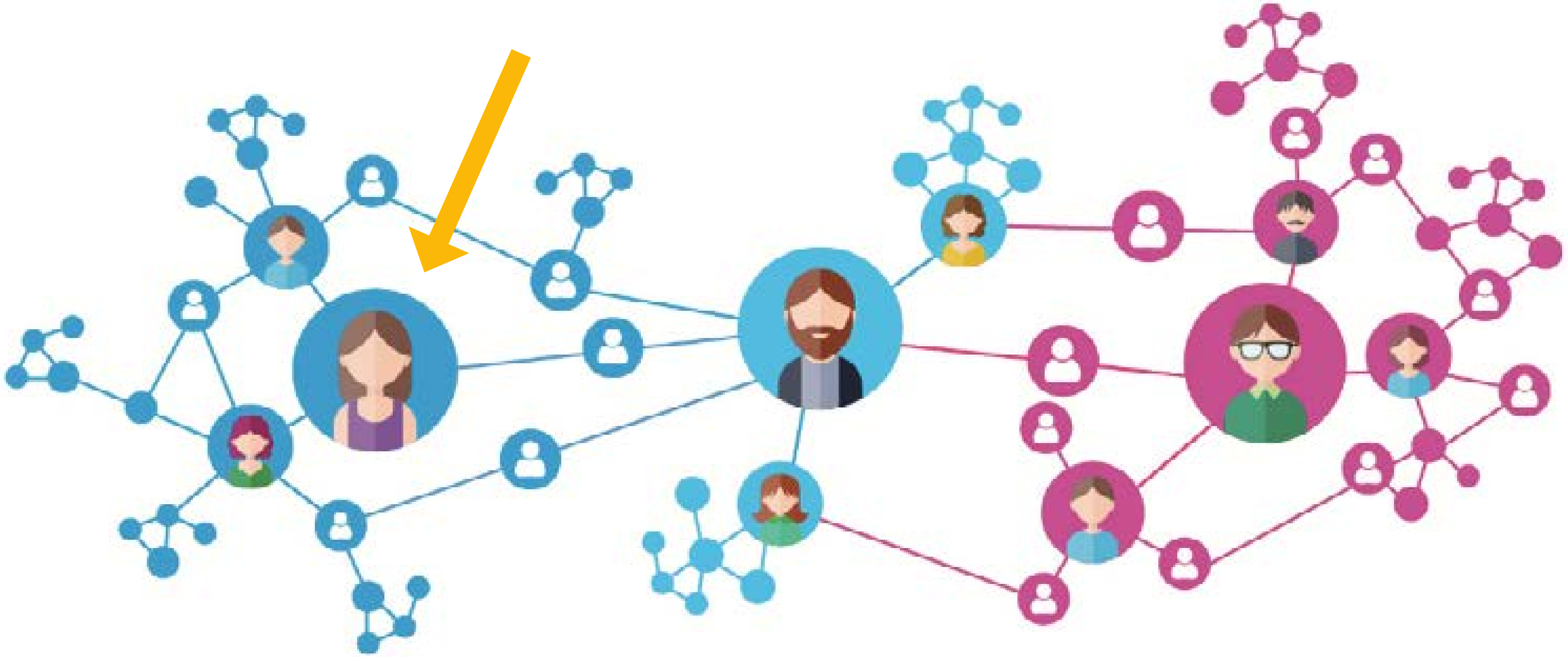
```
+---+-------+---+------+----------+-------------------+
| id|   name|age|gender|university|           pagerank|
+---+-------+---+------+----------+-------------------+
|  1|   Alex| 28|     M|      MIPT|   1.01935961862719|
|  2|  Emeli| 28|     F|      MIPT|   1.01935961862719|
|  4|  Pavel| 30|     M|      MIPT| 0.7202143452359088|
|  6|   Ivan| 30|     M|       MSU|0.43337602947710424|
|  3|Natasha| 27|     F|     SPbSU| 2.0084281384336697|
|  5|   Oleg| 35|     M|      MIPT| 0.7202143452359088|
|  7|   Ilya| 29|     M|       MSU|0.43337602947710424|
+---+-------+---+------+----------+-------------------+
results.edges.show()
```



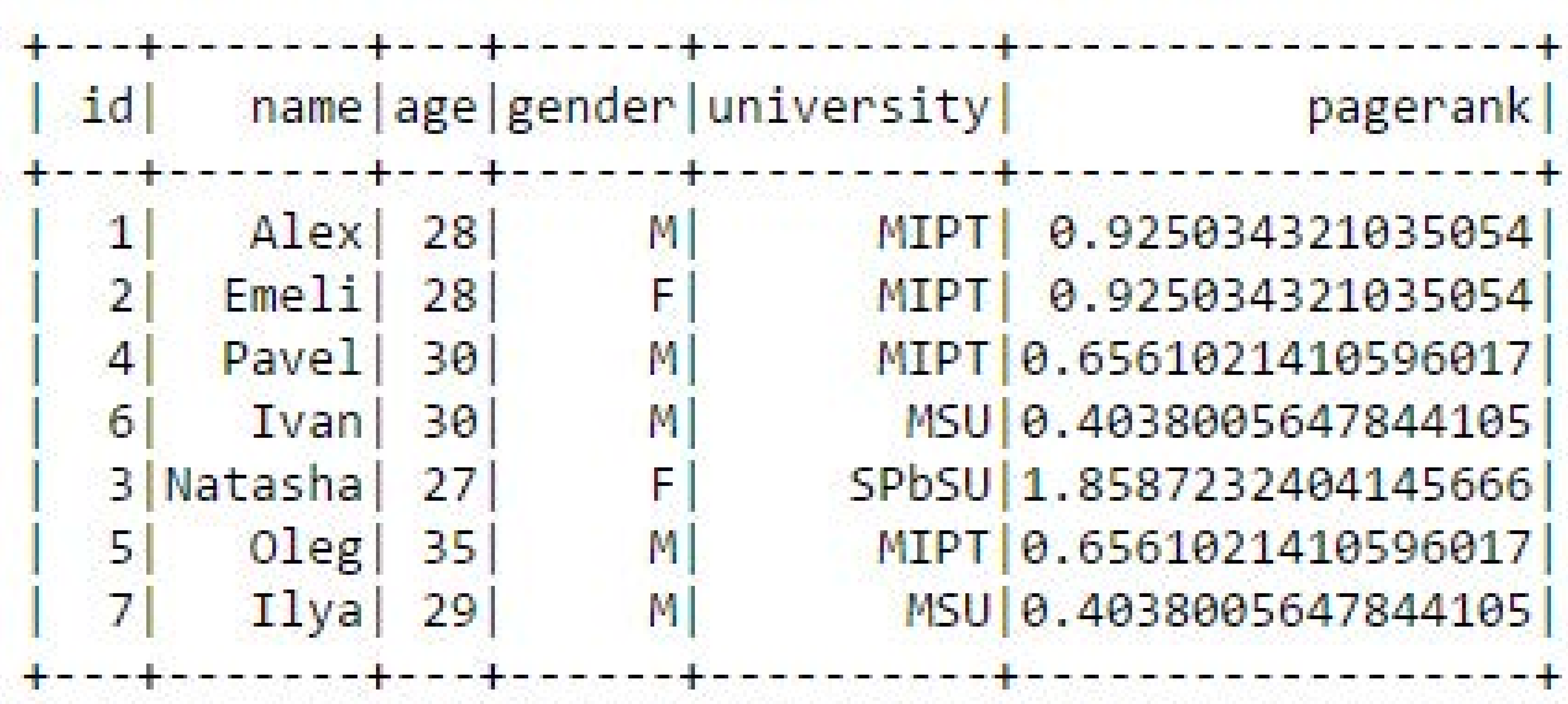
```
+---+---+------+------------------+
|src|dst|  type|            weight|
+---+---+------+------------------+
|  3|  1|friend|0.1666666666666666|
|  3|  2|friend|0.1666666666666666|
|  3|  4|friend|0.1666666666666666|
|  3|  6|friend|0.1666666666666666|
|  3|  5|friend|0.1666666666666666|
|  3|  7|friend|0.1666666666666666|
|  1|  2|friend| 0.3333333333333333|
|  1|  4|friend| 0.3333333333333333|
|  1|  3|friend| 0.3333333333333333|
|  5|  2|friend|               0.5|
|  5|  3|friend|               0.5|
|  2|  1|friend| 0.3333333333333333|
|  2|  3|friend| 0.3333333333333333|
|  2|  5|friend| 0.3333333333333333|
|  7|  3|friend|               1.0|
|  4|  1|friend|               0.5|
|  4|  3|friend|               0.5|
|  6|  3|friend|               1.0|
+---+---+------+------------------+
```
]



```
+---+-------+---+------+----------+-------------------+
| id|   name|age|gender|university|           pagerank|
+---+-------+---+------+----------+-------------------+
|  1|   Alex| 28|     M|      MIPT|   1.01935961862719|
|  2|  Emeli| 28|     F|      MIPT|   1.01935961862719|
|  4|  Pavel| 30|     M|      MIPT| 0.7202143452359088|
|  6|   Ivan| 30|     M|       MSU|0.43337602947710424|
|  3|Natasha| 27|     F|     SPbSU| 2.0084281384336697|
|  5|   Oleg| 35|     M|      MIPT| 0.7202143452359088|
|  7|   Ilya| 29|     M|       MSU|0.43337602947710424|
+---+-------+---+------+----------+-------------------+
```



```
# Run PageRank for a fixed number of iterations.
results2 = g.pageRank(resetProbability=0.15, maxIter=10)
results2.vertices.show()
```



```
+---+-------+---+------+----------+------------------+
| id|   name|age|gender|university|          pagerank|
+---+-------+---+------+----------+------------------+
|  1|   Alex| 28|     M|      MIPT| 0.925034321035054|
|  2|  Emeli| 28|     F|      MIPT| 0.925034321035054|
|  4|  Pavel| 30|     M|      MIPT|0.6561021410596017|
|  6|   Ivan| 30|     M|       MSU|0.4038005647844105|
|  3|Natasha| 27|     F|     SPbSU|1.8587232404145666|
|  5|   Oleg| 35|     M|      MIPT|0.6561021410596017|
|  7|   Ilya| 29|     M|       MSU|0.4038005647844105|
+---+-------+---+------+----------+------------------+
```

Summary

- Parameters GraphFrames implementation of Page Rank Algorithm has

Summary

- Parameters GraphFrames implementation of Page Rank Algorithm has
- How to tune them