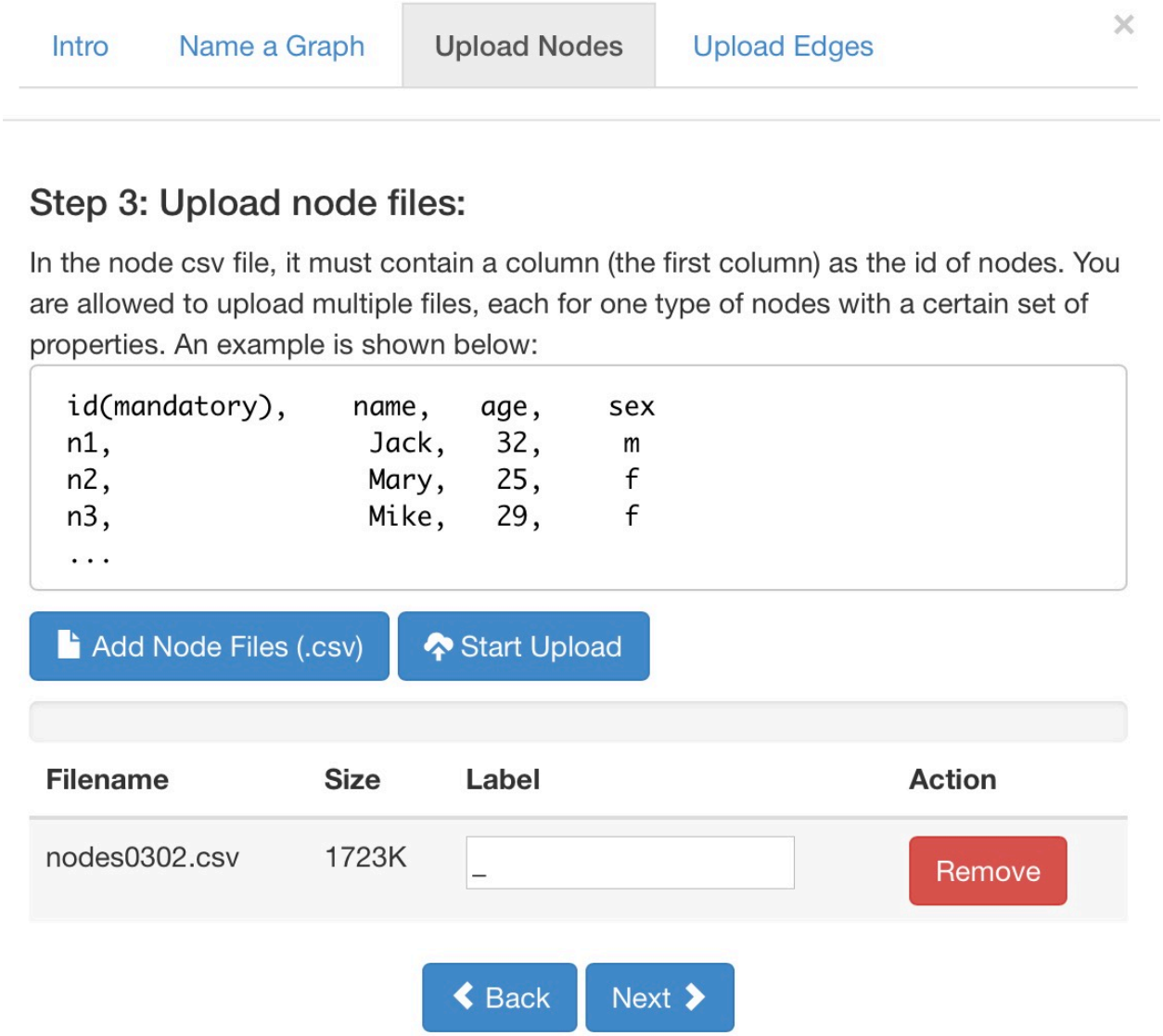
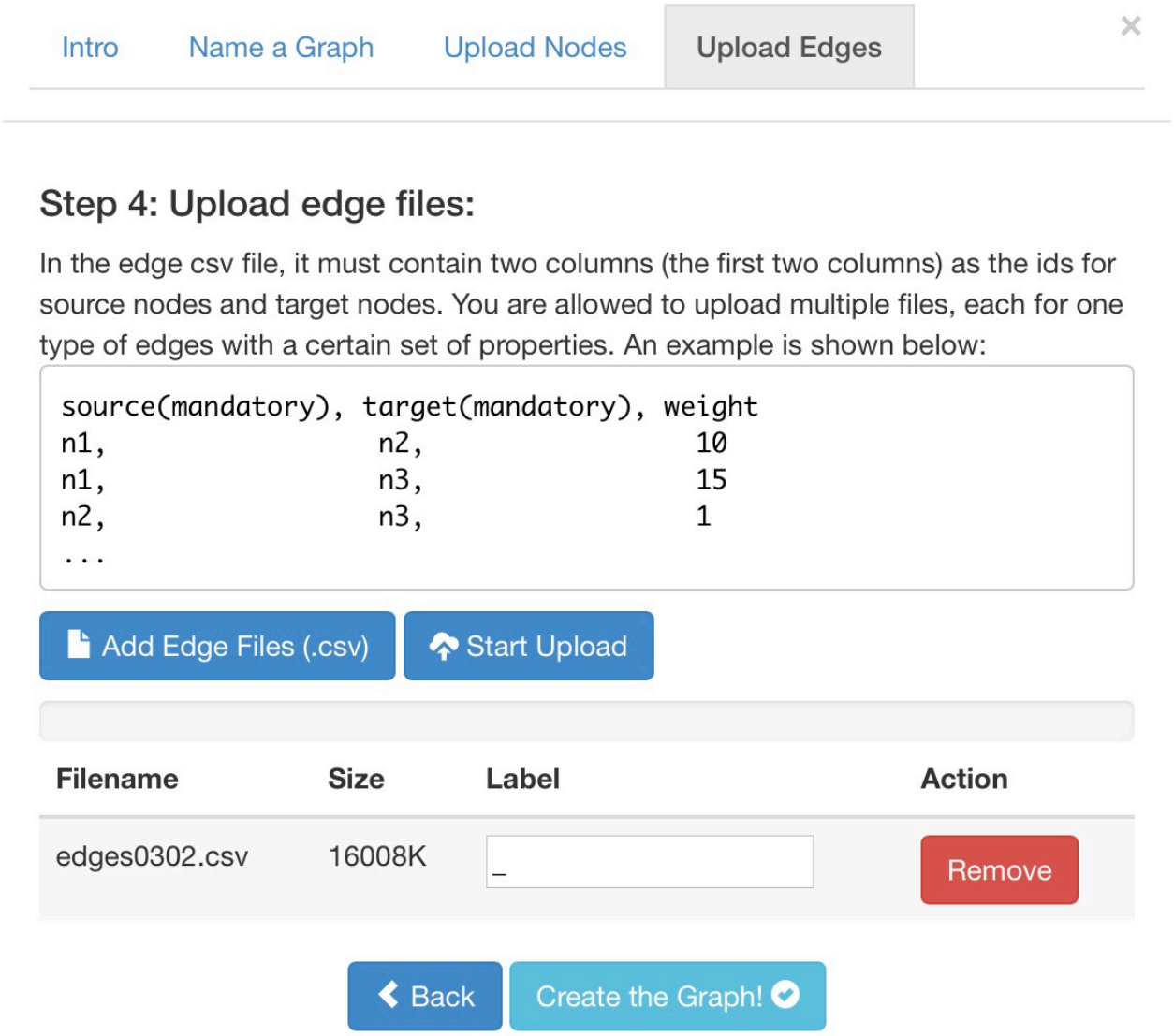
--code of part 2 item-to –item system G

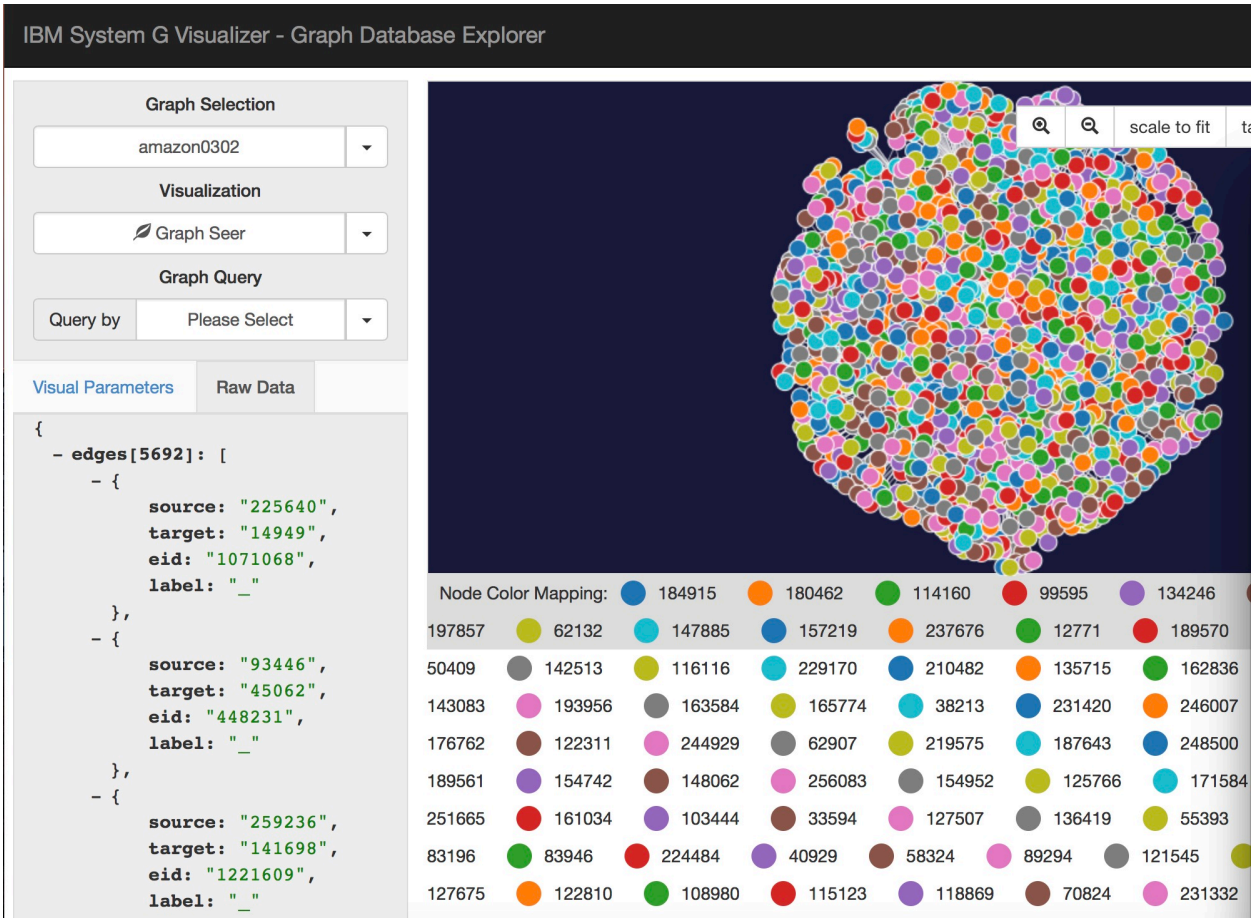
The graph name is named as amazon0302

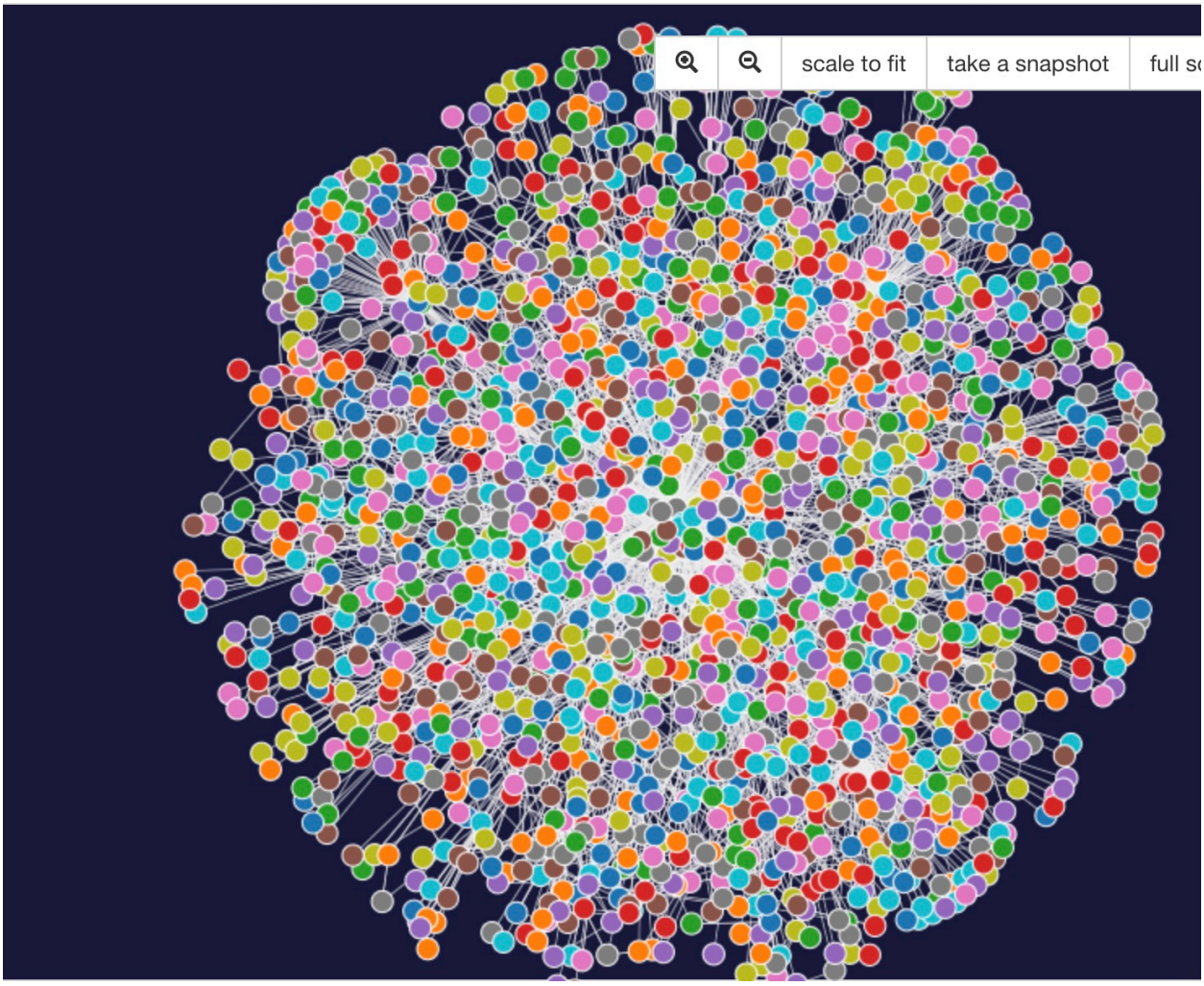
Load one of data amazon 0302 into System G:





The whole graph of data 0302: amazon0302


IBM System G Visualizer - Graph Database Explorer
Graph Selection
amazon0302
Visualization
Graph Seer
Graph Query
Query by
Please Select
Visual Parameters
Raw Data
{
- edges[5692]: [
- {
source: "225640",
target: "14949",
eid: "1071068",
label: "_"
},
- {
source: "93446",
target: "45062",
eid: "448231",
label: "_"
},
- {
source: "259236",
target: "141698",
eid: "1221609",
label: "_"
scale to fit
Node Color Mapping:
184915
180462
114160
99595
134246
197857
62132
147885
157219
237676
12771
189570
50409
142513
116116
229170
210482
135715
162836
143083
193956
163584
165774
38213
231420
246007
176762
122311
244929
62907
219575
187643
248500
189561
154742
148062
256083
154952
125766
171584
251665
161034
103444
33594
127507
136419
55393
83196
83946
224484
40929
58324
89294
121545
127675
122810
108980
115123
118869
70824
231332



For node 14949(with the max degree)

--Find the node with the max degree: (number of neighbors/edges): find_vertex_max_degree

find_vertex_max_degree --graph amazon0302 --showall

--Result: node id 14949

```
>>Query executed: [find_vertex_max_degree --graph amazon0302 --showall ]
>>Result received:
>>all-degree: 425
>>Rendering graph...
>>Rendering graph finished.
```

--The subgraph connected with id 14949(choose depth 2)

get_egonet –graph amazon0302 –id 14949 –depth 2

Use collaborative filtering for recommendation(top 10 items):

Collaborative filtering: analytic_colfilter.The analytic performs BFS for each vertex x up to --depth (must be an even number, default is 4), computes the number of paths N(x, y) from x to every vertex in Y, and ranks these vertices based on their N(x, y) values in a descending order. The top-ranked vertices up to the value specified by --topnum (must be at least 10, default is 100) are returned.

--For id 14949:

analytic_colfilter --graph amazon0302 –id 14949 --topnum 10 --nomultiedges --redirect Amazontest1

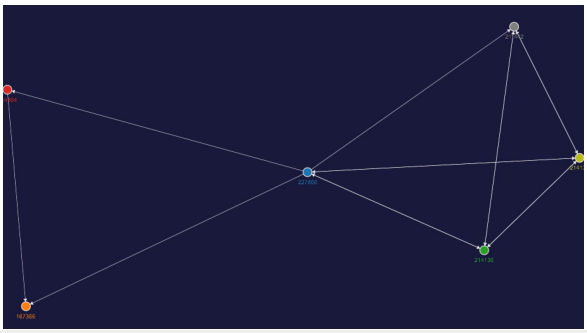
--Result:


```
{
"edges":[
{"source":"14949","target":"8459","count":3},
{"source":"14949","target":"9120","count":2},
{"source":"14949","target":"7604","count":2},
{"source":"14949","target":"7608","count":2},
{"source":"14949","target":"14311","count":2},
{"source":"14949","target":"52466","count":2},
{"source":"14949","target":"15725","count":2},
{"source":"14949","target":"15724","count":2},
{"source":"14949","target":"13222","count":2},
{"source":"14949","target":"18943","count":2}
],
"nodes":[
{"id":"8459","count":3},
{"id":"9120","count":2},
{"id":"7604","count":2},
{"id":"7608","count":2},
{"id":"14311","count":2},
{"id":"52466","count":2},
{"id":"15725","count":2},
{"id":"15724","count":2},
{"id":"13222","count":2},
{"id":"18943","count":2}
],
"summary":[{"number of nodes":10}],
"time":[{"TIME":"0.000736952"}]
}
```


For node 227866 as example

--The subgraph connected with id 14949(choose depth 1)

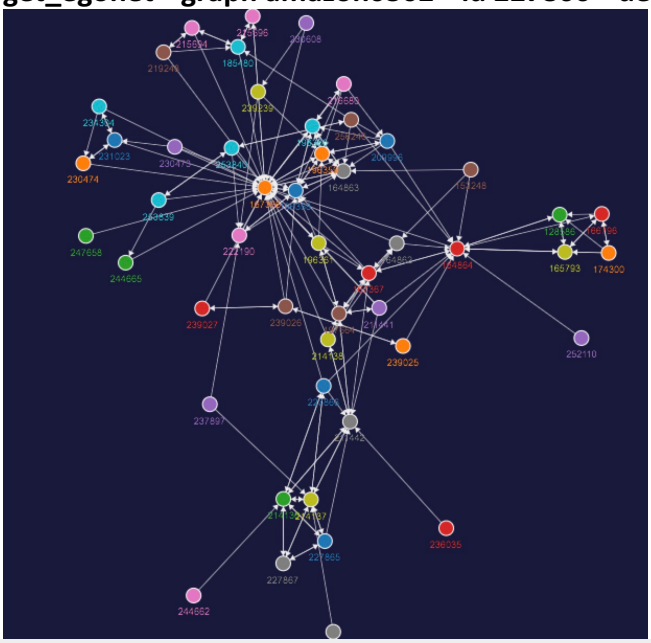
get_egonet --graph amazon0302 --id 227866 --depth 1



```
>>Result received:
>>number of nodes: 5
>>number of edges: 9
```

--The subgraph connected with id 14949(choose depth 2)

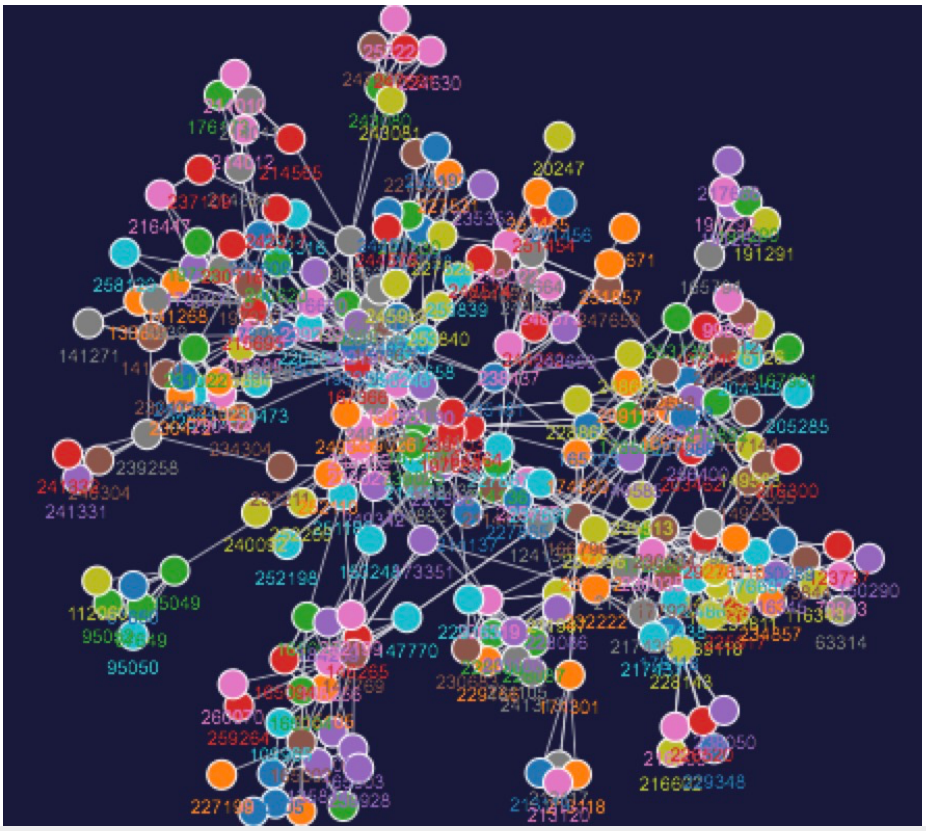
get_egonet --graph amazon0302 --id 227866 --depth 2



```
>>Result received:
>>number of nodes: 49
>>number of edges: 169
```

--The subgraph connected with id 14949(choose depth 4)

get_egonet --graph amazon0302 --id 227866 --depth 4



```
>>Result received:
>>number of nodes: 245
>>number of edges: 969
```