CSE 538 Graph Database and Graph Analytics

Project 2

Octavio Villalaz

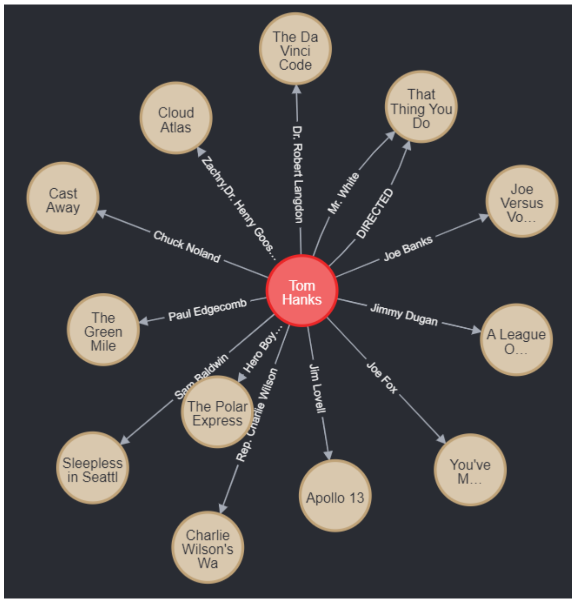
19/6/2024

1)

1.1)

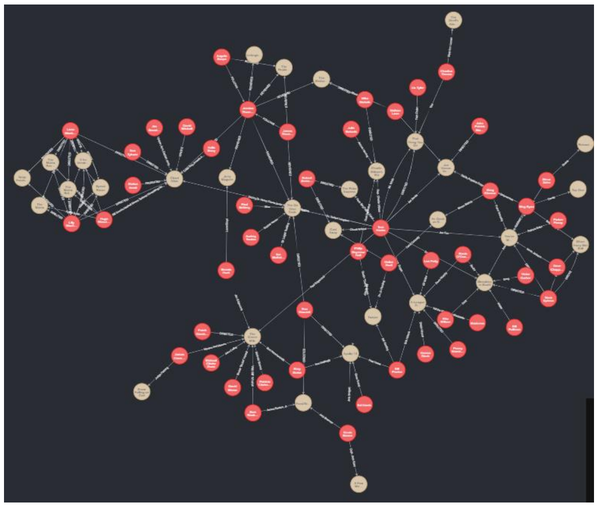
-Directional

```
MATCH p = (tom:Person {name: "Tom Hanks"})-[*1..3]->(reachable)
RETURN p
```



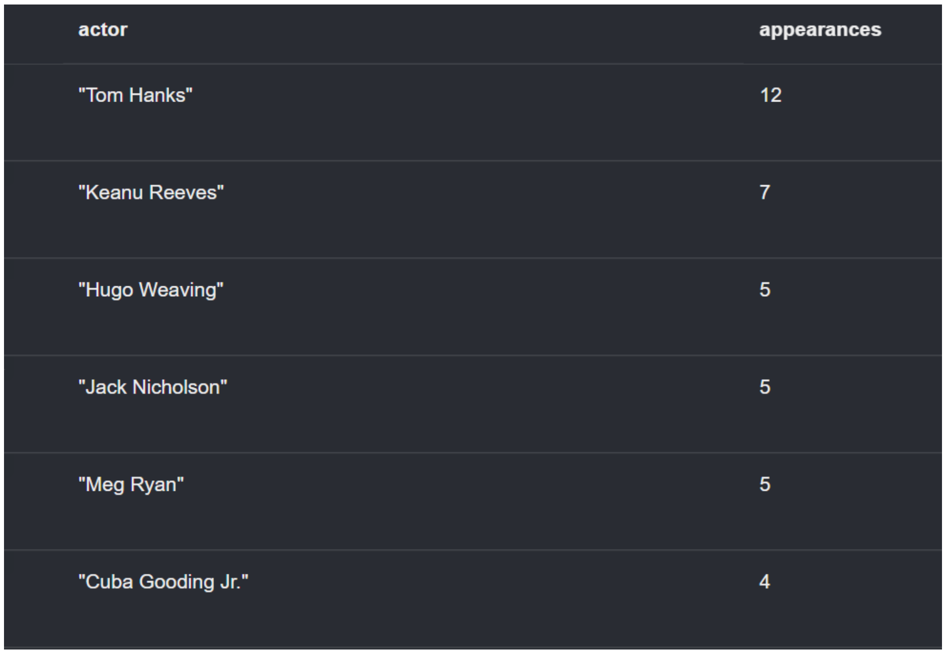
-Unidirectional

```
MATCH p = (tom:Person {name: "Tom Hanks"})-[*1..3]-(reachable)
RETURN p
```



1.2)

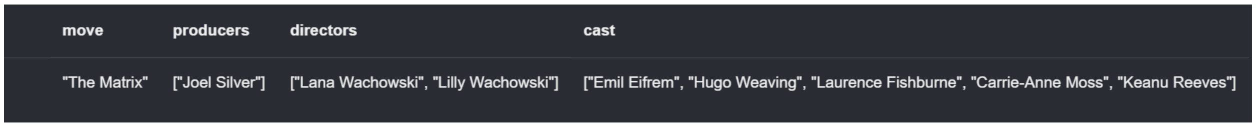
```
MATCH (a:Person)-[r:ACTED_IN]->(m:Movie)

WITH a, COUNT(r) AS appearances

RETURN a.name AS actor, appearances

ORDER BY appearances DESC
```



actor	appearances
"Tom Hanks"	12
"Keanu Reeves"	7
"Hugo Weaving"	5
"Jack Nicholson"	5
"Meg Ryan"	5
"Cuba Gooding Jr."	4

1.3)

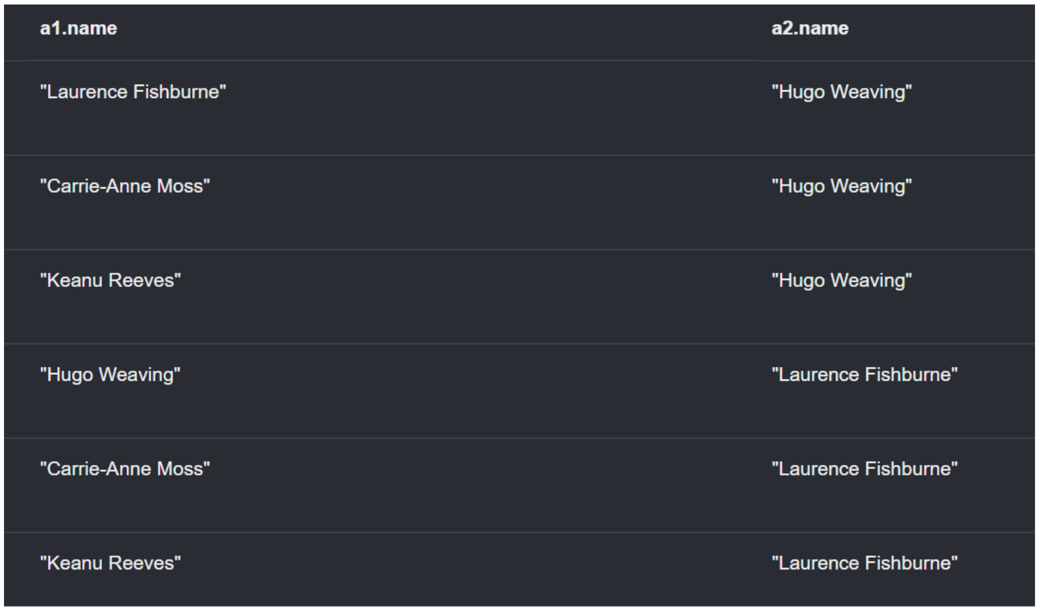
```
MATCH(m:Movie{title: "The Matrix"})

OPTIONAL MATCH (m)<- [:PRODUCED]-(p:Person)

OPTIONAL MATCH (m)<- [:DIRECTED]-(d:Person)

OPTIONAL MATCH (m)<- [:ACTED_IN]-(a:Person)

RETURN m.title AS move,

   collect(DISTINCT p.name) AS producers,

   collect(DISTINCT d.name) AS directors,

   collect(DISTINCT a.name) AS cast
```



move	producers	directors	cast
"The Matrix"	["Joel Silver"]	["Lana Wachowski", "Lilly Wachowski"]	["Emil Eifrem", "Hugo Weaving", "Laurence Fishburne", "Carrie-Anne Moss", "Keanu Reeves"]

1.4)

```
MATCH (a1:Person)-[:ACTED_IN]->(m:Movie)<-[:ACTED_IN]-(a2:Person)

WHERE a1<>a2

WITH a1, a2, COUNT(m) AS MovieCount

WHERE MovieCount = 3

RETURN a1.name, a2.name
```

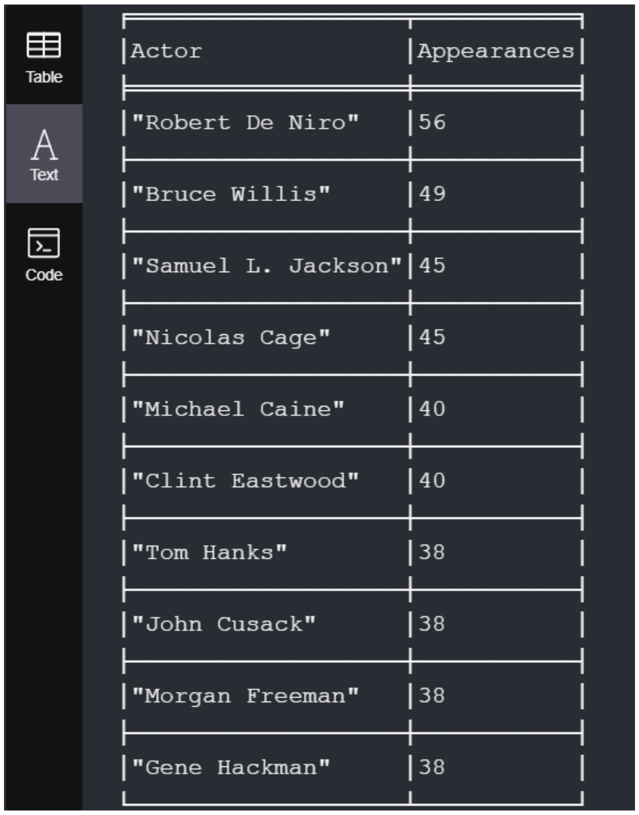


a1.name	a2.name
"Laurence Fishburne"	"Hugo Weaving"
"Carrie-Anne Moss"	"Hugo Weaving"
"Keanu Reeves"	"Hugo Weaving"
"Hugo Weaving"	"Laurence Fishburne"
"Carrie-Anne Moss"	"Laurence Fishburne"
"Keanu Reeves"	"Laurence Fishburne"

2)

2.1)

```
MATCH (a:Actor)-[:ACTED_IN]->(m:Movie)

RETURN a.name AS Actor, COUNT(m) AS Appearances

ORDER BY Appearances DESC

LIMIT 10
```

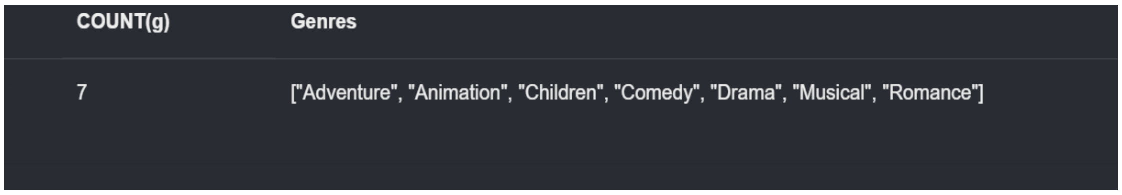


```
╒══════════════════╤═══════════╕
│Actor             │Appearances│
╞══════════════════╪═══════════╡
│"Robert De Niro"  │56         │
├──────────────────┼───────────┤
│"Bruce Willis"    │49         │
├──────────────────┼───────────┤
│"Samuel L. Jackson"│45        │
├──────────────────┼───────────┤
│"Nicolas Cage"    │45         │
├──────────────────┼───────────┤
│"Michael Caine"   │40         │
├──────────────────┼───────────┤
│"Clint Eastwood"  │40         │
├──────────────────┼───────────┤
│"Tom Hanks"       │38         │
├──────────────────┼───────────┤
│"John Cusack"     │38         │
├──────────────────┼───────────┤
│"Morgan Freeman"  │38         │
├──────────────────┼───────────┤
│"Gene Hackman"    │38         │
└──────────────────┴───────────┘
```

2.2)

MATCH(m:Movie{title: "Mulan"})-[:IN_GENRE]->(g:Genre)

RETURN COUNT(g), COLLECT(g.name) AS Genres



COUNT(g)	Genres
7	["Adventure", "Animation", "Children", "Comedy", "Drama", "Musical", "Romance"]

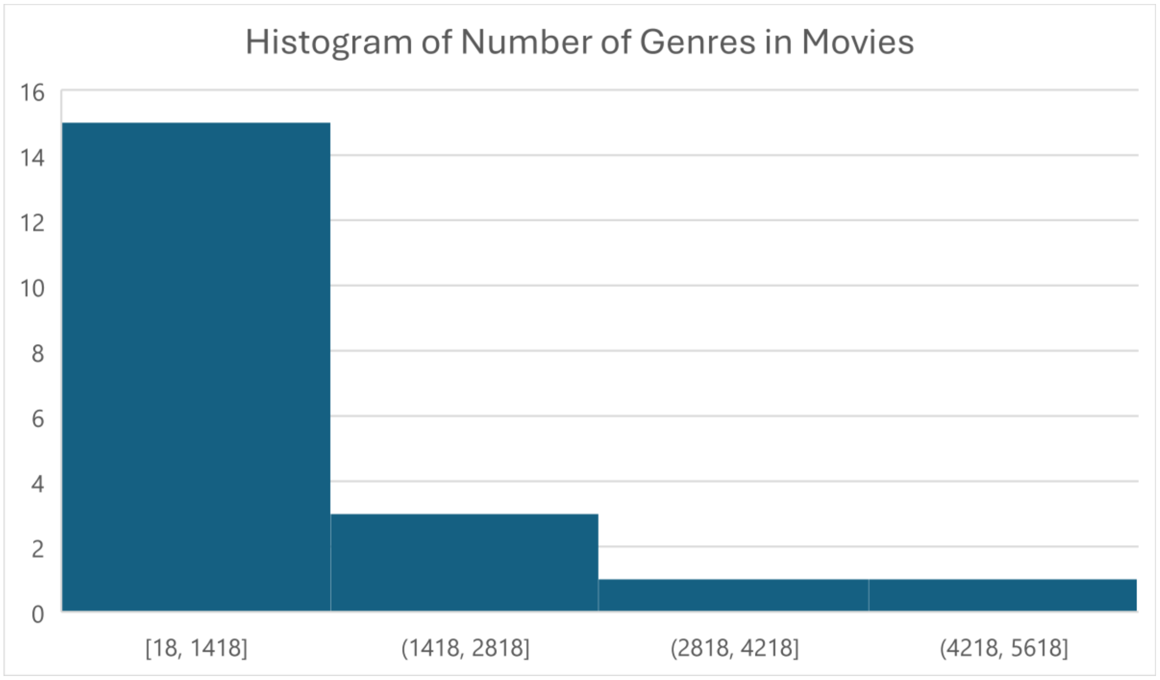
2.3)

MATCH (m:Movie)-[:IN_GENRE]->(g:Genre)

WITH m, g

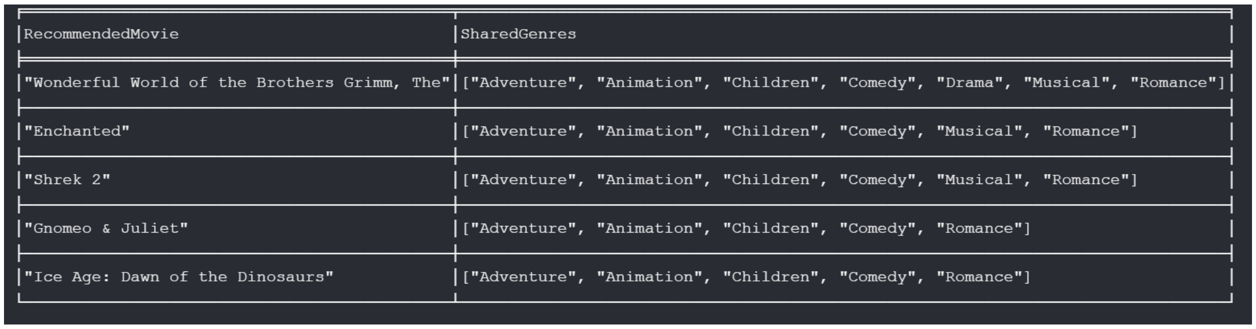
RETURN g.name AS Genres, COUNT(m) AS MovieCount

ORDER BY MovieCount



2.4)

```
MATCH(m:Movie{title:"Mulan"})-[:IN_GENRE]-> (g:Genre)<-[:IN_GENRE]-(rec:Movie)

WHERE m<>rec

WITH rec, COLLECT(g.name) AS SharedGenres

WHERE SIZE(SharedGenres) > 4

RETURN DISTINCT rec.title AS RecommendedMovie,SharedGenres

ORDER BY SIZE(SharedGenres) DESC

Limit 5
```



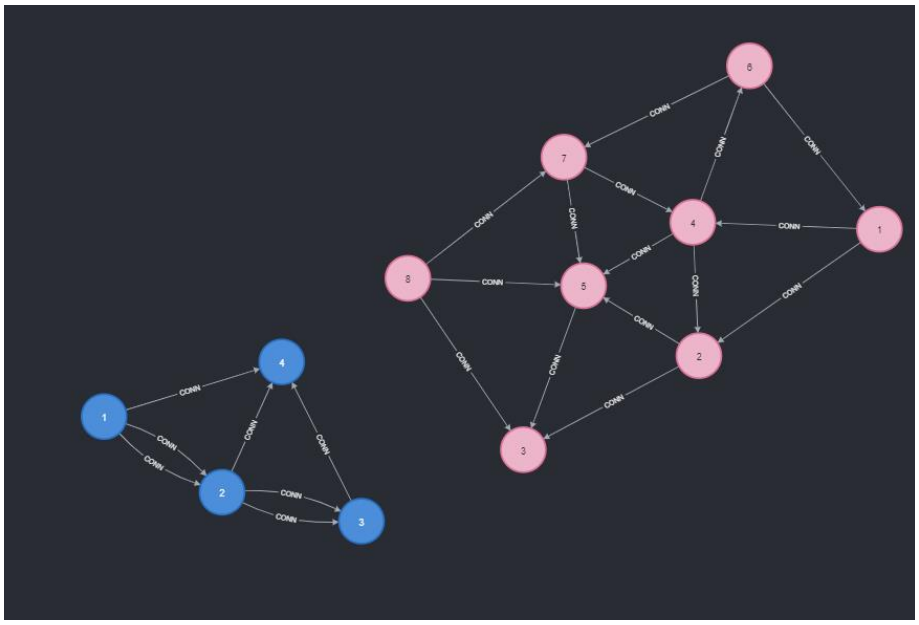
```
╒═════════════════════════════════════════╤═══════════════════════════════════════════════════════════════════════════════╕
│RecommendedMovie                         │SharedGenres                                                                   │
╞═════════════════════════════════════════╪═══════════════════════════════════════════════════════════════════════════════╡
│"Wonderful World of the Brothers Grimm, The"│["Adventure", "Animation", "Children", "Comedy", "Drama", "Musical", "Romance"]│
├─────────────────────────────────────────┼───────────────────────────────────────────────────────────────────────────────┤
│"Enchanted"                              │["Adventure", "Animation", "Children", "Comedy", "Musical", "Romance"]          │
├─────────────────────────────────────────┼───────────────────────────────────────────────────────────────────────────────┤
│"Shrek 2"                                │["Adventure", "Animation", "Children", "Comedy", "Musical", "Romance"]          │
├─────────────────────────────────────────┼───────────────────────────────────────────────────────────────────────────────┤
│"Gnomeo & Juliet"                        │["Adventure", "Animation", "Children", "Comedy", "Romance"]                     │
├─────────────────────────────────────────┼───────────────────────────────────────────────────────────────────────────────┤
│"Ice Age: Dawn of the Dinosaurs"         │["Adventure", "Animation", "Children", "Comedy", "Romance"]                     │
└─────────────────────────────────────────┴───────────────────────────────────────────────────────────────────────────────┘
```

Explanation:

This query helps the viewer who has seen the movie Mulan find recommendations of new movies to watch based on the genres. It recommends the top 5 movies that have more than 4 genres in common with the movie Mulan, showing the ones that have more genres in common first.
This query can be modified to show more recommendations by editing the Limit number.

3)

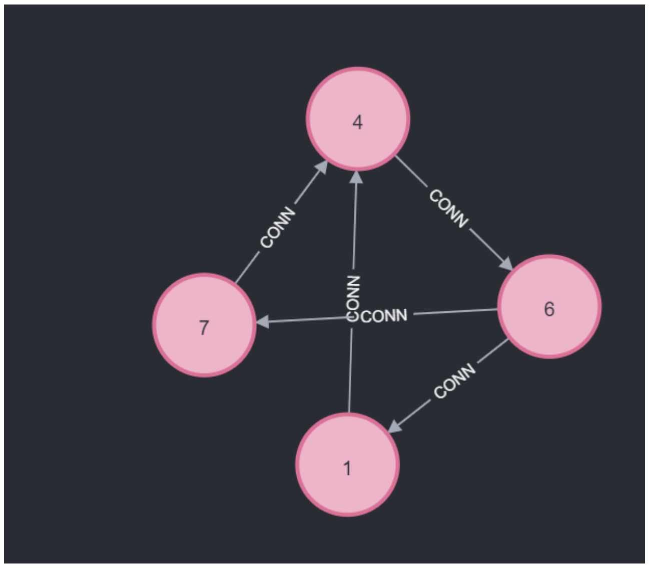
3.1)



3.2)

Directed Triangles

- (1,4,6)
- (4,6,7)

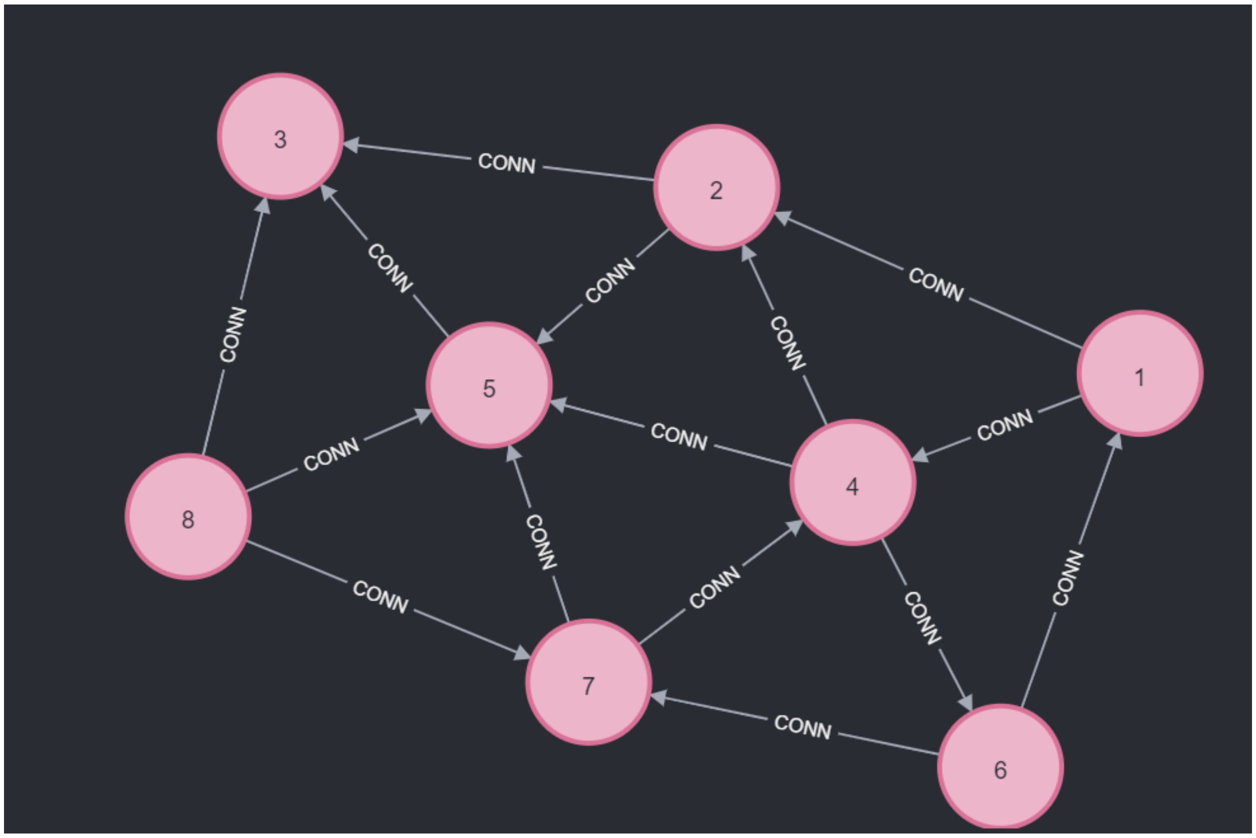


When using count(path) it counts all possible combinations of the triangle for example for the triangle (1,4,6) it will also count (4,6,1) and (6,1,4). That's why it appears a count of 6, because 2 triangles can make 6 possible combinations.

3.3)

match path = (v:G)-[:CONN*3]-(v:G)

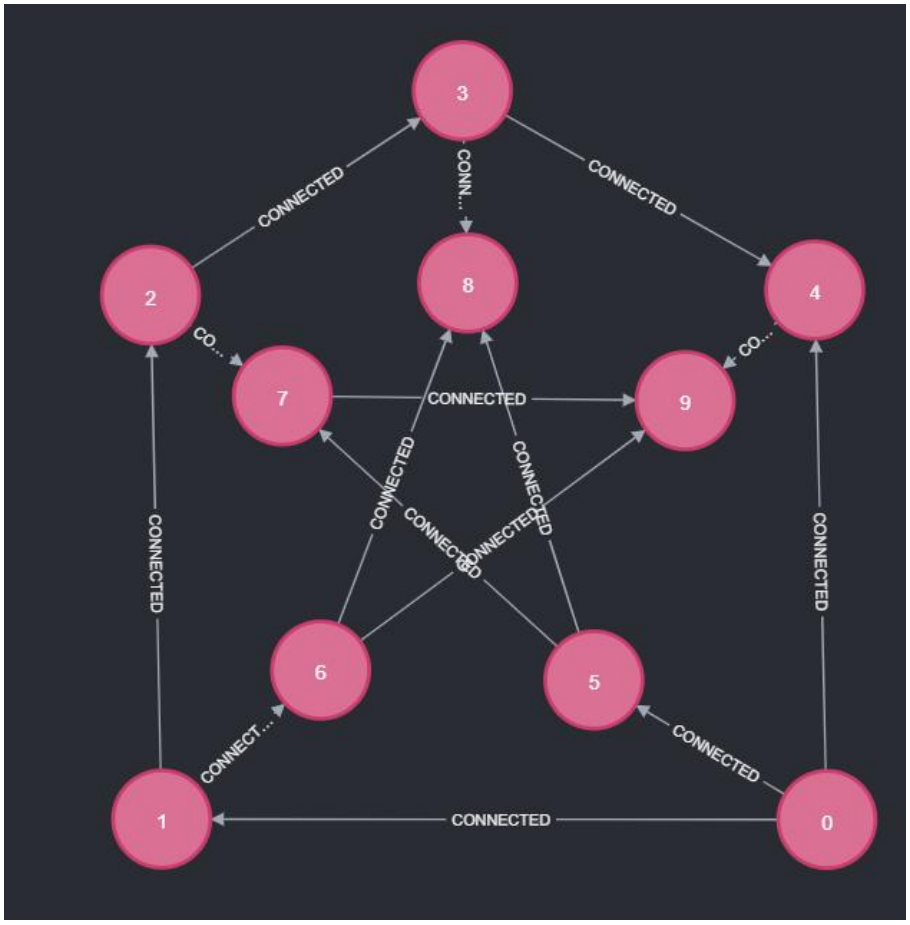
return path



It happens the same way as the previous problem the count includes all possible combinations of the triangles.

4)

```
LOAD CSV WITH HEADERS FROM 'file:/Petersen_Graph.csv' AS row
MERGE(n1:Petersens{id: toInteger(row.source)})
MERGE(n2:Petersens{id: toInteger(row.target)})
WITH row
MATCH (n1:Petersens{id: toInteger(row.source)})
MATCH (n2:Petersens{id: toInteger(row.target)})
MERGE(n1)-[:CONNECTED]->(n2)
```


3
CONNECTED
CONN...
CONNECTED
2
8
4
CO...
CO...
7
CONNECTED
9
CONNECTED
CONNECTED
CONNECTED
CONNECTED
CONNECTED
CONNECTED
CONNECTED
6
5
CONNECT...
CONNECTED
1
CONNECTED
0