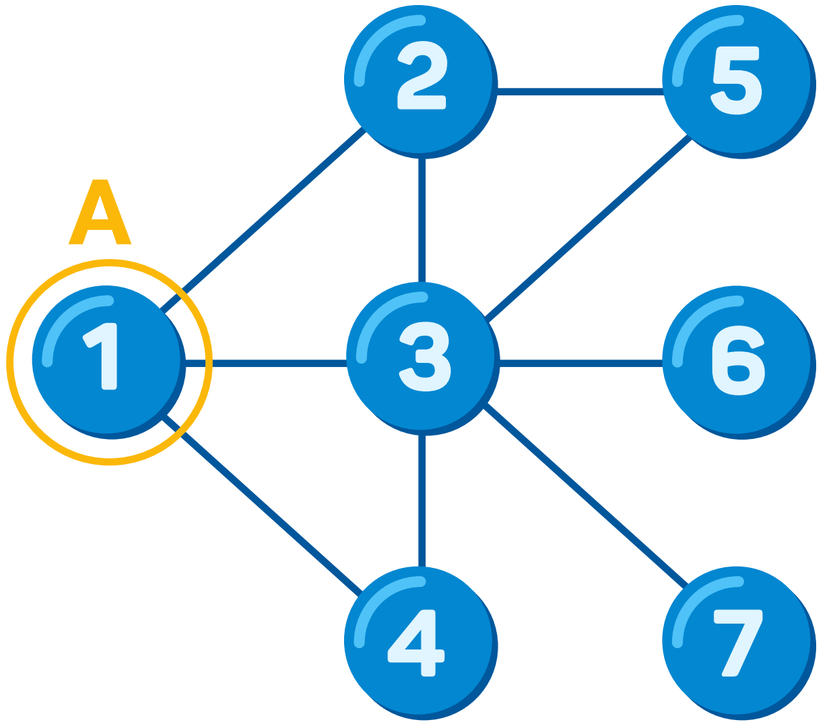
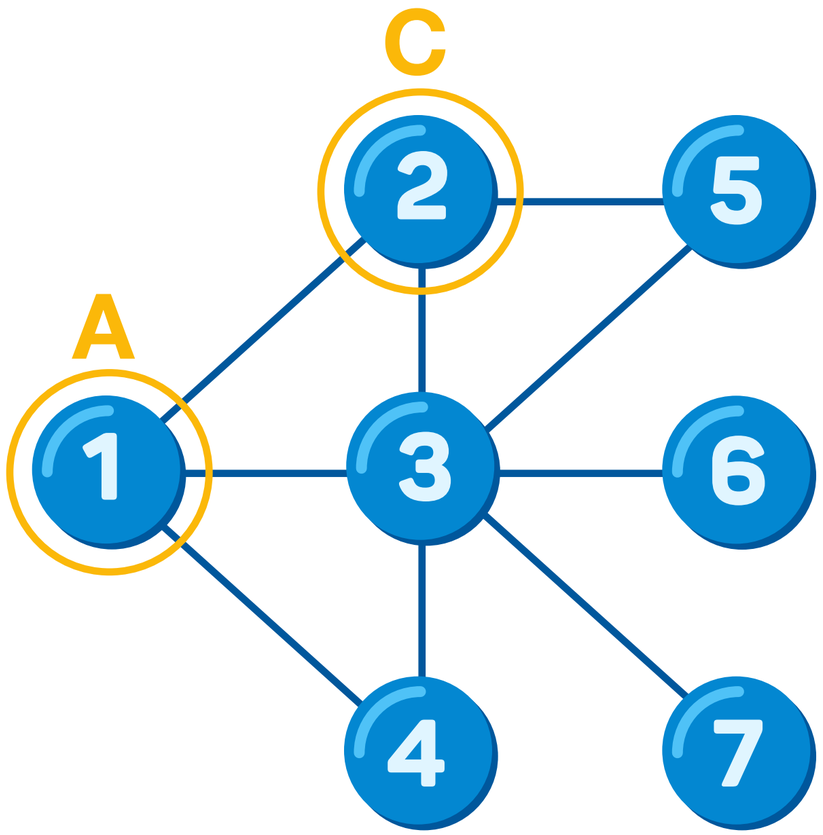
Motif finding:
counting mutual friends

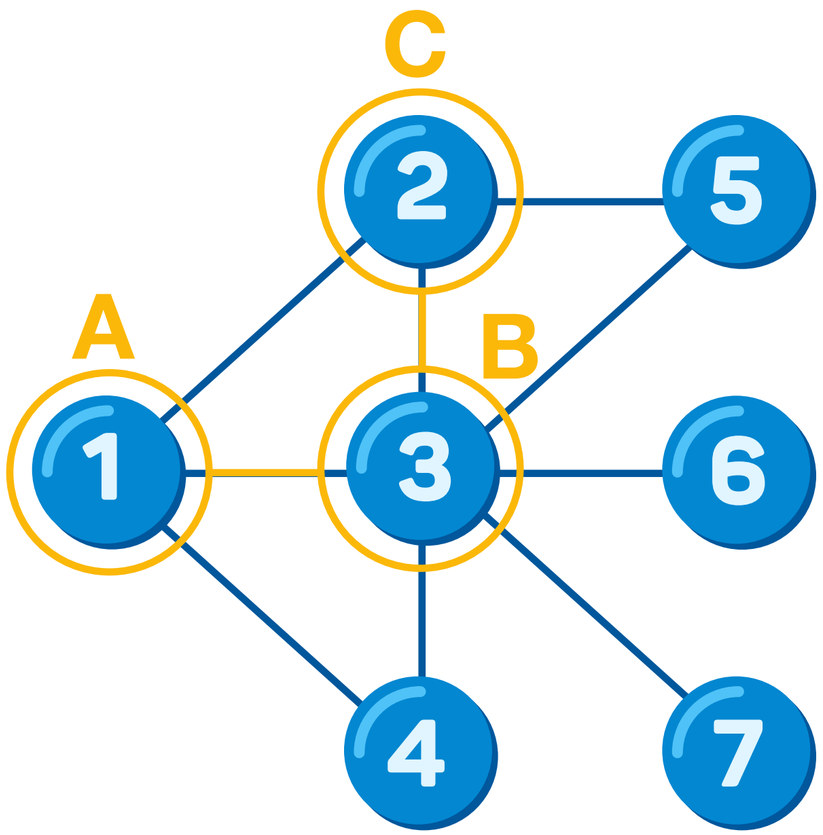
Mini social graph



Mini social graph



Mini social graph



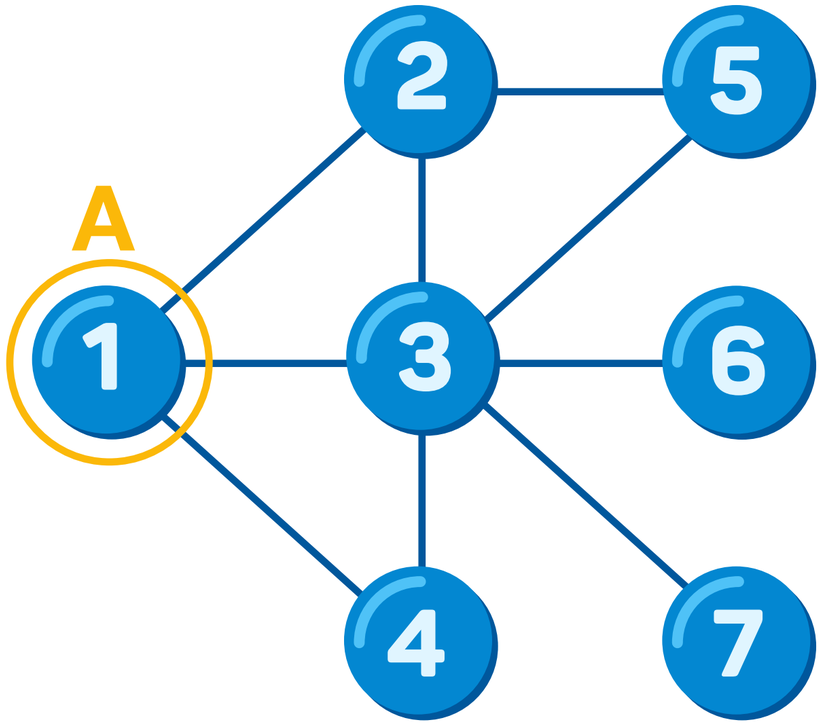
```
vertices = sparkSession.createDataFrame([
  ("1", "Alex", 28, "M" ),
  ("2", "Emeli", 28, "F" ),
  ("3", "Natasha", 27, "F" ),
  ("4", "Pavel", 30, "M" ),
  ("5", "Oleg", 35, "M" ),
  ("6", "Ivan", 30, "M" ),
  ("7", "Ilya", 29, "M" )], ["id", "name", "age", "gender"])

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"])
```

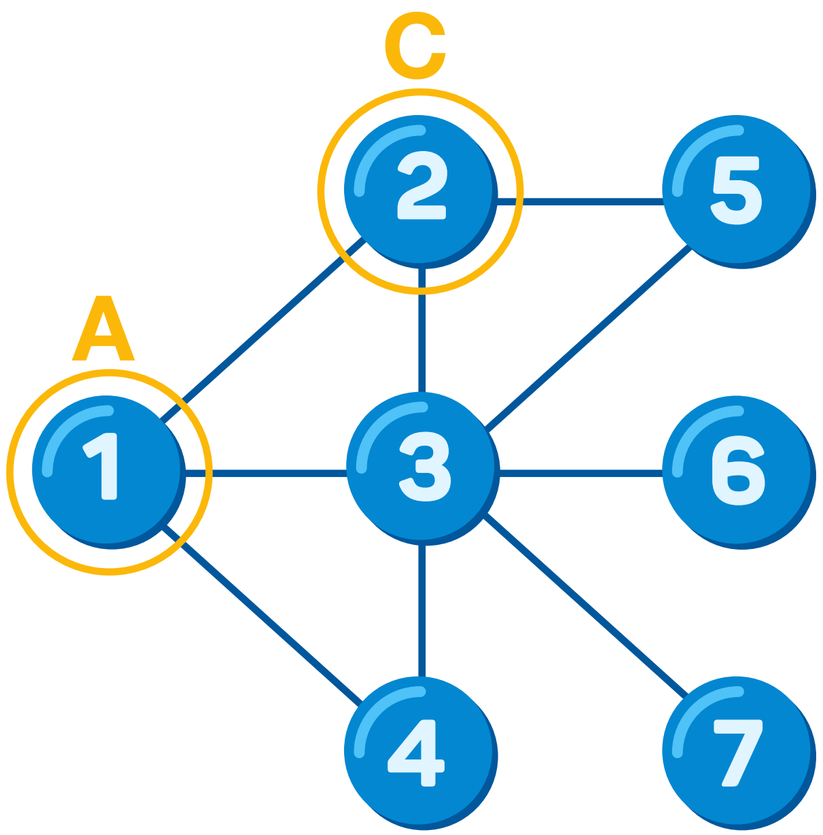
```
from graphframes import *

g = GraphFrame(vertices, edges)
```

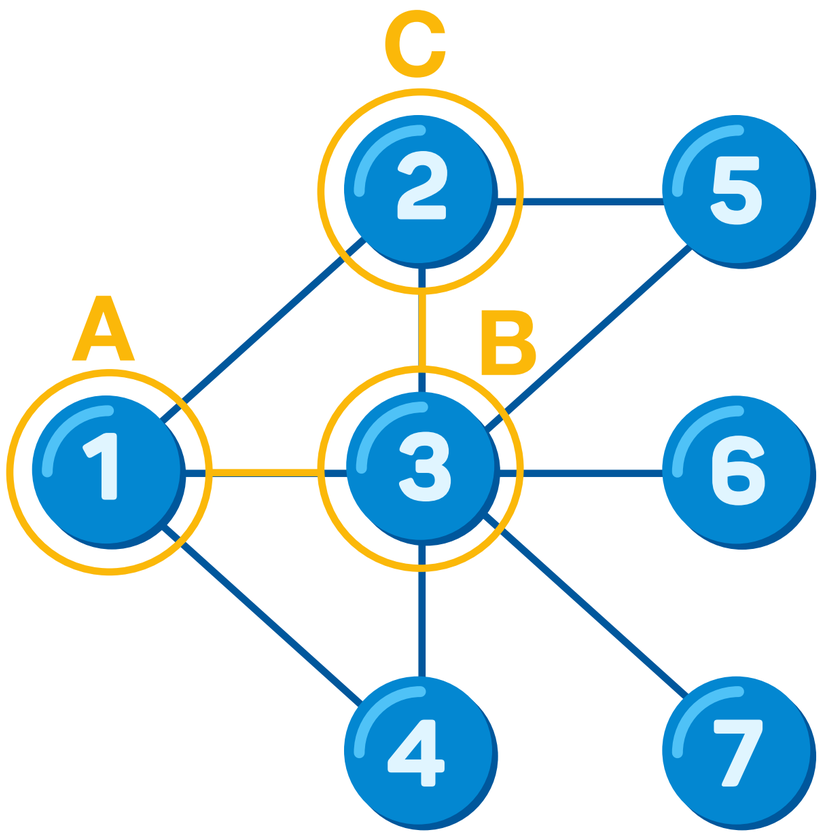
Mini social graph



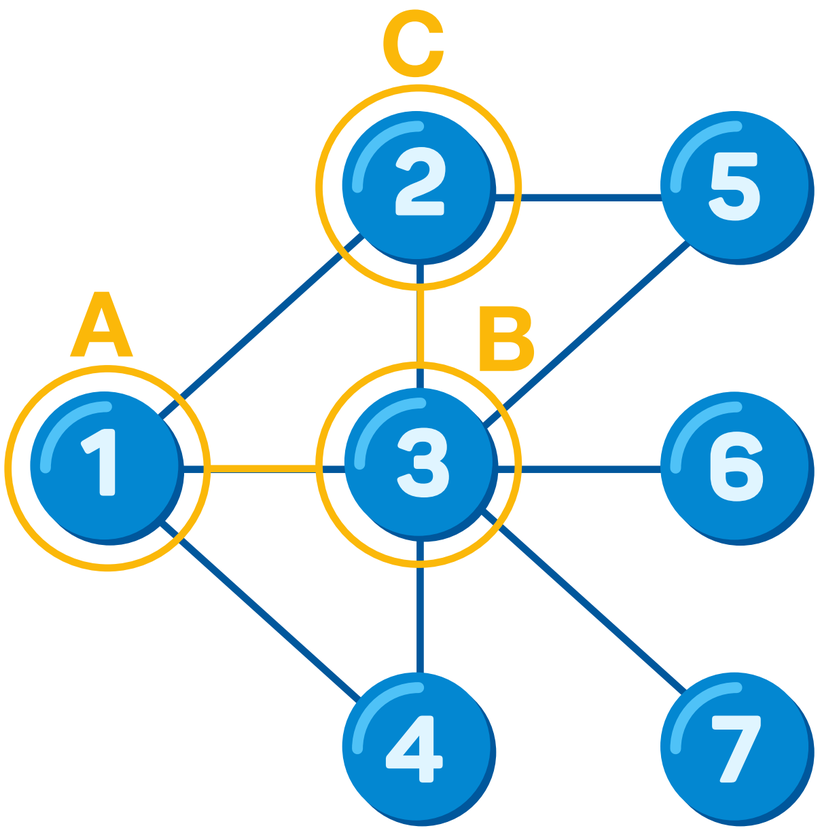
Mini social graph



Mini social graph



Mini social graph



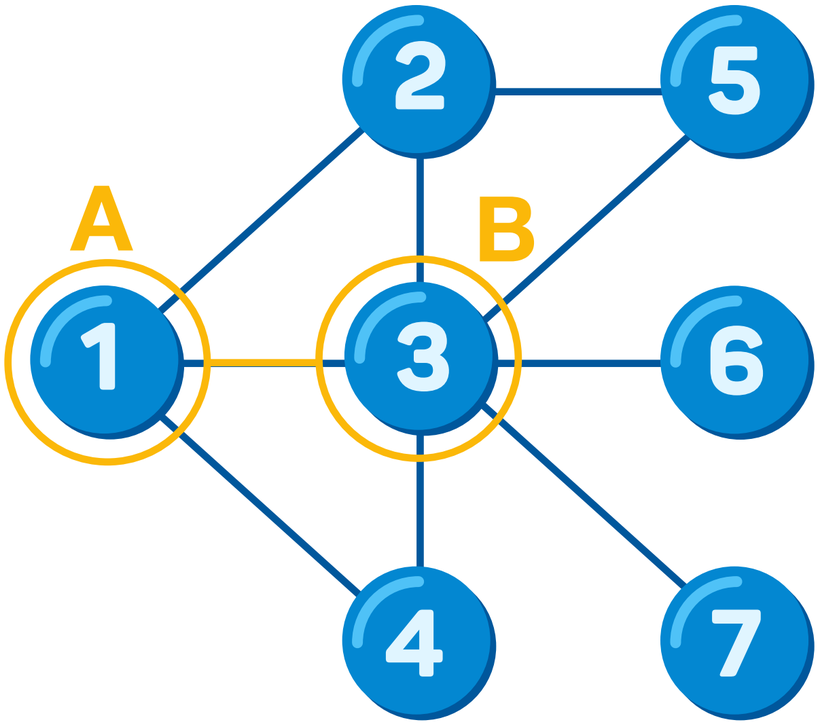
Pattern

(A)

(B)

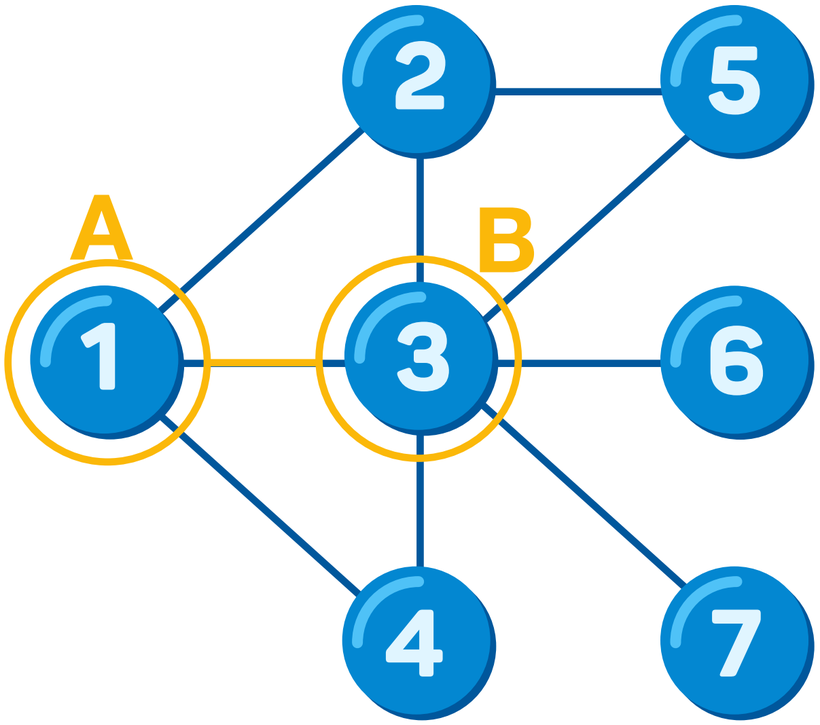
(C)

Mini social graph



Pattern

Mini social graph



Pattern

(A)-[e]->(B)

Pattern (A)-[e]->(B)

Result

A				e			B			
id	name	age	gender	src	dst	type	id	name	age	gender
1	Alex	28	M	1	2	friend	2	Emeli	28	F
1	Alex	28	M	1	3	friend	3	Natasha	27	F
3	Natasha	27	F	3	6	friend	6	Ivan	30	M
7	Ilya	29	M	7	3	friend	3	Natasha	27	F
5	Oleg	35	M	5	2	friend	2	Emeli	28	F
2	Emeli	28	F	2	5	friend	5	Oleg	35	M
6	Ivan	30	M	6	3	friend	3	Natasha	27	F

Pattern (A)-[~~e~~]->(B) → (A)-[]->(B)

Result

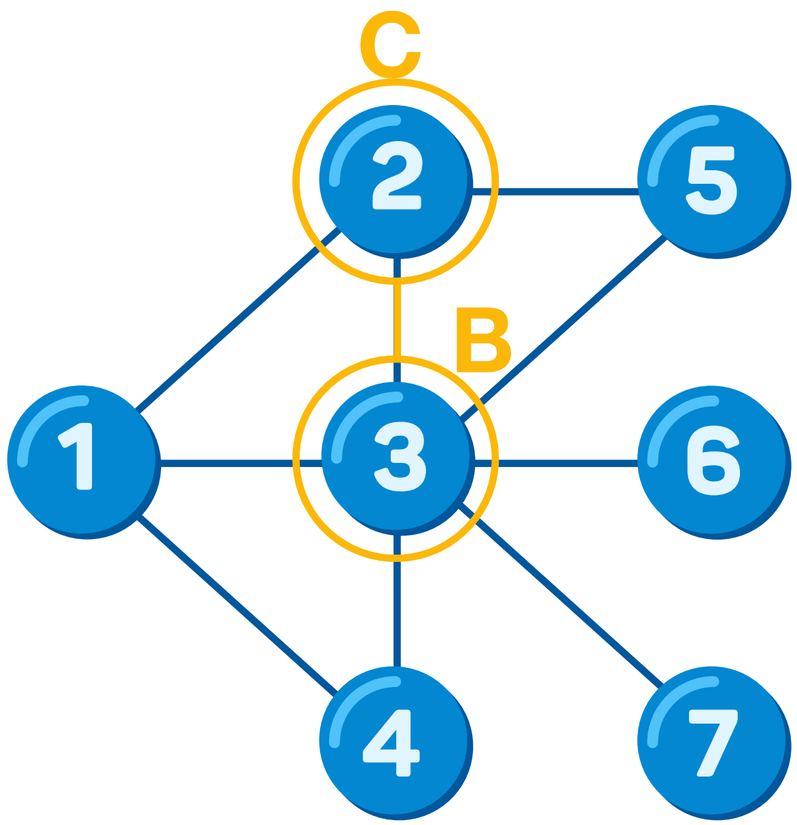
A				e			B			
id	name	age	gender	src	dst	type	id	name	age	gender
1	Alex	28	M	1	2	friend	2	Emeli	28	F
1	Alex	28	M	1	3	friend	3	Natasha	27	F
3	Natasha	27	F	3	6	friend	6	Ivan	30	M
7	Ilya	29	M	7	3	friend	3	Natasha	27	F
5	Oleg	35	M	5	2	friend	2	Emeli	28	F
2	Emeli	28	F	2	5	friend	5	Oleg	35	M
6	Ivan	30	M	6	3	friend	3	Natasha	27	F

Pattern (A)-[e]->(B) → (A)-[]->(B)

Result

A				e			B			
id	name	age	gender	src	dst	type	id	name	age	gender
1	Alex	28	M	1	2	friend	2	Emeli	28	F
1	Alex	28	M	1	3	friend	3	Natasha	27	F
3	Natasha	27	F	3	6	friend	6	Ivan	30	M
7	Ilya	29	M	7	3	friend	3	Natasha	27	F
5	Oleg	35	M	5	2	friend	2	Emeli	28	F
2	Emeli	28	F	2	5	friend	5	Oleg	35	M
6	Ivan	30	M	6	3	friend	3	Natasha	27	F

Mini social graph



Pattern

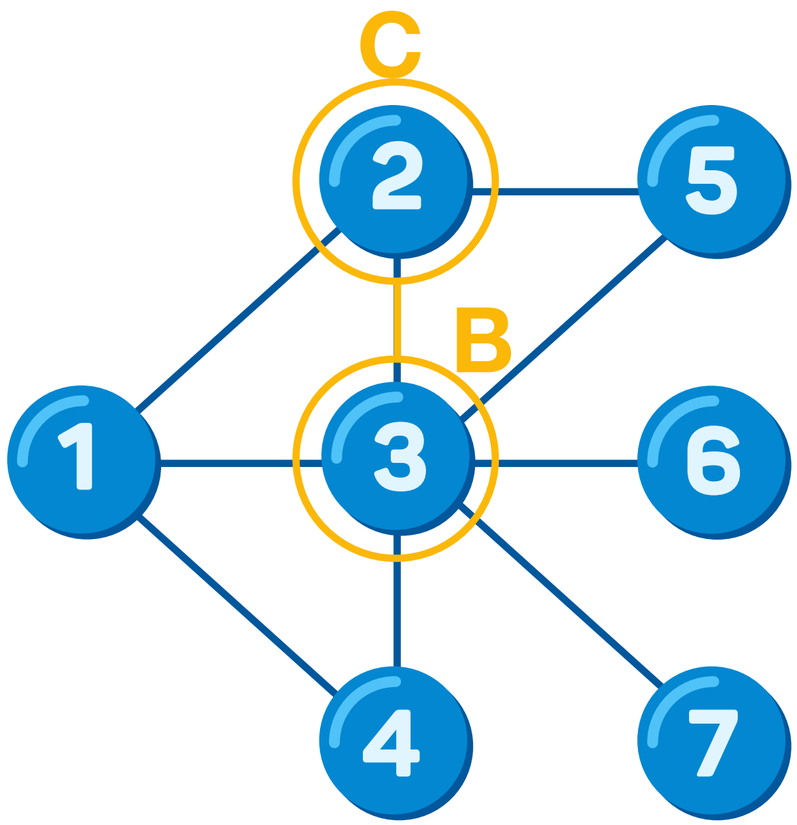
(B)-[e2]->(C)

Pattern (B)-[~~e2~~]->(C) → (B)-[e2]->(C)

Result

A				e			B			
id	name	age	gender	src	dst	type	id	name	age	gender
2	Emeli	28	F	2	1	friend	1	Alex	28	M
3	Natasha	27	F	3	1	friend	1	Alex	28	M
6	Ivan	30	M	6	3	friend	3	Natasha	27	F
3	Natasha	27	F	3	7	friend	7	Ilya	29	M
2	Emeli	28	F	2	5	friend	5	Oleg	35	M
5	Oleg	35	M	5	2	friend	2	Emeli	28	F
3	Natasha	27	F	3	6	friend	6	Ivan	30	M

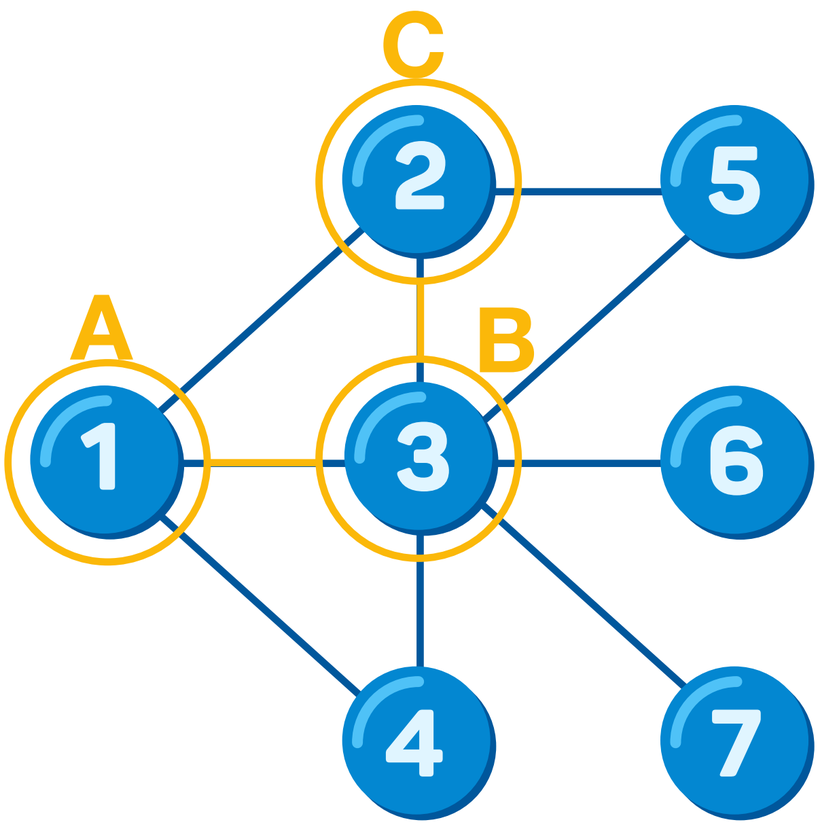
Mini social graph



Pattern

(B)-[]->(C)

Mini social graph



Pattern

(A)-[]->(B);
(B)-[]->(C)

```
motifs = g.find("(A)-[]->(B); (B)-[]->(C)")
motifs.show()
```

```
+----------------+----------------+--------------+
|               A|               B|             C|
+----------------+----------------+--------------+
|   [1,Alex,28,M]|  [2,Emeli,28,F]| [1,Alex,28,M]|
|[3,Natasha,27,F]|  [2,Emeli,28,F]| [1,Alex,28,M]|
|   [5,Oleg,35,M]|  [2,Emeli,28,F]| [1,Alex,28,M]|
|   [1,Alex,28,M]|[3,Natasha,27,F]| [1,Alex,28,M]|
|  [2,Emeli,28,F]|[3,Natasha,27,F]| [1,Alex,28,M]|
|  [4,Pavel,30,M]|[3,Natasha,27,F]| [1,Alex,28,M]|
|   [5,Oleg,35,M]|[3,Natasha,27,F]| [1,Alex,28,M]|
|   [6,Ivan,30,M]|[3,Natasha,27,F]| [1,Alex,28,M]|
|   [7,Ilya,29,M]|[3,Natasha,27,F]| [1,Alex,28,M]|
|   [1,Alex,28,M]|  [4,Pavel,30,M]| [1,Alex,28,M]|
|[3,Natasha,27,F]|  [4,Pavel,30,M]| [1,Alex,28,M]|
|  [2,Emeli,28,F]|   [1,Alex,28,M]|[2,Emeli,28,F]|
|[3,Natasha,27,F]|   [1,Alex,28,M]|[2,Emeli,28,F]|
|  [4,Pavel,30,M]|   [1,Alex,28,M]|[2,Emeli,28,F]|
|   [1,Alex,28,M]|[3,Natasha,27,F]|[2,Emeli,28,F]|
|  [2,Emeli,28,F]|[3,Natasha,27,F]|[2,Emeli,28,F]|
|  [4,Pavel,30,M]|[3,Natasha,27,F]|[2,Emeli,28,F]|
|   [5,Oleg,35,M]|[3,Natasha,27,F]|[2,Emeli,28,F]|
|   [6,Ivan,30,M]|[3,Natasha,27,F]|[2,Emeli,28,F]|
|   [7,Ilya,29,M]|[3,Natasha,27,F]|[2,Emeli,28,F]|
+----------------+----------------+--------------+
only showing top 20 rows
```

```
motifs = g.find("(A)-[]->(B); (B)-[]->(C)")
motifs.show()
+----------------+----------------+---------------+
|               A|               B|              C|
+----------------+----------------+---------------+
|   [1,Alex,28,M]|  [2,Emeli,28,F]| [1,Alex,28,M]|
|[3,Natasha,27,F]|  [2,Emeli,28,F]| [1,Alex,28,M]|
|   [5,Oleg,35,M]|  [2,Emeli,28,F]| [1,Alex,28,M]|
|   [1,Alex,28,M]|[3,Natasha,27,F]| [1,Alex,28,M]|
|  [2,Emeli,28,F]|[3,Natasha,27,F]| [1,Alex,28,M]|
|  [4,Pavel,30,M]|[3,Natasha,27,F]| [1,Alex,28,M]|
|   [5,Oleg,35,M]|[3,Natasha,27,F]| [1,Alex,28,M]|
|   [6,Ivan,30,M]|[3,Natasha,27,F]| [1,Alex,28,M]|
|   [7,Ilya,29,M]|[3,Natasha,27,F]| [1,Alex,28,M]|
|   [1,Alex,28,M]|  [4,Pavel,30,M]| [1,Alex,28,M]|
|[3,Natasha,27,F]|  [4,Pavel,30,M]| [1,Alex,28,M]|
|  [2,Emeli,28,F]|   [1,Alex,28,M]|[2,Emeli,28,F]|
|[3,Natasha,27,F]|   [1,Alex,28,M]|[2,Emeli,28,F]|
|  [4,Pavel,30,M]|   [1,Alex,28,M]|[2,Emeli,28,F]|
|   [1,Alex,28,M]|[3,Natasha,27,F]|[2,Emeli,28,F]|
|  [2,Emeli,28,F]|[3,Natasha,27,F]|[2,Emeli,28,F]|
|  [4,Pavel,30,M]|[3,Natasha,27,F]|[2,Emeli,28,F]|
|   [5,Oleg,35,M]|[3,Natasha,27,F]|[2,Emeli,28,F]|
|   [6,Ivan,30,M]|[3,Natasha,27,F]|[2,Emeli,28,F]|
|   [7,Ilya,29,M]|[3,Natasha,27,F]|[2,Emeli,28,F]|
+----------------+----------------+---------------+
only showing top 20 rows
```

```
motifs = g.find("(A)-[]->(B); (B)-[]->(C)").filter("A.id != C.id")
motifs.show()
```

```
+----------------+----------------+--------------+
|               A|               B|             C|
+----------------+----------------+--------------+
|   [1,Alex,28,M]|  [2,Emeli,28,F]| [1,Alex,28,M]|
|[3,Natasha,27,F]|  [2,Emeli,28,F]| [1,Alex,28,M]|
|   [5,Oleg,35,M]|  [2,Emeli,28,F]| [1,Alex,28,M]|
|   [1,Alex,28,M]|[3,Natasha,27,F]| [1,Alex,28,M]|
|  [2,Emeli,28,F]|[3,Natasha,27,F]| [1,Alex,28,M]|
|  [4,Pavel,30,M]|[3,Natasha,27,F]| [1,Alex,28,M]|
|   [5,Oleg,35,M]|[3,Natasha,27,F]| [1,Alex,28,M]|
|   [6,Ivan,30,M]|[3,Natasha,27,F]| [1,Alex,28,M]|
|   [7,Ilya,29,M]|[3,Natasha,27,F]| [1,Alex,28,M]|
|   [1,Alex,28,M]|  [4,Pavel,30,M]| [1,Alex,28,M]|
|[3,Natasha,27,F]|  [4,Pavel,30,M]| [1,Alex,28,M]|
|  [2,Emeli,28,F]|   [1,Alex,28,M]|[2,Emeli,28,F]|
|[3,Natasha,27,F]|   [1,Alex,28,M]|[2,Emeli,28,F]|
|  [4,Pavel,30,M]|   [1,Alex,28,M]|[2,Emeli,28,F]|
|   [1,Alex,28,M]|[3,Natasha,27,F]|[2,Emeli,28,F]|
|  [2,Emeli,28,F]|[3,Natasha,27,F]|[2,Emeli,28,F]|
|  [4,Pavel,30,M]|[3,Natasha,27,F]|[2,Emeli,28,F]|
|   [5,Oleg,35,M]|[3,Natasha,27,F]|[2,Emeli,28,F]|
|   [6,Ivan,30,M]|[3,Natasha,27,F]|[2,Emeli,28,F]|
|   [7,Ilya,29,M]|[3,Natasha,27,F]|[2,Emeli,28,F]|
+----------------+----------------+--------------+
only showing top 20 rows
```

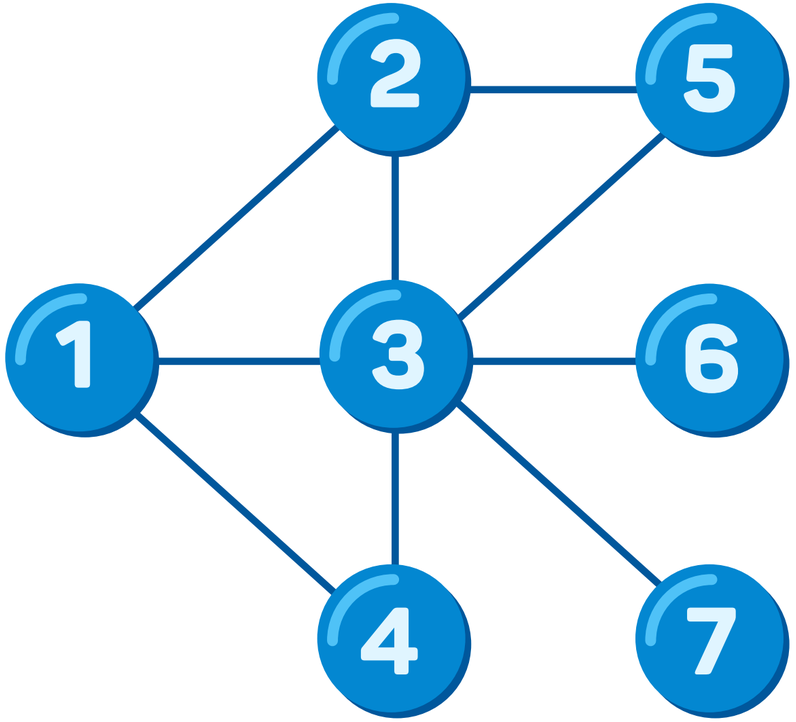
```
AC = motifs.selectExpr("A.id as A", "C.id as C")
AC.show()
+---+---+
|  A|  C|
+---+---+
|  3|  1|
|  5|  1|
|  2|  1|
|  4|  1|
|  5|  1|
|  6|  1|
|  7|  1|
|  3|  1|
|  3|  2|
|  4|  2|
|  1|  2|
|  4|  2|
|  5|  2|
|  6|  2|
|  7|  2|
|  3|  2|
|  2|  3|
|  4|  3|
|  1|  3|
|  5|  3|
+---+---+
only showing top 20 rows
```

```
AC.groupBy("A", "C").count().filter("A = 1").show()
```

```
+---+---+-----+
|  A|  C|count|
+---+---+-----+
|  1|  2|    1|
|  1|  3|    2|
|  1|  4|    1|
|  1|  5|    2|
|  1|  6|    1|
|  1|  7|    1|
+---+---+-----+
```

```
AC.groupBy("A", "C").count().filter("A = 1").show()
```

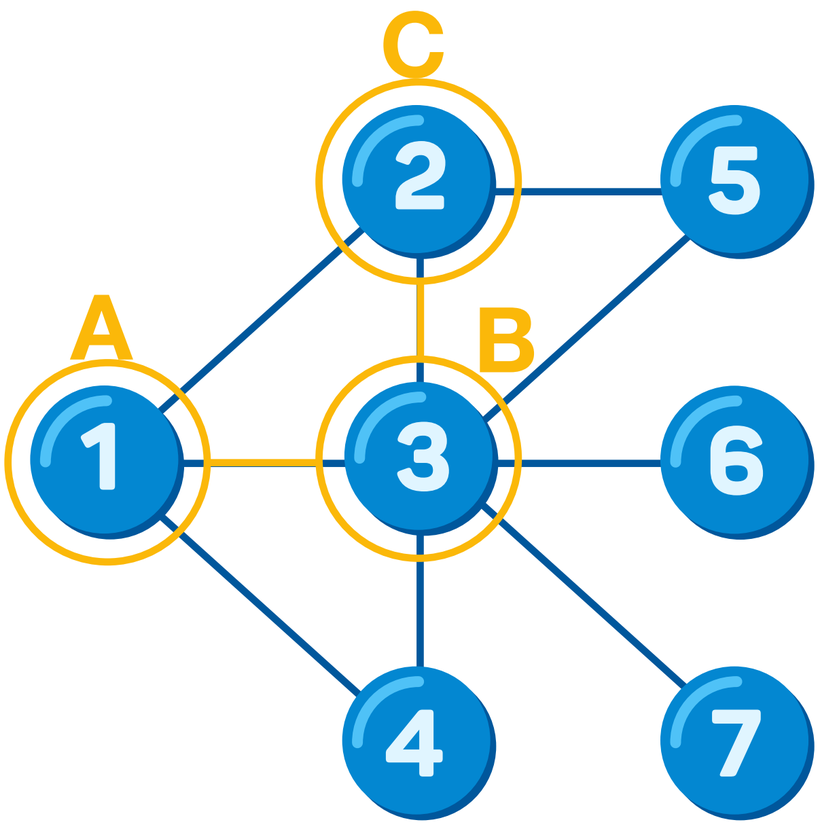
```
+---+---+-----+
|  A|  C|count|
+---+---+-----+
|  1|  2|    1|
|  1|  3|    2|
|  1|  4|    1|
|  1|  5|    2|
|  1|  6|    1|
|  1|  7|    1|
+---+---+-----+
```



```python
AC.groupBy("A", "C").count().show()
```

```
+---+---+-----+
|  A|  C|count|
+---+---+-----+
|  4|  1|    1|
|  4|  2|    2|
|  4|  3|    1|
|  4|  5|    1|
|  4|  6|    1|
|  4|  7|    1|
|  5|  1|    2|
|  5|  2|    1|
|  5|  3|    1|
|  5|  4|    1|
|  5|  6|    1|
|  5|  7|    1|
|  6|  1|    1|
|  6|  2|    1|
|  6|  4|    1|
|  6|  5|    1|
|  6|  7|    1|
|  1|  2|    1|
|  1|  3|    2|
|  1|  4|    1|
+---+---+-----+
only showing top 20 rows
```

Mini social graph



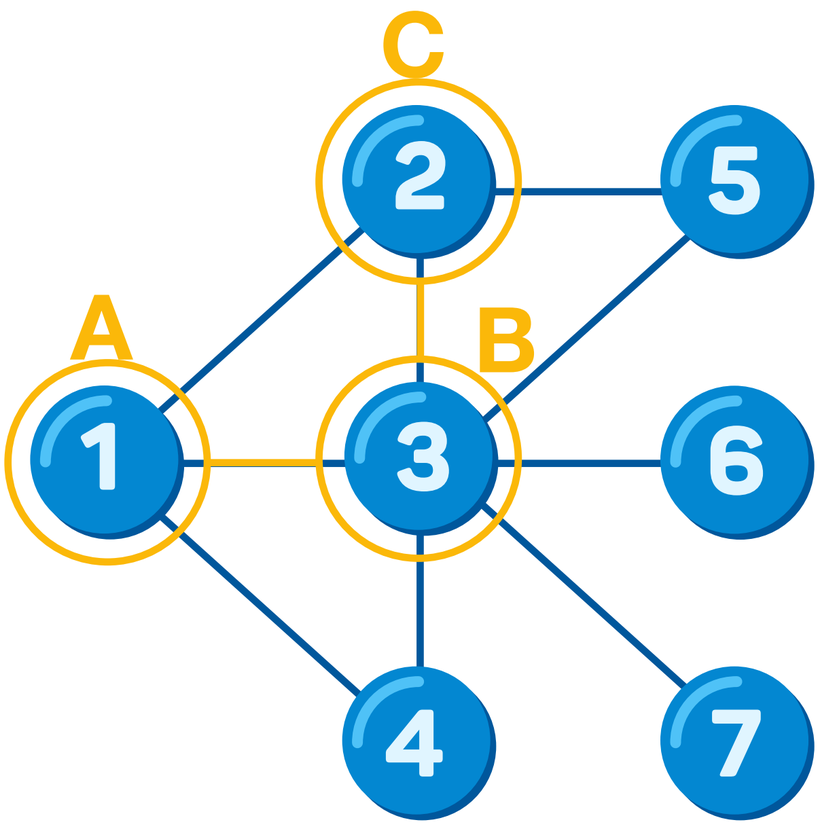
Pattern

(A)-[]->(B);
(B)-[]->(C)

↓ ?

(A)-[]->();
()-[]->(C)

Mini social graph



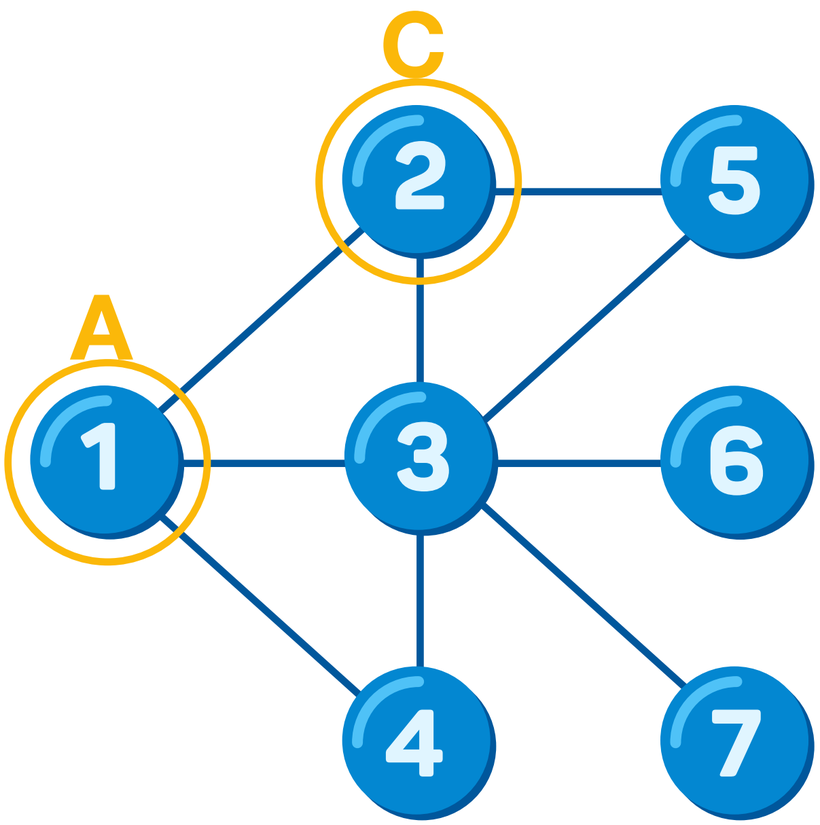
Pattern

(A)-[]->(B);
(B)-[]->(C)

Example

1->3->2
1->3->4

Mini social graph



Pattern

(A)-[]->();
()-[]->(C)

Example

1 -> 6
7 -> 3

Summary

- How to formulate tasks on GraphFrame DSL language.

Summary

- How to formulate tasks on GraphFrame DSL language.
- How to analyse if you should use anonymous or named edges and vertices in our patterns.

Summary

- How to formulate tasks on GraphFrame DSL language.
- How to analyse if you should use anonymous or named edges and vertices in our patterns.
- How to count mutual friends for each pair of users using motif finding algorithm of GraphFrames.