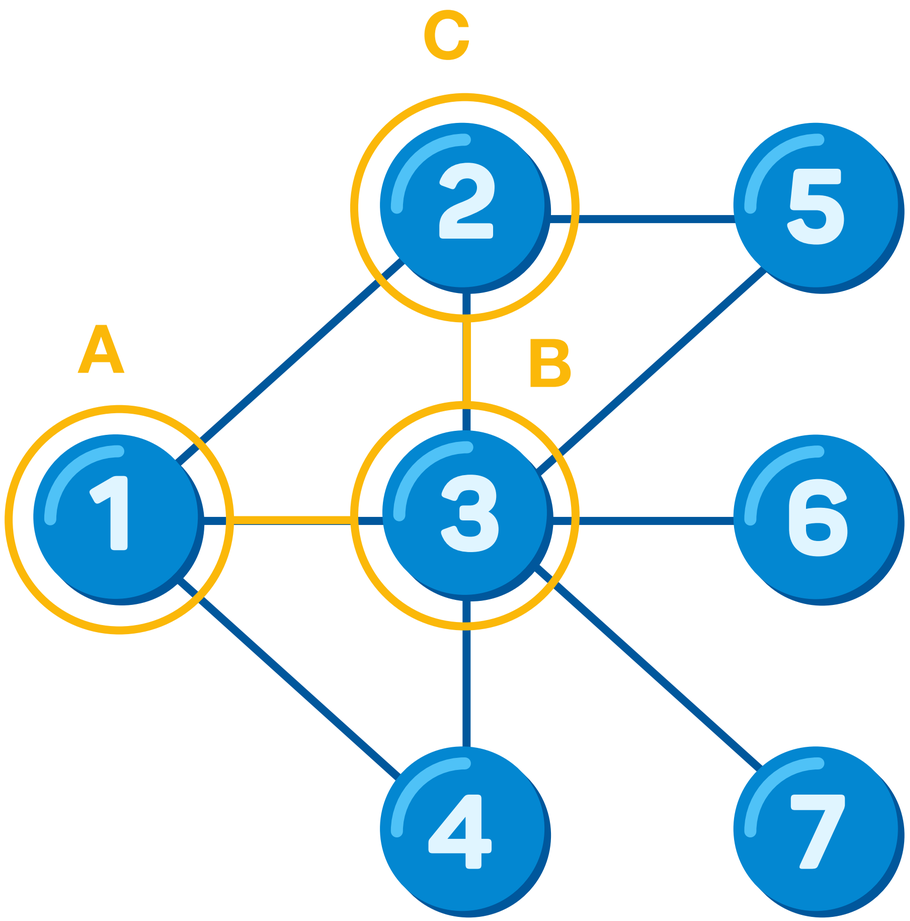
# Motif finding
# counting shuffles

# Mini social graph




## Pattern

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

# Creating a collection of patterns:

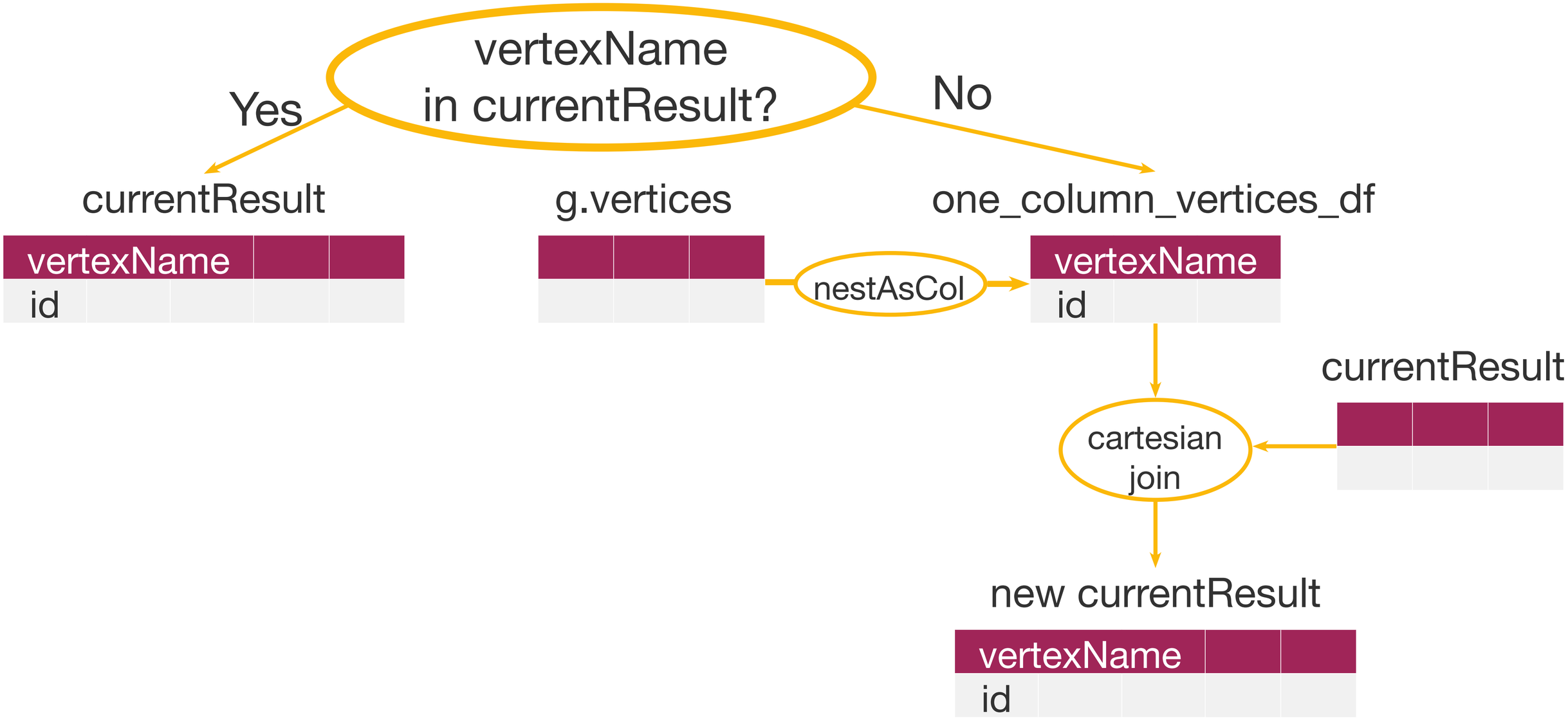
"(A)-[]->(B); (B)-[]->(C)"

[NamedVertex("A"),
AnonymousEdge(NamedVertex("A"), NamedVertex("B")),
NamedVertex("B"),
NamedVertex("B"),
AnonymousEdge(NamedVertex("B"), NamedVertex("C")),
NamedVertex("C")]

# Iterate over a collection of patterns:

```
[NamedVertex(“A”),
AnonymousEdge(NamedVertex(“A”), NamedVertex(“B”)),
NamedVertex(“B”),
NamedVertex(“B”),
AnonymousEdge(NamedVertex(“B”), NamedVertex(“C”)),
NamedVertex(“C”)]
```

# NamedVertex(VertexName)



# Iterate over a collection of patterns:

```
[NamedVertex("A"),
AnonymousEdge(NamedVertex("A"), NamedVertex("B")),
NamedVertex("B"),
NamedVertex("B"),
AnonymousEdge(NamedVertex("B"), NamedVertex("C")),
NamedVertex("C")]
```

```
JOINS: 0
```

# Iterate over a collection of patterns:

[NamedVertex(“A”),

AnonymousEdge(NamedVertex(“A”), NamedVertex(“B”)),

NamedVertex(“B”),

NamedVertex(“B”),

AnonymousEdge(NamedVertex(“B”), NamedVertex(“C”)),

NamedVertex(“C”)]

JOINS: 0

AnonymousEdge (src: AnonymousVertex|NamedVertex,
dst: AnonymousVertex|NamedVertex)



NamedEdge("__tmp", src: AnonymousVertex|NamedVertex,
dst: AnonymousVertex|NamedVertex)



new currentResult

| | | __tmp |
|---|---|---|
| | | |

**NamedEdge(edgeName,**
**src: NamedVertex(srcVertexName),**
**dst: NamedVertex(dstVertexName))**

**1** currentResult

| srcVertexName | dstVertexName | |
|---|---|---|
| | | |

**2** currentResult

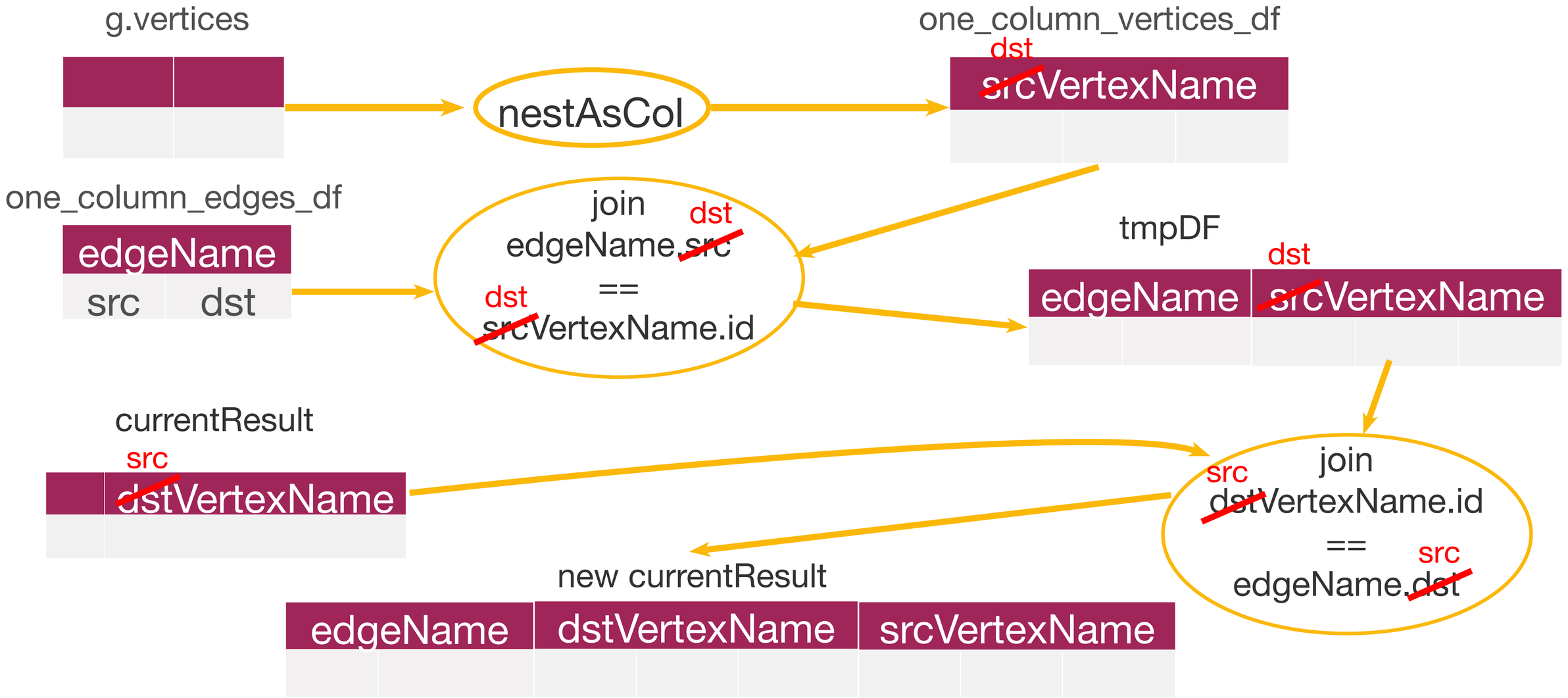
| | | |
|---|---|---|
| | | |

**3** currentResult

| | dstVertexName | |
|---|---|---|
| | | |

**4** currentResult

| srcVertexName | | |
|---|---|---|
| | | |


g.vertices
nestAsCol
one_column_vertices_df
dst
srcVertexName
one_column_edges_df
edgeName
src
dst
join
edgeName.src
==
dst
srcVertexName.id
tmpDF
dst
edgeName
srcVertexName
currentResult
src
dstVertexName
join
src
dstVertexName.id
==
src
edgeName.dst
new currentResult
edgeName
dstVertexName
srcVertexName

# Iterate over a collection of patterns:

[NamedVertex(“A”),
AnonymousEdge(NamedVertex(“A”), NamedVertex(“B”)),
NamedVertex(“B”),
NamedVertex(“B”),
AnonymousEdge(NamedVertex(“B”), NamedVertex(“C”)),
NamedVertex(“C”)]

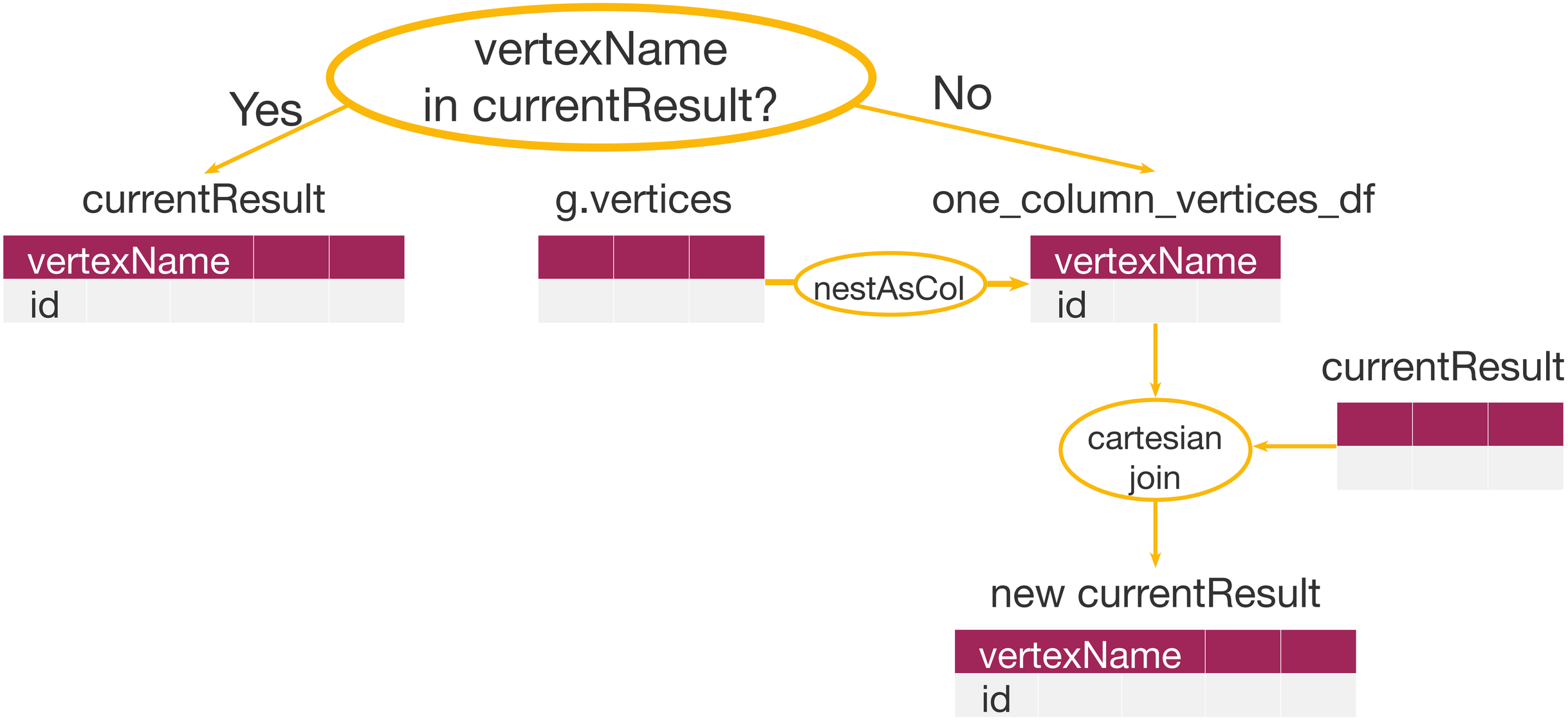
JOINS: 2

# Iterate over a collection of patterns:

[NamedVertex(“A”),
AnonymousEdge(NamedVertex(“A”), NamedVertex(“B”)),
NamedVertex(“B”),
NamedVertex(“B”),
AnonymousEdge(NamedVertex(“B”), NamedVertex(“C”)),
NamedVertex(“C”)]

JOINS: 2

# NamedVertex(VertexName)

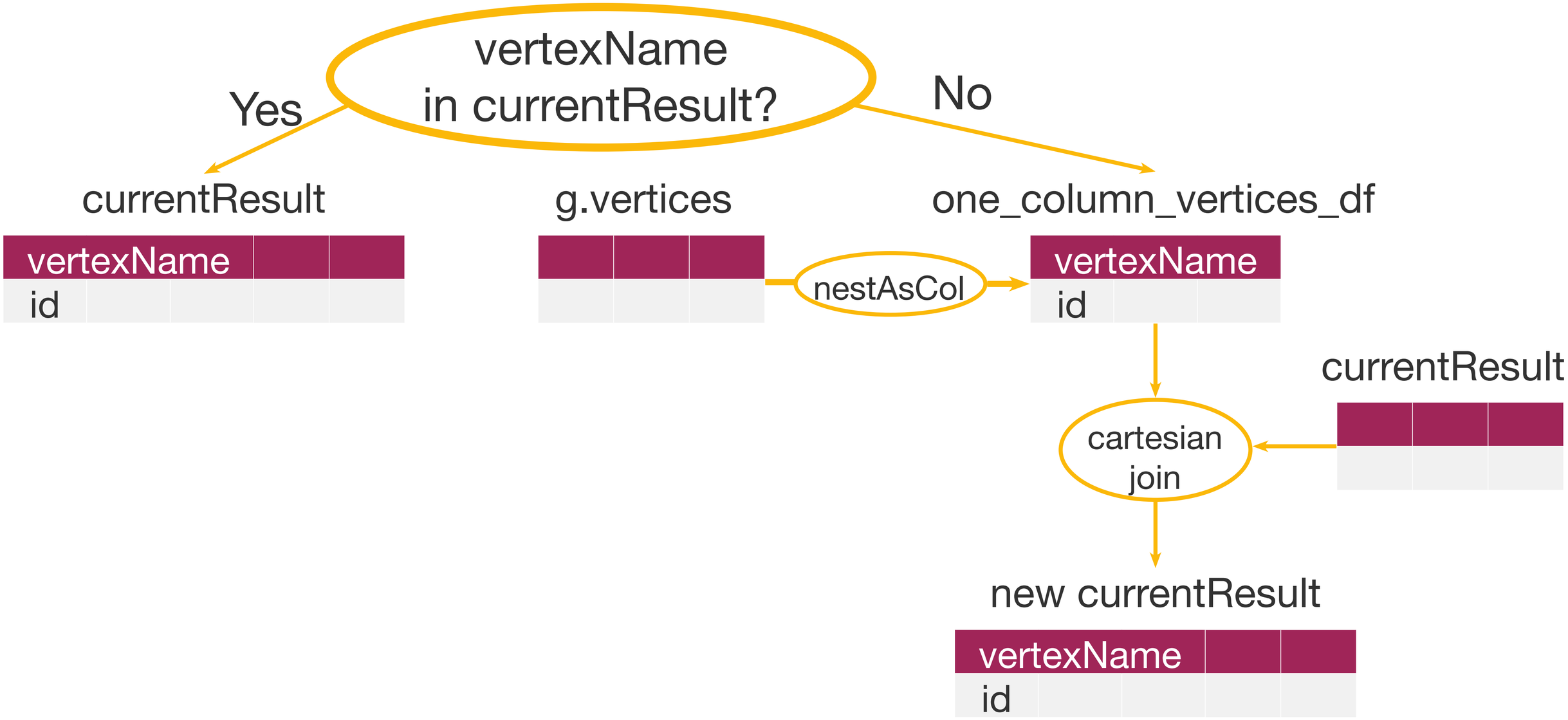


# Iterate over a collection of patterns:

[NamedVertex(“A”),
AnonymousEdge(NamedVertex(“A”), NamedVertex(“B”)),
NamedVertex(“B”),
NamedVertex(“B”),
AnonymousEdge(NamedVertex(“B”), NamedVertex(“C”)),
NamedVertex(“C”)]

JOINS: 2

# NamedVertex(VertexName)



# Iterate over a collection of patterns:

[NamedVertex(“A”),

AnonymousEdge(NamedVertex(“A”), NamedVertex(“B”)),

NamedVertex(“B”),

NamedVertex(“B”),

AnonymousEdge(NamedVertex(“B”), NamedVertex(“C”)),

NamedVertex(“C”)]

JOINS: 2

AnonymousEdge (src: AnonymousVertex|NamedVertex,
dst: AnonymousVertex|NamedVertex)



NamedEdge(«__tmp», src: AnonymousVertex|NamedVertex,
dst: AnonymousVertex|NamedVertex)



new currentResult

| | | |
|---|---|---|
| | | __tmp |
| | | |

**NamedEdge(edgeName,**
**src: NamedVertex(srcVertexName),**
**dst: NamedVertex(dstVertexName))**

**1** currentResult

| srcVertexName | dstVertexName | |
|---|---|---|
| | | |

**2** currentResult

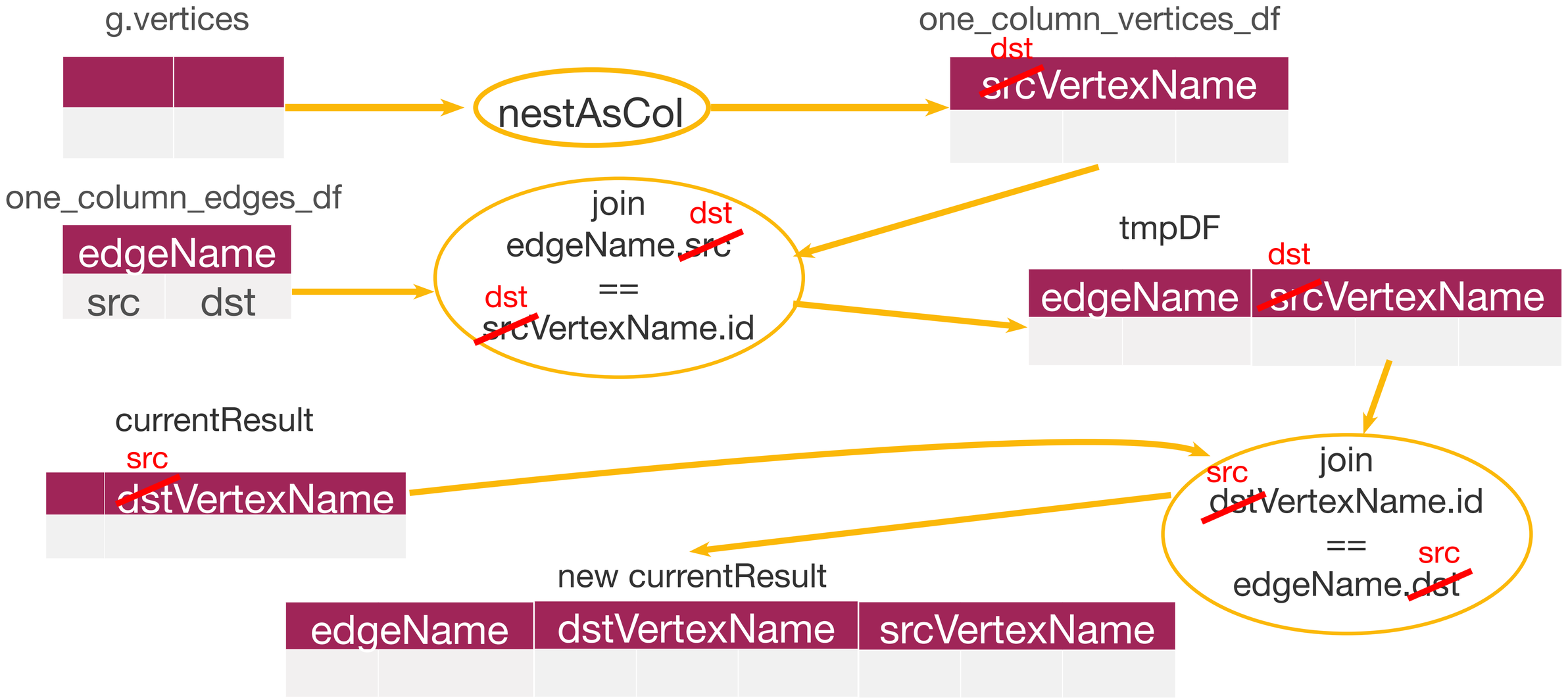
| | | |
|---|---|---|
| | | |

**3** currentResult

| | dstVertexName | |
|---|---|---|
| | | |

**4** currentResult

| srcVertexName | | |
|---|---|---|
| | | |


g.vertices
nestAsCol
one_column_vertices_df
dst
srcVertexName
one_column_edges_df
edgeName
src
dst
join
edgeName.src
==
dst
srcVertexName.id
tmpDF
dst
edgeName
srcVertexName
currentResult
src
dstVertexName
join
src
dstVertexName.id
==
src
edgeName.dst
new currentResult
edgeName
dstVertexName
srcVertexName

# Iterate over a collection of patterns:

[NamedVertex(“A”),

AnonymousEdge(NamedVertex(“A”), NamedVertex(“B”)),

NamedVertex(“B”),

NamedVertex(“B”),

AnonymousEdge(NamedVertex(“B”), NamedVertex(“C”)),

NamedVertex(“C”)]

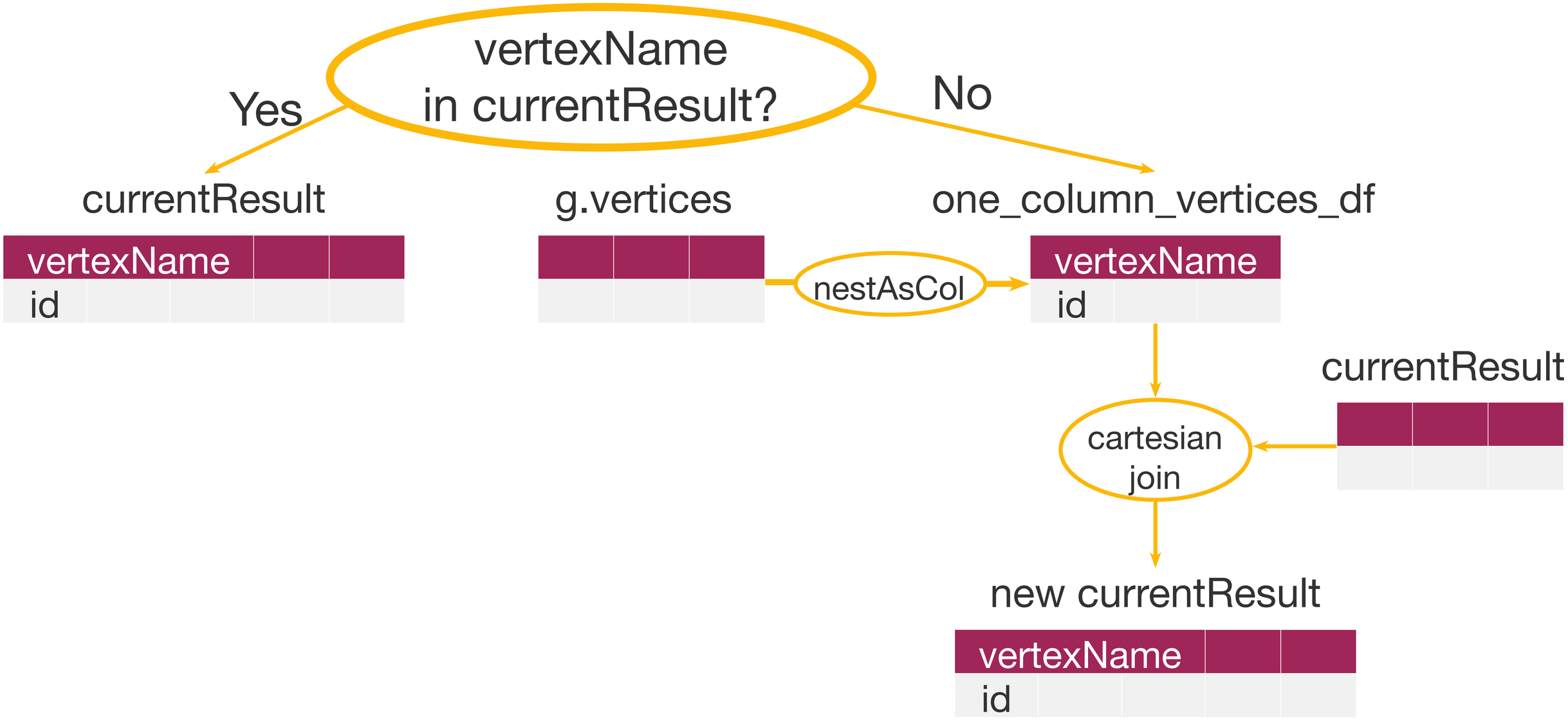
JOINS: 4

# Iterate over a collection of patterns:

[NamedVertex(“A”),
AnonymousEdge(NamedVertex(“A”), NamedVertex(“B”)),
NamedVertex(“B”),
NamedVertex(“B”),
AnonymousEdge(NamedVertex(“B”), NamedVertex(“C”)),
NamedVertex(“C”)]

JOINS: 4

# NamedVertex(VertexName)



# Counting mutual friends

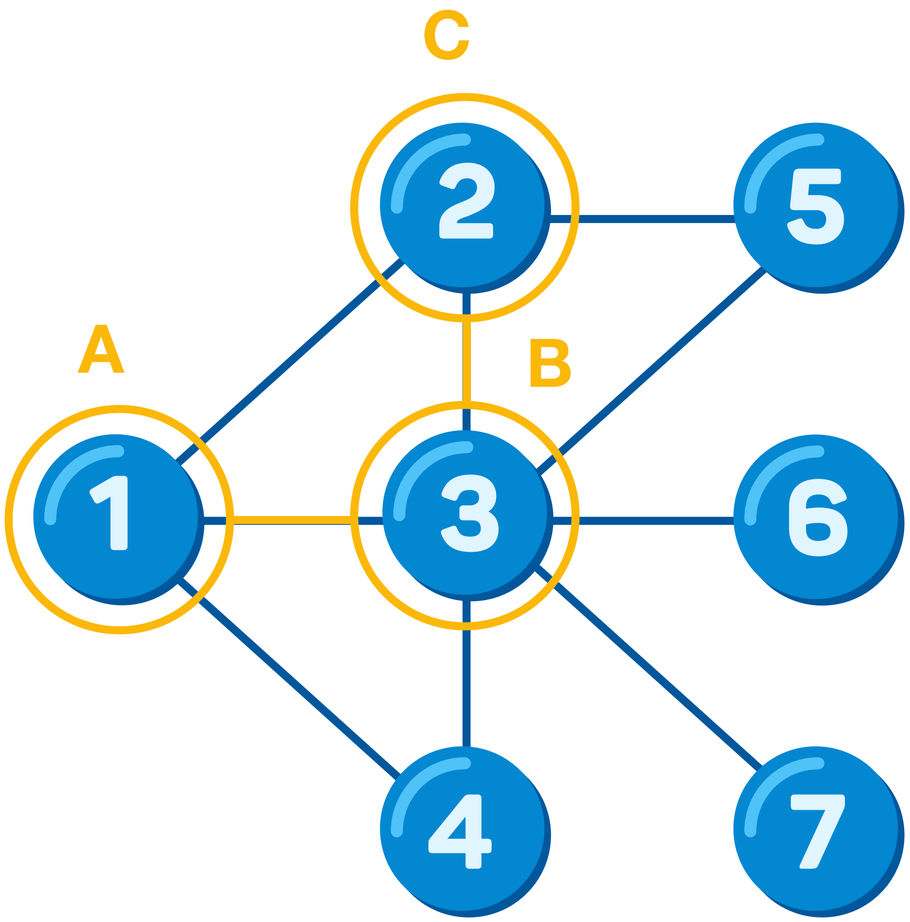
```
abcDF = abDF.join(bcDF, "B").filter("A = 1")
abcDF.show()
+---+---+---+
|  B|  A|  C|
+---+---+---+
|  3|  1|  4|
|  3|  1|  5|
|  3|  1|  6|
|  3|  1|  7|
+---+---+---+
```

JOINS: 1

```
motifs = gf.find("(A)-[]->(B); (B)-[]->(C)")
motifs.show()
+--------------------+--------------------+--------------------+
|                   A|                   B|                   C|
+--------------------+--------------------+--------------------+
|  [1,Alex,28,M,MIPT]| [2,Emeli,28,F,MIPT]| [1,Alex,28,M,MIPT]|
|[3,Natasha,27,F,S...| [2,Emeli,28,F,MIPT]| [1,Alex,28,M,MIPT]|
|  [5,Oleg,35,M,MIPT]| [2,Emeli,28,F,MIPT]| [1,Alex,28,M,MIPT]|
|  [1,Alex,28,M,MIPT]|[3,Natasha,27,F,S...| [1,Alex,28,M,MIPT]|
```

JOINS: 4

# Mini social graph




## Pattern

A

| | |
|---|---|
| id | 1 |
| name | Alex |
| age | 28 |
| gender | M |
| university | MIPT |

B

| | |
|---|---|
| id | 2 |
| name | Emeli |
| age | 28 |
| gender | F |
| university | MIPT |

C

| | |
|---|---|
| id | 3 |
| name | Natasha |
| age | 28 |
| gender | F |
| university | MIPT |

# Counting mutual friends

```
abcDF = abDF.join(bcDF, "B").filter("A = 1")
abcDF.show()
```

JOINS: 1

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

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

JOINS: 4

```
+--------------------+--------------------+--------------------+
|                   A|                   B|                   C|
+--------------------+--------------------+--------------------+
|  [1,Alex,28,M,MIPT]| [2,Emeli,28,F,MIPT]| [1,Alex,28,M,MIPT]|
|[3,Natasha,27,F,S...| [2,Emeli,28,F,MIPT]| [1,Alex,28,M,MIPT]|
|  [5,Oleg,35,M,MIPT]| [2,Emeli,28,F,MIPT]| [1,Alex,28,M,MIPT]|
|  [1,Alex,28,M,MIPT]|[3,Natasha,27,F,S...| [1,Alex,28,M,MIPT]|
```

# Summary

- How to split motif finding algorithm in steps

# Summary

- How to split motif finding algorithm in steps
- Estimate the amount of joins motif finding algorithm will do for particular patterns