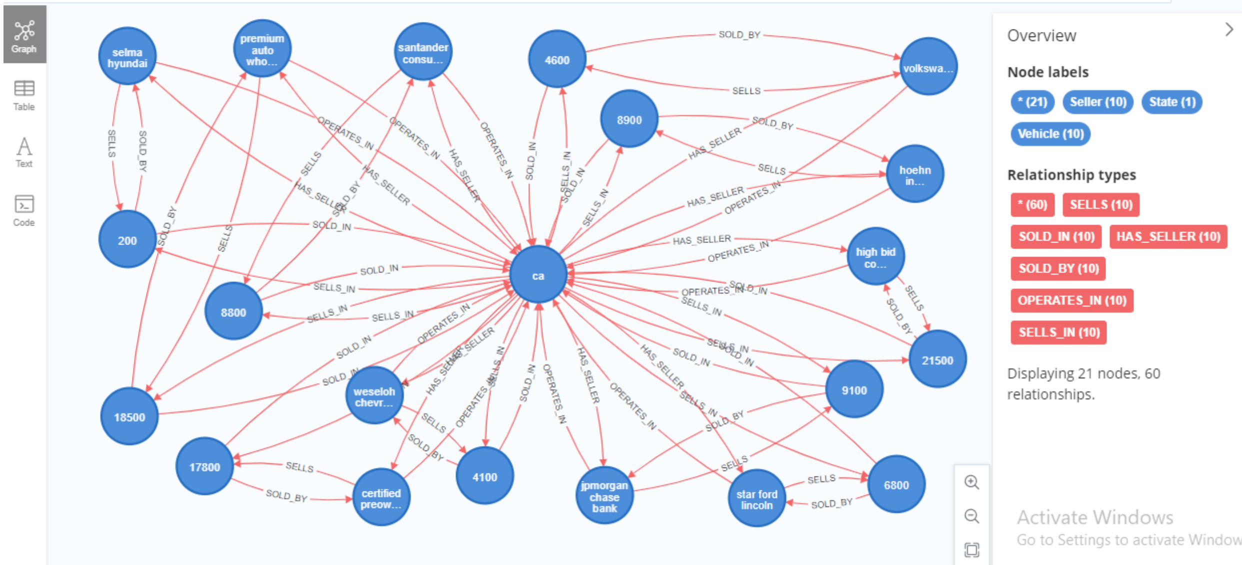
➢ Information Travel Through Network:

finding the paths most taken by vehicles from sellers to states:

```
MATCH path=(s:Seller)-[:SELLS]→(v:Vehicle)-[:SOLD_IN]→(st:State)
RETURN path
LIMIT 10
```



➢ Influential Nodes in Information Spread:

Query to rank sellers by the number of vehicles they have sold:

```
MATCH (s:Seller)-[:SELLS]->(v:Vehicle)
RETURN s.name as SellerName, COUNT(v) as VehiclesSold ORDER BY VehiclesSold DESC
```

	SellerName	VehiclesSold
1	"ford motor credit company llc"	719
2	"santander consumer"	639
3	"nissan infiniti lt"	587
4	"wells fargo dealer services"	528
5	"jpmorgan chase bank n.a."	506
6	"financial services remarketing (lease)"	464
7	"avis corporation"	463
8	"nissan-infiniti lt"	375

Activate Windows
Go to Settings to activate Windows.

Started streaming 4877 records after 27 ms and completed after 170 ms, displaying first 1000 rows.

States which have sold the most vehicles thus carrying the most influence on the spread of information (e.g., market trends, vehicle popularity).

```
MATCH (v:Vehicle)-[:SOLD_IN]->(st:State)
RETURN st.name AS State, count(v) AS VehiclesSold
ORDER BY VehiclesSold DESC
LIMIT 7
```

	State	VehiclesSold
1	"ca"	8485
2	"fl"	3316
3	"tx"	2262
4	"pa"	2237
5	"il"	1854
6	"ga"	1377
7	"va"	1362

Started streaming 7 records after 20 ms and completed after 67 ms.

- **Which nodes interact the most with another certain node and why?**

To find which states have the most vehicles sold from a specific seller:

```
MATCH (s:Seller {name: 'ford motor credit company llc'})-[:SELLS]->(v:Vehicle)-[:SOLD_IN]->(st:State)
RETURN st.name AS State, count(v) AS VehiclesSold
ORDER BY VehiclesSold DESC
LIMIT 10
```

State	VehiclesSold
"mi"	275
"il"	85
"tx"	64
"ca"	56
"mo"	42
"pa"	39
"mn"	39
"fl"	34
"ny"	30
"oh"	25

```
MATCH (s:Seller {name: 'nissan infiniti lt'})-[:SELLS]->(v:Vehicle)-[:SOLD_IN]->(st:State)
RETURN st.name AS State, count(v) AS VehiclesSold
ORDER BY VehiclesSold DESC
LIMIT 10
```

State	VehiclesSold
"ca"	352
"tn"	84
"il"	53
"tx"	26
"nj"	24
"fl"	14
"mo"	14
"ga"	8
"pa"	8
"wa"	3

- **Are there any patterns visible in the graph network?**

To identify patterns, such as which vehicle makes are most commonly sold by each seller:

```
MATCH (s:Seller)-[:SELLS]→(v:Vehicle)
RETURN s.name AS Seller, v.make AS Make, count(*) AS Count
ORDER BY Count DESC
LIMIT 10
```

Seller	Make	Count
"ford motor credit company llc"	"Ford"	688
"nissan infiniti lt"	"Infiniti"	587
"financial services remarketing (lease)"	"BMW"	417
"nissan-infiniti lt"	"Nissan"	375
"ahfc/honda lease trust/hvt inc."	"Honda"	271
"avis budget group"	"Ford"	242
"hyundai motor finance"	"Hyundai"	208
"kia motors america inc"	"Kia"	198
"ford motor credit company llc pd"	"Ford"	189
"toyota financial services"	"Toyota"	175

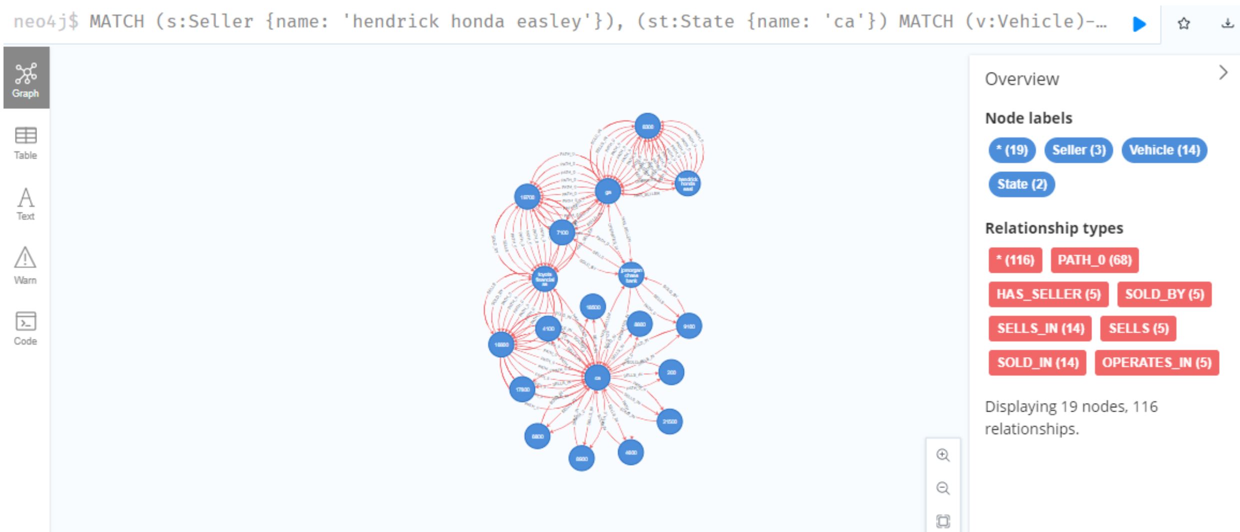
Creating a Graph Projection

```
CALL gds.graph.project(
    'myGraphProjection',
    {
        Vehicle: {},
        Seller: {},
        State: {}
    },
    {
        SOLD_BY: {
            type: 'SOLD_BY',
            orientation: 'REVERSE' // Since the direction is from Vehicle to Seller.
        },
        SELLS: {
            type: 'SELLS',
            orientation: 'NATURAL' // it's the inverse of SOLD_BY.
        },
        OPERATES_IN: {
            type: 'OPERATES_IN',
            orientation: 'NATURAL' // From Seller to State.
        },
        HAS_SELLER: {
            type: 'HAS_SELLER',
            orientation: 'REVERSE' // Inverse of OPERATES_IN, from State to Seller.
        },
        SOLD_IN: {
            type: 'SOLD_IN',
            orientation: 'NATURAL' // From Vehicle to State.
        },
        SELLS_IN: {
            type: 'SELLS_IN',
            orientation: 'REVERSE' // Inverse of SOLD_IN, from State to Vehicle.
        }
    }
)
YIELD graphName, nodeCount, relationshipCount
RETURN graphName, nodeCount, relationshipCount;
```

- **Path Finding in the Graph**

Find the shortest path between two nodes

```
MATCH (s:Seller {name: 'hendrick honda easley'}), (st:State {name: 'ca'})
MATCH (v:Vehicle)-[:SOLD_IN]->(st)
CALL gds.shortestPath.dijkstra.stream('myGraphProjection', {
  sourceNode: s,
  targetNode: v,
  relationshipWeightProperty: null
})
YIELD sourceNode, targetNode, totalCost, nodeIds, costs, path
RETURN
  gds.util.asNode(sourceNode).name AS Start,
  gds.util.asNode(targetNode).vin AS End,
  totalCost,
  [nodeId IN nodeIds | gds.util.asNode(nodeId).name] AS NodesOnPath,
  costs,
  path
LIMIT 10
```



Start	End	totalCost	NodesOnPath	costs
"hendrick honda easley"	"1ftne1ew4dda47458"	7.0	["hendrick honda easley", *null*, "ga", *null*, "toyota financial services", *null*, "ca", *null*]	[0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0]

➢ Node Centrality

Identify nodes with the most influence, using Degree Centrality.

```
CALL gds.degree.stream('myGraphProjection')
YIELD nodeId, score
RETURN gds.util.asNode(nodeId).name AS Node, score AS Centrality
ORDER BY score DESC
LIMIT 10
```

"ca"	8485.0
"fl"	3316.0
"tx"	2262.0
"pa"	2237.0
"il"	1854.0
"ga"	1377.0
"va"	1362.0
"mi"	1312.0
"nc"	1071.0
"az"	1021.0

- **Community Detection**

Detect communities within the graph using the Louvain algorithm:

```
CALL gds.louvain.stream('myGraphProjection')
YIELD nodeId, communityId
RETURN communityId, collect(gds.util.asNode(nodeId).name) AS Members
ORDER BY size(Members) DESC
LIMIT 10
```

	communityId	Members
1	33443	["kia motors america inc", "financial services remarketing (lease)", "enterprise vehicle exchange / tra / rental / tulsa", "audi mission viejo", "d/m auto sales inc", "desert
2	35184	["the hertz corporation/gm", "enterprise fm exchange/tra/lease", "lease plan usa/tra", "coastal credit llc / jacksonville / orange park", "transfleet us llc", "remarketing by g
3	33486	["avis tra", "tdaf remarketing", "hyundai motor finance", "jpmorgan chase bank n.a.", "mercedes-benz finc services premier program", "us bank", "fiserv/usb dealer serv
4	34404	["mercedes-benz financial services", "caprock auto remarketing", "vw credit", "chicago motor car corporation", "donlen corporation", "remarketing services of el paso",
5	33484	["enterprise veh exchange/rental", "hertz corporation/gdp", "pv holding inc/gdp", "navy fcu", "aq motors", "chrysler group llc", "dt credit corporation", "prestige financial s
6	36731	["volvo na rep/world omni", "enterprise fleet management exchange inc.", "primeritus remarketing/oregon comm credit", "bmw alphera/alphera financial services", "str
7	[illegible]	[illegible]

⚠ Record fields have been truncated. Started streaming 10 records after 36 ms and completed after 9942 ms.

Activate Windows
Go to Settings to activate Windows.

➢ Influence Spread (using PageRank)

To identify which sellers or states have the most influence over the vehicle sales network:

```
CALL gds.pageRank.stream('myGraphProjection')
YIELD nodeId, score
RETURN gds.util.asNode(nodeId).name AS nodeName, score AS pageRankScore
ORDER BY score DESC
LIMIT 10;
```

nodeName	pageRankScore
"ca"	1179.0026526424892
"fl"	483.5614020144815
"tx"	327.38269364760606
"pa"	325.09876192824544
"il"	265.43993966829447
"ga"	201.12535284115592
"va"	195.6633615290157
"mi"	180.47490861446676
"nc"	155.25909351997805
"ny"	148.56049214030568

- **Interaction Patterns (using Degree Centrality)**

To quantify the number of sales interactions for each seller or vehicle:

```
CALL gds.degree.stream('myGraphProjection')
YIELD nodeId, score
RETURN gds.util.asNode(nodeId).name AS nodeName, score AS interactionCount
ORDER BY score DESC
LIMIT 10;
```

nodeName	interactionCount
"ford motor credit company llc"	1468.0
"santander consumer"	1306.0
"nissan infiniti lt"	1196.0
"wells fargo dealer services"	1084.0
"jpmorgan chase bank n.a."	1040.0
"avis corporation"	968.0
"financial services remarketing (lease)"	944.0
"nissan-infiniti lt"	772.0
"enterprise veh exchange/rental"	754.0
"ge fleet services for itself/servicer"	674.0

➢ Identifying Strongly Connected Components

To find clusters or groups of nodes that frequently interact with each other:

```
CALL gds.wcc.stream('myGraphProjection')
YIELD nodeId, componentId
RETURN componentId, collect(gds.util.asNode(nodeId).name) AS ComponentMembers
ORDER BY size(ComponentMembers) desc
LIMIT 10
```

	componentId	ComponentMembers
1	0	["kia motors america inc", "financial services remarketing (lease)", "volvo na rep/world omni", "enterprise vehicle exchange / tra / rental / tulsa", "the hertz corporation",
2	9463	["canadian auto remarketing", "coconut grove fleet and lease", "mercedes-benz", "auction direct", "malfara's automotive", "auto leasing limited", "bel auto sales", "era cla
3	9098	["fca canada inc", "chrysler canada inc", "qc"]

⚠ Record fields have been truncated. Started streaming 3 records after 1 ms and completed after 150 ms.

Activate Windows
Go to Settings to activate Windows.

➢ **Similarity**

Node similarity algorithms can be used to find nodes that share similar characteristics or patterns of connectivity. This can be particularly useful in a sales network to identify sellers who operate similarly or vehicles that often follow similar sales patterns.

```
CALL gds.nodeSimilarity.stream('myGraphProjection')
YIELD node1, node2, similarity
RETURN gds.util.asNode(node1).name AS Node1, gds.util.asNode(node2).name AS Node2, similarity
ORDER BY similarity asc
LIMIT 10
```

Node1	Node2	similarity
"select remarketing group llc/loan max title"	"ca"	0.00011706860220088972
"caprock auto remarketing"	"ca"	0.000117096018735363
"mid city mcandrew motors"	"ca"	0.00011749500646222535
"chrysler group llc"	"ca"	0.00011760555098200635
"chrysler group/hertz/pv holding/000gd"	"ca"	0.00011763321962122103
"united acceptance"	"ca"	0.00011764705882352942
"peoples credit company inc"	"ca"	0.00011766090128250382
"prestige financial services"	"ca"	0.00011767474699929395
"progressive remarketing"	"ca"	0.00011768859597505002
"new city funding"	"ca"	0.00011770244821092278