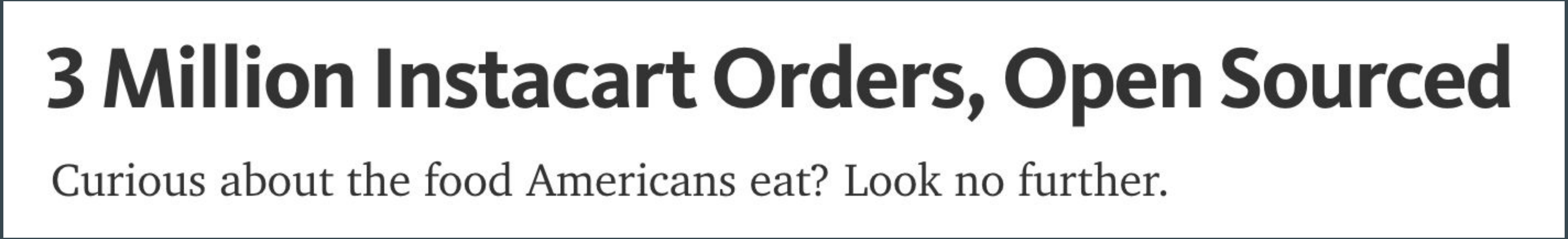
400 bags of grocery receipts + Neo4j …

An exploration of Instacart data using Neo4j



3 Million Instacart Orders, Open Sourced

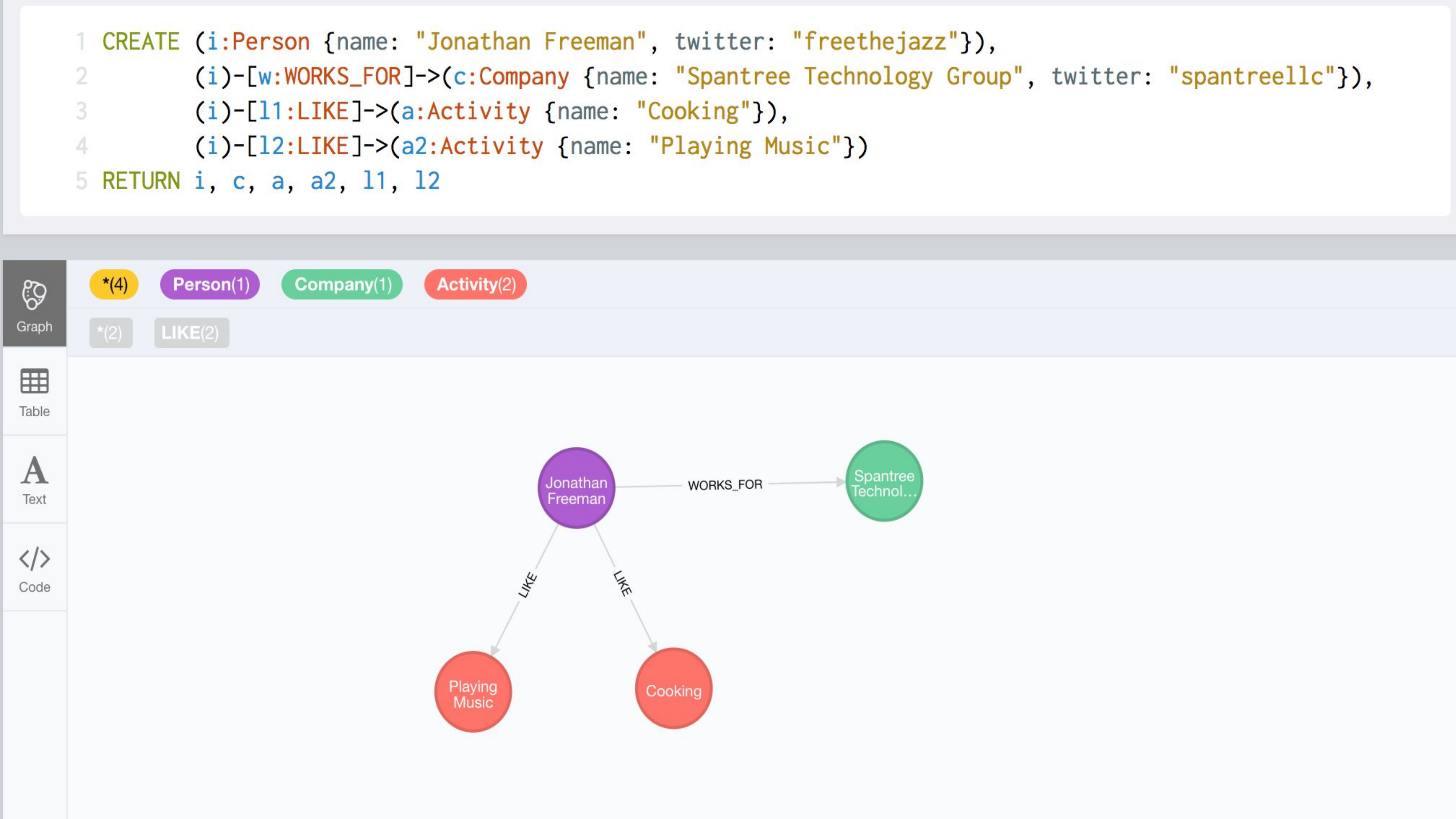
Curious about the food Americans eat? Look no further.

Data and Data Modeling

Mechanics of getting data into Neo4j

Dumpster Diving

Production Concerns (Sizing, import strategy)



```
CREATE (i:Person {name: "Jonathan Freeman", twitter: "freethejazz"}),
       (i)-[w:WORKS_FOR]->(c:Company {name: "Spantree Technology Group", twitter: "spantreellc"}),
       (i)-[l1:LIKE]->(a:Activity {name: "Cooking"}),
       (i)-[l2:LIKE]->(a2:Activity {name: "Playing Music"})
RETURN i, c, a, a2, l1, l2
```

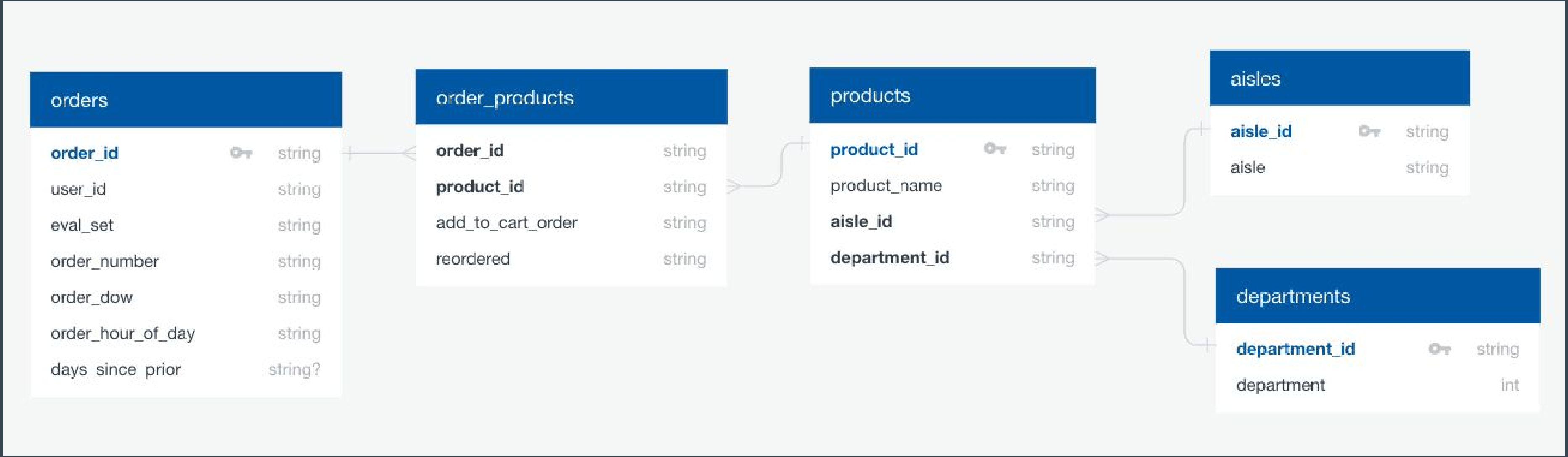
Data and Data Modeling

Mechanics of getting data into Neo4j

Dumpster Diving

Production Concerns (Sizing, import strategy)

CSV Model



orders

order_id	🔑	string
user_id		string
eval_set		string
order_number		string
order_dow		string
order_hour_of_day		string
days_since_prior		string?

order_products

order_id	string
product_id	string
add_to_cart_order	string
reordered	string

products

product_id	🔑	string
product_name		string
aisle_id		string
department_id		string

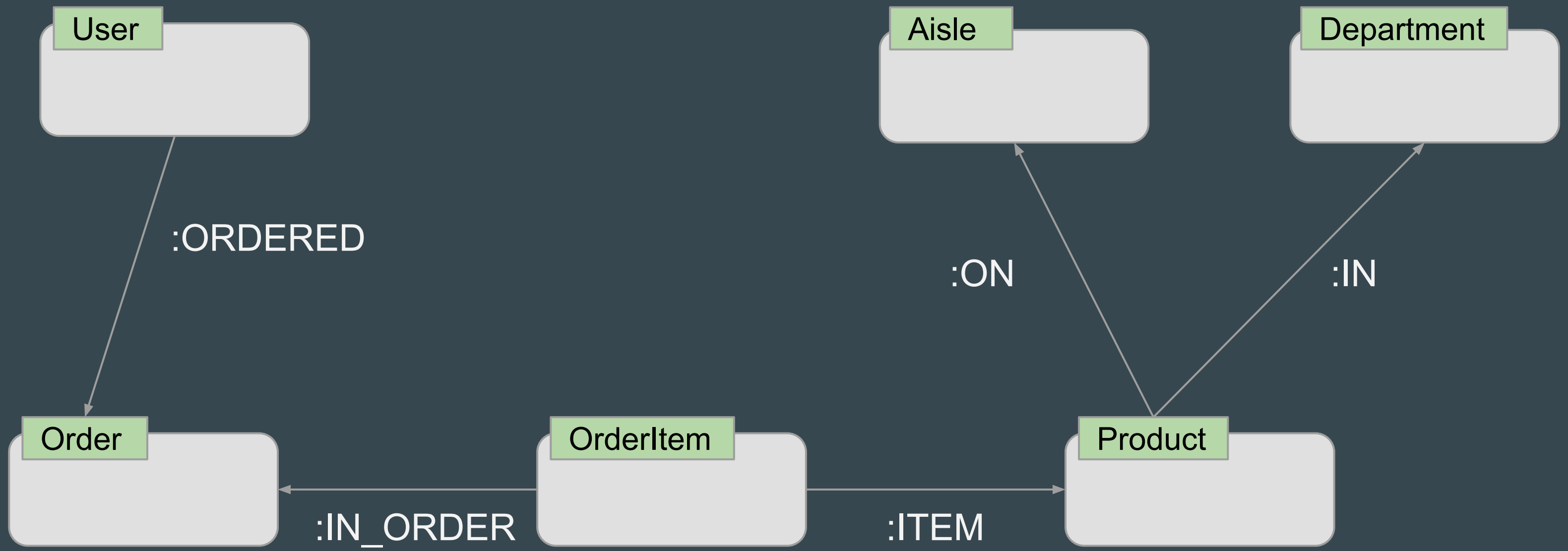
aisles

aisle_id	🔑	string
aisle		string

departments

department_id	🔑	string
department		int

Graph Model


User
Aisle
Department
:ORDERED
:ON
:IN
Order
OrderItem
Product
:IN_ORDER
:ITEM

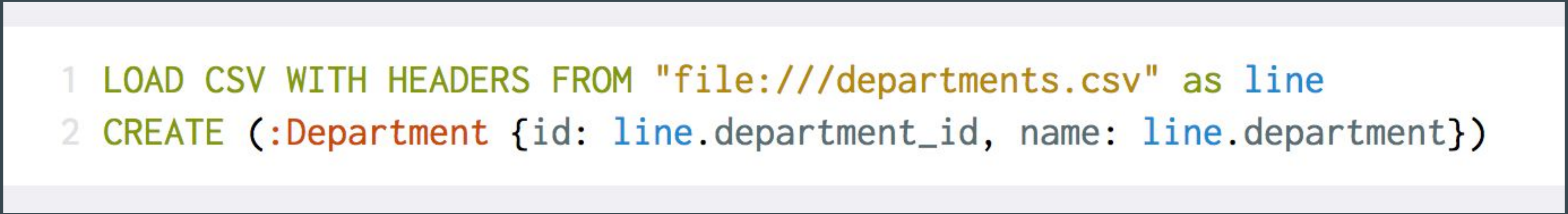
~~Data and Data Modeling~~

Mechanics of getting data into Neo4j

Dumpster Diving

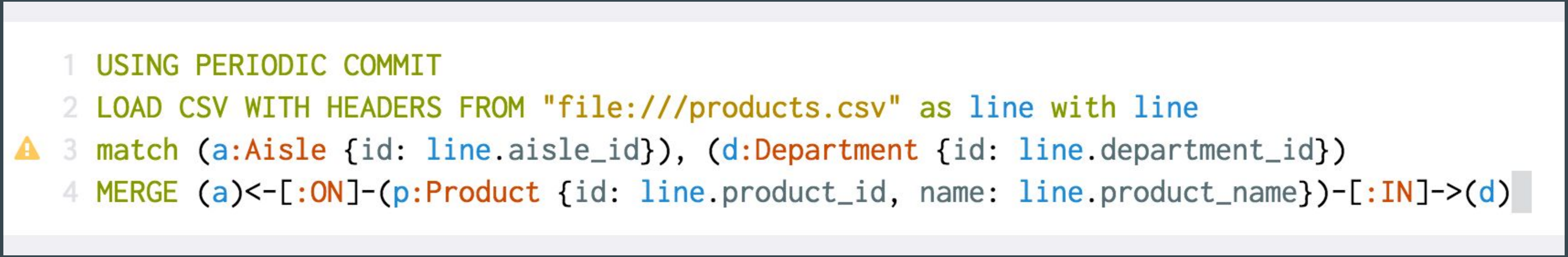
Production Concerns (Sizing, import strategy)

IMPORT FROM CSV WITH HEADERS

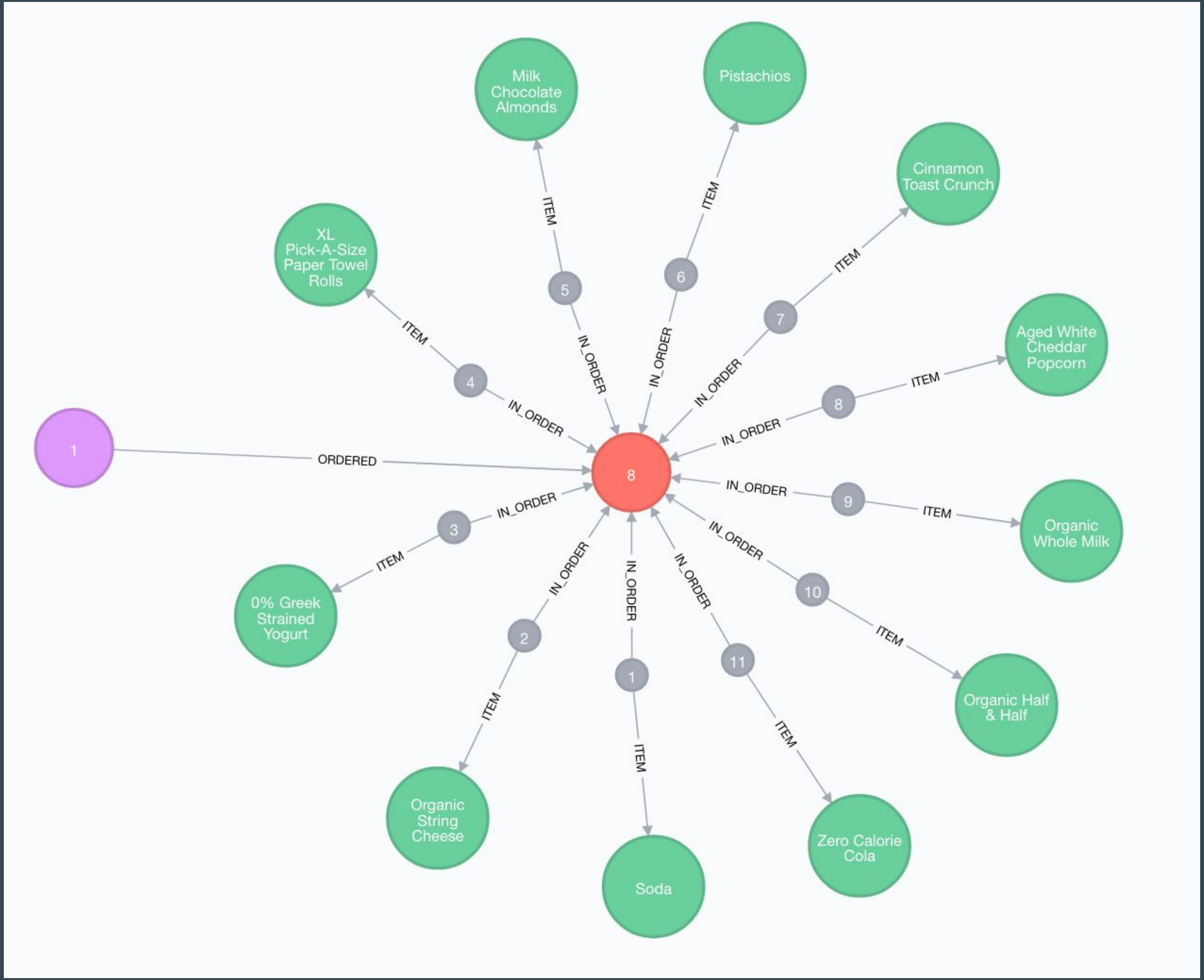


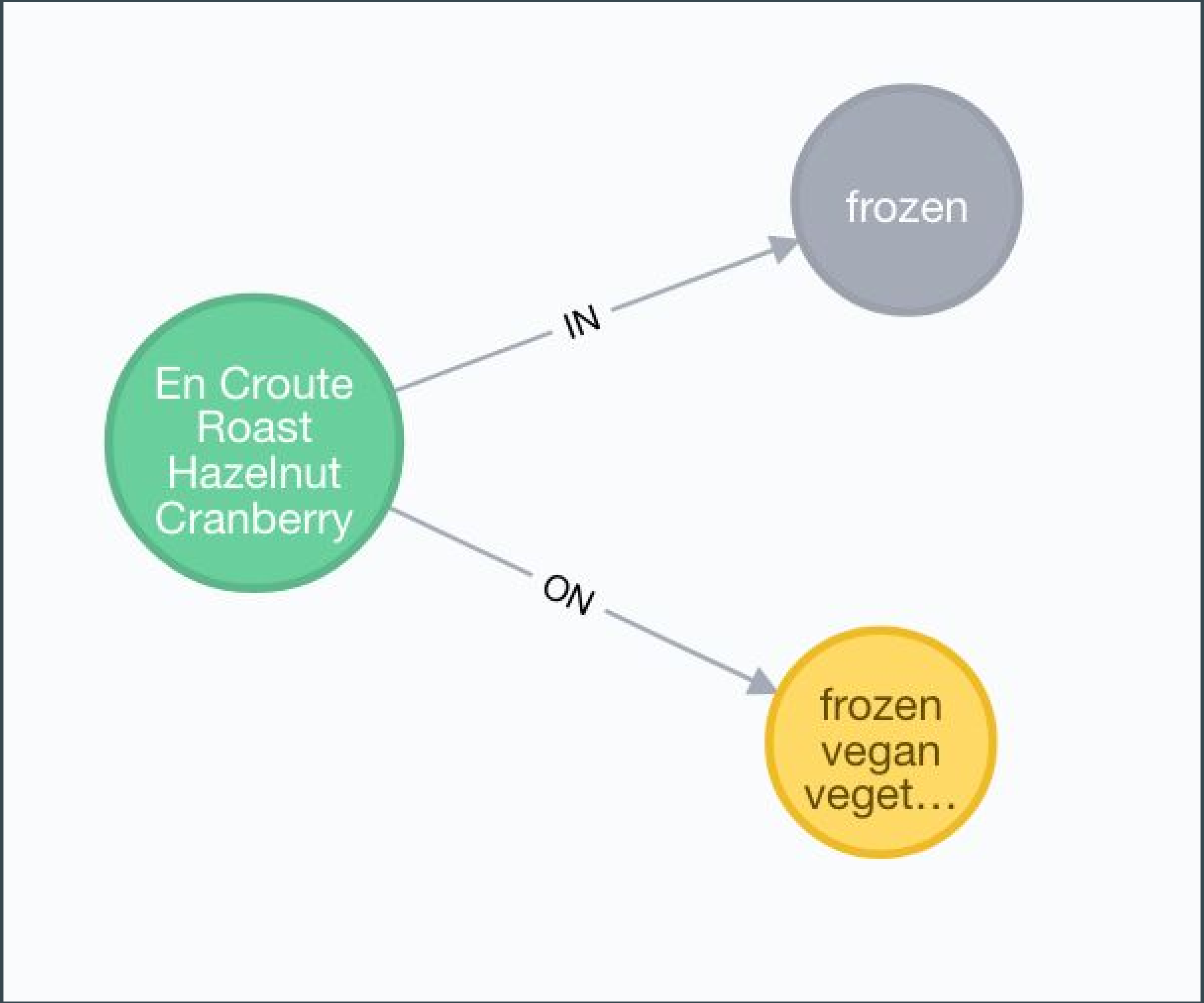
```
LOAD CSV WITH HEADERS FROM "file:///departments.csv" as line
CREATE (:Department {id: line.department_id, name: line.department})
```

Works with relationships, too



```
USING PERIODIC COMMIT
LOAD CSV WITH HEADERS FROM "file:///products.csv" as line with line
match (a:Aisle {id: line.aisle_id}), (d:Department {id: line.department_id})
MERGE (a)<-[:ON]-(p:Product {id: line.product_id, name: line.product_name})-[:IN]->(d)
```


Milk Chocolate Almonds
Pistachios
Cinnamon Toast Crunch
XL Pick-A-Size Paper Towel Rolls
Aged White Cheddar Popcorn
Organic Whole Milk
0% Greek Strained Yogurt
Organic Half & Half
Organic String Cheese
Soda
Zero Calorie Cola
1
ORDERED
8
IN_ORDER
ITEM
1
2
3
4
5
6
7
8
9
10
11


En Croute Roast Hazelnut Cranberry
IN
frozen
ON
frozen vegan veget…

~~Data and Data Modeling~~

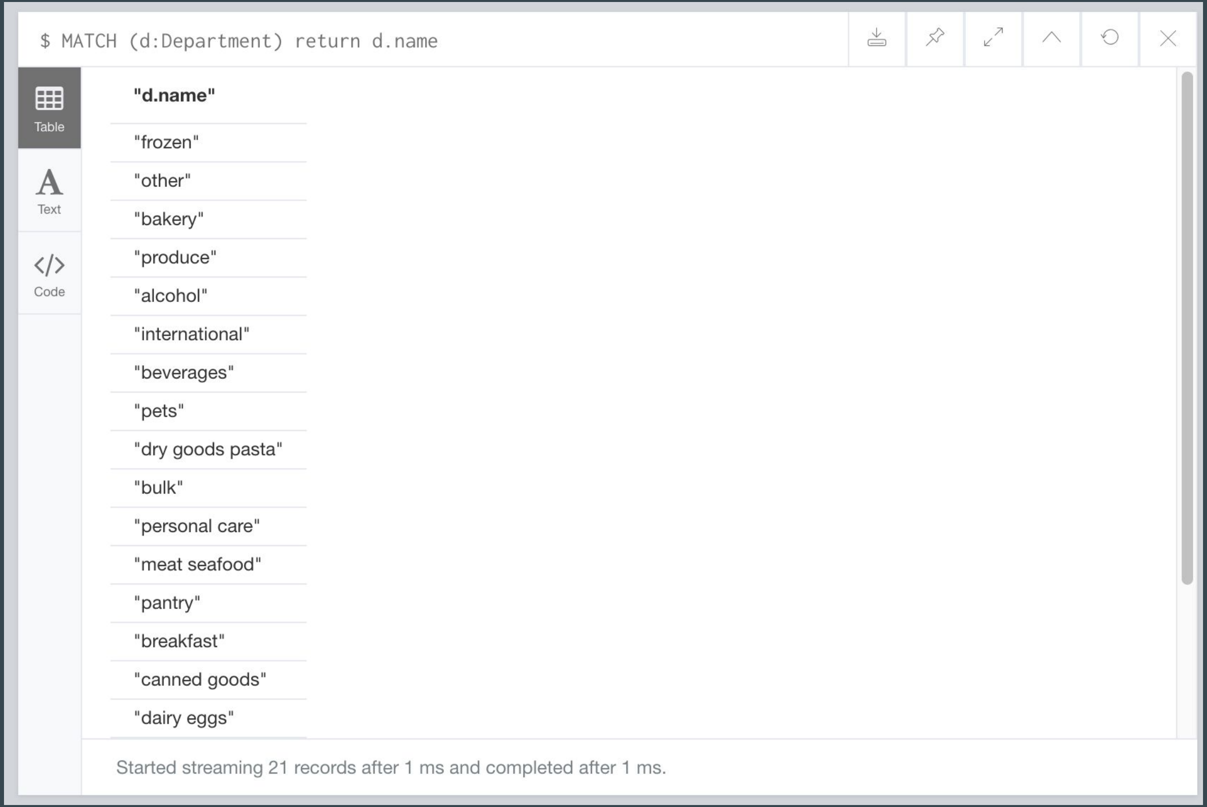
~~Mechanics of getting data into Neo4j~~

Dumpster Diving

Production Concerns (Sizing, import strategy)

Dietary Restrictions

Get Departments



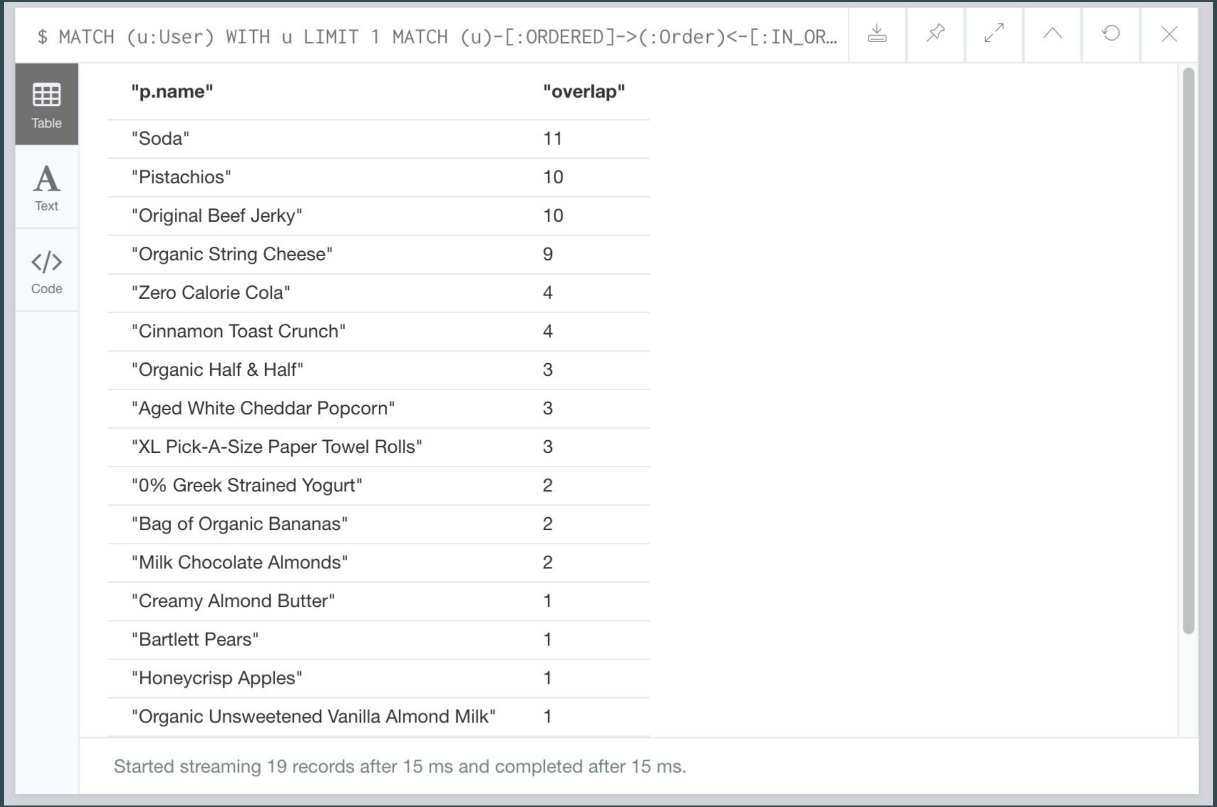
```
$ MATCH (d:Department) return d.name
```

Table
Text
Code

```
"d.name"
"frozen"
"other"
"bakery"
"produce"
"alcohol"
"international"
"beverages"
"pets"
"dry goods pasta"
"bulk"
"personal care"
"meat seafood"
"pantry"
"breakfast"
"canned goods"
"dairy eggs"
```

Started streaming 21 records after 1 ms and completed after 1 ms.

Department-based Vegetarian

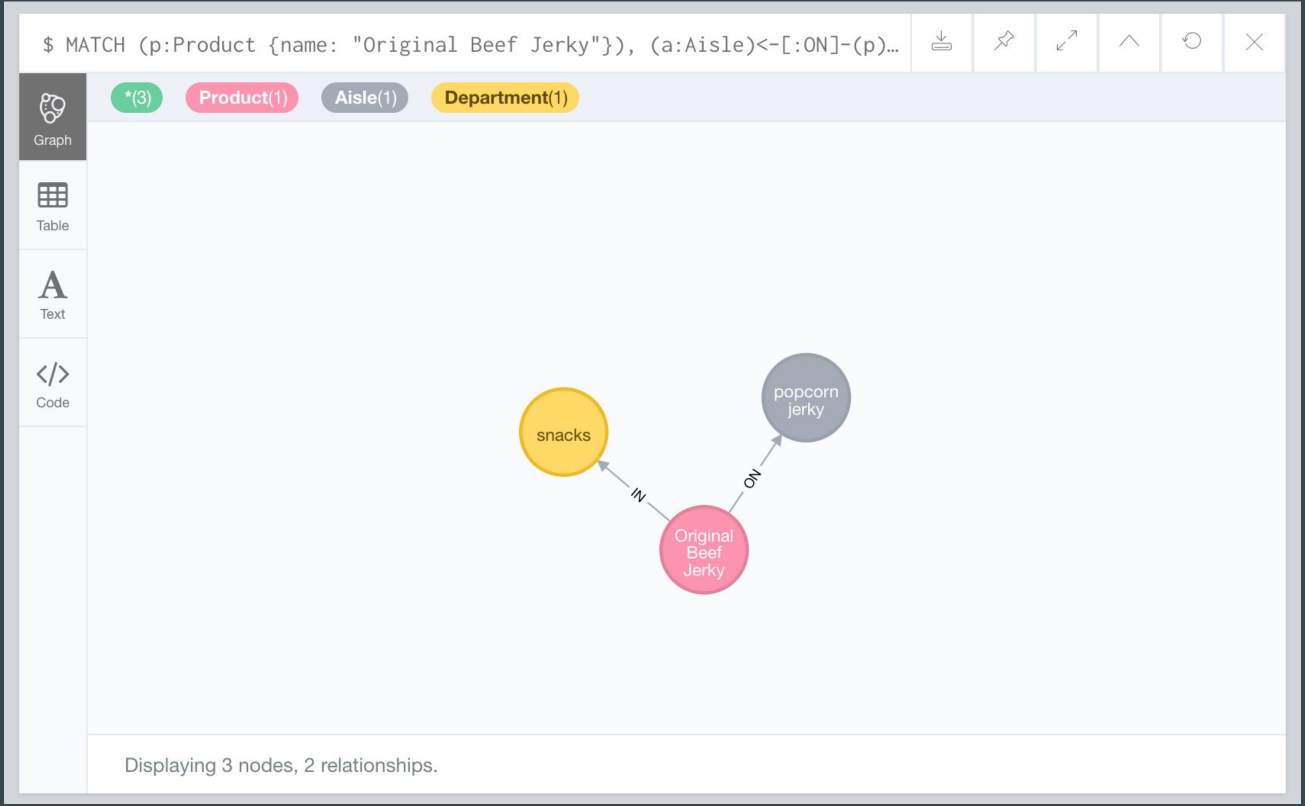


```
$ MATCH (u:User) WITH u LIMIT 1 MATCH (u)-[:ORDERED]->(:Order)<-[:IN_OR…
```

"p.name"	"overlap"
"Soda"	11
"Pistachios"	10
"Original Beef Jerky"	10
"Organic String Cheese"	9
"Zero Calorie Cola"	4
"Cinnamon Toast Crunch"	4
"Organic Half & Half"	3
"Aged White Cheddar Popcorn"	3
"XL Pick-A-Size Paper Towel Rolls"	3
"0% Greek Strained Yogurt"	2
"Bag of Organic Bananas"	2
"Milk Chocolate Almonds"	2
"Creamy Almond Butter"	1
"Bartlett Pears"	1
"Honeycrisp Apples"	1
"Organic Unsweetened Vanilla Almond Milk"	1

Started streaming 19 records after 15 ms and completed after 15 ms.

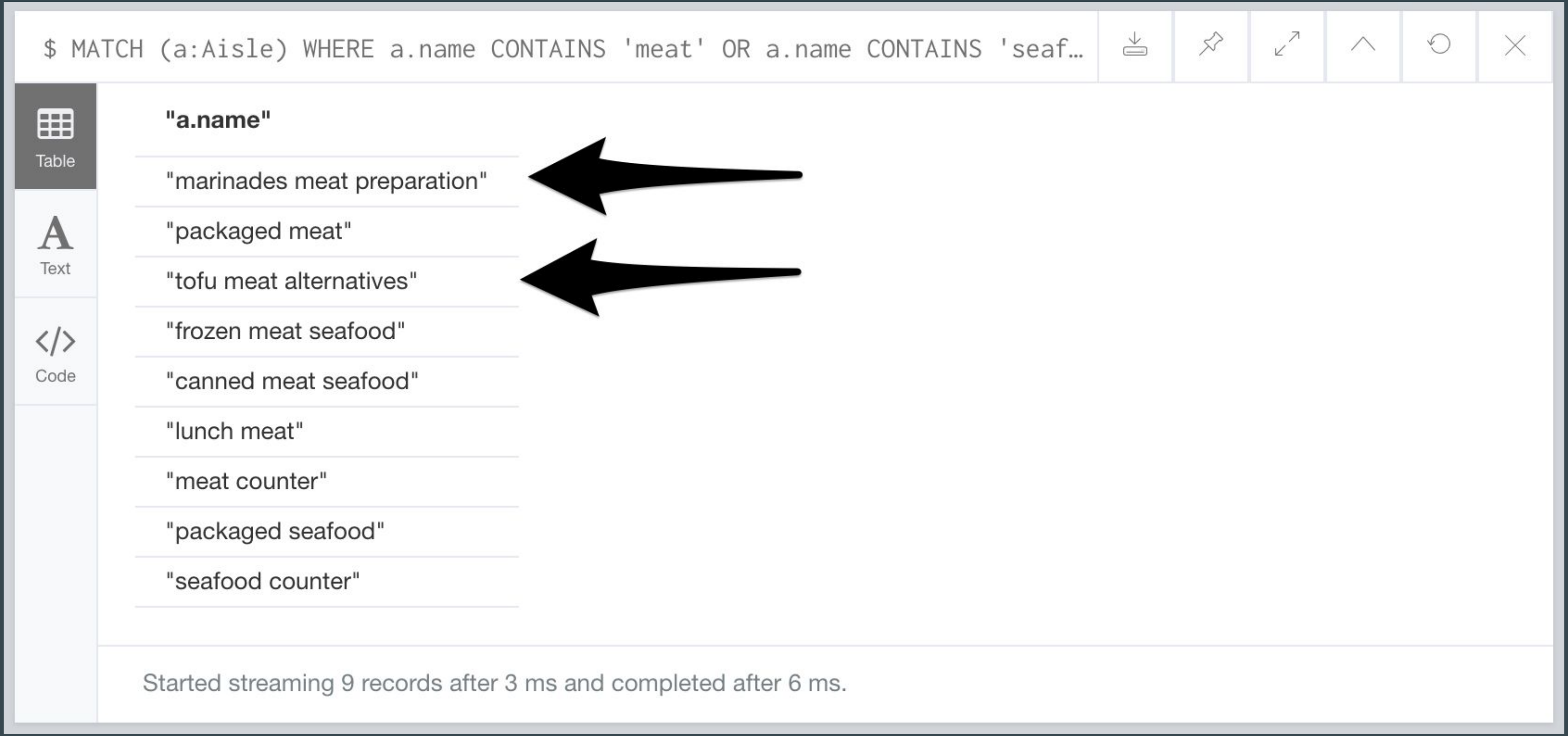
Original Beef Jerky



```
$ MATCH (p:Product {name: "Original Beef Jerky"}), (a:Aisle)<-[:ON]-(p)…
```

Displaying 3 nodes, 2 relationships.

Aisles with Meat or Seafood in the name

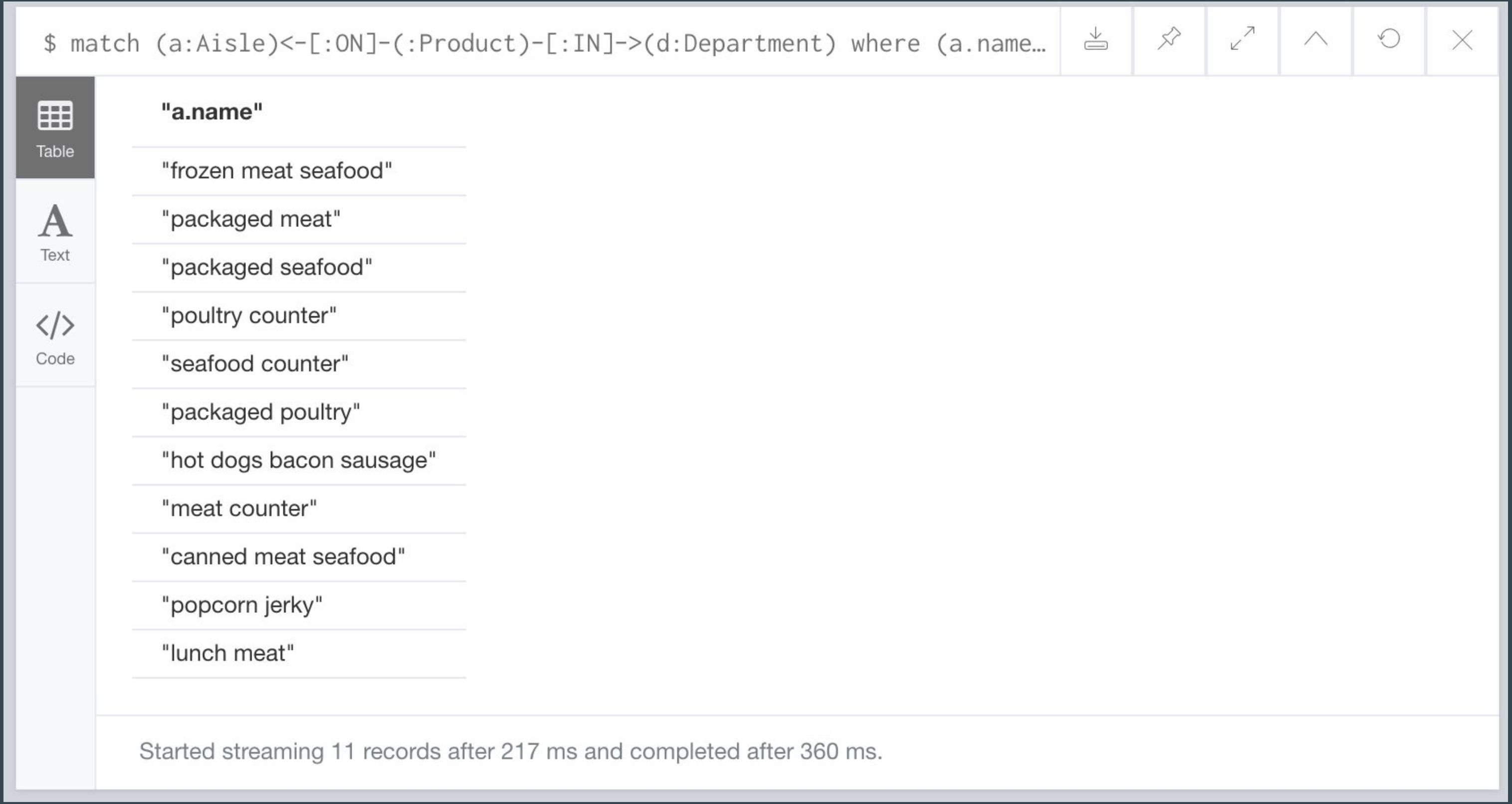


```
$ MATCH (a:Aisle) WHERE a.name CONTAINS 'meat' OR a.name CONTAINS 'seaf…
```

"a.name"
"marinades meat preparation"
"packaged meat"
"tofu meat alternatives"
"frozen meat seafood"
"canned meat seafood"
"lunch meat"
"meat counter"
"packaged seafood"
"seafood counter"

Started streaming 9 records after 3 ms and completed after 6 ms.

Aisles to Avoid



```
$ match (a:Aisle)<-[:ON]-(:Product)-[:IN]->(d:Department) where (a.name…
```

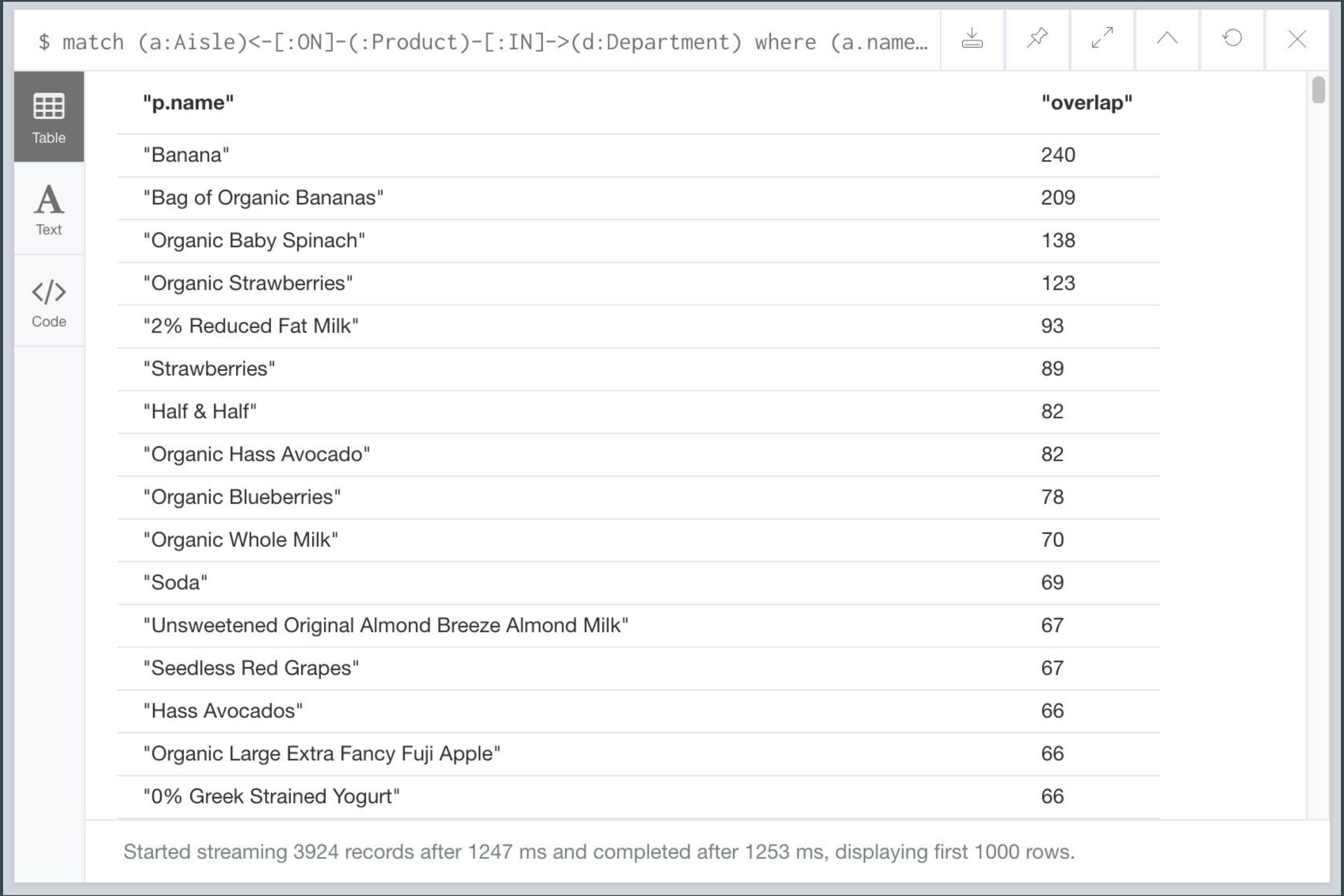
Table

Text

Code

"a.name"
"frozen meat seafood"
"packaged meat"
"packaged seafood"
"poultry counter"
"seafood counter"
"packaged poultry"
"hot dogs bacon sausage"
"meat counter"
"canned meat seafood"
"popcorn jerky"
"lunch meat"

Started streaming 11 records after 217 ms and completed after 360 ms.



```
$ match (a:Aisle)<-[:ON]-(:Product)-[:IN]->(d:Department) where (a.name…
```

Table

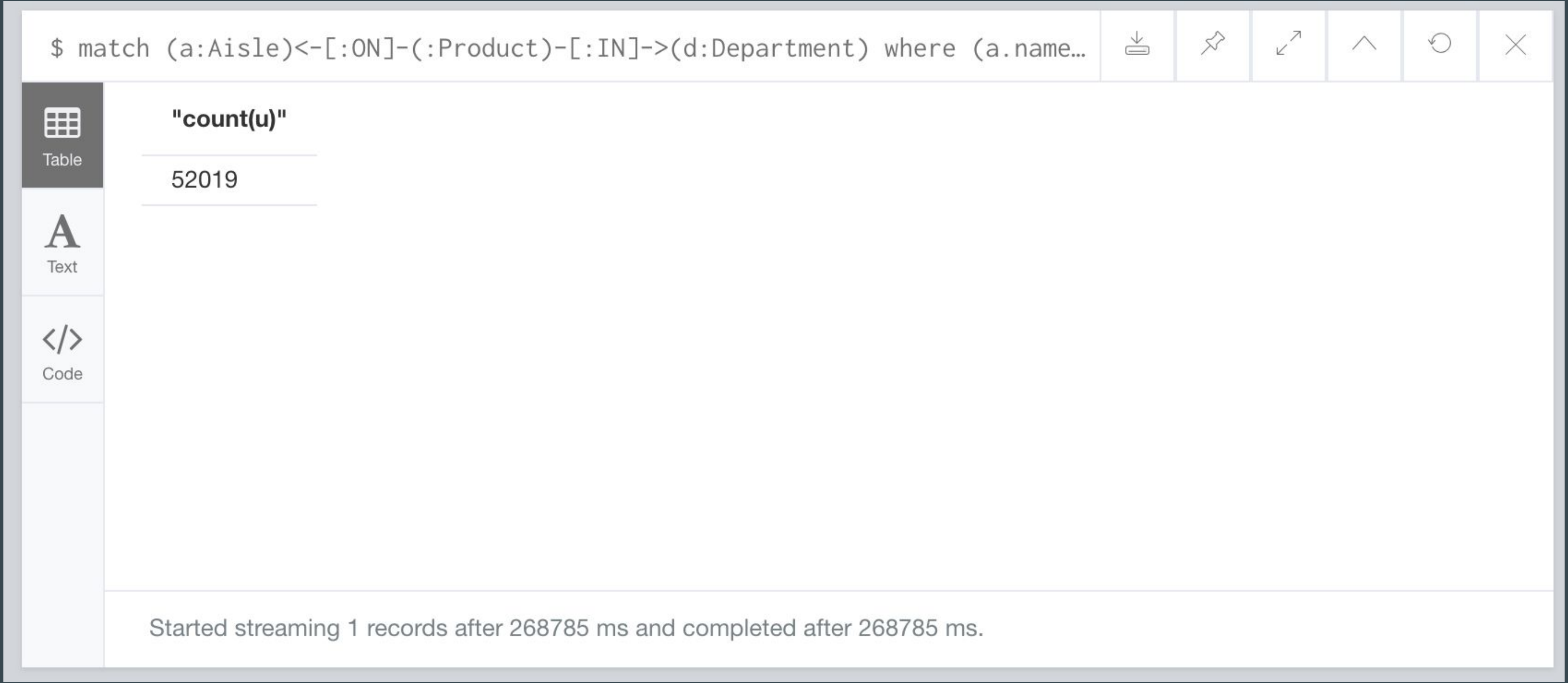
Text

Code

"p.name"	"overlap"
"Banana"	240
"Bag of Organic Bananas"	209
"Organic Baby Spinach"	138
"Organic Strawberries"	123
"2% Reduced Fat Milk"	93
"Strawberries"	89
"Half & Half"	82
"Organic Hass Avocado"	82
"Organic Blueberries"	78
"Organic Whole Milk"	70
"Soda"	69
"Unsweetened Original Almond Breeze Almond Milk"	67
"Seedless Red Grapes"	67
"Hass Avocados"	66
"Organic Large Extra Fancy Fuji Apple"	66
"0% Greek Strained Yogurt"	66

Started streaming 3924 records after 1247 ms and completed after 1253 ms, displaying first 1000 rows.

How Many Vegetarians?

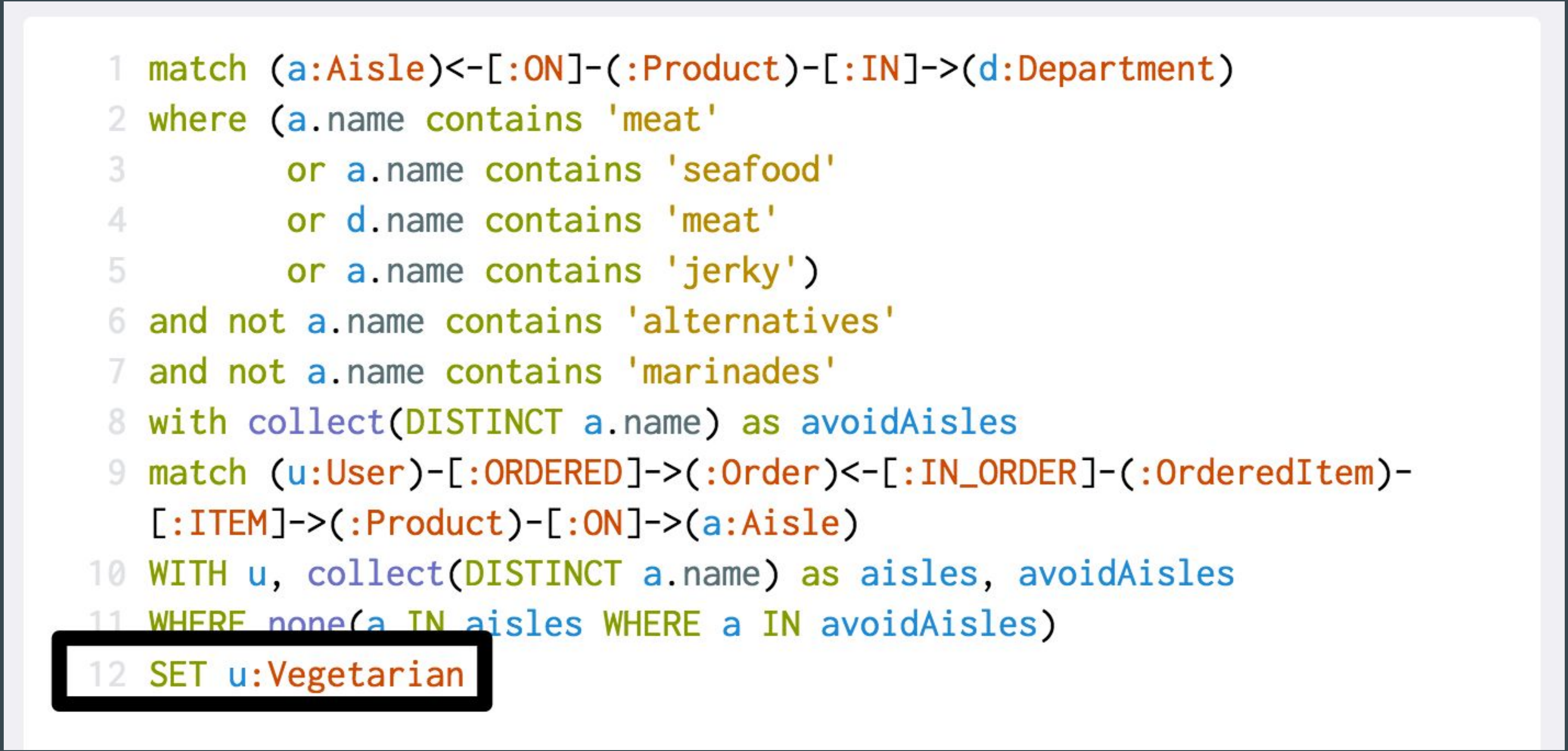


```
$ match (a:Aisle)<-[:ON]-(:Product)-[:IN]->(d:Department) where (a.name…
```

```
"count(u)"
52019
```

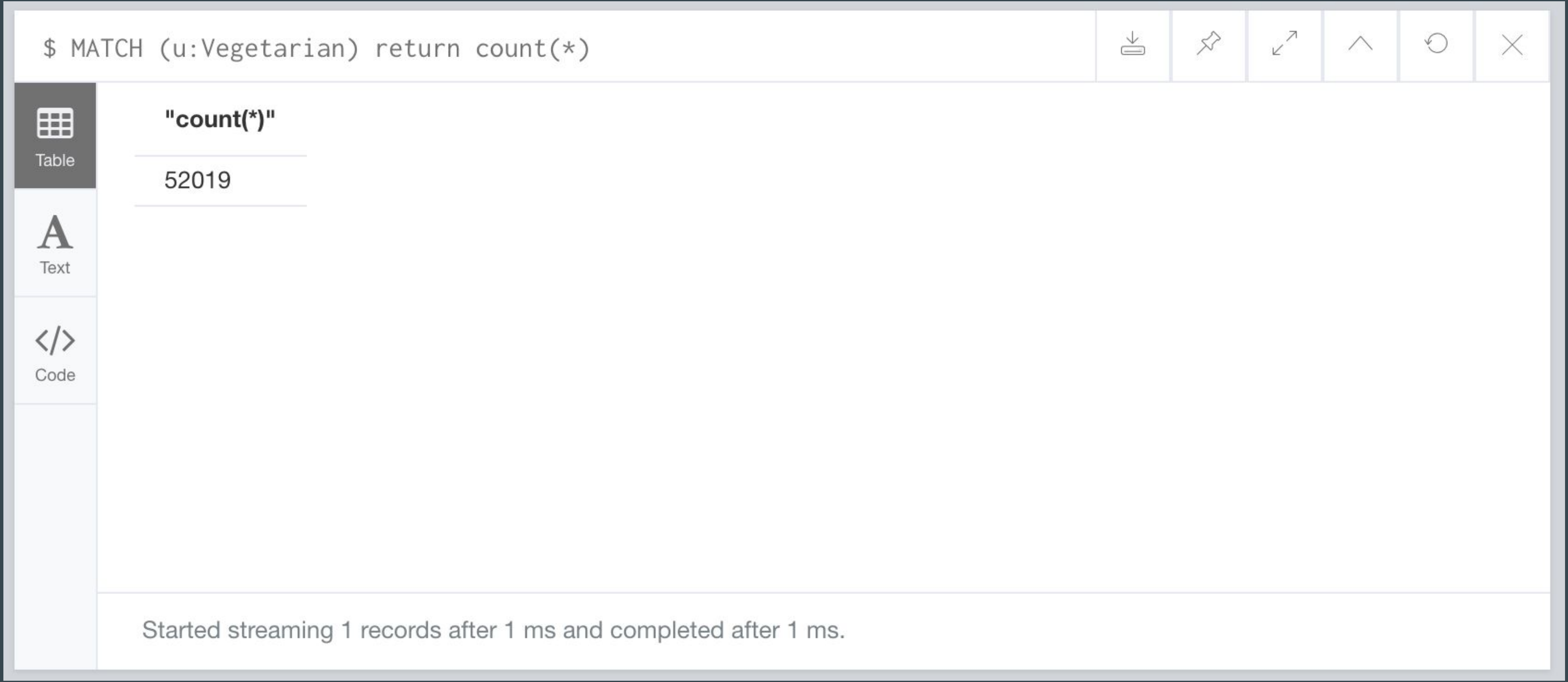
Started streaming 1 records after 268785 ms and completed after 268785 ms.

Evolve the model



```
match (a:Aisle)<-[:ON]-(:Product)-[:IN]->(d:Department)
where (a.name contains 'meat'
       or a.name contains 'seafood'
       or d.name contains 'meat'
       or a.name contains 'jerky')
and not a.name contains 'alternatives'
and not a.name contains 'marinades'
with collect(DISTINCT a.name) as avoidAisles
match (u:User)-[:ORDERED]->(:Order)<-[:IN_ORDER]-(:OrderedItem)-[:ITEM]->(:Product)-[:ON]->(a:Aisle)
WITH u, collect(DISTINCT a.name) as aisles, avoidAisles
WHERE none(a IN aisles WHERE a IN avoidAisles)
SET u:Vegetarian
```

Add a label, et voila

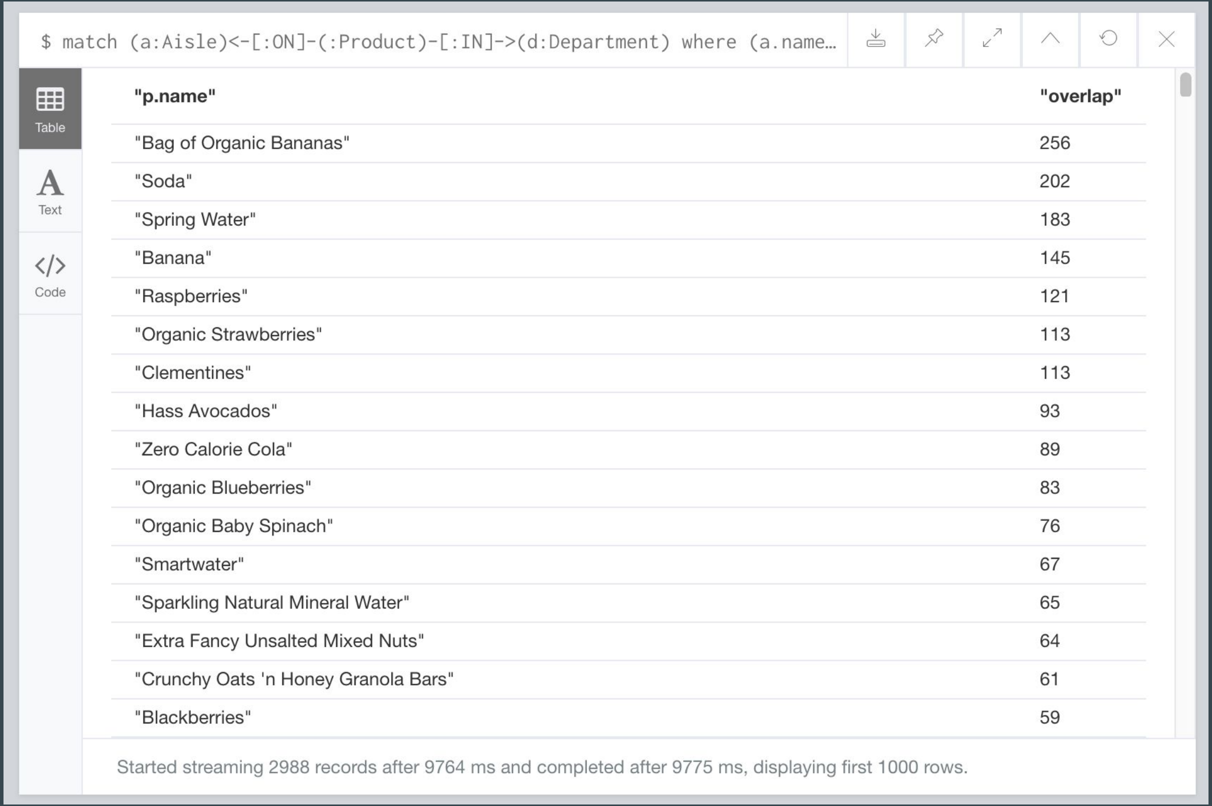


```
$ MATCH (u:Vegetarian) return count(*)
```

"count(*)"
52019

Started streaming 1 records after 1 ms and completed after 1 ms.

Products “Vegans” buy



```
$ match (a:Aisle)<-[:ON]-(:Product)-[:IN]->(d:Department) where (a.name…
```

Table

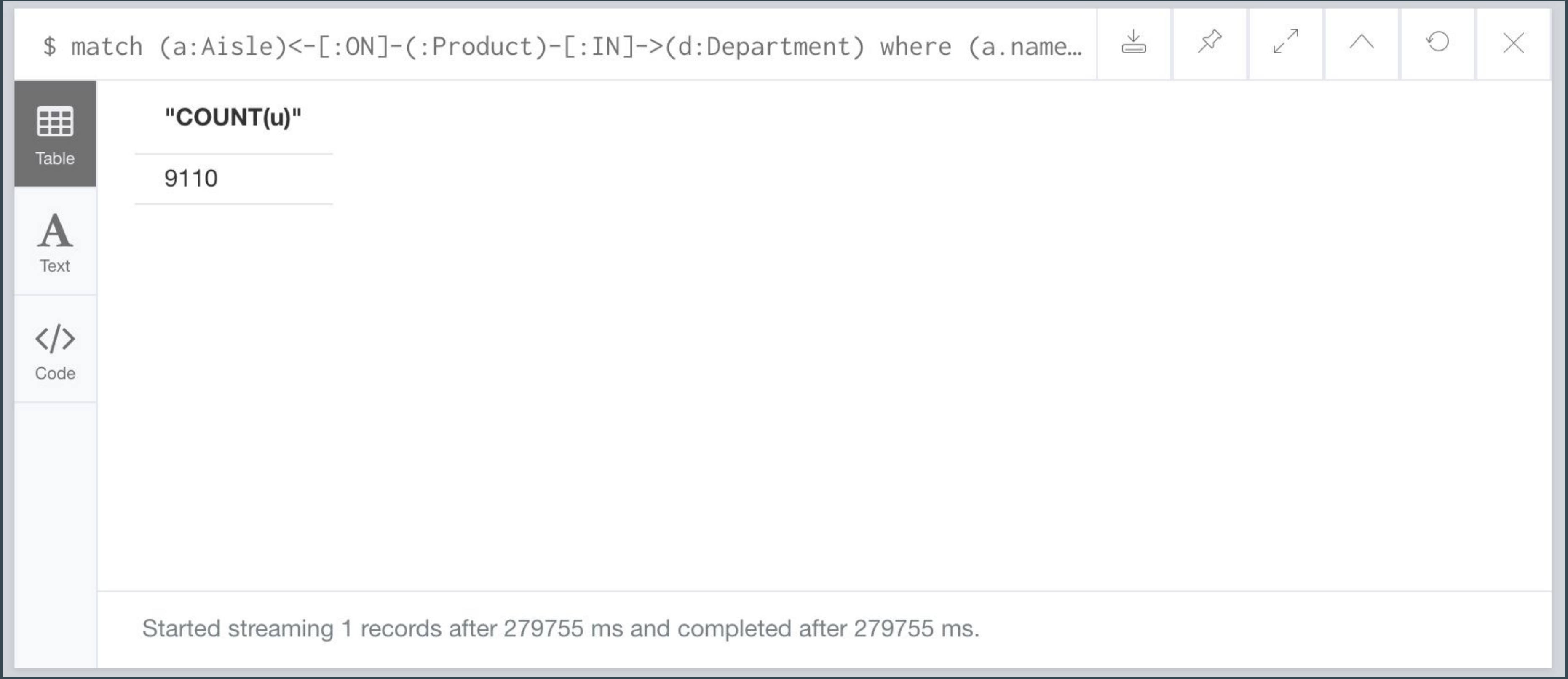
Text

Code

"p.name"	"overlap"
"Bag of Organic Bananas"	256
"Soda"	202
"Spring Water"	183
"Banana"	145
"Raspberries"	121
"Organic Strawberries"	113
"Clementines"	113
"Hass Avocados"	93
"Zero Calorie Cola"	89
"Organic Blueberries"	83
"Organic Baby Spinach"	76
"Smartwater"	67
"Sparkling Natural Mineral Water"	65
"Extra Fancy Unsalted Mixed Nuts"	64
"Crunchy Oats 'n Honey Granola Bars"	61
"Blackberries"	59

Started streaming 2988 records after 9764 ms and completed after 9775 ms, displaying first 1000 rows.

How Many "Vegans"?

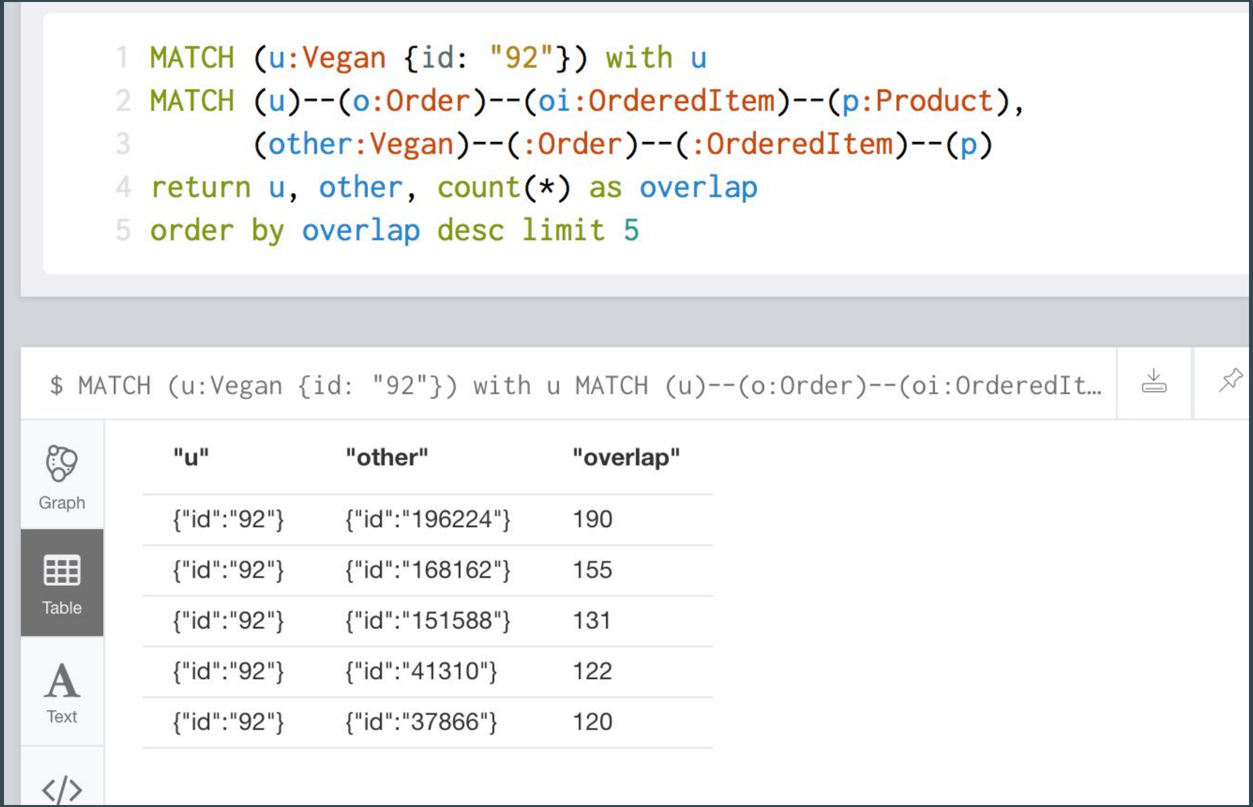


```
$ match (a:Aisle)<-[:ON]-(:Product)-[:IN]->(d:Department) where (a.name…
```

"COUNT(u)"
9110

Started streaming 1 records after 279755 ms and completed after 279755 ms.

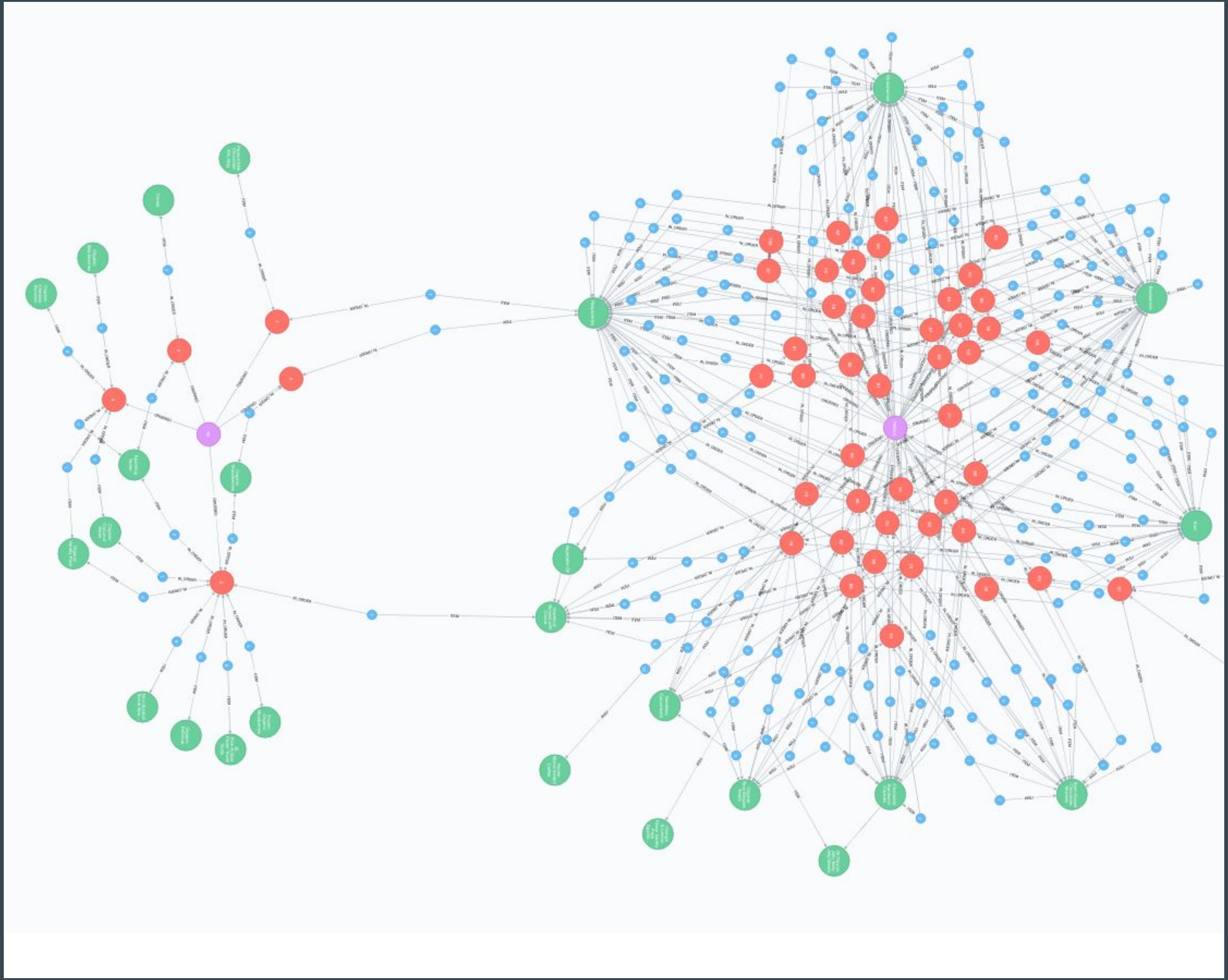
Vegans like me



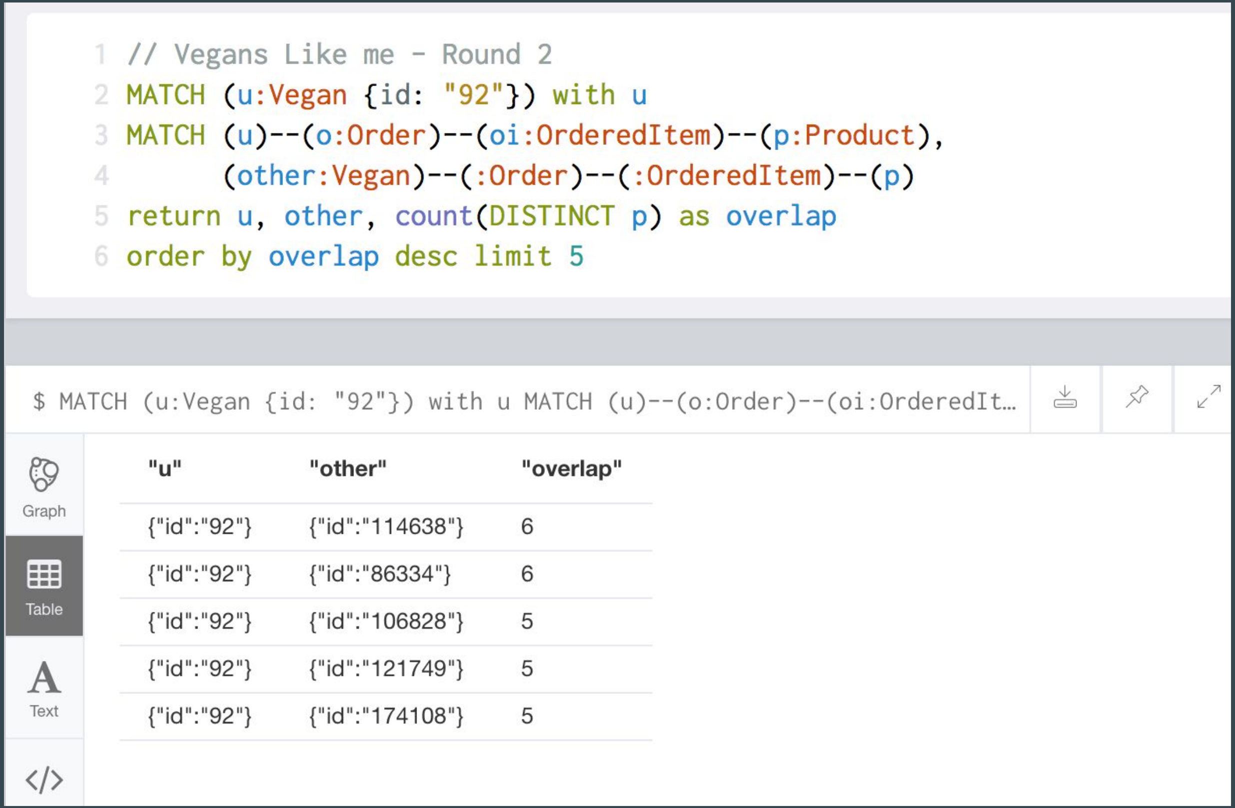
```
MATCH (u:Vegan {id: "92"}) with u
MATCH (u)--(o:Order)--(oi:OrderedItem)--(p:Product),
      (other:Vegan)--(:Order)--(:OrderedItem)--(p)
return u, other, count(*) as overlap
order by overlap desc limit 5
```

```
$ MATCH (u:Vegan {id: "92"}) with u MATCH (u)--(o:Order)--(oi:OrderedIt…
```

"u"	"other"	"overlap"
{"id":"92"}	{"id":"196224"}	190
{"id":"92"}	{"id":"168162"}	155
{"id":"92"}	{"id":"151588"}	131
{"id":"92"}	{"id":"41310"}	122
{"id":"92"}	{"id":"37866"}	120



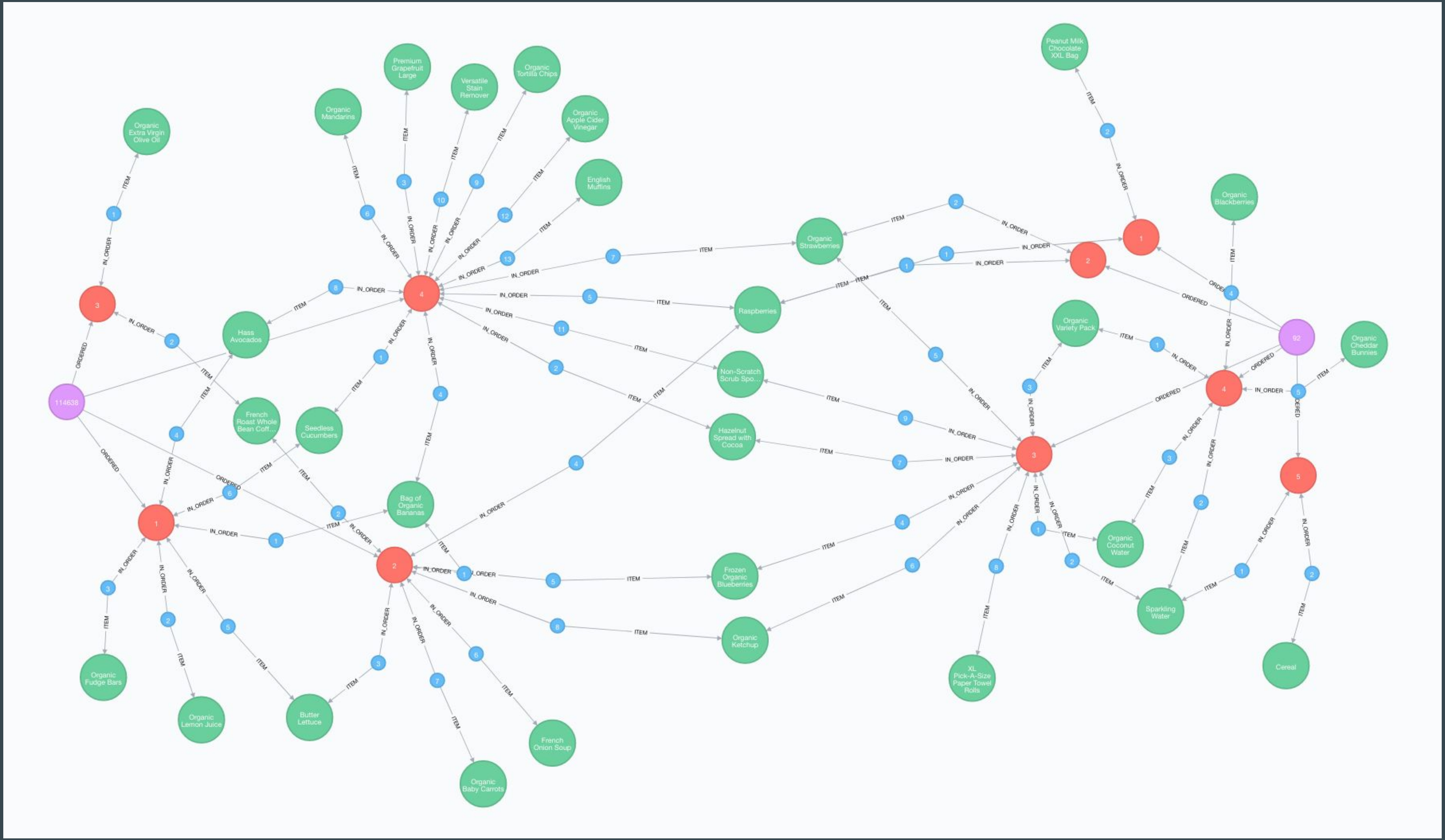
Vegans like me - Round 2



```
// Vegans Like me - Round 2
MATCH (u:Vegan {id: "92"}) with u
MATCH (u)--(o:Order)--(oi:OrderedItem)--(p:Product),
      (other:Vegan)--(:Order)--(:OrderedItem)--(p)
return u, other, count(DISTINCT p) as overlap
order by overlap desc limit 5
```

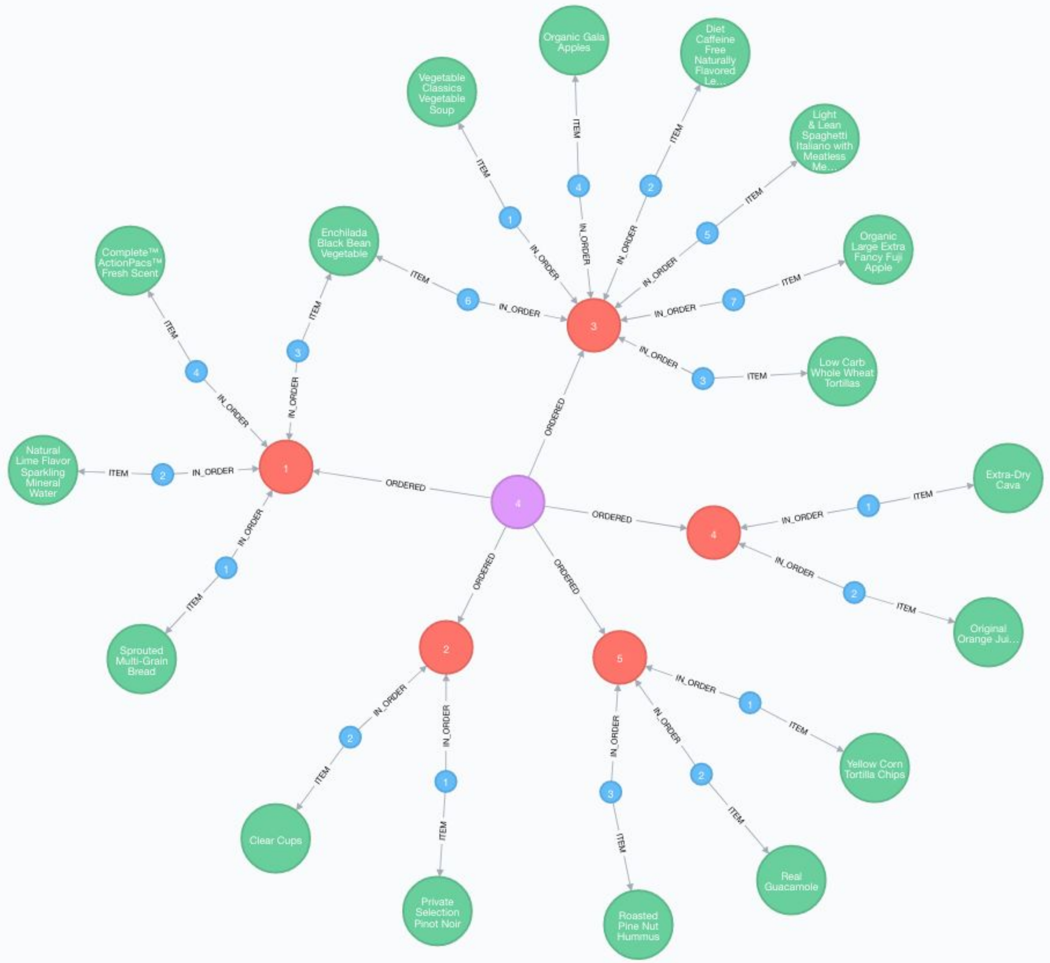
```
$ MATCH (u:Vegan {id: "92"}) with u MATCH (u)--(o:Order)--(oi:OrderedIt…
```

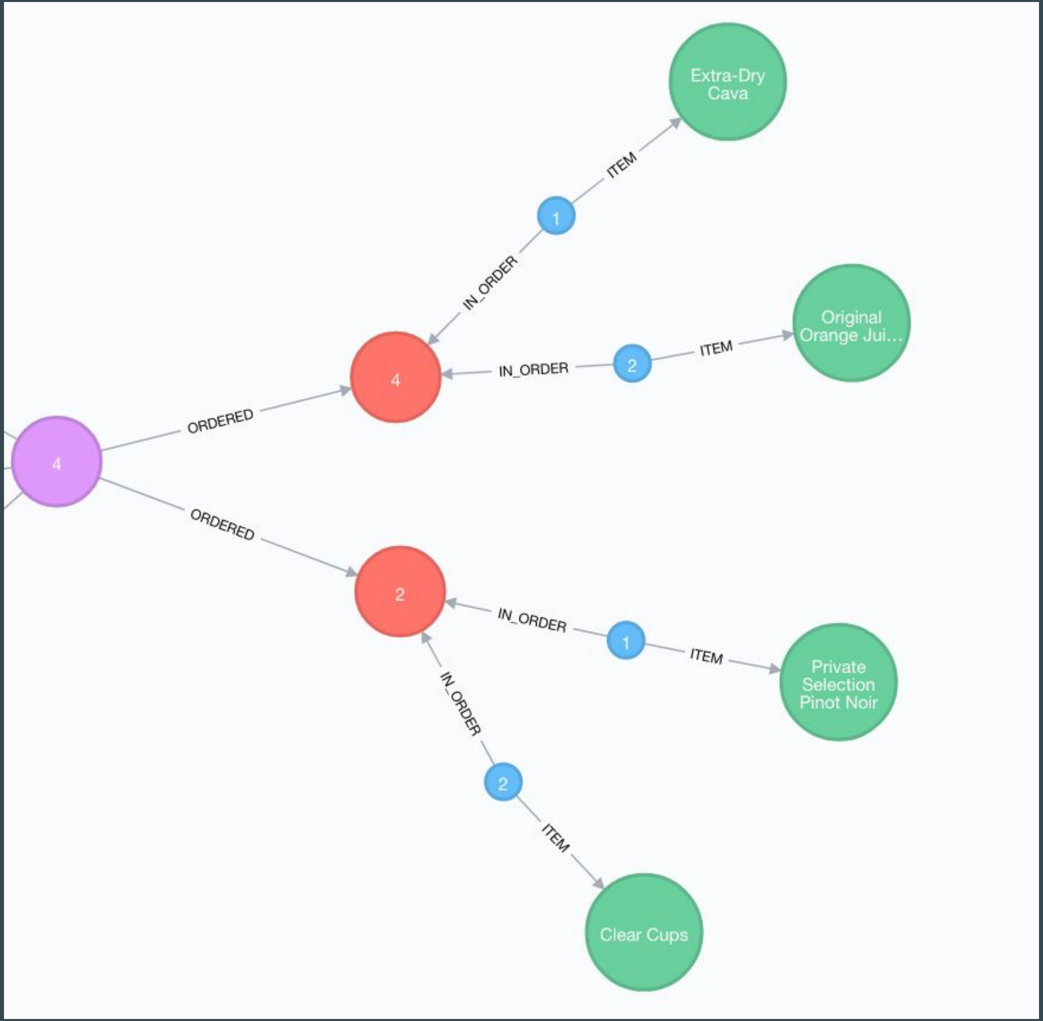
"u"	"other"	"overlap"
{"id":"92"}	{"id":"114638"}	6
{"id":"92"}	{"id":"86334"}	6
{"id":"92"}	{"id":"106828"}	5
{"id":"92"}	{"id":"121749"}	5
{"id":"92"}	{"id":"174108"}	5

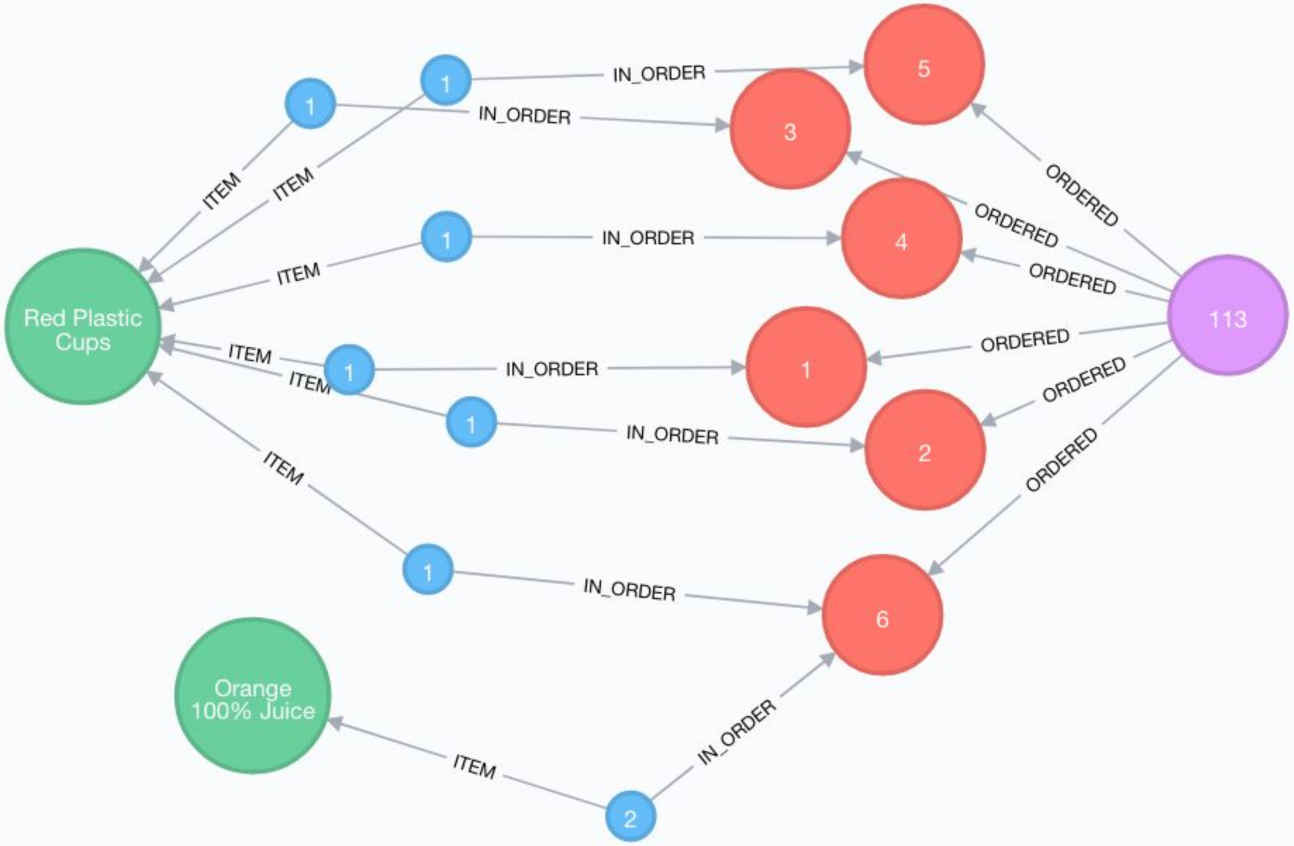

114638
92
ORDERED
IN_ORDER
ITEM
Organic Extra Virgin Olive Oil
Organic Mandarins
Premium Grapefruit Large
Versatile Stain Remover
Organic Tortilla Chips
Organic Apple Cider Vinegar
English Muffins
Hass Avocados
French Roast Whole Bean Coff…
Seedless Cucumbers
Bag of Organic Bananas
Organic Strawberries
Raspberries
Non-Scratch Scrub Spo…
Hazelnut Spread with Cocoa
Frozen Organic Blueberries
Organic Ketchup
Organic Fudge Bars
Organic Lemon Juice
Butter Lettuce
Organic Baby Carrots
French Onion Soup
Peanut Milk Chocolate XXL Bag
Organic Blackberries
Organic Variety Pack
Organic Cheddar Bunnies
Organic Coconut Water
Sparkling Water
XL Pick-A-Size Paper Towel Rolls
Cereal

Instadate

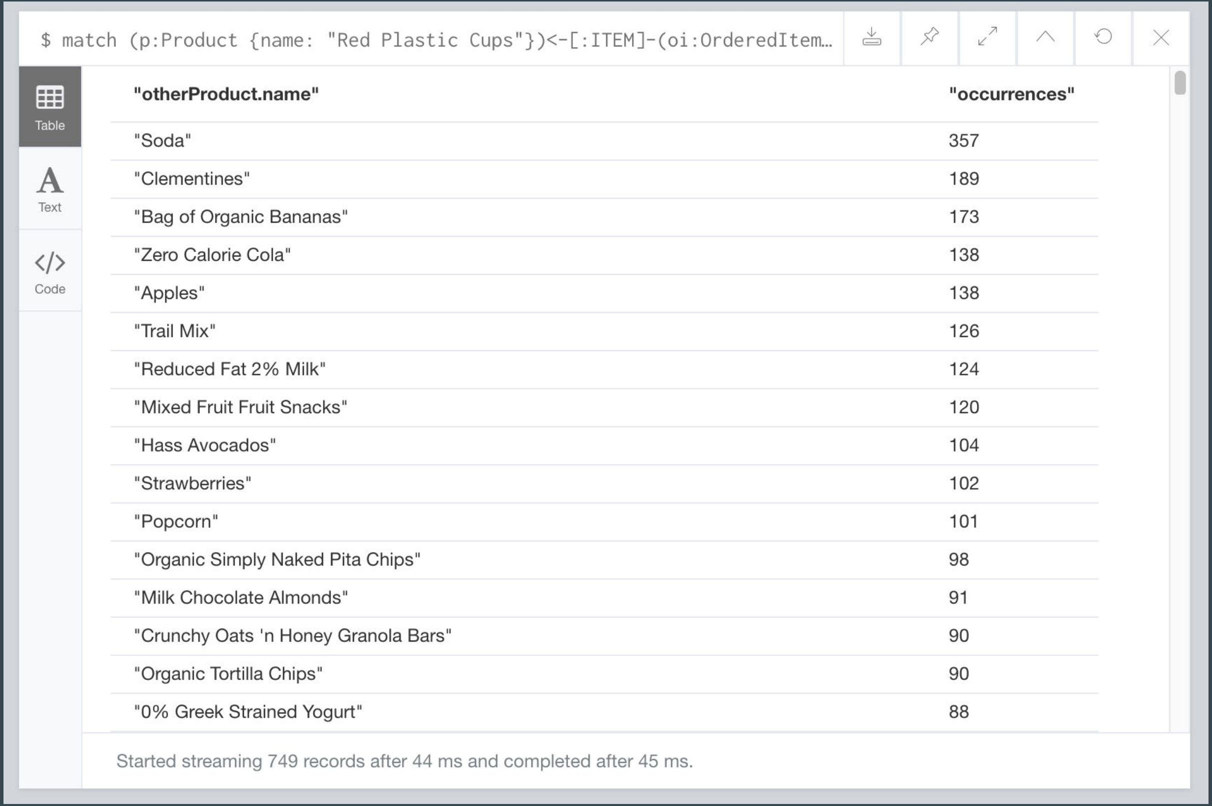
One more thing…


4
ORDERED
1
2
3
4
5
IN_ORDER
ITEM
Complete™ ActionPacs™ Fresh Scent
Enchilada Black Bean Vegetable
Natural Lime Flavor Sparkling Mineral Water
Sprouted Multi-Grain Bread
Vegetable Classics Vegetable Soup
Organic Gala Apples
Diet Caffeine Free Naturally Flavored Le...
Light & Lean Spaghetti Italiano with Meatless Me...
Organic Large Extra Fancy Fuji Apple
Low Carb Whole Wheat Tortillas
Extra-Dry Cava
Original Orange Jui...
Clear Cups
Private Selection Pinot Noir
Roasted Pine Nut Hummus
Real Guacamole
Yellow Corn Tortilla Chips
6
7


Extra-Dry Cava
ITEM
1
IN_ORDER
Original Orange Jui…
ITEM
2
IN_ORDER
4
ORDERED
4
ORDERED
2
IN_ORDER
1
ITEM
Private Selection Pinot Noir
IN_ORDER
2
ITEM
Clear Cups


Red Plastic Cups
Orange 100% Juice
113
ITEM
IN_ORDER
ORDERED
1
2
3
4
5
6

Cups and ....

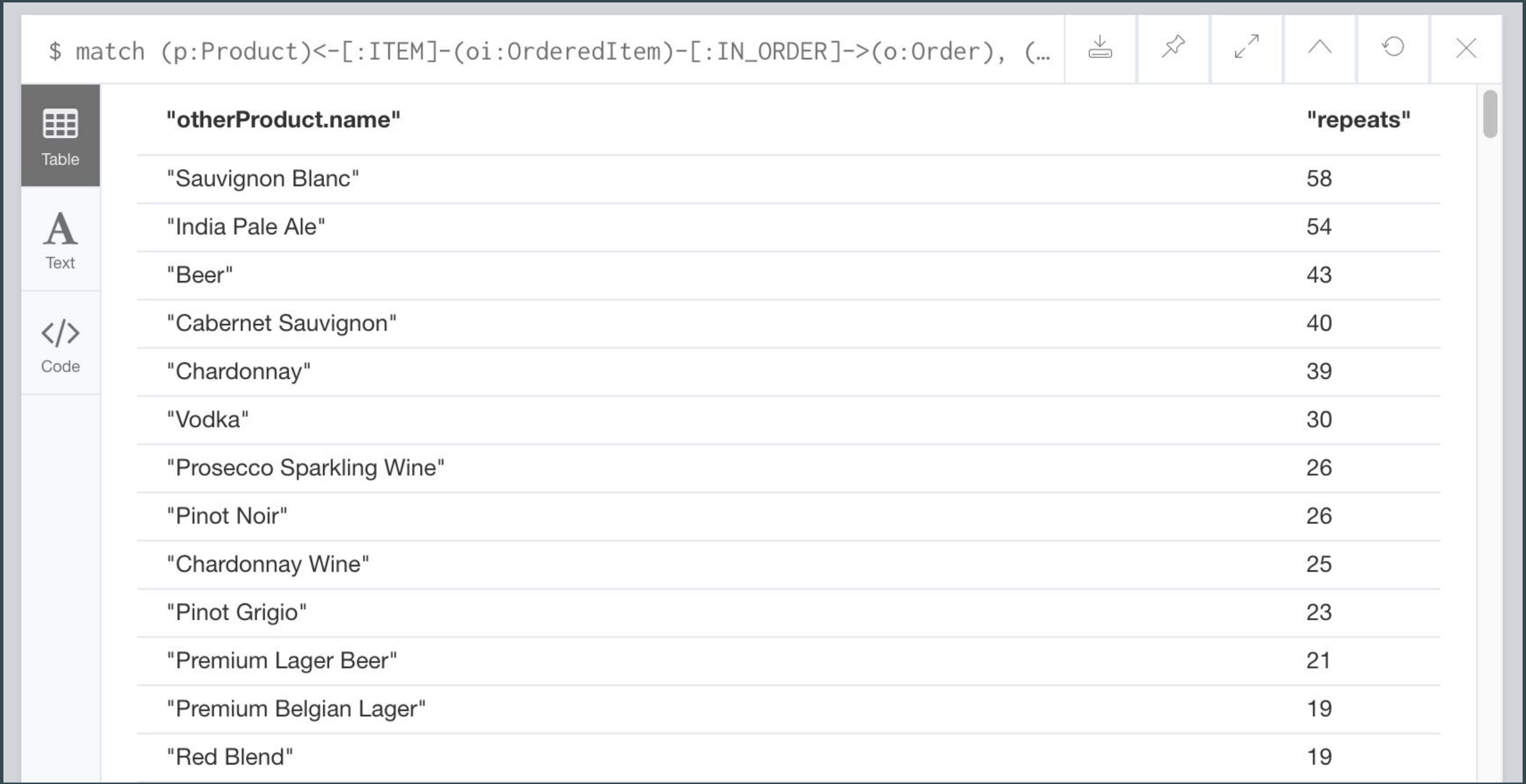


```
$ match (p:Product {name: "Red Plastic Cups"})<-[:ITEM]-(oi:OrderedItem…
```

"otherProduct.name"	"occurrences"
"Soda"	357
"Clementines"	189
"Bag of Organic Bananas"	173
"Zero Calorie Cola"	138
"Apples"	138
"Trail Mix"	126
"Reduced Fat 2% Milk"	124
"Mixed Fruit Fruit Snacks"	120
"Hass Avocados"	104
"Strawberries"	102
"Popcorn"	101
"Organic Simply Naked Pita Chips"	98
"Milk Chocolate Almonds"	91
"Crunchy Oats 'n Honey Granola Bars"	90
"Organic Tortilla Chips"	90
"0% Greek Strained Yogurt"	88

Started streaming 749 records after 44 ms and completed after 45 ms.

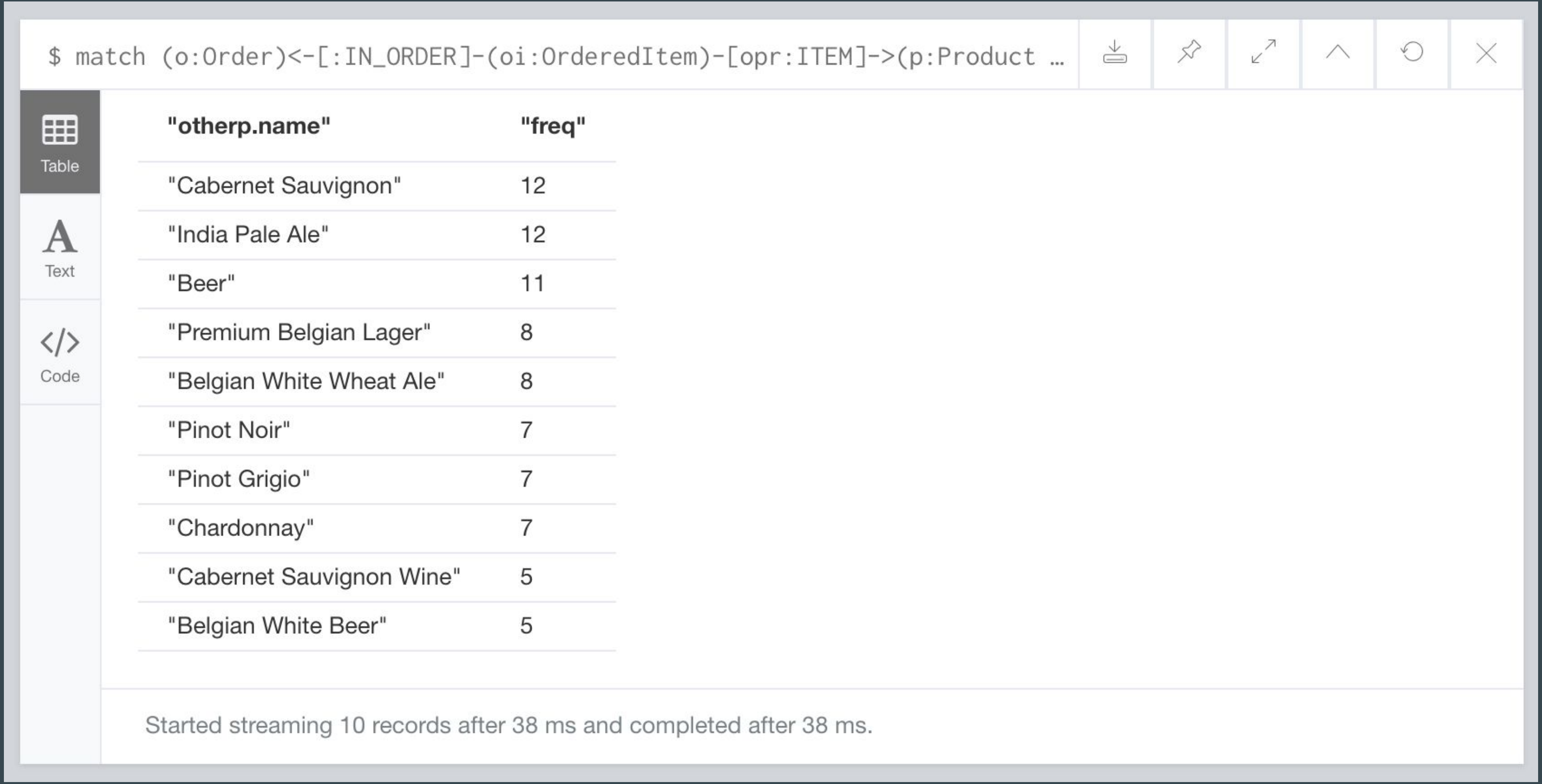
Cups and Drinks



```
$ match (p:Product)<-[:ITEM]-(oi:OrderedItem)-[:IN_ORDER]->(o:Order), (…
```

"otherProduct.name"	"repeats"
"Sauvignon Blanc"	58
"India Pale Ale"	54
"Beer"	43
"Cabernet Sauvignon"	40
"Chardonnay"	39
"Vodka"	30
"Prosecco Sparkling Wine"	26
"Pinot Noir"	26
"Chardonnay Wine"	25
"Pinot Grigio"	23
"Premium Lager Beer"	21
"Premium Belgian Lager"	19
"Red Blend"	19

What else could you want?



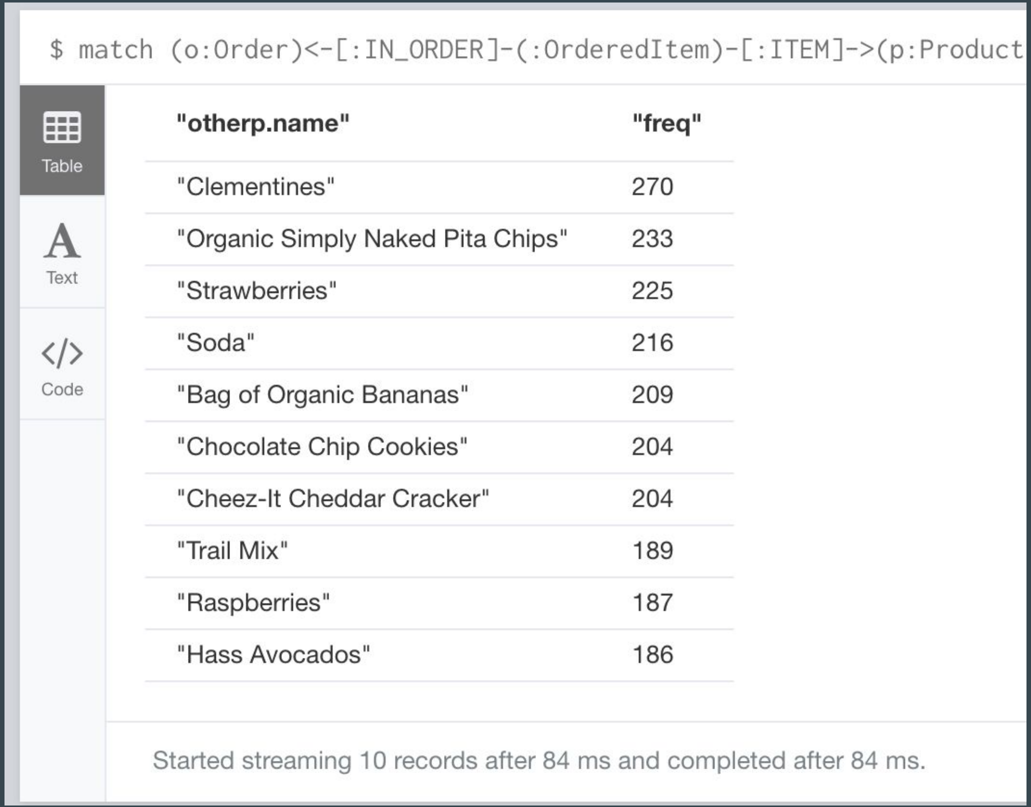
```
$ match (o:Order)<-[:IN_ORDER]-(oi:OrderedItem)-[opr:ITEM]->(p:Product …
```

"otherp.name"	"freq"
"Cabernet Sauvignon"	12
"India Pale Ale"	12
"Beer"	11
"Premium Belgian Lager"	8
"Belgian White Wheat Ale"	8
"Pinot Noir"	7
"Pinot Grigio"	7
"Chardonnay"	7
"Cabernet Sauvignon Wine"	5
"Belgian White Beer"	5

Started streaming 10 records after 38 ms and completed after 38 ms.

Recommendations

Problem w/ recommendations



```
$ match (o:Order)<-[:IN_ORDER]-(:OrderedItem)-[:ITEM]->(p:Product
```

Table

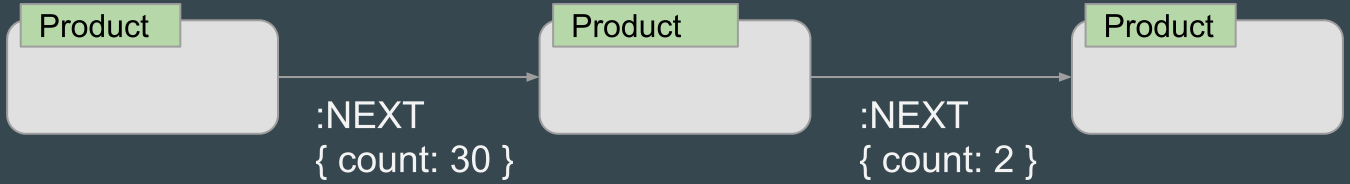
Text

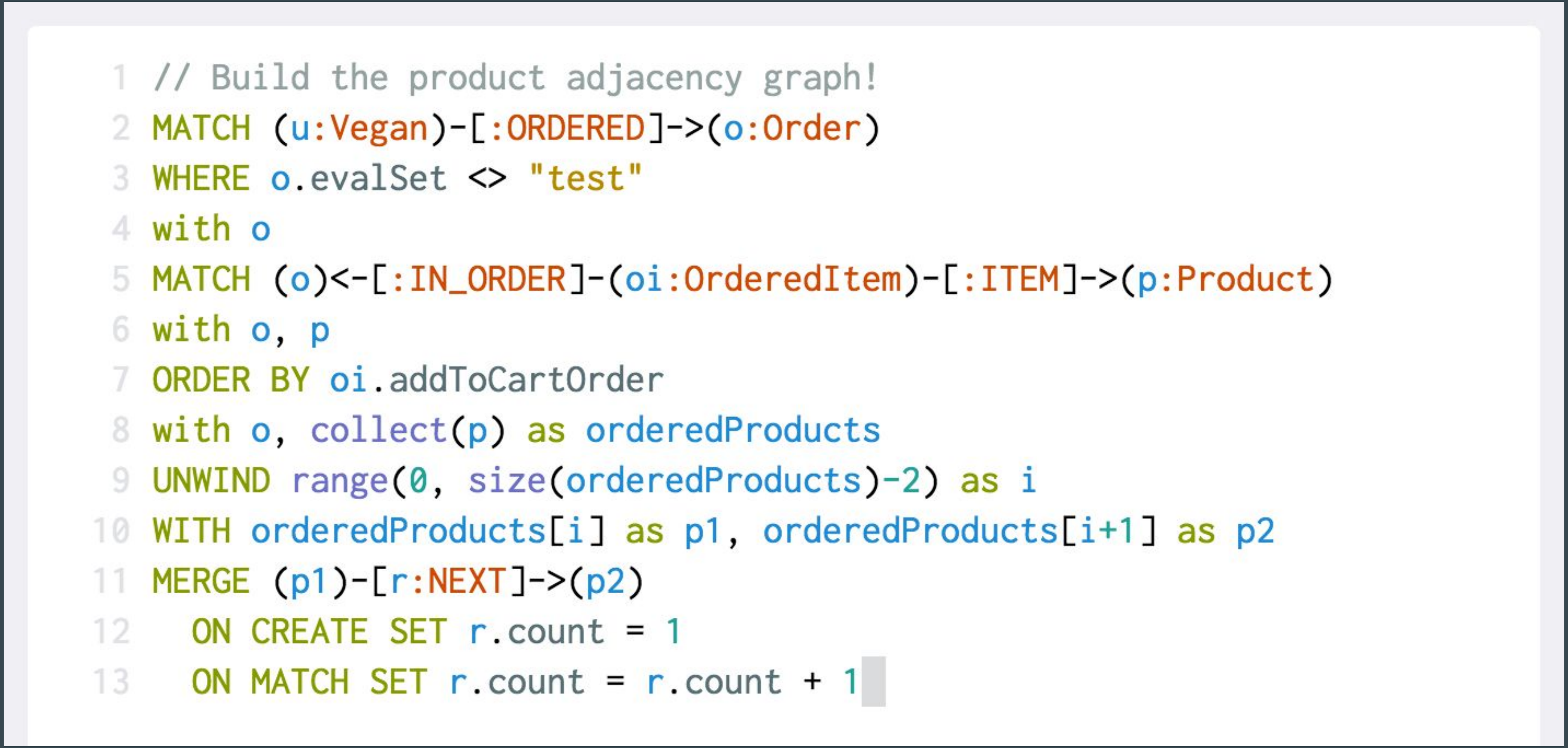
Code

"otherp.name"	"freq"
"Clementines"	270
"Organic Simply Naked Pita Chips"	233
"Strawberries"	225
"Soda"	216
"Bag of Organic Bananas"	209
"Chocolate Chip Cookies"	204
"Cheez-It Cheddar Cracker"	204
"Trail Mix"	189
"Raspberries"	187
"Hass Avocados"	186

Started streaming 10 records after 84 ms and completed after 84 ms.

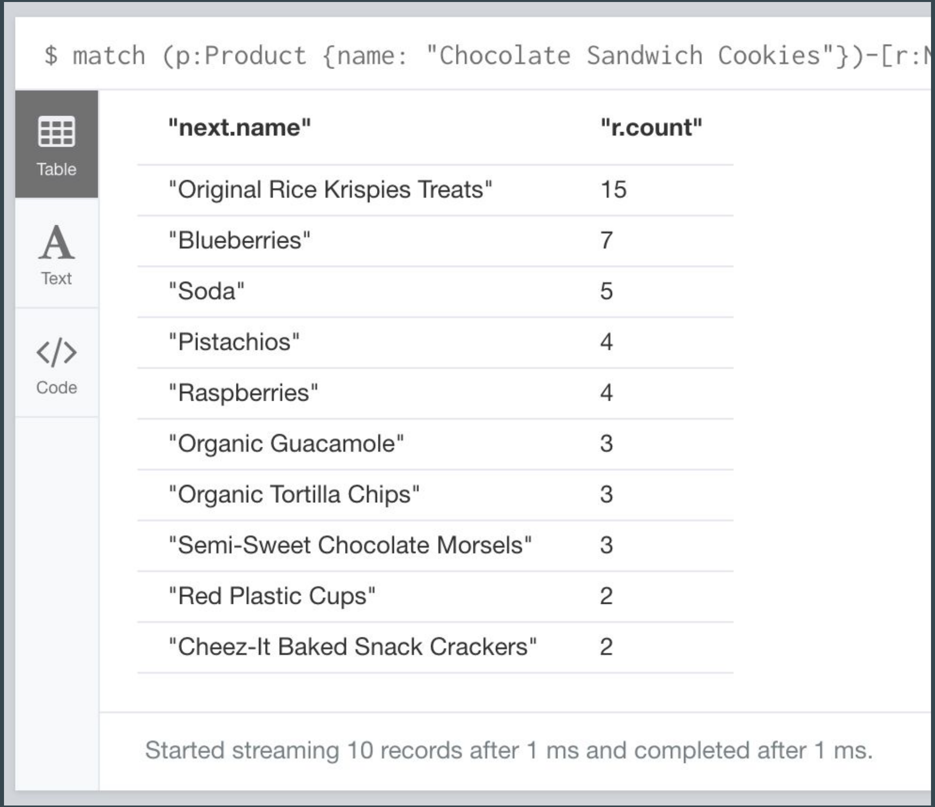
Product Adjacency





```
// Build the product adjacency graph!
MATCH (u:Vegan)-[:ORDERED]->(o:Order)
WHERE o.evalSet <> "test"
with o
MATCH (o)<-[:IN_ORDER]-(oi:OrderedItem)-[:ITEM]->(p:Product)
with o, p
ORDER BY oi.addToCartOrder
with o, collect(p) as orderedProducts
UNWIND range(0, size(orderedProducts)-2) as i
WITH orderedProducts[i] as p1, orderedProducts[i+1] as p2
MERGE (p1)-[r:NEXT]->(p2)
  ON CREATE SET r.count = 1
  ON MATCH SET r.count = r.count + 1
```

What to buy after Chocolate Cookies?

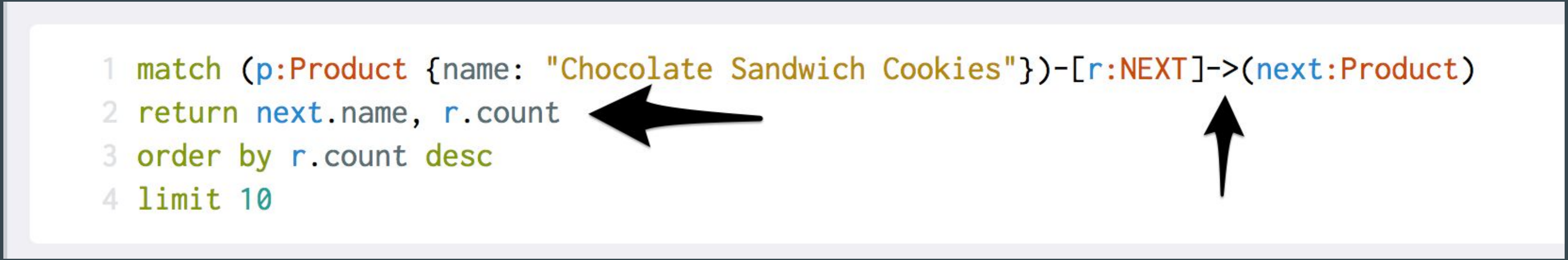


```
$ match (p:Product {name: "Chocolate Sandwich Cookies"})-[r:N
```

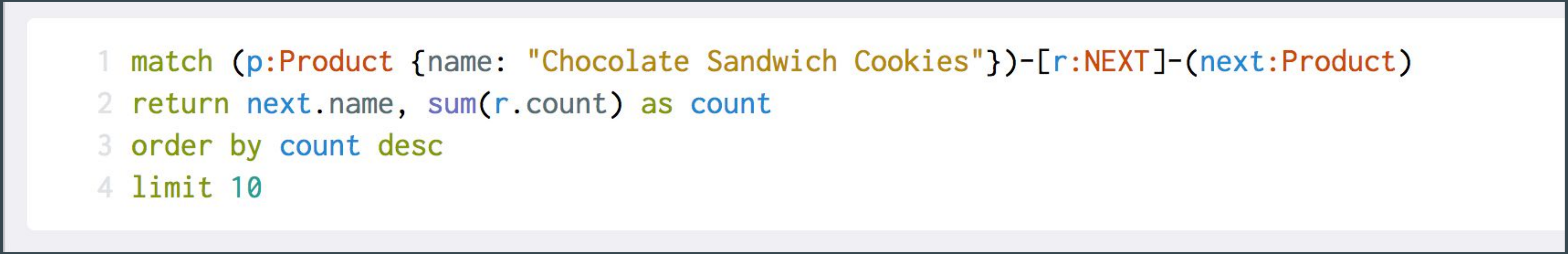
"next.name"	"r.count"
"Original Rice Krispies Treats"	15
"Blueberries"	7
"Soda"	5
"Pistachios"	4
"Raspberries"	4
"Organic Guacamole"	3
"Organic Tortilla Chips"	3
"Semi-Sweet Chocolate Morsels"	3
"Red Plastic Cups"	2
"Cheez-It Baked Snack Crackers"	2

Started streaming 10 records after 1 ms and completed after 1 ms.

What people buy around Chocolate Cookies?

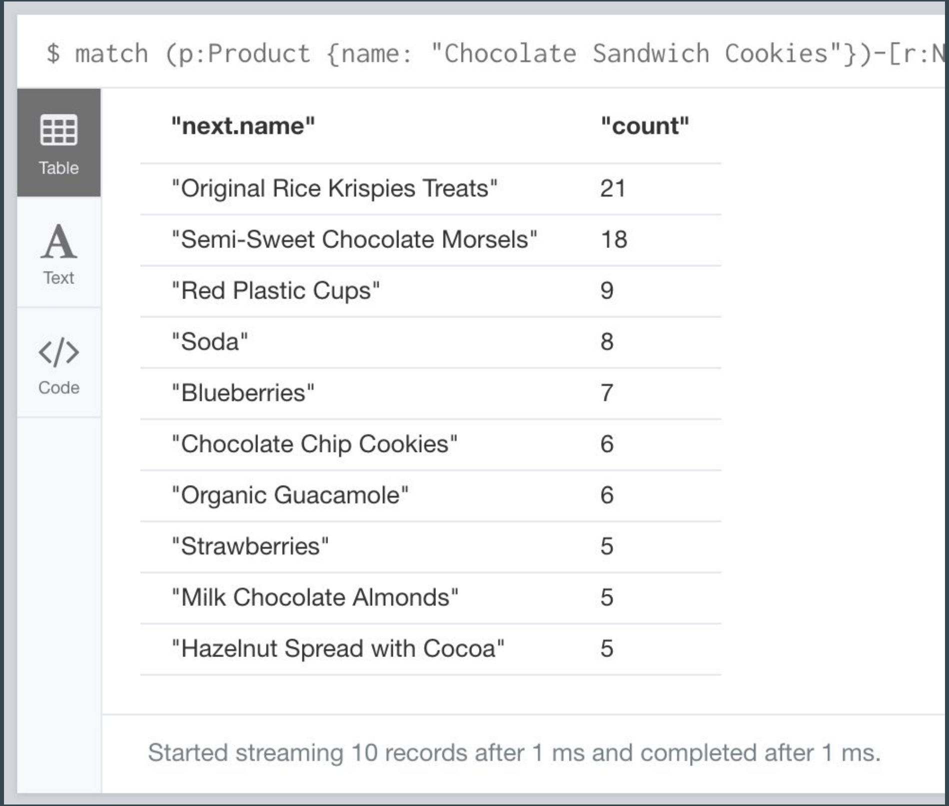


```
match (p:Product {name: "Chocolate Sandwich Cookies"})-[r:NEXT]->(next:Product)
return next.name, r.count
order by r.count desc
limit 10
```



```
match (p:Product {name: "Chocolate Sandwich Cookies"})-[r:NEXT]-(next:Product)
return next.name, sum(r.count) as count
order by count desc
limit 10
```

What people buy around Chocolate Cookies?



```
$ match (p:Product {name: "Chocolate Sandwich Cookies"})-[r:N
```

"next.name"	"count"
"Original Rice Krispies Treats"	21
"Semi-Sweet Chocolate Morsels"	18
"Red Plastic Cups"	9
"Soda"	8
"Blueberries"	7
"Chocolate Chip Cookies"	6
"Organic Guacamole"	6
"Strawberries"	5
"Milk Chocolate Almonds"	5
"Hazelnut Spread with Cocoa"	5

Started streaming 10 records after 1 ms and completed after 1 ms.

What Else?

What to people like to eat/drink when they're sick?

Match a person ordering frozen meals with someone ordering the ingredients in those meals

Can you find out if people have "unhealthy" habits? i.e., order flu medicine an bottle of whiskey

Match people who just ordered pasta and sauce with someone who just ordered a bottle of red wine

Find out if it's a single person, a couple, or a family

~~Data and Data Modeling~~

~~Mechanics of getting data into Neo4j~~

~~Dumpster Diving~~

Production Concerns (Sizing, import strategy)

Production Concerns

1. Hardware Sizing
2. Initial Data Load Strategy

Make sure you have the memory!

Data size on disk for FS cache + 8-16 GB JVM heap + 1 GB for OS and misc stuff

Total Memory Ballpark

1. Load a representative subset of the data
2. Extrapolate for full data size on disk

34,000 Orders (~1%)

Total Nodes: 412k

Total Rels: 1.5m

Total Props: 1.2m

Total Size on Disk: 333 MB

Extrapolated full set

Total Nodes: 41m

Total Rels: 150m

Total Props: 120m

Total Size on Disk: 33 GB

Total Memory Ballpark

Data size on disk for FS cache + 8 - 16 GB Java heap + 1 GB for OS and misc stuff

33 GB + 16 GB + 1GB = 50GB

m4.4xl with 64GB of memory for $0.80/hour

Extrapolated full set

Total Nodes: 41m -> 40.7m

Total Rels: 150m -> 71m

Total Props: 120m -> 128m

Total Size on Disk: 33 GB -> 5.18GB

Importing a larger dataset

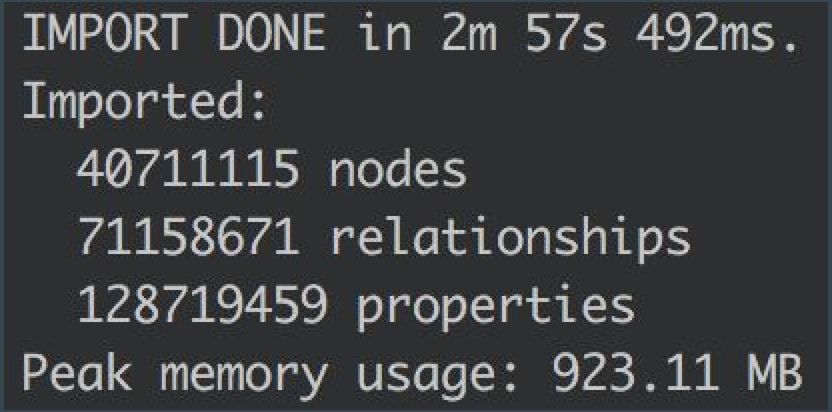
LOAD CSV

Importing a larger dataset

~~LOAD CSV~~

neo4j-import

```
neo4j-import --into instacart.db --id-type string \
  --skip-duplicate-nodes true \
  --bad-tolerance 3300000 \
  --nodes:User users_headers.csv,orders.csv \
  --nodes:Department departments.csv \
  --nodes:Aisle aisles.csv \
  --nodes:Product products_headers.csv,products.csv \
  --nodes:Order orders_headers.csv,orders.csv \
  --nodes:OrderedItem order_products_headers.csv,order_products__prior_id.csv,order_products__train_id.csv \
  --relationships:ON products_aisles_headers.csv,products.csv \
  --relationships:IN products_departments_headers.csv,products.csv \
  --relationships:ORDERED users_orders_headers.csv,orders.csv \
  --relationships:IN_ORDER order_products_order_headers.csv,order_products__prior_id.csv,order_products__train_id.csv \
  --relationships:ITEM order_products_products_headers.csv,order_products__prior_id.csv,order_products__train_id.csv
```



```
IMPORT DONE in 2m 57s 492ms.
Imported:
  40711115 nodes
  71158671 relationships
  128719459 properties
Peak memory usage: 923.11 MB
```

Thanks!

Questions?

Resources

Instagram Blog Post

https://tech.instacart.com/3-million-instacart-orders-open-sourced-d40d29ead6f2

Data Set

https://www.instacart.com/datasets/grocery-shopping-2017

Neo4j Hardware Sizing

https://neo4j.com/news/video-hardware-sizing-for-neo4j/

Data Citation

"The Instacart Online Grocery Shopping Dataset 2017", Accessed from https://www.instacart.com/datasets/grocery-shopping-2017 on May 7th, 2017