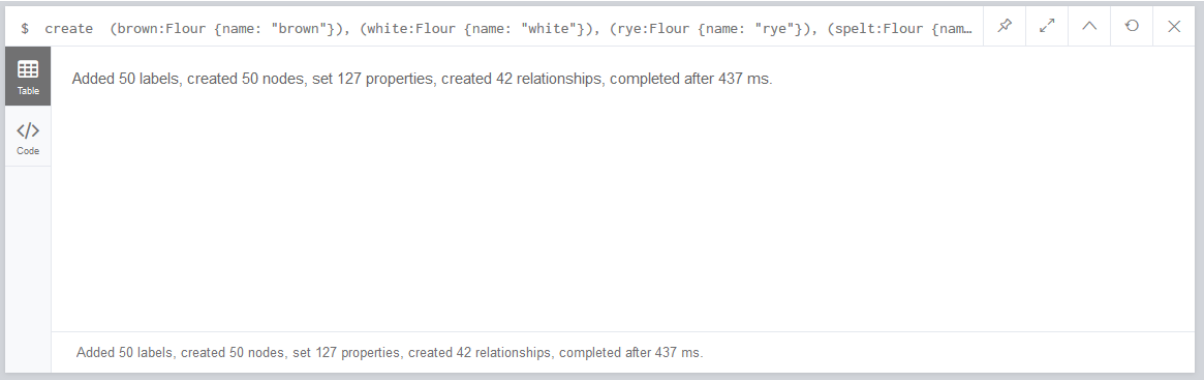
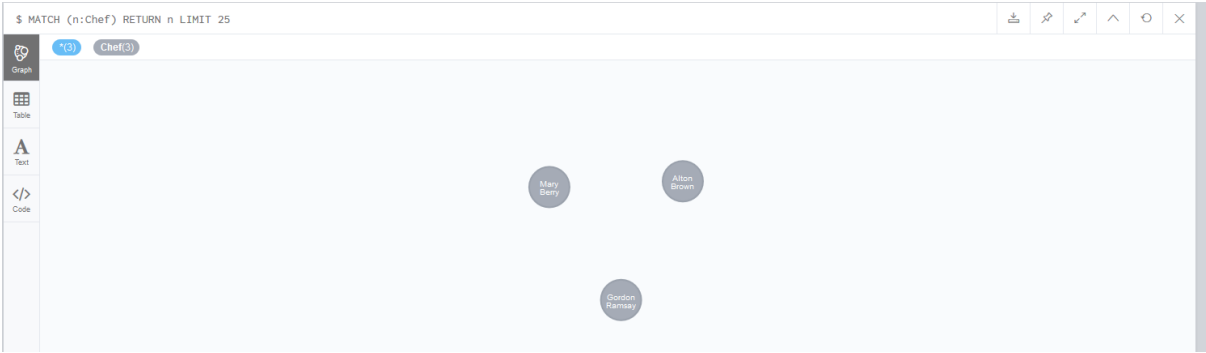
/* PART 2 */

// Initial script



// Create 3 chefs

```
CREATE
        (gordon:Chef { name: 'Gordon Ramsay', title: 'Chef' }),
        (alton:Chef { name: 'Alton Brown', title: 'Chef' }),
        (mary:Chef { name: 'Mary Berry', title: 'Chef' })
```



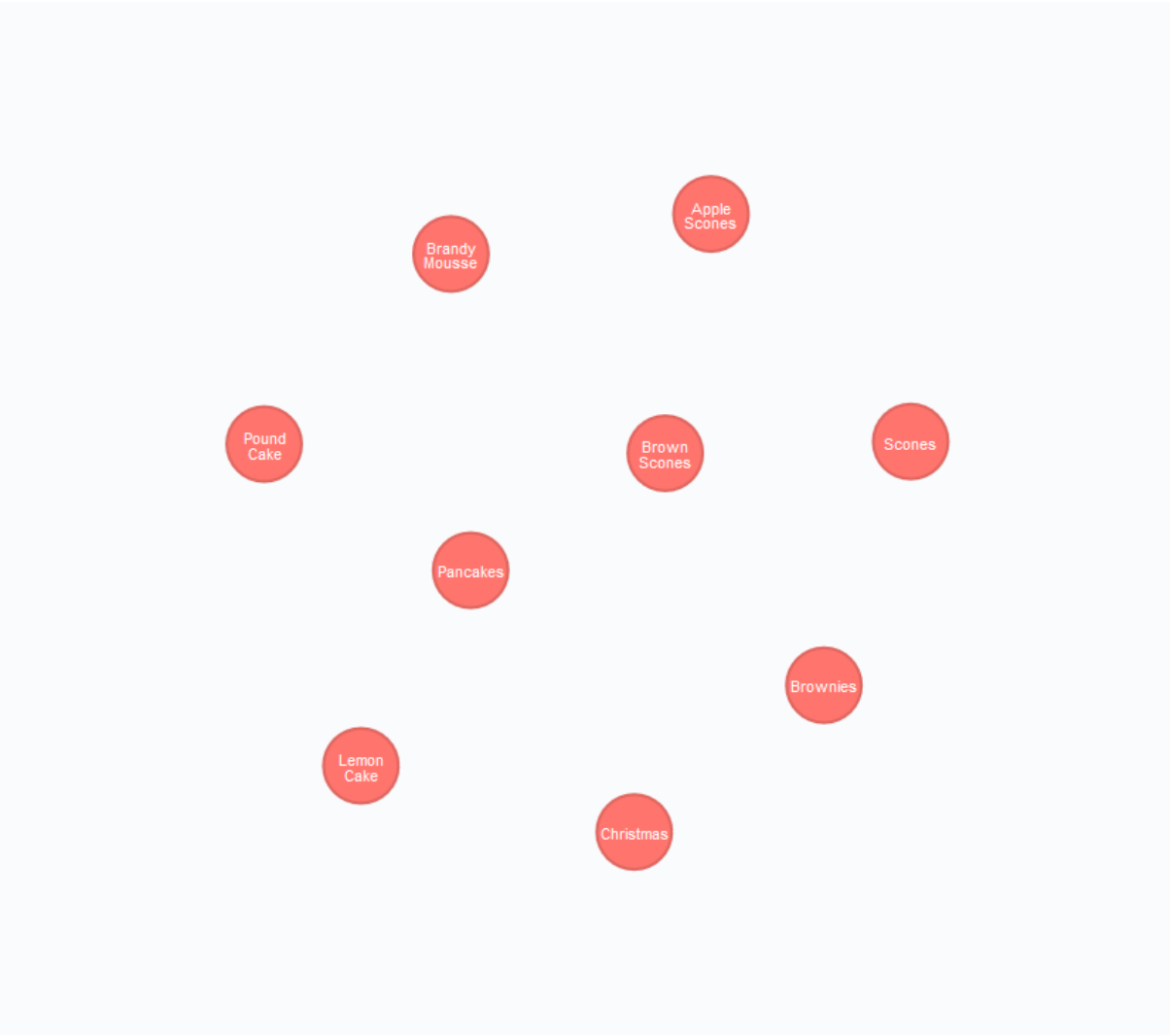
```
// Creating recipes

        CREATE

         // Pound Cake
         (poundCake:Cake {name: "Pound Cake"}),
         (poundCake)-[:CONTAINS {quantity: 400, unit: "grams"}]->(selfraising),
         (poundCake)-[:CONTAINS {quantity: .5, unit: "grams"}]->(salt),
         (poundCake)-[:CONTAINS {quantity: 160, unit: "grams"}]->(sugar),
         (poundCake)-[:CONTAINS {quantity: 300, unit: "grams"}]->(milk),
         (poundCake)-[:CONTAINS {quantity: 4, unit: "grams"}]->(egg),
         (poundCake)-[:GARNISHED_WITH {how: "poured on top"} ]->(cream),


         // Pancakes
         (pancakes:Cake {name: "Pancakes"}),
         (pancakes)-[:CONTAINS {quantity: 100, unit: "grams"}]->(white),
         (pancakes)-[:CONTAINS {quantity: 150, unit: "grams"}]->(brown),
         (pancakes)-[:CONTAINS {quantity: .5, unit: "grams"}]->(salt),
         (pancakes)-[:CONTAINS {quantity: 80, unit: "grams"}]->(sugar),
         (pancakes)-[:CONTAINS {quantity: 300, unit: "grams"}]->(milk),
         (pancakes)-[:CONTAINS {quantity: 1, unit: "grams"}]->(egg),
         (pancakes)-[:GARNISHED_WITH {how: "sprinkled on top"} ]->(sugar),


         // Lemon Cake
         (lemonCake:Cake {name: "Lemon Cake"}),
         (lemonCake)-[:CONTAINS {quantity: 250, unit: "grams"}]->(selfraising),
         (lemonCake)-[:CONTAINS {quantity: 80, unit: "grams"}]->(margarine),
         (lemonCake)-[:CONTAINS {quantity: .5, unit: "grams"}]->(salt),
         (lemonCake)-[:CONTAINS {quantity: 125, unit: "grams"}]->(sugar),
         (lemonCake)-[:CONTAINS {quantity: 100, unit: "grams"}]->(currants),
         (lemonCake)-[:CONTAINS {quantity: 100, unit: "grams"}]->(sultanas),
         (lemonCake)-[:CONTAINS {quantity: 25, unit: "grams"}]->(brandy),
         (lemonCake)-[:CONTAINS {quantity: 3, unit: "grams"}]->(lemon),
         (lemonCake)-[:CONTAINS {quantity: .5, unit: "grams"}]->(mixedspice)
```

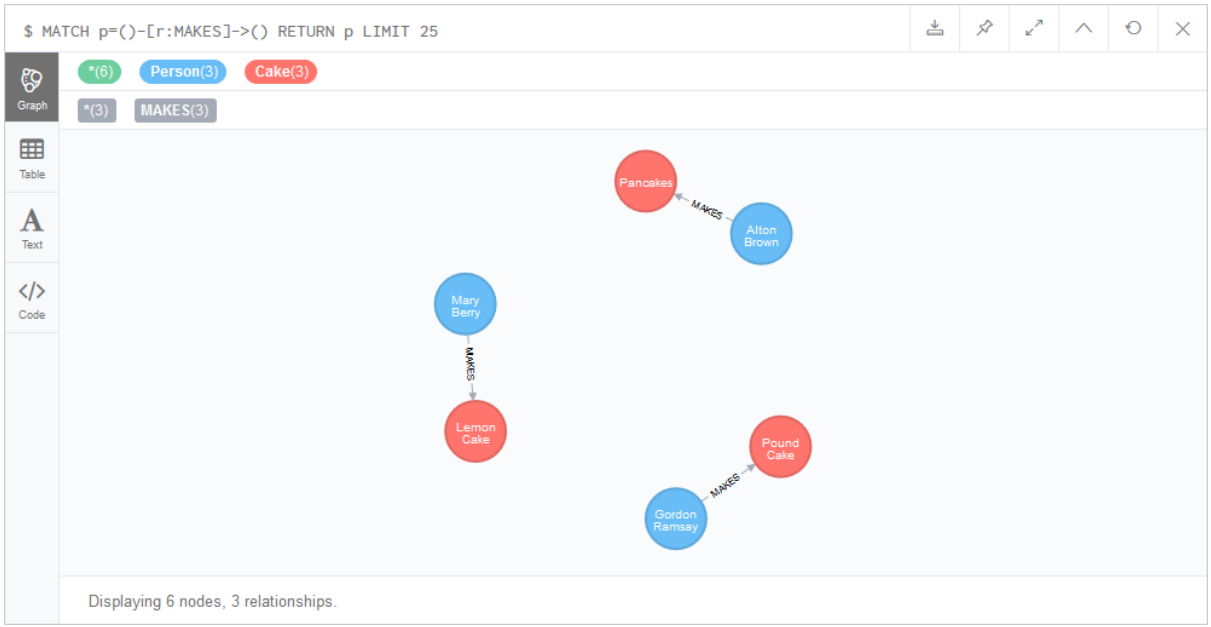


// Creating relationships between chefs and their recipes with popularity property

```
MATCH (gordon:Chef),(poundCake:Cake)
WHERE gordon.name = 'Gordon Ramsay' AND poundCake.name = 'Pound Cake'
CREATE (gordon)-[r:MAKES {Popularity : 6}]->(poundCake)
RETURN gordon, poundCake;
```

```
MATCH (alton:Chef),(pancakes:Cake)
WHERE alton.name = 'Alton Brown' AND pancakes.name = 'Pancakes'
CREATE (alton)-[r:MAKES {Popularity : 9}]->(pancakes)
RETURN alton, pancakes;
```

```
MATCH (mary:Chef),(lemonCake:Cake)
WHERE mary.name = 'Mary Berry' AND lemonCake.name = 'Lemon Cake'
CREATE (mary)-[r:MAKES {Popularity : 7}]->(lemonCake)
RETURN mary, lemonCake;
```



// Match existing cakes to chefs

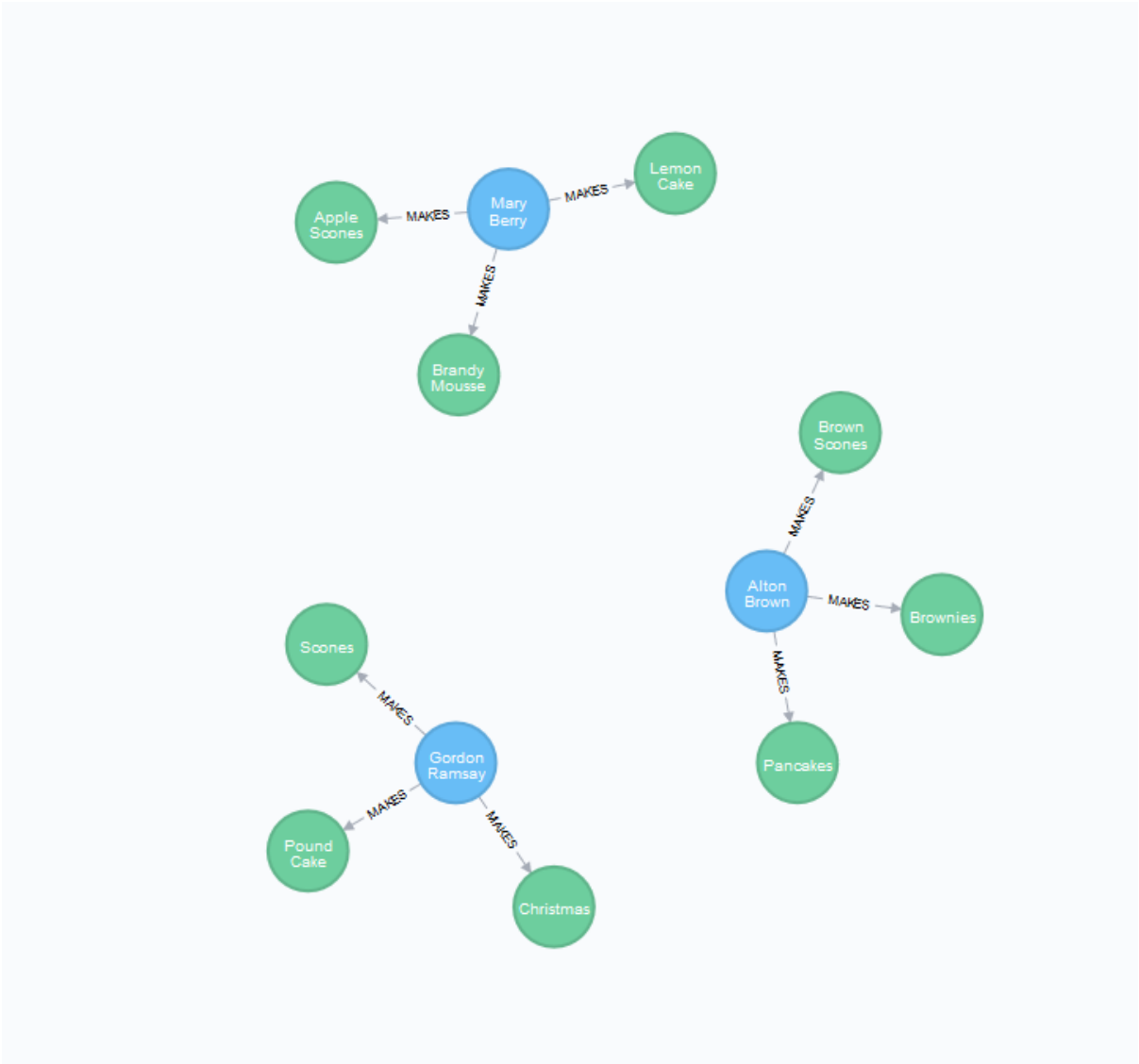
```
MATCH (gordon:Chef),(scones:Cake)
WHERE gordon.name = 'Gordon Ramsay' AND scones.name = 'Scones'
CREATE (gordon)-[r:MAKES {Used : 150}]->(scones)
RETURN gordon, scones;
```

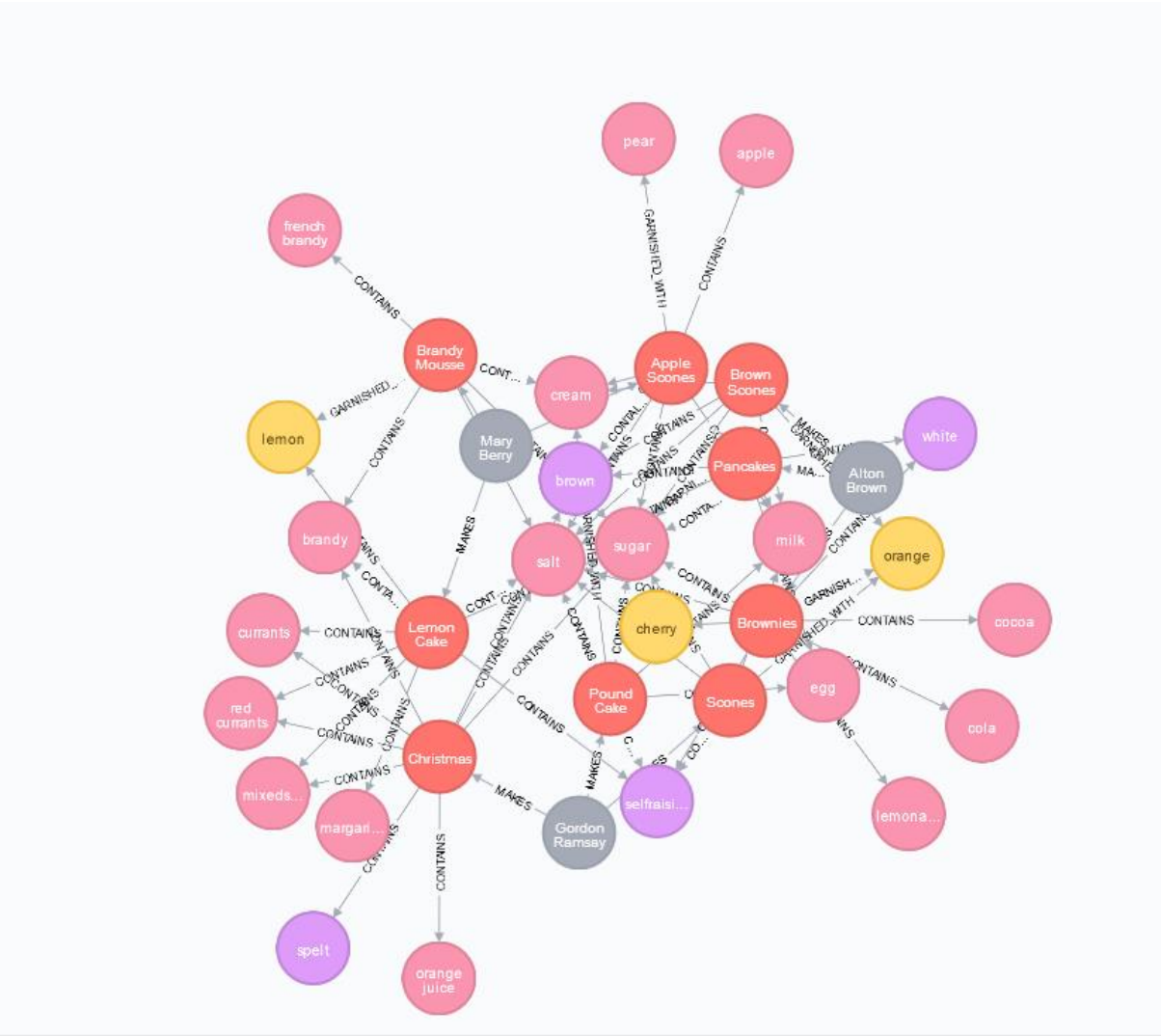
```
MATCH (alton:Chef),(brownies:Cake)
WHERE alton.name = 'Alton Brown' AND brownies.name = 'Brownies'
CREATE (alton)-[r:MAKES {Used : 210}]->(brownies)
RETURN alton, brownies;
```

```
MATCH (alton:Chef),(brownscones:Cake)
WHERE alton.name = 'Alton Brown' AND brownscones.name = 'Brown Scones'
CREATE (alton)-[r:MAKES {Used : 210}]->(brownscones)
RETURN alton, brownscones;
```

```
MATCH (mary:Chef),(applescones:Cake)
WHERE mary.name = 'Mary Berry' AND applescones.name = 'Apple Scones'
CREATE (mary)-[r:MAKES {Used : 90}]->(applescones)
RETURN mary, applescones;
```

```
MATCH (mary:Chef),(bmousse:Cake)
WHERE mary.name = 'Mary Berry' AND bmousse.name = 'Brandy Mousse'
CREATE (mary)-[r:MAKES {Used : 90}]->(bmousse)
RETURN mary, bmousse;
```


Mary Berry
Apple Scones
Lemon Cake
Brandy Mousse
MAKES
MAKES
MAKES
Alton Brown
Brown Scones
Brownies
Pancakes
MAKES
MAKES
MAKES
Gordon Ramsay
Scones
Pound Cake
Christmas
MAKES
MAKES
MAKES



// Q2. List all of the cakes and the ingredients for each cake.

```
MATCH (c:Cake)-[r:CONTAINS]->(i)
RETURN c.name, i.name
```

c.name	i.name
"Christmas Cake"	"red currants"
"Christmas Cake"	"brandy"
"Christmas Cake"	"orange juice"
"Christmas Cake"	"mixedspice"
"Christmas Cake"	"brown"
"Christmas Cake"	"salt"
"Christmas Cake"	"sugar"
"Christmas Cake"	"currants"
"Christmas Cake"	"spelt"
"Brownies"	"cocoa"
"Brownies"	"lemonade"
"Brownies"	"cola"
"Brownies"	"white"
"Brownies"	"selfraising"
"Brownies"	"salt"
"Brownies"	"sugar"
"Scones"	"sugar"
"Scones"	"salt"
"Scones"	"selfraising"
"Scones"	"milk"
"Brown Scones"	"cream"
"Brown Scones"	"milk"
"Brown Scones"	"sugar"
"Brown Scones"	"salt"
"Brown Scones"	"brown"
"Apple Scones"	"cream"
"Apple Scones"	"apple"
"Apple Scones"	"milk"
"Apple Scones"	"sugar"
"Apple Scones"	"salt"

// Q3. List the ingredients in a brownie recipe.

```
MATCH (c:Cake{name:'Brownies'})-[r:CONTAINS]->(i)
RETURN c.name, i.name
```

c.name	i.name
"Brownies"	"cocoa"
"Brownies"	"lemonade"
"Brownies"	"cola"
"Brownies"	"white"
"Brownies"	"selfraising"
"Brownies"	"salt"
"Brownies"	"sugar"

// Q4. List all the cakes with flour in them.

```
MATCH (c:Cake)-[r:CONTAINS]->(i:Flour)
RETURN c.name, i.name
```

c.name	i.name
"Pancakes"	"brown"
"Apple Scones"	"brown"
"Brown Scones"	"brown"
"Christmas Cake"	"brown"
"Pancakes"	"white"
"Brownies"	"white"
"Christmas Cake"	"spelt"
"Lemon Cake"	"selfraising"
"Pound Cake"	"selfraising"
"Scones"	"selfraising"
"Brownies"	"selfraising"

// Q5. List which Cakes have both Cream and Milk in them.

```
MATCH (c:Cake)-[r:CONTAINS]->(i:Ingredient)
WHERE i.name = 'cream' OR i.name = 'milk'
RETURN c.name, i.name
ORDER BY c.name
```

c.name	i.name
"Apple Scones"	"milk"
"Apple Scones"	"cream"
"Brandy Mousse"	"cream"
"Brown Scones"	"milk"
"Brown Scones"	"cream"
"Pancakes"	"milk"
"Pound Cake"	"milk"
"Scones"	"milk"

// Q6. List which recipes are associated with each chef.

```
MATCH (p:Chef)-[]->(c:Cake)
RETURN p.name, c.name
```

p.name	c.name
"Gordon Ramsay"	"Scones"
"Gordon Ramsay"	"Christmas Cake"
"Gordon Ramsay"	"Pound Cake"
"Alton Brown"	"Brown Scones"
"Alton Brown"	"Brownies"
"Alton Brown"	"Pancakes"
"Mary Berry"	"Brandy Mousse"
"Mary Berry"	"Apple Scones"
"Mary Berry"	"Lemon Cake"

// Q7. What chef has the most popular recipes?

```
MATCH (p:Chef)-[q]->()
WHERE q.Popularity IS NOT NULL
RETURN p.name, q.Popularity
ORDER BY q.Popularity DESC
```

p.name	q.Popularity
"Alton Brown"	9
"Mary Berry"	7
"Gordon Ramsay"	6

// Q8. What are the 5 most useful ingredients?

```
MATCH (c:Cake)-[r]->(i)
RETURN count(r) AS num, i.name
ORDER BY num DESC
LIMIT 5
```

num	i.name
10	"sugar"
9	"salt"
5	"milk"
4	"cream"
4	"selfraising"

// Q9. What are the 5 least useful ingredients?

```
MATCH (c:Cake)-[r]->(i)
RETURN count(r) AS num, i.name
ORDER BY num
LIMIT 5
```

num	i.name
1	"cola"
1	"cocoa"
1	"pear"
1	"cherry"
1	"french brandy"

// Q10. What recipe is the most similar to apple scones - output list of ingredients from other recipes in order of similarity to this recipe?

```
MATCH (a:Cake{name:'Apple Scones'})-[r]->(b), (c:Cake)-[d]->(e)
WHERE b = e
RETURN a.name, c.name, count(r) AS num
ORDER BY num DESC
```

a.name	c.name	num
"Apple Scones"	"Pancakes"	5
"Apple Scones"	"Brown Scones"	5
"Apple Scones"	"Pound Cake"	4
"Apple Scones"	"Christmas Cake"	3
"Apple Scones"	"Scones"	3
"Apple Scones"	"Brandy Mousse"	3
"Apple Scones"	"Brownies"	2
"Apple Scones"	"Lemon Cake"	2

/* PART 3 */

// Q1. What kind of cuisine do New Yorkers prefer?

```
> db.restaurants.aggregate([{$project:{_id:0, cuisine:1}},{$sortByCount:"$cuisine"}]).pretty()
{ "_id" : "American", "count" : 6183 }
{ "_id" : "Chinese", "count" : 2418 }
{ "_id" : "Café/Coffee/Tea", "count" : 1214 }
{ "_id" : "Pizza", "count" : 1163 }
{ "_id" : "Italian", "count" : 1069 }
{ "_id" : "Other", "count" : 1011 }
{
        "_id" : "Latin (Cuban, Dominican, Puerto Rican, South & Central American)",
        "count" : 850
}
{ "_id" : "Japanese", "count" : 760 }
{ "_id" : "Mexican", "count" : 754 }
{ "_id" : "Bakery", "count" : 691 }
{ "_id" : "Caribbean", "count" : 657 }
{ "_id" : "Spanish", "count" : 637 }
{ "_id" : "Donuts", "count" : 479 }
{ "_id" : "Pizza/Italian", "count" : 468 }
{ "_id" : "Sandwiches", "count" : 459 }
{ "_id" : "Hamburgers", "count" : 433 }
{ "_id" : "Chicken", "count" : 410 }
{ "_id" : "Ice Cream, Gelato, Yogurt, Ices", "count" : 348 }
{ "_id" : "French", "count" : 344 }
{ "_id" : "Delicatessen", "count" : 321 }
Type "it" for more
>
```

// Q2. Which area represents the biggest market opportunity for opening a new restaurant of this kind of cuisine?

```
> db.restaurants.aggregate([{$match:{cuisine:"American"}},{$project:{_id:0, cuisine:1, borough:1}},{$sortByCount:"$borough"}]).pretty()
{ "_id" : "Manhattan", "count" : 3205 }
{ "_id" : "Brooklyn", "count" : 1273 }
{ "_id" : "Queens", "count" : 1040 }
{ "_id" : "Bronx", "count" : 411 }
{ "_id" : "Staten Island", "count" : 244 }
{ "_id" : "Missing", "count" : 10 }
>
```

???

```
> db.restaurants.aggregate([{$match:{cuisine:"American"}},{$project:{_id:0, cuisine:1, borough:1, zipcode:1}},{$sortByCount:"$zipcode"}]).pretty()
{ "_id" : null, "count" : 6183 }
>
```

// Q3. Who are the biggest competitors in this area?

???