

Team FoodScape

“Find Your Food”



Paul Cabasag, Adelle Driker, Bo Yan

DSE 203: Data Integration and ETL

Introduction

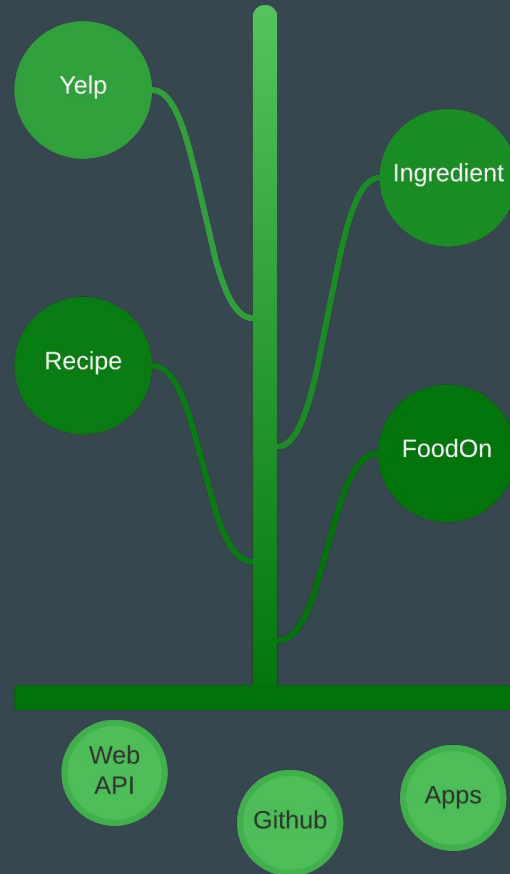
- What is “Find Your Food”?
 - “Find Your Food” is a comprehensive and easily accessible ontology about food, recipes, ingredients, restaurants as well as the diets, menus, seasons and occasions users may be suitable for.
 - We created a simple lightweight ontology that uses the shared terminology for types, properties and relationships about food concepts, and thus can help users to find the best food.

Information Integration Problem

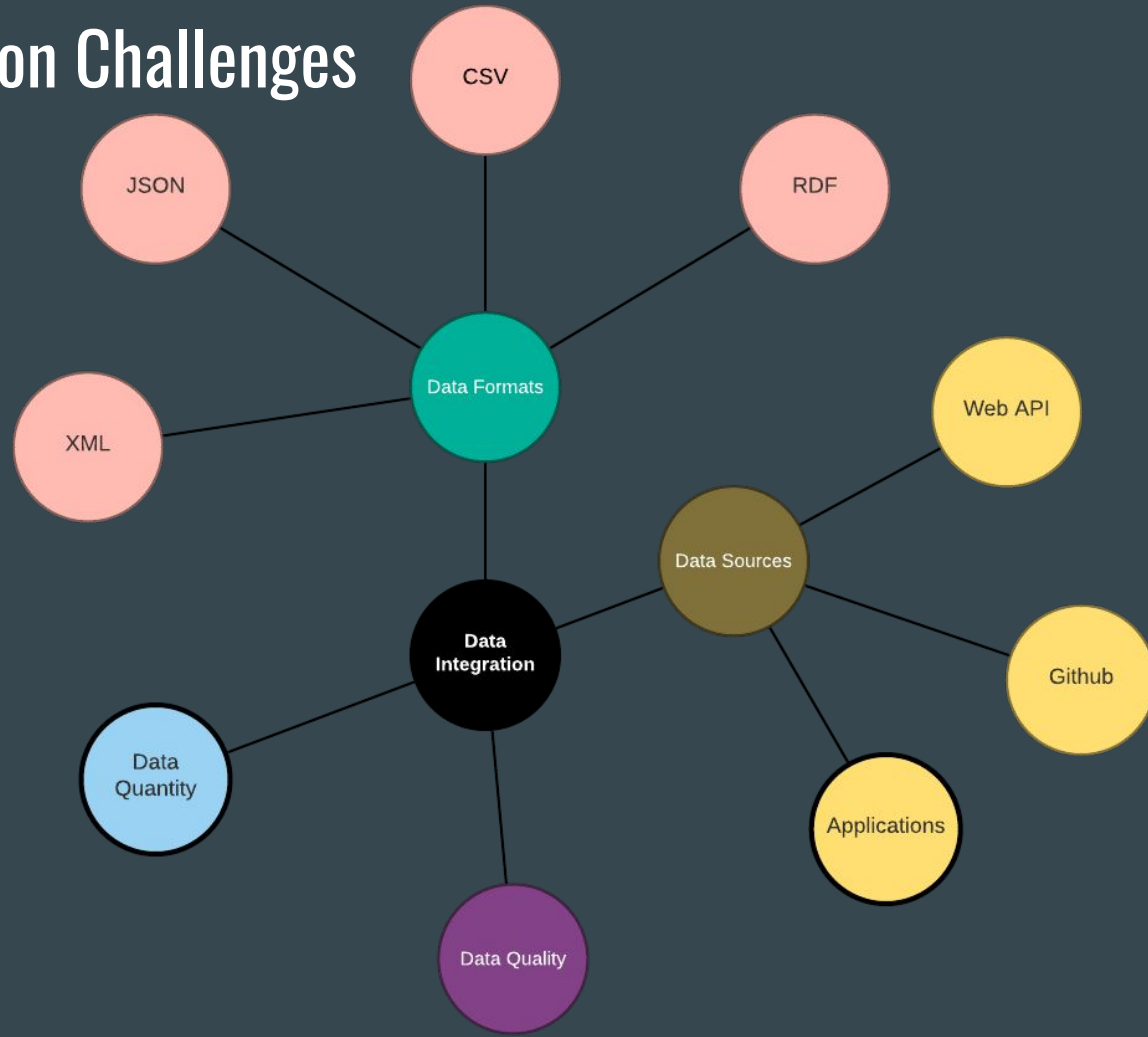
- Eat-Out: Provide a restaurant recommendation query tool depending on personal nutrients, ingredients, dietary, instructions, and cost preferences.
- Eat-In: Provide “cookbook” recommendation queries based on food recipe/ingredient.

Data Sources

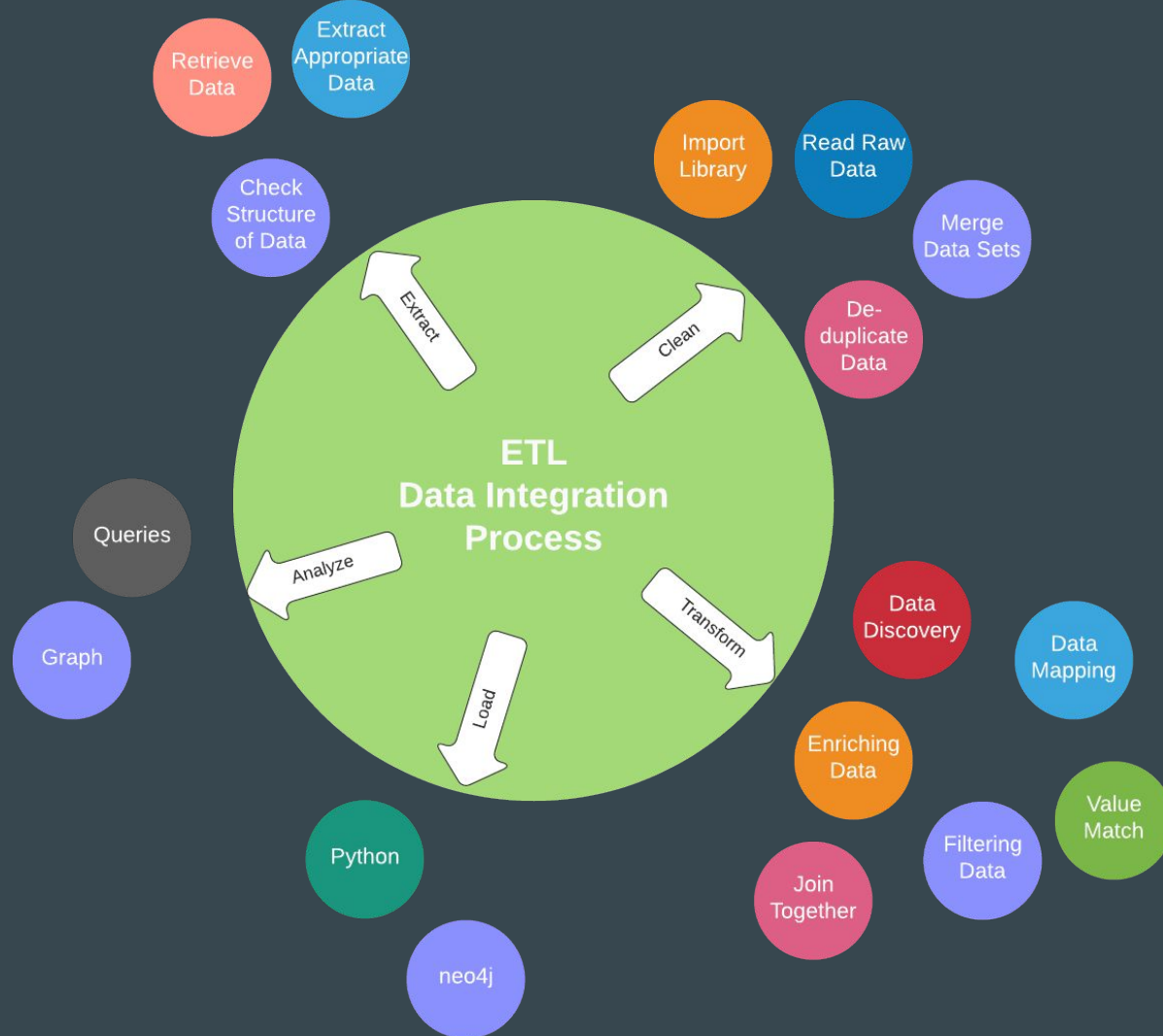
- FoodOn
 - <https://github.com/FoodOntology/foodon>
 - Based on a conversion of the LanguaL.org food indexing thesaurus
 - Over 9,000 food products available
 - Encompasses materials in natural ecosystems, food webs, and human-centric categorization and handling of food.
- Recipe
 - <https://rapidapi.com/spoonacular/api/recipe-food-nutrition/>
 - Over 1,000 recipes data available
- Ingredient
 - https://raw.githubusercontent.com/foodkg/foodkg.github.io/master/ontologies/WhatToMake_Individuals.rdf
 - Over 200 food ingredients data available
- Yelp
 - <https://www.yelp.com/dataset/documentation/main>
 - Over 160,000 restaurants data available



Data Integration Challenges



Approach



Data Extraction and Cleaning

- Modules Used
 - Py2Neo
 - NLTK
- Data Preparation
 - Clean FoodOn Labels to exclude inedible items (plastic, metal, chemicals), scientific/Latin names
- Noun Extraction
 - All foods/ingredients are assumed to be Nouns
 - Utilize NLTK's text preprocessing functions
 - E.g. "The steak and salad hit the spot!" → ['steak', 'salad', 'spot']

Data Transformation and Combining

- String Matching
 - For all Nouns in a Yelp Tip or Recipe Summary
 - Compare against list of FoodOn or Ingredient items
 - E.g. ['steak', 'salad', 'spot'] → ['steak', 'salad']
- Attribute Construction (Update Node Properties)
 - Create a new property that contains a list of matched foods → used to create new edges
- Data Combining
 - Use Append, Merge, and Join to combine data

Structure of Knowledge Graph: Nodes

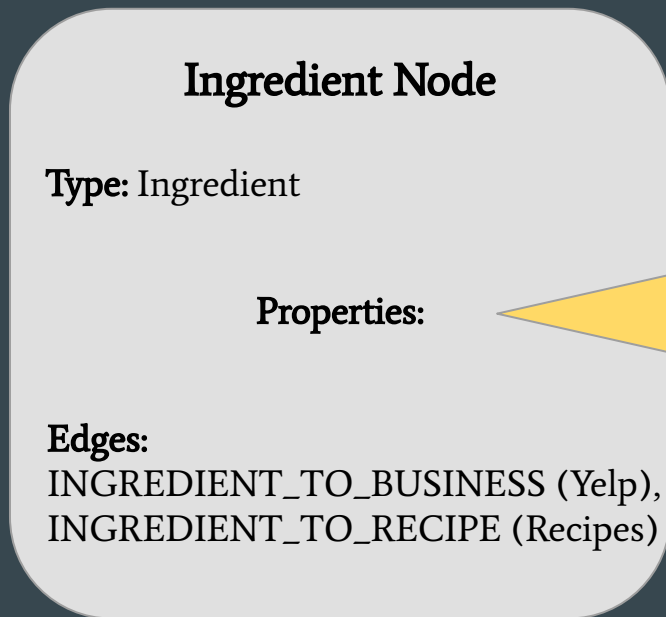
FoodOn Node

Type: owl__class

Properties: rdfs__label

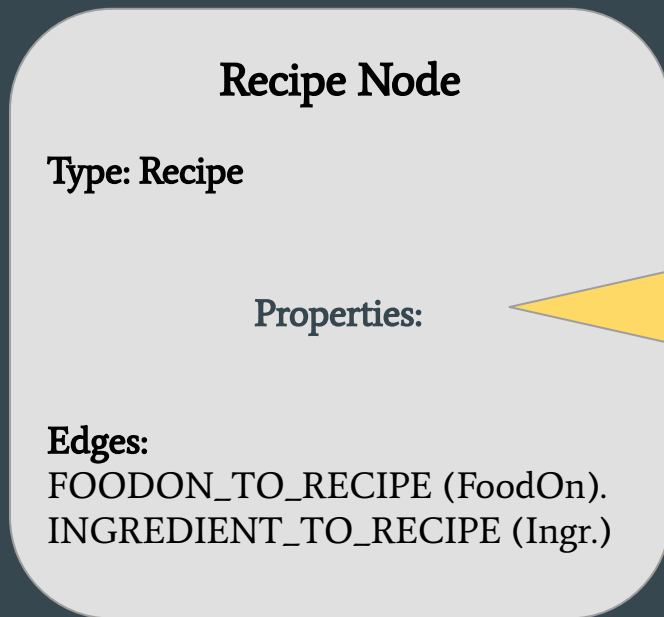
Edges:
FOODON_TO_BUSINESS (Yelp),
FOODON_TO_RECIPE (Recipes)

Structure of Knowledge Graph: Nodes



Ingredient
Ingredient_id *
Ingredient_Name
Ingredient_Num_Measurement
Ingredient_Unit_Measurement
Recipe_id

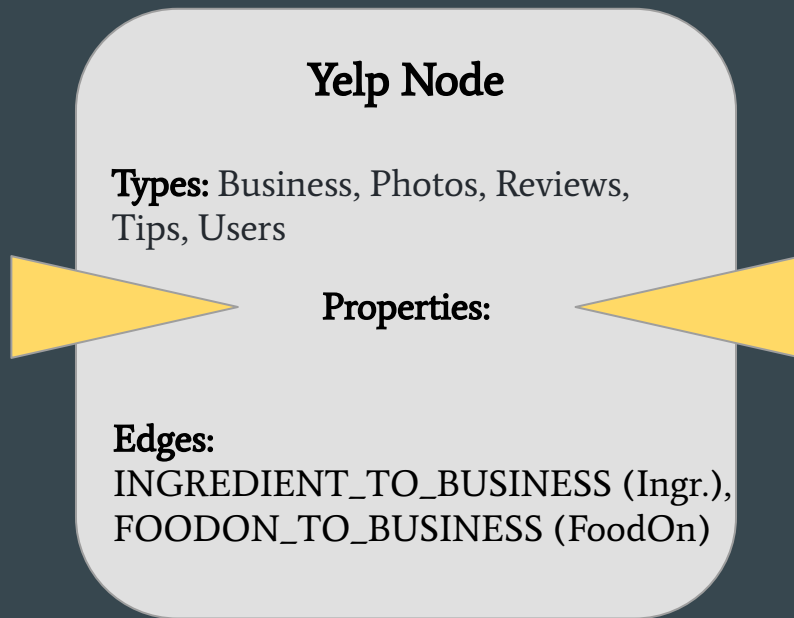
Structure of Knowledge Graph: Nodes



Recipe
Recipe_id *
Title
Summary
ReadyInMinutes
PricePerServing
GlutenFree
DairyFree
Vegan
...

Structure of Knowledge Graph: Nodes

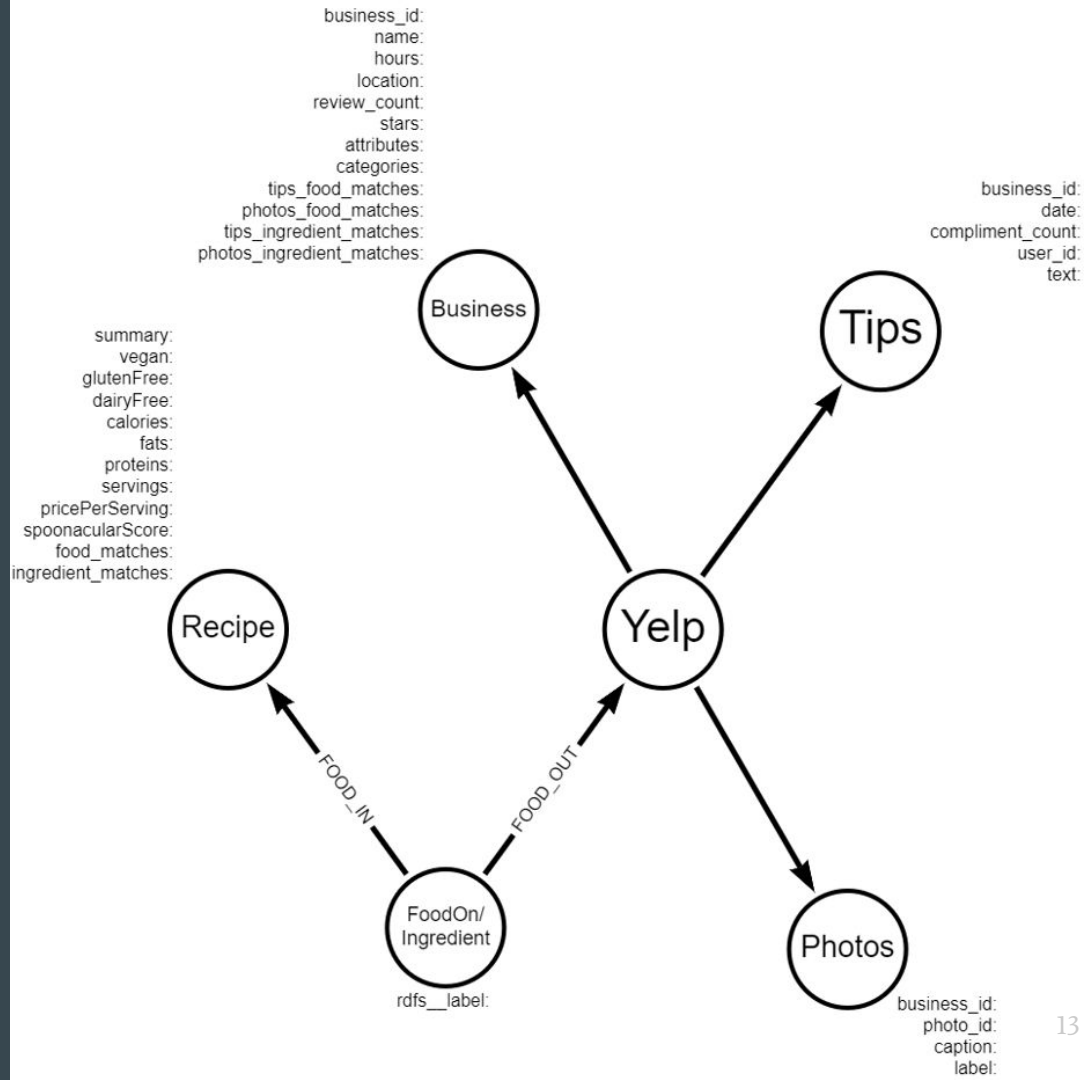
Tips
Business_id *
Date
Compliment_Count
User_id
Text
Photos
Business_id *
Photo_id
Caption
Label



* = Primary Key
** = Inserted Property

Business
Business_id *
Name
Hours
Location (address, city, latitude, longitude, state, postal code)
Review_count
Stars
Attributes (e.g. Happy Hour)
Categories (e.g. Food)
Tips_food_matches, Photos_food_matches **
Tips_ingredient_matches, Photos_ingredient_matches **

Structure of Knowledge Graph (Connected)



Query 1 (Cypher):

Question: What are the highly-rated restaurants (at least 4.5 stars) and recipes (at least 60%) allowed for people that are diabetic (no sugar) and are available as a breakfast?

Cypher:

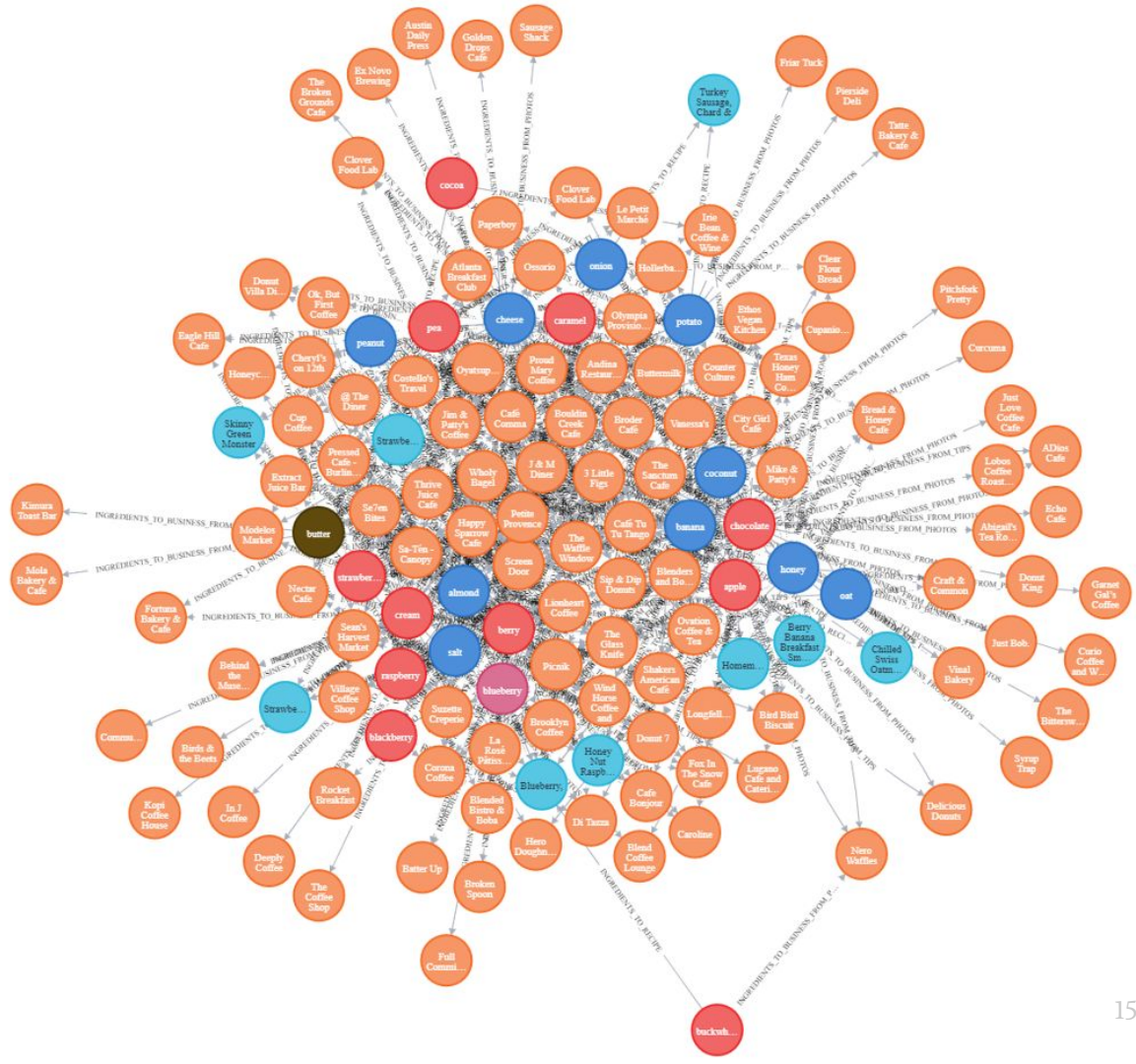
MATCH

```
(c:Business)<-[s:INGREDIENTS_TO_BUSINESS_FROM_PHOTOS]-(a:Ingredients)-[r:INGREDIENTS_TO_RECIPE]->(b:Recipe)
```

```
WHERE (a.rdfs__label<>'sugar' AND b.ingredient_matches CONTAINS a.rdfs__label) AND  
(a.rdfs__label<>'sugar' AND c.photos_ingredient_matches CONTAINS a.rdfs__label) AND  
(b.dishTypes_0='breakfast' OR b.dishTypes_1='breakfast' OR b.dishTypes_2='breakfast' OR  
b.dishTypes_3='breakfast') AND (c.categories CONTAINS 'Breakfast') AND (c.stars>=4.5) AND  
(b.spoonacularScore>=60)
```

RETURN *

Query 1 (Knowledge Graph):



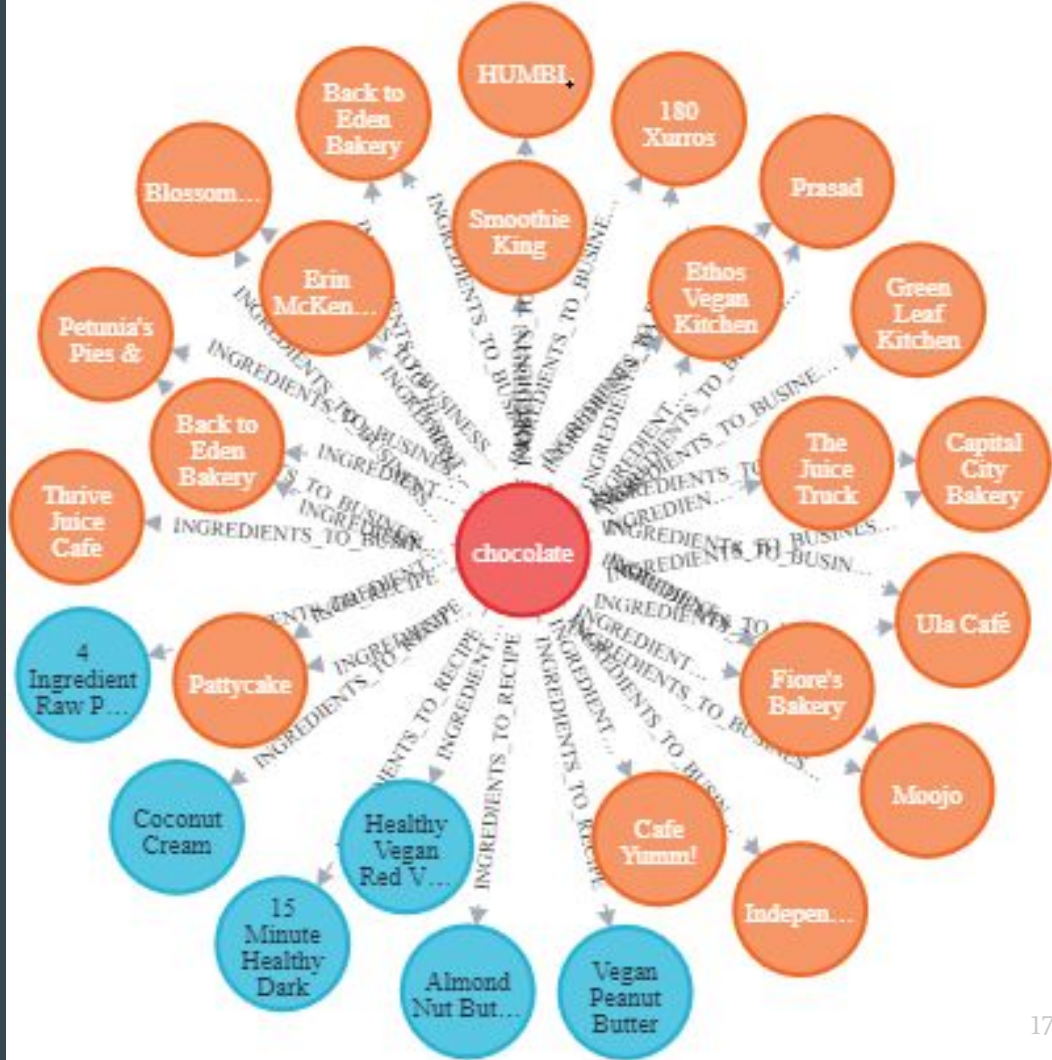
Query 2 (Cypher):

Question: What are the restaurants and recipes that are suitable to vegans and also contain chocolate?

Cypher:

```
MATCH
(c:Business)<-[s:INGREDIENTS_TO_BUSINESS_FROM_PHOTOS]-(a:Ingredients)-[r:INGREDIENTS_TO_RECIPE]->(b:Recipe)
WHERE (a.rdfs__label='chocolate' AND b.vegan=true AND b.ingredient_matches CONTAINS a.rdfs__label) AND (a.rdfs__label='chocolate' AND c.categories CONTAINS 'Vegan' AND c.photos_ingredient_matches CONTAINS a.rdfs__label)
RETURN *
```


Query 2 (Knowledge Graph):



Query 3 (Cypher):

Question: What are the cheap restaurants (cheap tips) and recipes (cost less than \$50), with more than 5 servings or good for groups, and gluten-free?

Cypher:

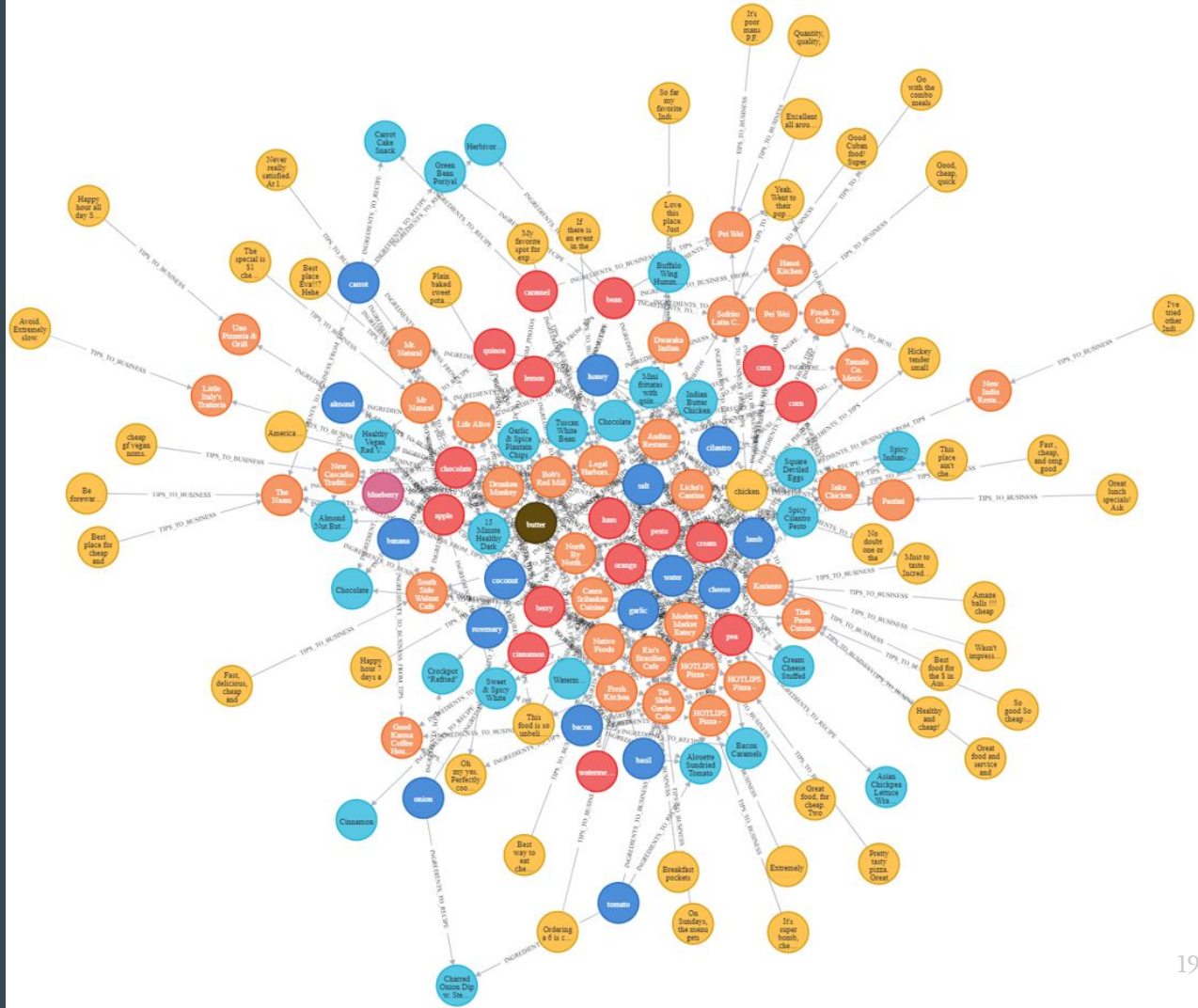
```
MATCH
```

```
(d:Tips)-[t:TIPS_TO_BUSINESS]->(c:Business)<-[s:INGREDIENTS_TO_BUSINESS_FROM_TIPS]-(a:Ingredient  
s)-[r:INGREDIENTS_TO_RECIPE]->(b:Recipe)
```

```
WHERE (d.text CONTAINS 'cheap' AND c.attributes CONTAINS '"RestaurantsGoodForGroups': 'True'" AND  
c.categories CONTAINS 'Gluten-Free') AND (b.pricePerServing<50 AND b.servings>5 AND  
b.glutenFree=true)
```

```
RETURN *
```

Query 3 (Knowledge Graph):



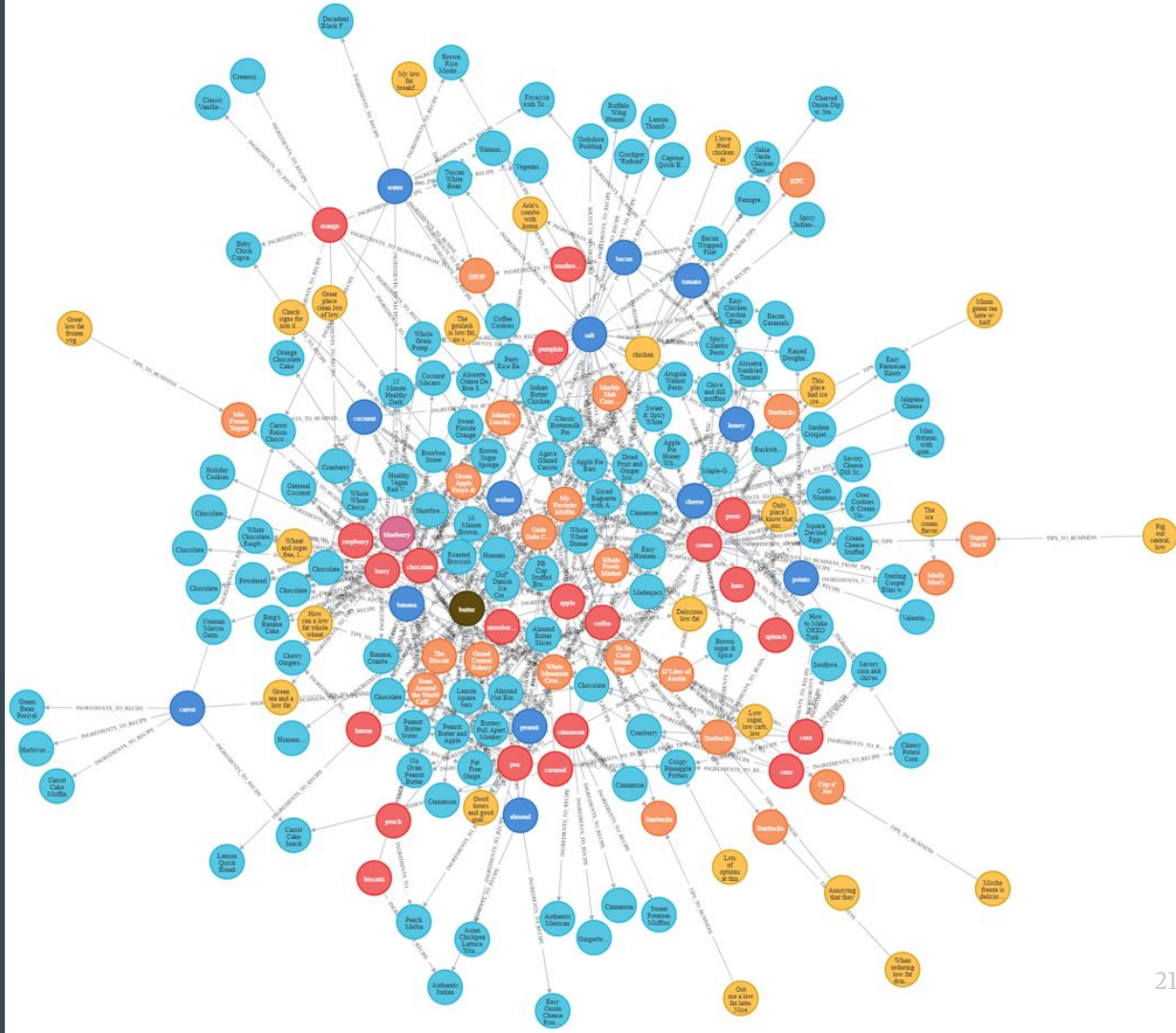
Query 4 (Cypher):

Question: What are the cheap restaurants (cheap tips) and recipes (cost less than \$50) that are low-calorie or low-fat?

Cypher:

```
MATCH
(d:Tips)-[t:TIPS_TO_BUSINESS]->(c:Business)<-[s:INGREDIENTS_TO_BUSINESS_FROM_TIPS]-(a:Ingredient
s)-[r:INGREDIENTS_TO_RECIPE]->(b:Recipe)
WHERE (d.text CONTAINS 'cheap' AND d.text CONTAINS 'low calorie' OR d.text CONTAINS 'low fat') AND
(b.pricePerServing<50) AND (toInteger(b.calories)/b.servings<41 OR toInteger(b.fats)/b.servings<4)
RETURN *
```

Query 4 (Knowledge Graph):



Query 5 (Cypher):

Question: What are the restaurants and recipes that serve duck or quail that allow for dogs, and what other ingredients can be found?

Cypher:

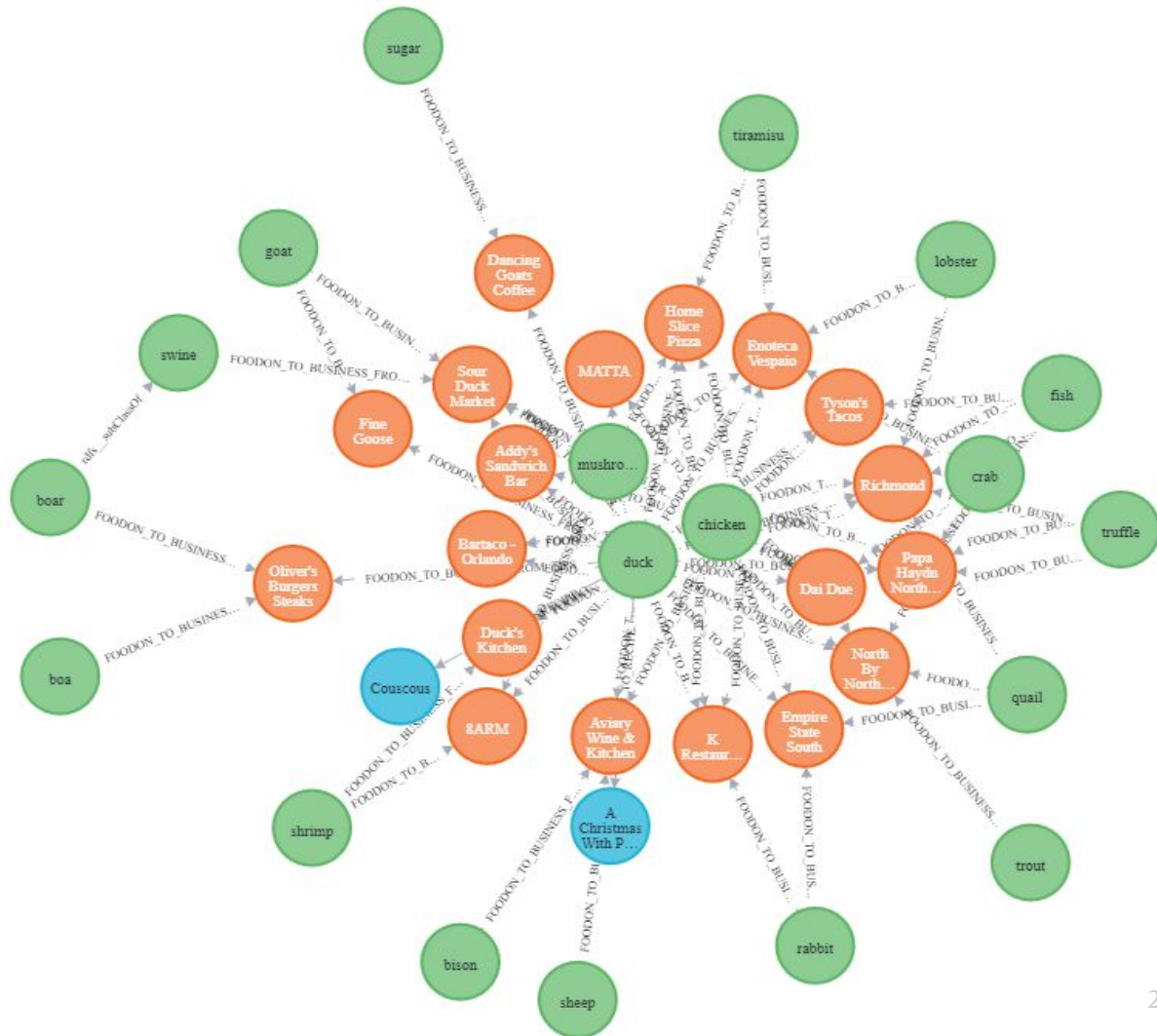
MATCH

```
(d:Recipe)<-[t:FOODON_TO_RECIPE]-(a:owl__Class)-[r:FOODON_TO_BUSINESS_FROM_TIPS]->(b:Business)<-[s:FOODON_TO_BUSINESS_FROM_TIPS]-(c:owl__Class)
```

```
WHERE (a.rdfs__label CONTAINS 'duck' OR a.rdfs__label CONTAINS 'quail' AND b.tips_food_matches CONTAINS a.rdfs__label) AND (a.rdfs__label CONTAINS 'duck' OR a.rdfs__label CONTAINS 'quail' AND d.food_matches CONTAINS a.rdfs__label) AND (b.attributes CONTAINS '"DogsAllowed": 'True"')
```

```
RETURN *
```

Query 5 (Knowledge Graph):



Demo

Lessons Learned

- Create graphs with different data sources
- Identify reasonable associations between different data sources
- Extract information from varying text properties
- Apply value matching methods
- Dataset preprocessing acceptable for graph creation
- Creating edge relationships with graphs
- Creating new properties on existing nodes

Conclusion and Future Work

- Conclusion
 - Given a set of Yelp reviews, it is possible to predict a restaurant's menu offerings
 - Successfully performed data integration between Yelp, FoodOn/Ingredients, and Recipe graphs
 - Successfully query outside and inside food options from food or ingredient inputs
- Future Work
 - Implementing knowledge graphs using Python graph visualizer
 - Creating queries involving Users and Reviews subgraphs from the Yelp graph
 - Connecting actual photos from id per restaurant

References

- <https://www.cancer.org/healthy/eat-healthy-get-active/take-control-your-weight/understanding-food-labels.html>
- <https://www.healthline.com/nutrition/how-much-protein-per-day>

Thank you!