Grade: **

VRIJE
UNIVERSITEIT
BRUSSEL

Could you elaborate on the choice of xsd:token?
It would have been nice to see some snippets of the ontology. How did you document the ontology?
Some (informed?) decisions have not been explained such as the namespace of your ontology.
While the changes are minimal, the goal is not to change your ERD as to reflect your ontology. The only changes we "allow" are those when your ERD is flawed. The mapping from database to ontology is otherwise trivial.

THE LAST RECIPE

OIS Project Report

Khaïm Berkane, Stijn Vissers, Kelvin Schoofs, Aaron Lippeveldts

December 20, 2019

Science & Bio-engineering Sciences

Contents

Abstract

This report contains the ontologies and rules of our system. The ontologies have been created by using the Protégé tool[1]. A visualisation of these ontologies has also been generated using the WebVOWL[2] visualization tool. The remainder of this report will discuss the relevance of the rules in our system.

en_GB: visualisation, en_US: visualization
Do not use both ;)
(you can keep it in the reference as it is the official title of the web page)

1 Ontology

For the design of the ontology, we made a small alteration to our conceptual schema (which we made by taking some inspiration of spoonacular[3]). We changed the `title` property of a recipe to a `name`. `CookingStep` now has an `amount` property referring to the amount of ingredients used in the step. The amount will be a float, and the unit it expresses can be found as a property of the associated Ingredient. This also meant that a step could only refer to one Ingredient. We therefore changed the cardinality of the `hasIngredient` relation. The WebVOWL visualization of our ontology can be found in the attachments, as Figure 1, and our updated ER diagram can also be found as Figure 2.

2 Rules

The isVegetarian (and similar diet-related properties) is an inferred property that can be written as the following rule:

$$\forall recipe \forall s \forall i \forall c.$$
$$consistOf(recipe, s), hasIngredient(s, i), belongsTo(i, c), \neg forbids(c, "Vegetarian")$$
$$\rightarrow isVegetarian(recipe)$$

where the literal `"Vegetarian"` can be replaced by any other diet.

The price property in Recipe is an inferred property, calculated from all the used ingredients, based on their price and the used amount. A recipe consists of several cooking steps, each indicating an ingredient and how much of that ingredient gets used. The ingredient itself contains the price per amount, which can then be multiplied with the amount, and summed for all the steps, to know the total price of the recipe.

rather informal

The inferred nutritional properties of Recipe (calories, proteins, ...) can similarly be calculated by summing them from the ingredients listed by the cooking steps, multiplied by the amount used in the cooking step. A SPARQL query for this could look like:

TIP: I think you use minted here?
You can change the default coloring
using \usemintedstyle{borland}
after \usepackage{minted}

```
PREFIX recipe: <http://example.com/thelastrecipe/>
SELECT ?name (SUM(?cals * ?amount) as ?calories)
WHERE {
  ?rec a recipe:Recipe.
  ?rec recipe:name ?name.
  ?rec recipe:ConsistsOf ?cookingstep.
  ?cookingstep recipe:HasIngredient ?ingredient.
  ?cookingstep recipe:amount ?amount.
  ?ingredient recipe:calories ?cals.
}
GROUP BY ?name
```

For data Properties such as `pricePerUnit` or `proteins` we also need a default value.

$$\forall i.proteins(i,n) \wedge \neg(\exists i.proteins(i,n)) \rightarrow proteins(i,0)$$

Finding an alternative ingredient consists of finding another ingredient that belongs to one of the substitution categories.

$$\forall i' \forall cat.canBeSubstBy(i,cat) \wedge belongsTo(i',cat) \rightarrow alternativeIngredient(i,i')$$

`hasIngredient` implies an amount in `CookingStep` is bigger than 0.

$$\forall step \exists i.hasIngredient(step,i), amount(step,n) \rightarrow n > 0$$

3 Conclusion

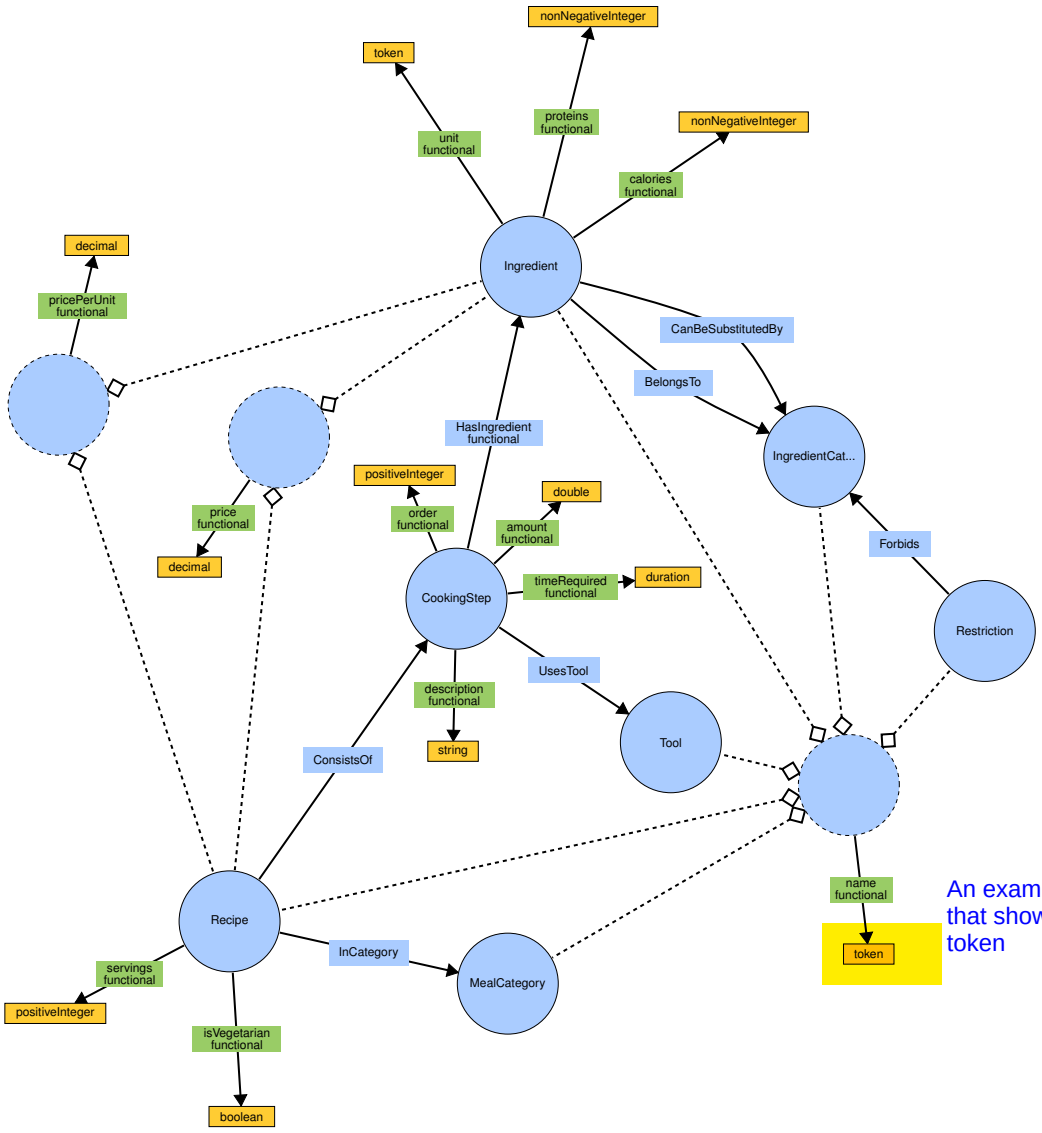
With some minor changes to our original conceptual schema, we were able to construct an ontology according to our system.

Thanks to the design of our schema, we were able to infer some properties, and these properties have been actualized in the rules found in this report.

Data properties have the correct data types. You can clearly see that proteins needs to be positive, that timeRequires is a duration...

Appendices

4 ~~Attachments~~



An example would be nice that shows you opted for token

Figure 1: WebVOWL Visualization of the ontology

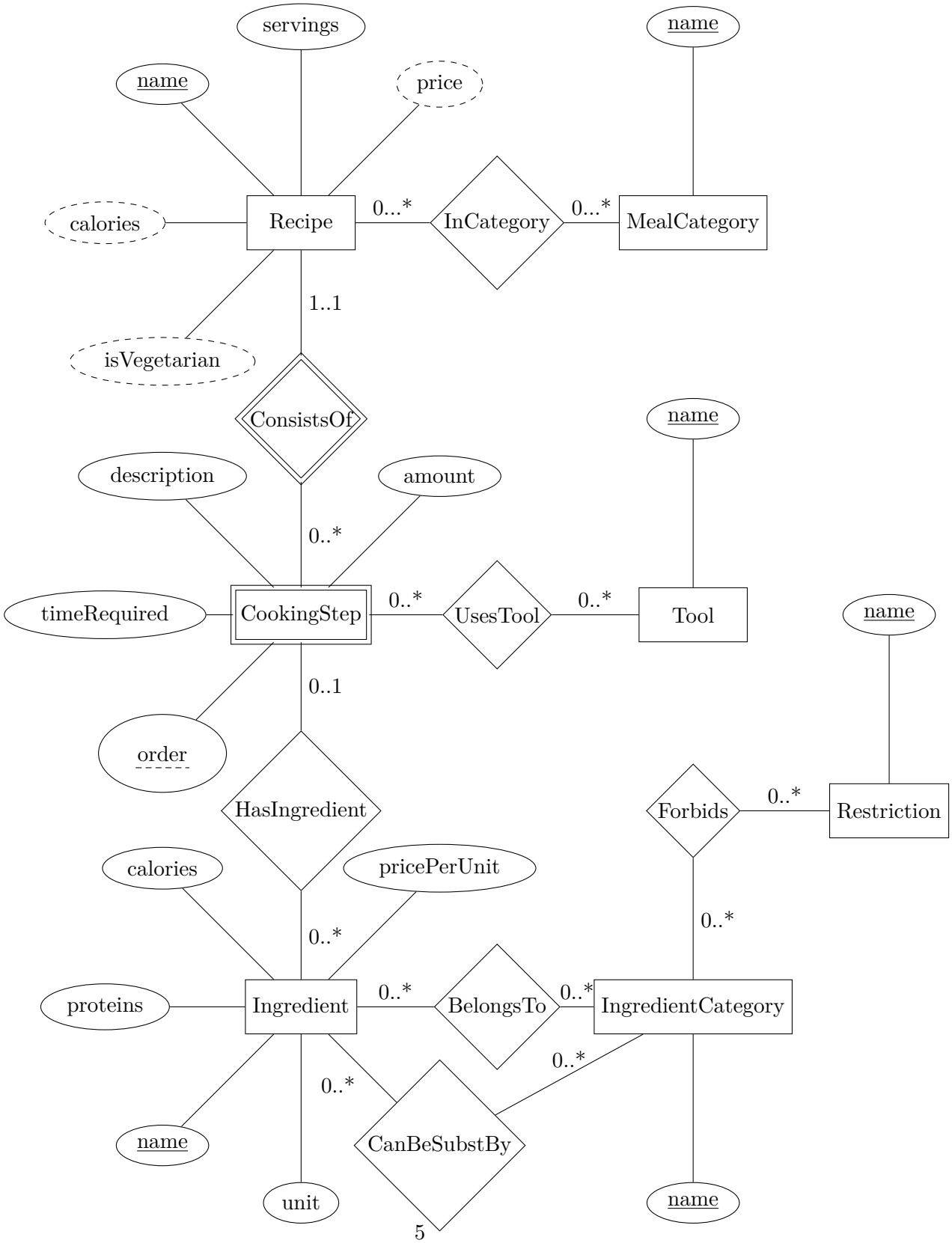


Figure 2: Updated ER Diagram

References

[1] Rafael Gonçalves, Josef Hardi, Matthew Horridge, Samson tu, and Mark Musen. *Protégé Desktop v5.5.0*. `https://protege.stanford.edu/about.php`, 2016. [Online; accessed 22-November-2019].

[2] Vincent Link, Steffen Lohmann, Eduard Marbach, Stefan Negru, and Vitalis Wiens. *WebVOWL: Web-based Visualization of Ontologies*. `http://www.visualdataweb.de/webvowl/`, 2014. [Online; accessed 22-November-2019].

[3] Crystal Schlegelmilch, David Urbansk, Philipp Katz, and Allie Hillmann. *Spoonacular recipe and food API*. `https://spoonacular.com/food-api`, 2018. [Online; accessed 22-November-2019].