**Kolegium Nauk Przyrodniczych  
Uniwersytet Rzeszowski**

**Sieci semantyczne**

Projekt Fluent Editor:   
Ontologia postaci z gry League of Legends

|  |  |
| --- | --- |
| **Prowadzący:** | **Autor:** |
| dr inż. Wojciech Kozioł | Artur Pas |
|  | 122961 |
|  | Lab3 |

**Rzeszów 2024**

Spis treści

[1. Opis projektu 3](#_Toc177596495)

[2. Źródła danych 3](#_Toc177596496)

[3. Struktura bazy wiedzy 3](#_Toc177596497)

[3.1. Opis klas 3](#_Toc177596498)

[3.2. Opis atrybutów 12](#_Toc177596499)

[3.3. Opis relacji 12](#_Toc177596500)

[3.4. Diagram CNL 13](#_Toc177596501)

[4. Sposób wypełnienia bazy wiedzy 14](#_Toc177596502)

[5. Eksploracja bazy za pomocą Reasoner’a oraz SPARQL’a 15](#_Toc177596503)

[5.1. Zapytania Reasoner’a 15](#_Toc177596504)

[5.2. Zapytania w języku SPARQL 16](#_Toc177596505)

[6. Możliwe wykorzystanie bazy wiedzy 20](#_Toc177596506)

[6.1. Generator zapytań SPARQL do gry Loldle 20](#_Toc177596507)

[7. Podsumowanie 23](#_Toc177596508)

# 1. Opis projektu

Projekt dotyczył utworzenia własnej ontologicznej bazy wiedzy na podstawie realnych danych. Do jej stworzenia wybrano środowisko Ontorion Fluent Editor 2015.

Tematem projektu jest stworzenie bazy wiedzy na temat postaci z gry League of Legends i wykorzystanie jej oraz środowiska Fluent Editor do znalezienia rozwiązania codziennej gry [Loldle w trybie klasycznym](https://loldle.net/classic).

Baza wiedzy zawiera około 1700 zdań napisanych w CNL (ang. Controlled Natural Language), czyli kontrolowanym języku naturalnym środowiska Fluent Editor. W skład zawartych danych wchodzą m.in. nazwy postaci, regiony, zasób używany przez bohatera, zasięg, płeć, rasa, rok wydania oraz pozycja na jakiej postać jest grana w grze. Sama baza była tworzona automatycznie na podstawie danych zebranych ręcznie ze strony internetowej gry Loldle.

# 2. Źródła danych

Projekt został wykonany bazując na informacjach zawartych na stronie <https://loldle.net/classic>

# **3. Struktura bazy wiedzy**

Elementami najważniejszymi w ontologicznej bazie wiedzy są klasy, ich instancje czyli indywidua oraz relacje, jakie między tymi elementami zachodzą. Baza posiada płaską strukturę. Składa się z 7 klas, 6 relacji oraz 2 atrybutów.

## 3.1. Opis klas

* **Klasa „gender”**

Opisuje płeć bohaterów. Obecnie w grze występują 3 płci.



*Rysunek 1. Klasa „gender” oraz jej indywidua*

* **Klasa „position”**

Opisuje pozycje na jakiej dany bohater jest obecnie grany. W grze jest 5 pozycji do obsadzenia przez graczy. Bohaterowie często są grani na wielu pozycjach.



*Rysunek 2. Klasa „position” oraz jej indywidua*

* **Klasa „resource”**

Opisuje zasób wykorzystywany przez bohatera w grze. Na chwilę obecną bohaterowie używają jednego z 13 typów zasobów.



*Rysunek 3. Klasa „resource” oraz jej indywidua*

* **Klasa „species”**

Opisuje rasę/gatunek bohatera. Bohaterowie mogą mieć wiele spośród 29 ras.



*Rysunek 4. Klasa „species” oraz jej indywidua*

* **Klasa „range-type”**

Opisuje zasięg bohatera.



*Rysunek 5. Klasa „range-type” oraz jej indywidua*

* **Klasa „region”**

Opisuje miejsca z którymi związani są bohaterowie. W historii gry występuje 16 głównych lokacji.



*Rysunek 6. Klasa „region” oraz jej indywidua*

* **Klasa „champion”**

Opisuje istniejących bohaterów. Gracze mają do dyspozycji 167 bohaterów.













*Rysunek 7, 8, 9, 10, 11, 12. Klasa „champion” oraz jej indywidua*

## 3.2. Opis atrybutów

* **Atrybut „released-in”**

Opisuje rok wydania bohatera.

* **Atrybut „have-importance”**

Opisuje ilość informacji o bohaterze. Atrybut ten nie występuje w grze, jest sztucznie wygenerowany na potrzeby wyszukiwania najlepszego wyboru podczas rozwiązywania gry Loldle.

## 3.3. Opis relacji

* **Relacja „have-gender”**

Wiąże bohatera z posiadaną przez niego płcią.

* **Relacja „play-in”**

Wiąże bohatera z pozycjami na jakich jest grany.

* **Relacja „use-resource”**

Wiąże bohatera z używanym przez niego zasobem.

* **Relacja „have-range-type”**

Wiąże bohatera z jego rodzajami ataków.

* **Relacja „have-species”**

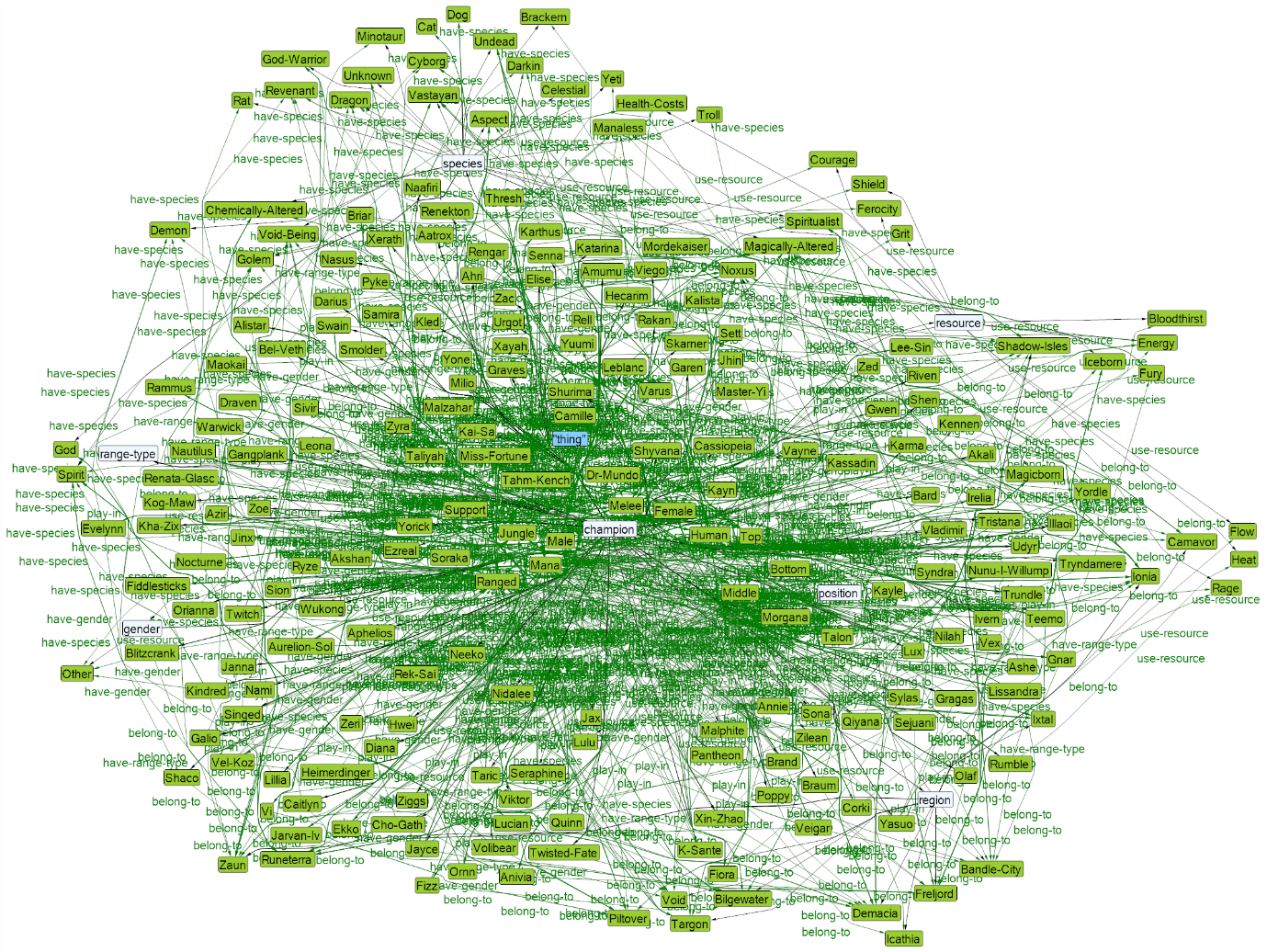
Wiąże bohatera z rasą lub rasami do jakich należy.

* **Relacja „belong-to”**

Wiąże bohatera do lokacji z jakimi jest związany.

## 3.4. Diagram CNL

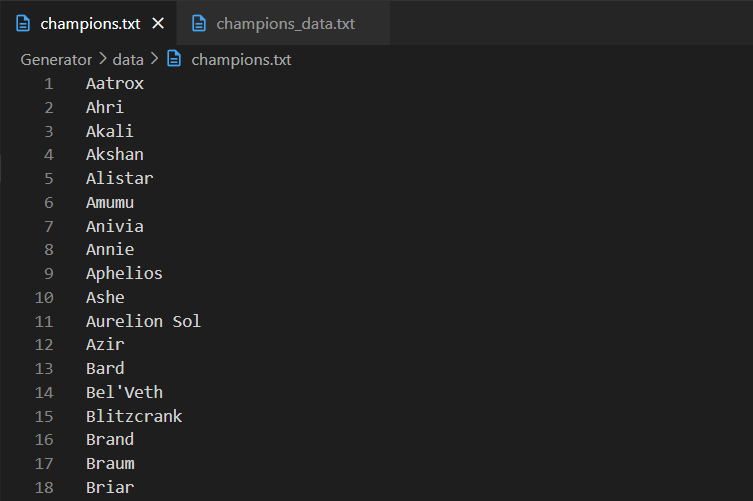
Poniżej znajduje się diagram CNL ontologicznej bazy wiedzy. Białe prostokąty reprezentują klasy. Zielone prostokąty to instancje klas czy indywidua. Czarne połączenia oznaczają zawieranie się jednej klasy w drugiej lub instancji w klasie. Zielone połączenia to są własne zdefiniowane relacje.



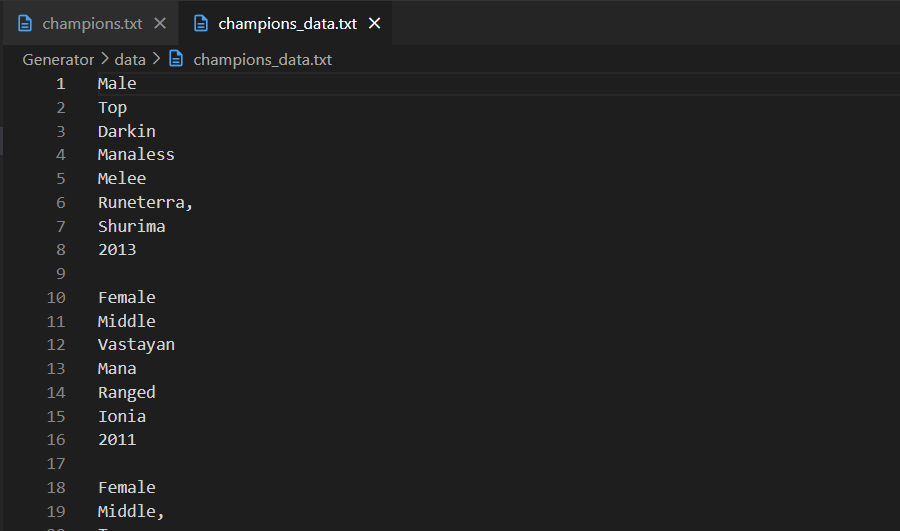
*Rysunek 13. Diagram CNL utworzony przez Fluent Editor*

# 4. Sposób wypełnienia bazy wiedzy

Dane zostały zebrane ręcznie poprzez wpisywanie wszystkich możliwych odpowiedzi do gry Loldle. Składają się na nie nazwy bohaterów oraz ich dane które następnie zostały użyte do automatycznego wypełnienia bazy wiedzy z pomocą skryptu napisanego w JavaScript.



*Rysunek 14. Zebrane nazwy bohaterów*



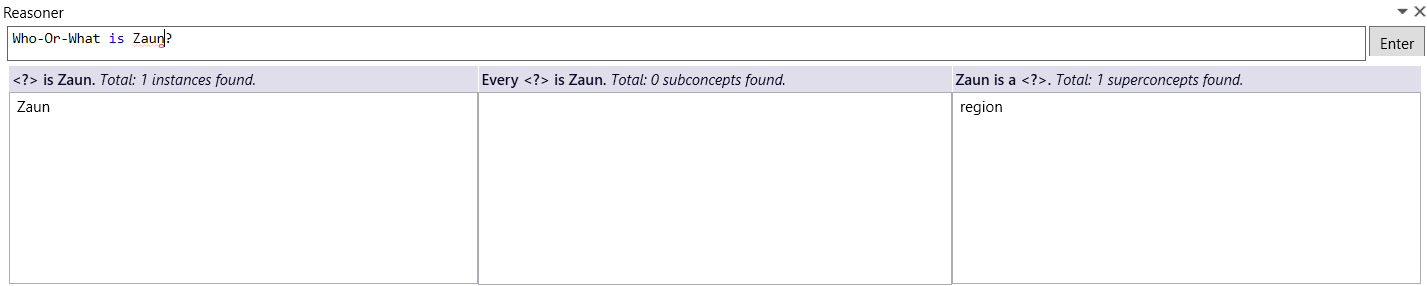
*Rysunek 15. Zebrane dane bohaterów*

# 5. Eksploracja bazy za pomocą Reasoner’a oraz SPARQL’a

## 5.1. Zapytania Reasoner’a

**1. Zapytanie pytające czym jest Zaun.**

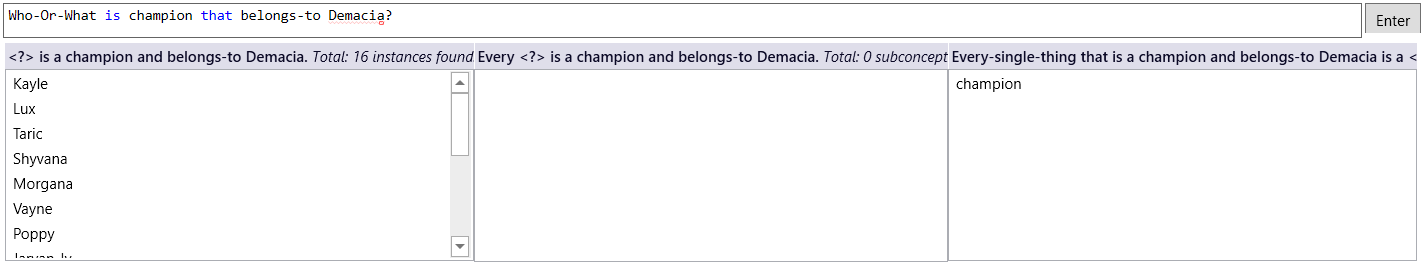
Who-Or-What is Zaun?



*Rysunek 16. Rezultat zapytania 1*

**2. Zapytanie wybierające bohaterów którzy są związani z regionem Demacja.**

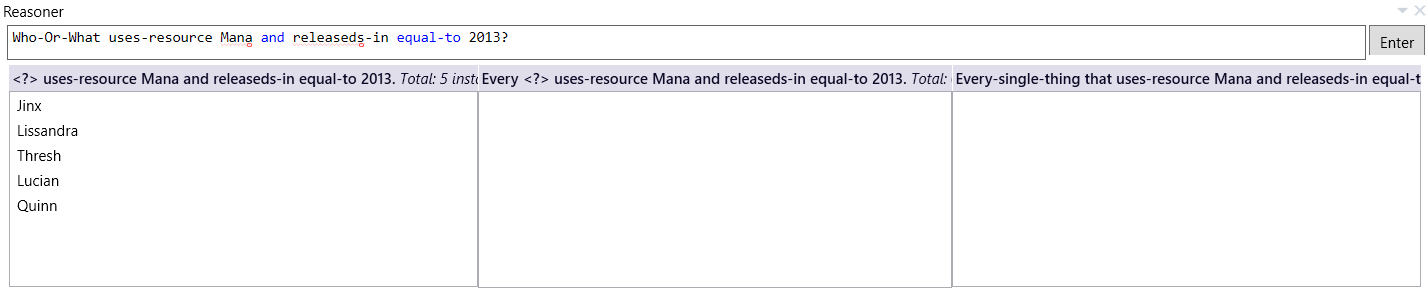
Who-Or-What is champion that belongs-to Demacia?



*Rysunek 17. Rezultat zapytania 2*

**3. Zapytanie wybierające bohaterów wydanych w 2013 roku i używających many.**

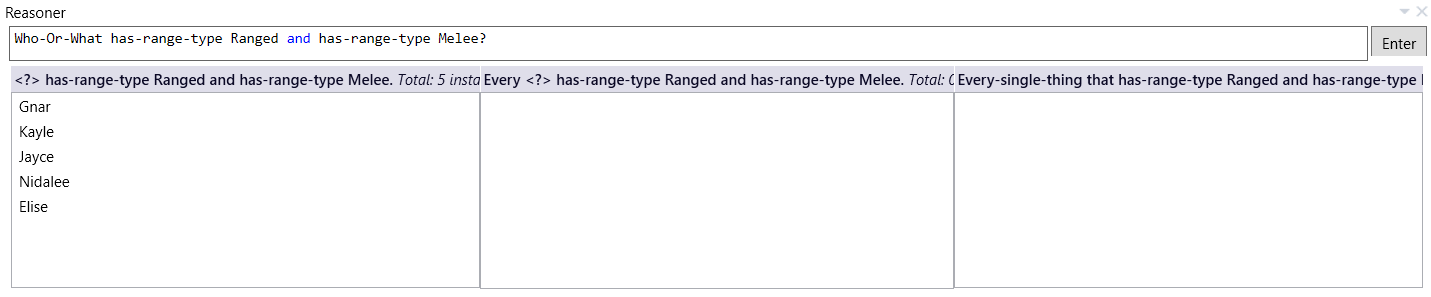
Who-Or-What uses-resource Mana and releaseds-in equal-to 2013?



*Rysunek 18. Rezultat zapytania 3*

**4. Zapytanie wybierające bohaterów mających jednocześnie typ ataku w zwarciu i dystansowy.**

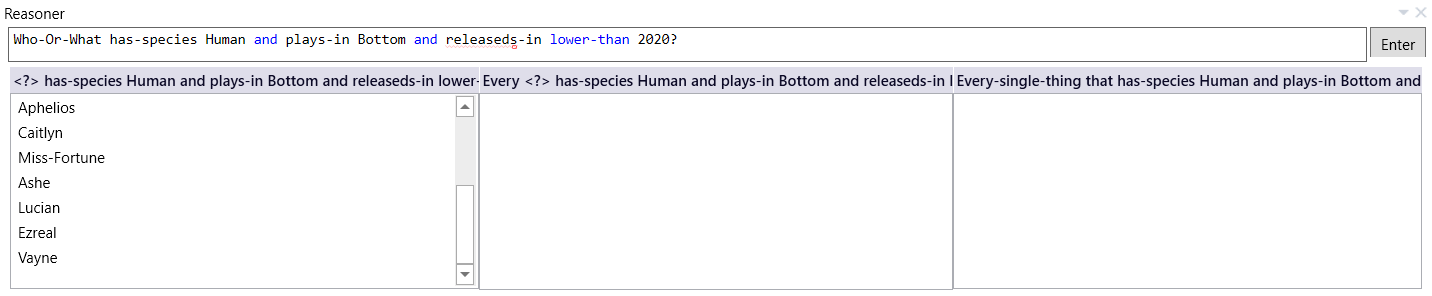
Who-Or-What has-range-type Ranged and has-range-type Melee?



*Rysunek 19. Rezultat zapytania 4*

**5. Zapytanie wybierające bohaterów którzy są ludźmi, są grani na dolnej linii oraz zostali wydani przed 2020 rokiem.**

Who-Or-What has-species Human and plays-in Bottom and releaseds-in lower-than 2020?



*Rysunek 20. Rezultat zapytania 5*

## 5.2. Zapytania w języku SPARQL

**6. Zapytanie listujące bohaterów wraz z regionem z którym są związani oraz ich rasą, którzy są mężczyznami.**

select ?x ?region ?species {

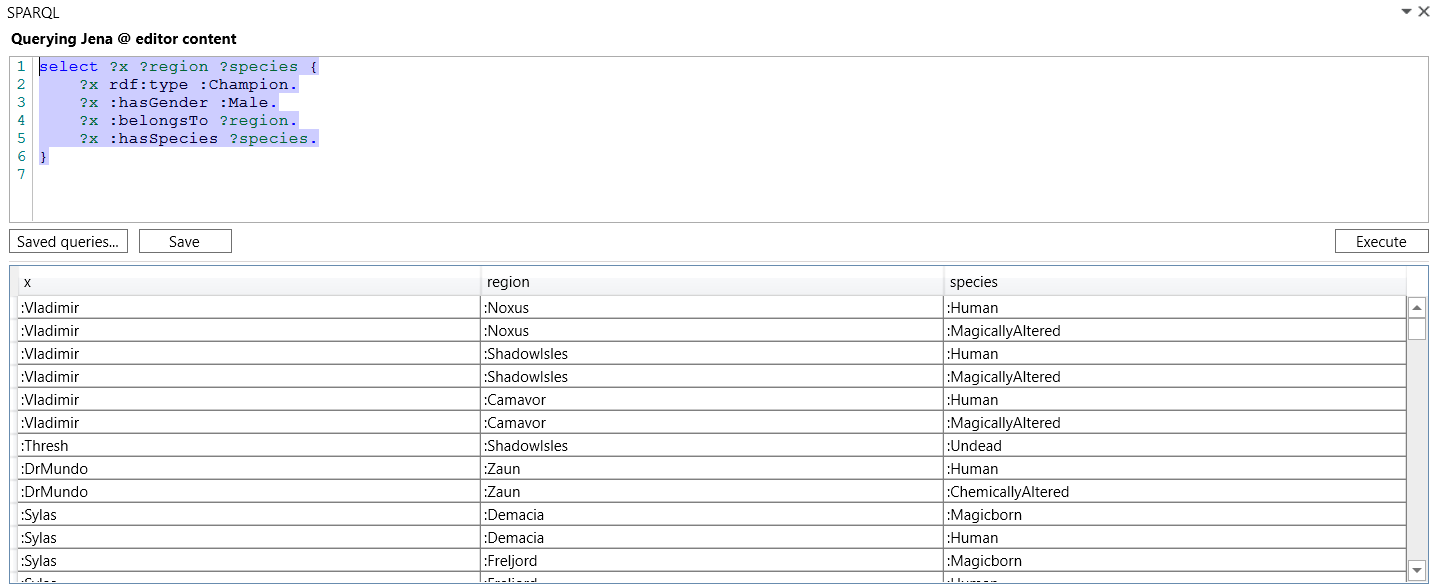
?x rdf:type :Champion.

?x :hasGender :Male.

?x :belongsTo ?region.

?x :hasSpecies ?species.

}



*Rysunek 21. Rezultat zapytania 6*

**7. Zapytanie listujące bohaterów walczących w zwarciu którzy nie pochodzą z regionów Runeterra i Shurima.**

select ?x {

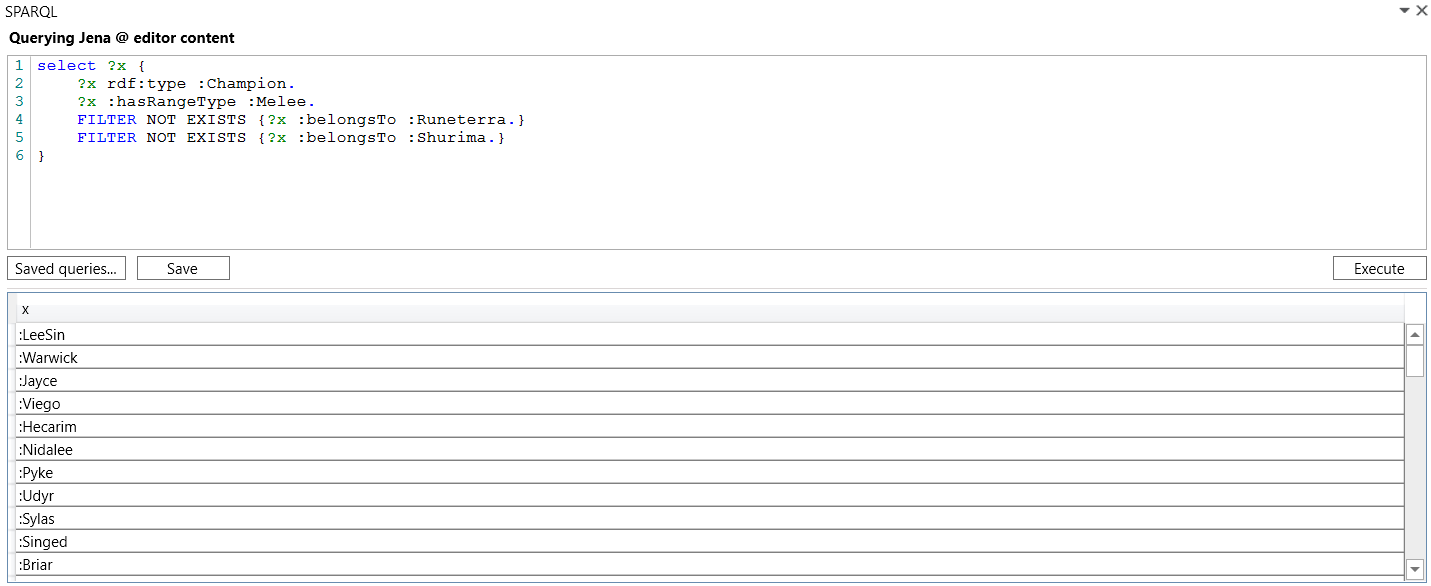
?x rdf:type :Champion.

?x :hasRangeType :Melee.

FILTER NOT EXISTS {?x :belongsTo :Runeterra.}

FILTER NOT EXISTS {?x :belongsTo :Shurima.}

}



*Rysunek 22. Rezultat zapytania 7*

**8. Zapytanie listujące bohaterów którzy są Darkinami lub Boskimi Wojownikami.**

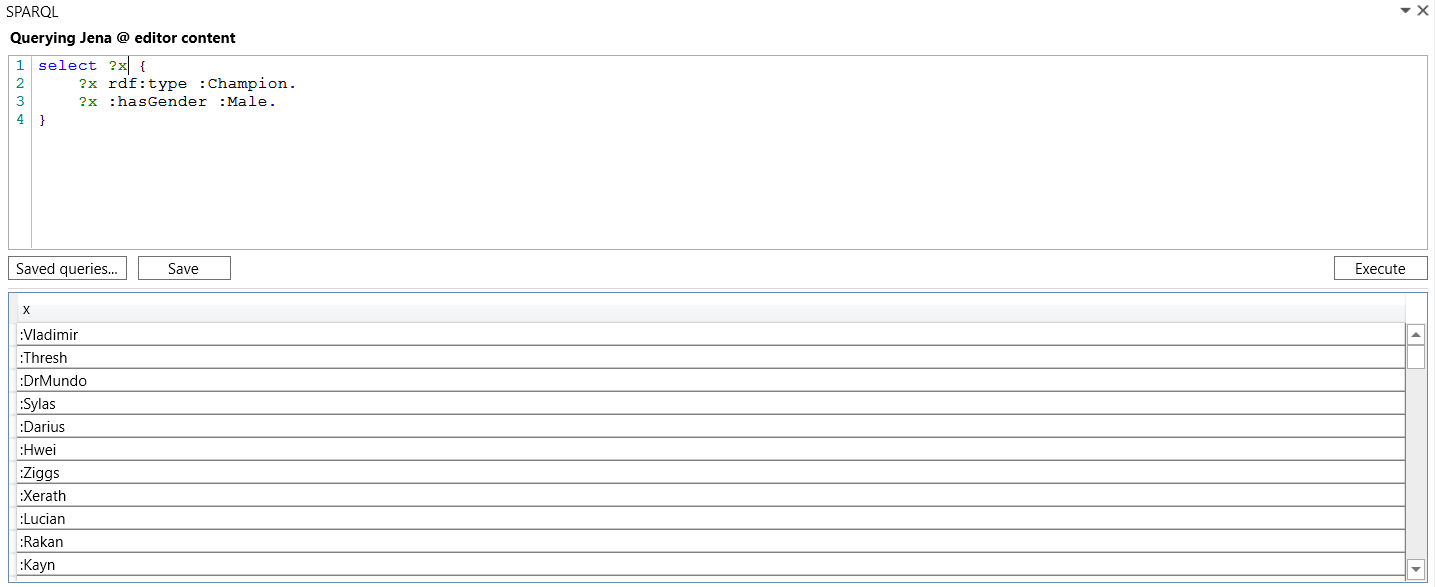
select ?x {

?x rdf:type :Champion.

?x :hasSpecies ?sp.

FILTER (?sp IN (:Darkin, :GodWarrior))

}



*Rysunek 23. Rezultat zapytania 8*

**9. Zapytanie listujące bohaterów posortowanych po roku wydania, wraz z rokiem ich wydania, którzy zostali wydani do roku 2015 włącznie.**

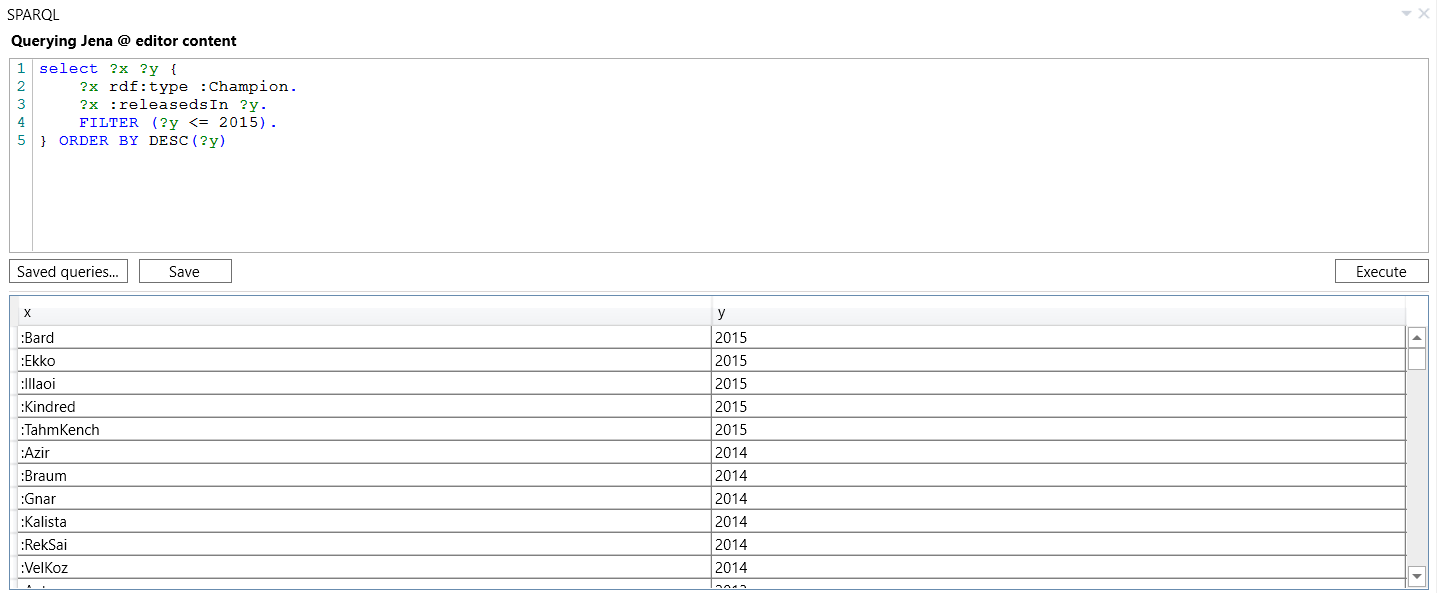
select ?x ?y {

?x rdf:type :Champion.

?x :releasedsIn ?y.

FILTER (?y <= 2015).

} ORDER BY DESC(?y)



*Rysunek 24. Rezultat zapytania 9*

**10. Zapytanie listujące bohaterów posortowanych po ważności, którzy są mężczyznami, nie są grani na dżungli, są grani na wsparciu lub na górnej linii, nie są nieumarli, ani nie są demonami, ani nie są duchami, używają many, walczą w zwarciu, nie należą do regionu Camavor, ani do Wysp Cienia, ani do Bilgewater ani do Runeterry, zostali wydani powyżej roku 2015.**

SELECT ?x {

?x rdf:type :Champion.

?x :hasGender :Male.

FILTER NOT EXISTS {?x :playsIn :Jungle}.

?x :playsIn ?pos.

FILTER (?pos IN (:Support, :Top)).

FILTER NOT EXISTS {?x :hasSpecies :Undead}.

FILTER NOT EXISTS {?x :hasSpecies :Demon}.

FILTER NOT EXISTS {?x :hasSpecies :Spirit}.

?x :usesResource :Mana.

?x :hasRangeType :Melee.

FILTER NOT EXISTS {?x :belongsTo :Camavor}.

FILTER NOT EXISTS {?x :belongsTo :ShadowIsles}.

FILTER NOT EXISTS {?x :belongsTo :Bilgewater}.

FILTER NOT EXISTS {?x :belongsTo :Runeterra}.

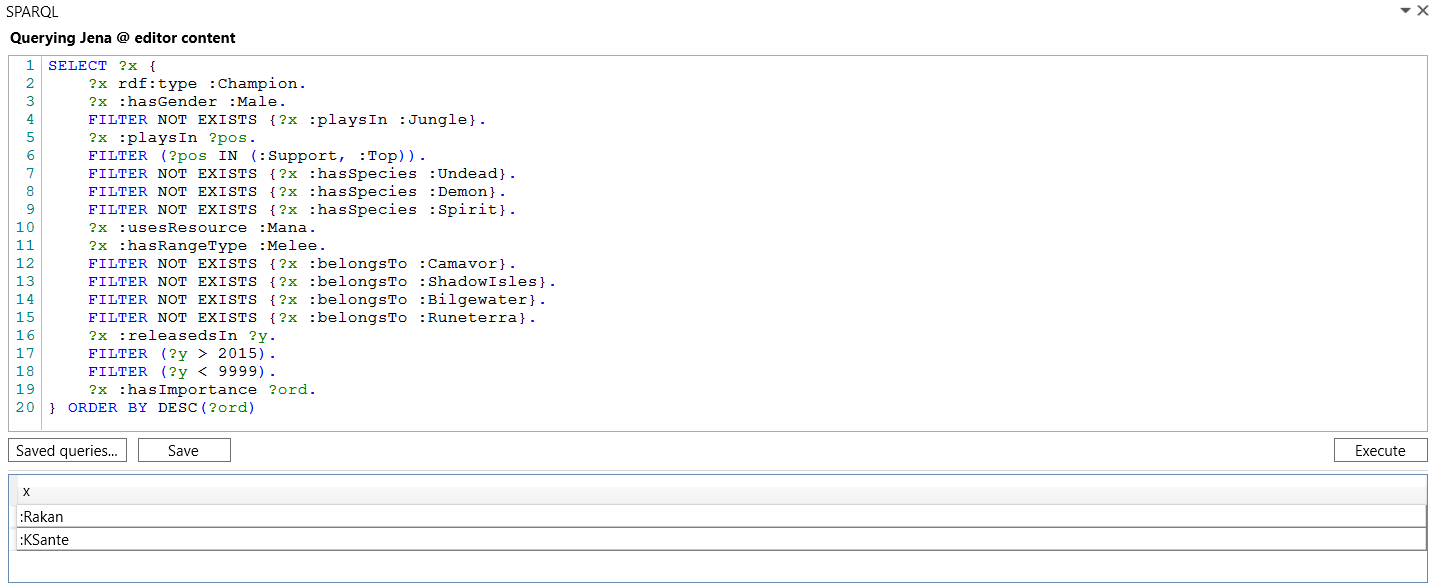
?x :releasedsIn ?y.

FILTER (?y > 2015).

FILTER (?y < 9999).

?x :hasImportance ?ord.

} ORDER BY DESC(?ord)



*Rysunek 25. Rezultat zapytania 10*

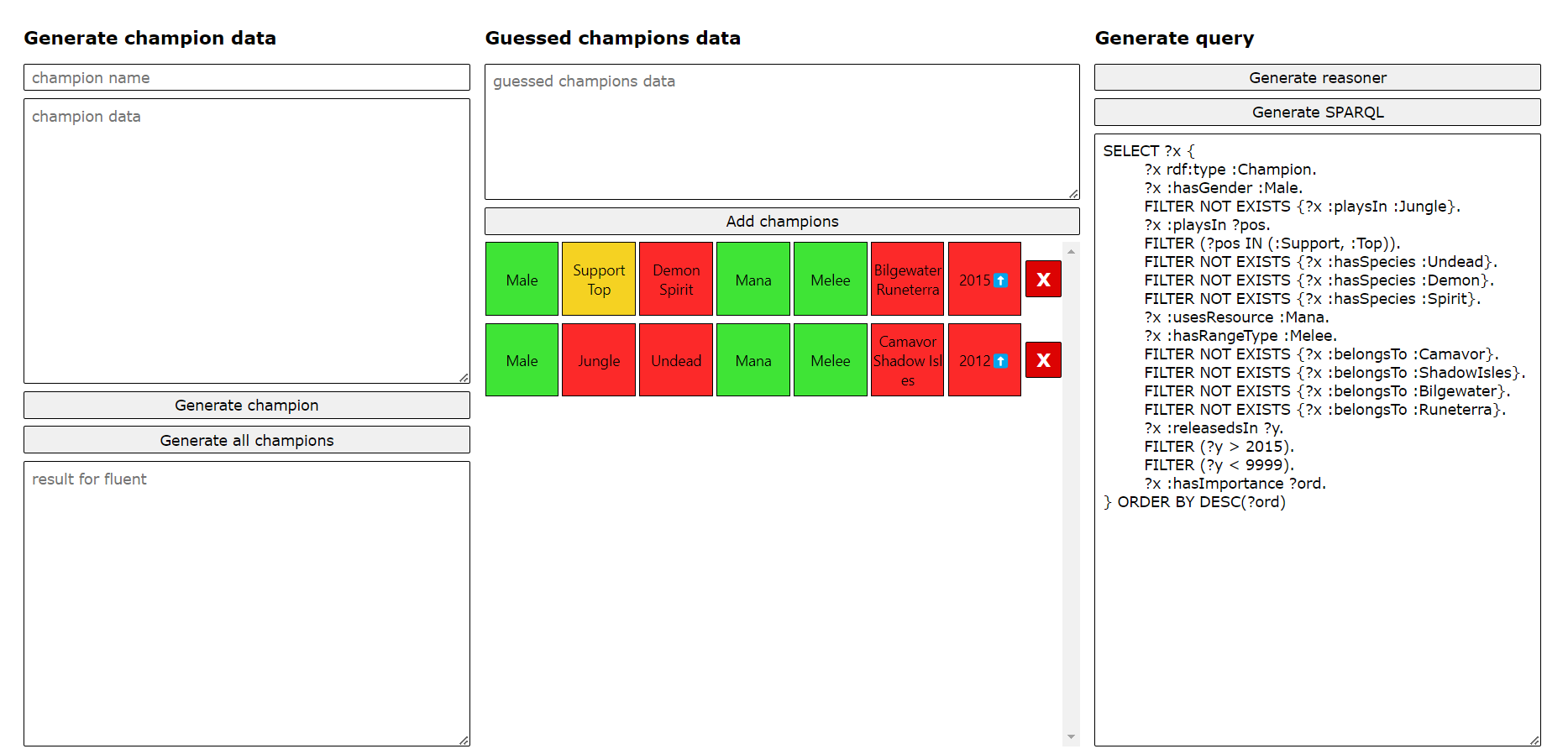
# 6. Możliwe wykorzystanie bazy wiedzy

Bazę wiedzy można wykorzystać do zwykłego wyszukania informacji na temat bohaterów lub znalezienia rozwiązania w codziennej grze Loldle za pomocą zapytań SPARQL w środowisku Fluent Editor.

## 6.1. Generator zapytań SPARQL do gry Loldle

Tworzenie takiego zapytania ręcznie jest zadaniem wymagającym czasu, znajomości SPARQL oraz poprawności, dlatego przygotowany został generator pozwalający na wygenerowanie zapytań na postawie danych skopiowanych ze strony Loldle.

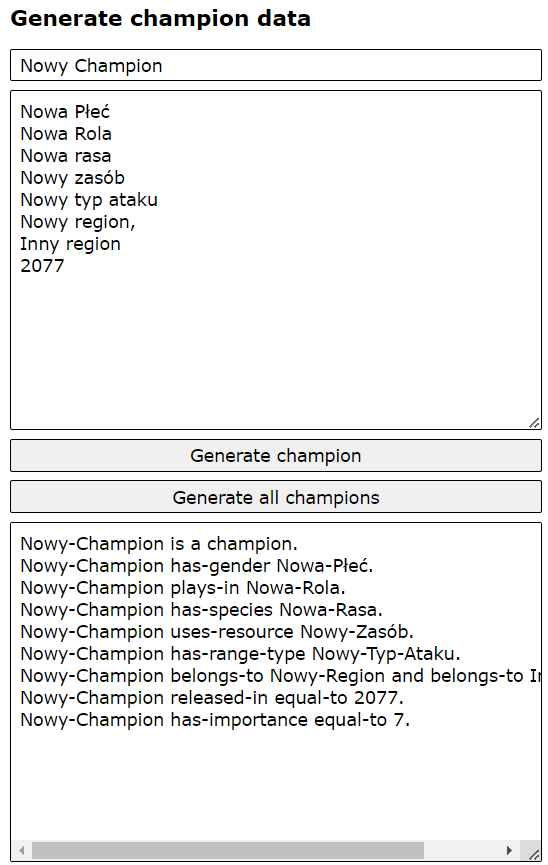
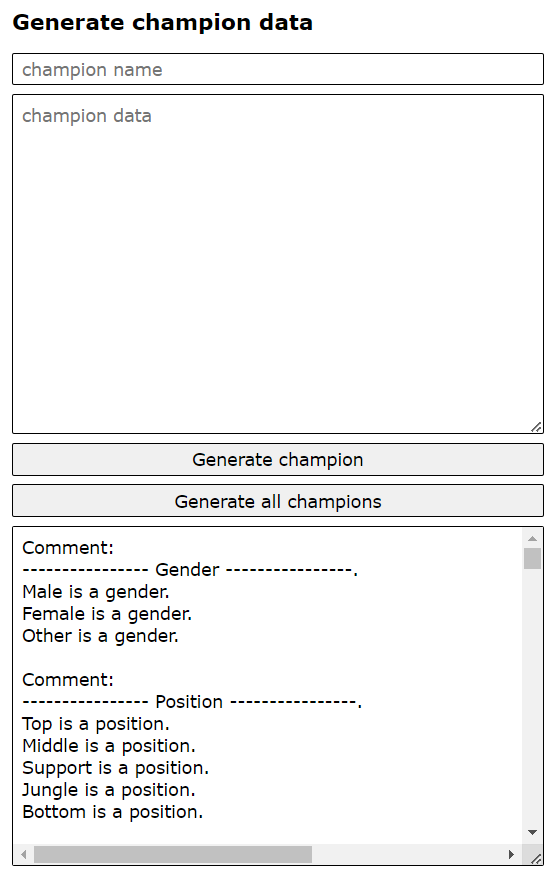
Do użycia generatora potrzebny jest serwer w celu odczytania plików z poziomu JavaScript. Idealnie nadaje się do tego rozszerzenie [Live Server](https://marketplace.visualstudio.com/items?itemName=ritwickdey.LiveServer) do Visual Studio Code



*Rysunek 26. Okno generatora*

**Generowanie danych**

Lewa kolumna generatora pozwala wygenerować ontologie nowego bohatera poprzez wprowadzenie jego nazwy oraz skopiowanych danych lub całą bazę wiedzy na postawie danych w plikach champions.txt oraz champions\_data.txt w folderze /data.

*Rysunek 26 i 27. Wygenerowane dane nowego bohatera oraz wszystkich bohaterów*

**Generowanie zapytań**

Środkowa kolumna służy do utworzenia kontenerów z danymi bohaterów poprzez wklejenie danych skopiowanych ze strony.

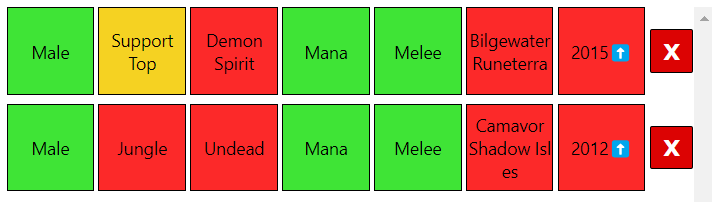
Możliwe jest wklejenie wielu bohaterów na raz jeśli np. gracz chciałby skorzystać z programu jako podpowiedzi mając już kilka nietrafionych odgadnięć, lub wklejanie danych pojedynczo w miarę odgadywania bohaterów z pomocą programu.

Następnie klikając na kontenery trzeba odwzorować informację o poprawności danych odgadniętego bohatera. Dane mają 3 typy:

* Czerwony – błędne, żadne dane w kwadracie nie są poprawne
* Żółty – częściowo poprawne, co najmniej jedna rzecz jest poprawna ale coś nie pasuje lub czegoś brakuje
* Zielony – poprawne trafienie

Kolejność bohaterów nie ma znaczenia.   
Powtórzenia nie wpływają na działanie programu.   
Po oznaczeniu poprawnych danych w kolumnie nie trzeba zaznaczać danych ponownie w innych odgadnięciach.  
Niechciane kontenery można usunąć przyciskiem z ikoną X.





*Rysunek 28 i 29. Porównanie kontenerów z danymi na temat odgadniętych bohaterów*

Prawa kolumna programu służy do wygenerowania zapytania na podstawie dostarczonych danych poprzez wciśnięcie przycisku z napisem „Generate SPARQL”.

Wygenerowane zapytanie należy wkleić do zakładki SPARQL i je wykonać. Następnie trzeba wpisać nazwę pierwszego bohatera z listy do gry Loldle. Jeśli trafienie nie było poprawne trzeba dane nowo odgadniętego bohatera skopiować, wkleić do generatora, oznaczyć poprawność i ponownie wygenerować i wykonać zapytanie.

# 7. Podsumowanie

Stworzona ontologiczna baza wiedzy jest prostym źródłem danych na temat bohaterów z gry League of Legends. Mogłaby ona zostać poszerzona o różne dane takie jak umiejętności, powiązania między bohaterami, opisy, dokładniejsze lokacje w regionach, daty narodzin, statystyki bohatera w grze, przedmioty występujące w grze itp. lecz takie dane nic nie wnoszą w kontekście znalezienia rozwiązania do gry Loldle.

Zaletą ograniczenia danych jest możliwość dostosowania projektu dla innej gry w tym typie, która nie zawiera danych obrazkowych, takiej jak np. [Pokedle](https://pokedle.net/classic) (odgadnięcie pokemona) czy [Dotadle](https://dotadle.net/classic) (odgadnięcie postaci z gry Dota2), przy zachowaniu funkcjonalności generatora.