



# **Knowledge Representation and Reasoning Project Report**

## **Genshin Impact Knowledge Graph**

Course Instructor: **Dr. Amna Basharat**

### **Team Members**

Sarim Aeyzaz (i21-0328)

Raffay Khan (i21-0335)

Zaraar Malik (i21-2705)

## Table of Contents

- 1) Domain Description
- 2) Competency Questions
- 3) Dataset Description
  - a) Description of each Table
- 4) Conceptual Model
  - a) Identified Vocabularies
- 5) External Linked Datasets
- 6) Designing the Ontology
  - a) Ontology Class Hierarchy
  - b) Ontology 11 Requirements
- 7) Mapping
- 8) Linking
- 9) Reasoning
  - a) SWRL Rules
  - b) Normal Reasoning
  - c) SWRL Reasoning
  - d) Linked Data Reasoning
- 10) Publishing
  - a) GraphDB
  - b) Fuseki
- 11) Validation using SPARQL Endpoint
  - a) Competency Questions Answered
- 12) Visualization
- 13) Reflect

## Domain Description

Our KRR Knowledge Base will be about Genshin Impact. It's a game where you have 7 elements (Pyro, Cryo, Hydro, Electro, Dendro, Anemo & Geo) and each character is aligned to one of those elements only. Using the elements, the character is able to use their abilities and defeat enemies/bosses found throughout the world.

Each character can equip a specific type of weapon and certain characters are strong against certain enemies depending on the element of the character and the element of the respective enemy. In terms of progression, each character needs specific materials to upgrade themselves and their weapon.

As you play the game, you are able to control 4 characters at the same time. These characters can be of any element and of any role (Damage Dealer, Shielder, Healer, Support). When you are using multiple characters to defeat enemies, different elements are able to react with one another causing a reaction (Electro-charged, Vaporize, Melt, Crystalize, Freeze, Swirl, Spread, Aggravate). These reactions amplify the damage you deal to the enemy so you are incentivized to run specific team combinations to maximize a reaction type.

In terms of explorations, there are tons of chests and enemies you encounter along the way. These enemies can be of different tiers (Normal, Elite, Boss and Weekly Boss) and as such, the player has to defeat them in order to progress with the story.

## Competency Questions

Here are a few competency questions we expected our knowledge graph to answer:

- What materials do I need to level up X character? (e.g. Kaeya)
- What materials do I need to upgrade X weapon? (e.g. Aquila Favonia)
- List me all characters of X element. (e.g. Hydro)
- Which element is involved in the most reactions?
- Which element has the most healers?
- List me all the weapons X character can use? (e.g. Kazuha)
- What characters are strong against X boss? (e.g. Pyro Regisvine)
- Create a team of 2 people which can do a vaporize reaction?
- List me an electro-charged only team with all 4 characters having a different role
- (Bonus Federated query) Get me the voice actor names for the characters from Wikidata

## Dataset Description

Our data is unstructured as it is represented in the form of a game. Using this information, We extracted majority of the game's information and then gave it a formal representation via CSV files. Here is the final iteration of the CSV files and the attributes of each one:

- Characters
- Weapons
- Enemies
- Reactions

**Note:** A lot of this data was extracted from Project Amber website (<https://ambr.top/en>)

### Description of each Table

- **Characters:**
  - character\_name (Name of the character)
  - rarity (Rarity of the character, either 4-star or 5-star)
  - region (Location where the character resides at)
  - vision (One of the 7 elements possessed by the character)
  - weapon\_type (One of the 5 weapon types used by the character)
  - constellation (Unique per character)
  - birthday (Birthday of the character)
  - ascension\_boss (Material used by the character to level up)
  - limited (Boolean value representing if the character is limited or not)
  - ascension (Ascension passive granted when you level the character up)
  - role (Role of the character. Can be DPS, Healer, Shielder or Support)
  - gender (Male or Female)
  - archon (Boolean value representing if a character is an archon)
  - release\_date (Release date of the character in the game)

	character_name	rarity	region	vision	weapon_type	constellation	birthday	ascension_boss	limited	ascension	role	gender	archon
0	Albedo	5	Mondstadt	Geo	Sword	Princeps_Cretaceus	2023-09-13	Basalt_Pillar	True	Geo_DMG_Bonus	Support	Male	False
1	Alhaitham	5	Sumeru	Dendro	Sword	Vultur_Volans	2023-02-11	Pseudo-Stamens	True	Dendro_DMG_Bonus	DPS	Male	False
2	Aloy	5	Mondstadt	Cryo	Bow	Nora_Fortis	2023-04-04	Crystalline_Bloom	True	Cryo_DMG_Bonus	Support	Female	False
3	Amber	4	Mondstadt	Pyro	Bow	Lepus	2023-08-10	Everflame_Seed	False	ATK	Support	Female	False
4	Arataki_Itto	5	Inazuma	Geo	Claymore	Taurus_Iracundus	2023-06-01	Riftborn_Regalia	True	CRIT_Rate	DPS	Male	False

- **Weapons:**
- Name (Name of the weapon)
- Type (One of the 5 types of the weapon)
- Rarity (Can be either 1-Star till 5-Star)
- UpgradeMaterial (Material used to upgrade the weapon)

	Name	Type	Rarity	UpgradeMaterial
0	Wolf's_Gravestone	Claymore	5	Tusk_of_the_Wolf
1	Skyward_Harp	Bow	5	Luminous_Sands_from_Guyun
2	Primordial_Jade_Winged-Spear	Polearm	5	Jade_Wing
3	Aquila_Favonia	Sword	5	Tusk_of_the_Wolf
4	Lost_Prayer_to_the_Sacred_Winds	Catalyst	5	Tile_of_Decarabian's_Tower

- **Reactions:**
- Element (Element involved in a reaction)
- causesReaction (Reaction caused by that element)

	Element	causesReaction
0	Pyro	Overloaded
1	Electro	Overloaded
2	Pyro	Melt
3	Cryo	Melt
4	Pyro	Vaporize

- **Enemies:**
- Name (Name of the enemy)
- Element (Element possessed by the enemy)
- WeakAgainst (Element the enemy is weak against)
- EnemyType (One of the 4 different types an enemy can be)
- DropsMaterial (Material dropped by the enemy)

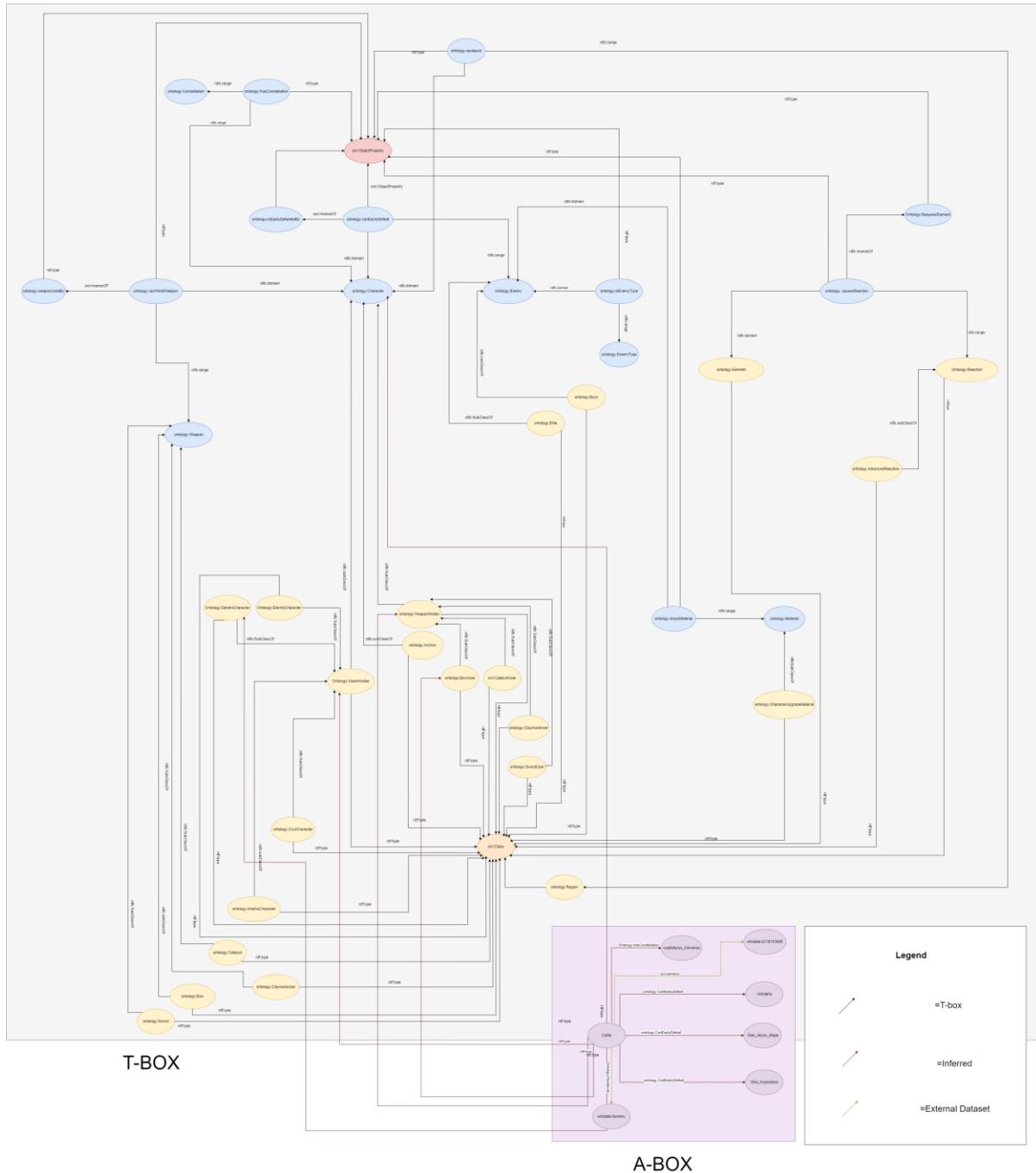
	Name	Element	WeakAgainst	EnemyType	DropsMaterial
0	Hilichurl	Anemo	Electro	Normal	Firm_Arrowhead
1	Anemo_Samachurl	Anemo	Electro	Elite	Divining_Scroll
2	Hydro_Samachurl	Hydro	Pyro	Elite	Divining_Scroll
3	Cryo_Samachurl	Cryo	Pyro	Elite	Divining_Scroll
4	Mitachurl	Anemo	Electro	Elite	Firm_Arrowhead

# Conceptual Model

There are 2 forms of conceptual models we have made, the first one is the true conceptual model but it will be extremely hard to interpret because it's too large. Here it is:

It also has the Wikidata relations included.

**Conceptual Model of Genshin Impact Knowledge Graph (KRR Project)**  
**Sarim Aeyzaz (i210328)**  
**Zaraar Malik (i212705)**  
**Raffay Khan(i210335)**



Link to conceptual model:

<https://drive.google.com/file/d/1STxGt5KberewhSH27bZgwpptAY0CPybP/view?usp=sharing>

Here is the second one which is taken from Protégé (it's missing the object properties but at-least it is readable:



## Identified Vocabularies

- canEasilyDefeat
- canWeildWeapon
- causesReaction
- dropsMaterial
- elementAlignment
- enemyWeakAgainst
- hasConstellation
- hasResident
- hasRole
- isEasilyDefeatedBy
- isEnemyType
- isWeaponType
- requiresElement
- residesAt
- usesCharacterUpgradeMaterials
- usesWeaponType
- usesWeaponUpgradeMaterial
- weaponUsedBy
- birthday
- hasAscensionPassive
- hasGender
- isArchon
- isLimitedCharacter
- rarity
- releaseDate
- foaf:name

## External Linked Dataset

I tried to search a lot for an external dataset related to Genshin but sadly, I couldn't find a single thing. The only source I could find is partial information on character only in Wikidata and thus, just decided to link all character of Genshin listed on Wikidata to my characters.

However, do note that Wikidata doesn't have every character of Genshin impact on its website so not every single character could be linked :(.

Here it is: <https://www.wikidata.org/wiki/Q65059474>

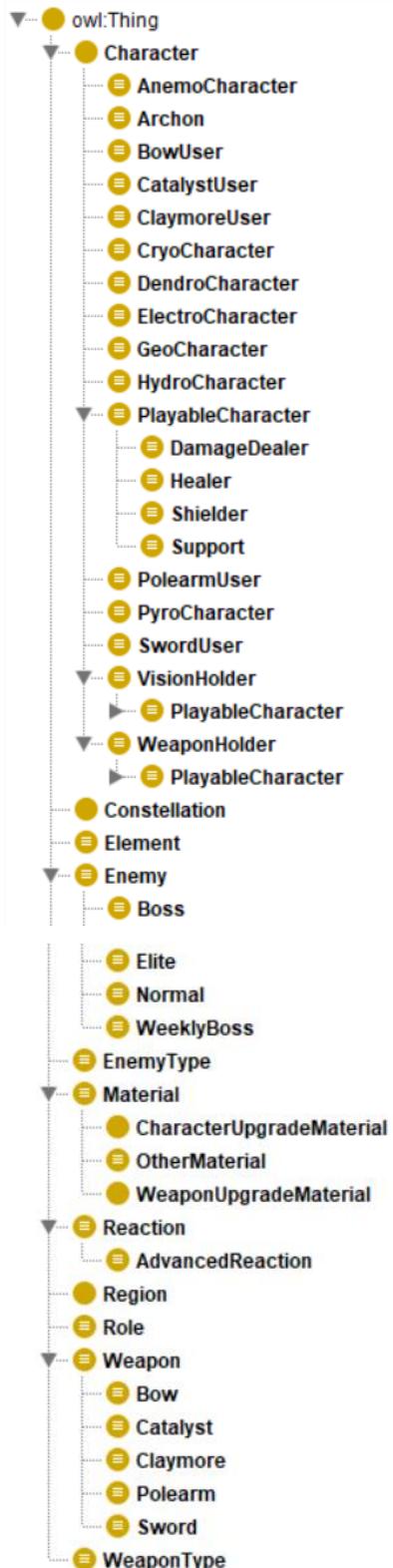
The screenshot shows the Wikidata item page for the character Candace. The top navigation bar includes links for English, Not logged in, Talk, Contributions, Create account, and Log in. A banner at the top encourages participation in Data Modelling Days from November 30th to December 2nd. The main content area starts with the title "Candace" and its ID "Q121919991". Below the title is a brief description: "playable character in Genshin Impact". There is a link to "In more languages" and a "Configure" button. A table lists language labels and descriptions: English (Candace, playable character in Genshin Impact), Urdu (No label defined, No description defined), and Pashto (No label defined, No description defined). A "All entered languages" section follows. The "Statements" section contains three rows of data:

Statement Type	Value	Label	Description	Action Buttons
instance of	video game character	video game character	playable character in Genshin Impact	edit, + 0 references, + add reference, + add value
sex or gender	female	female	No label defined	edit, + 0 references, + add reference, + add value
country of citizenship	Sumeru	Sumeru	No label defined	edit

On the right side of the statements, there are boxes for linking to other Wikidata projects: Wikipedia, Wikibooks, Wikinews, Wikiquote, Wikisource, Wikiversity, Wikivoyage, Wiktionary, and Multilingual sites. A note at the bottom of the statements section mentions "Category:Cosplay of Candace (Genshin Impact)".

# Designing the Ontology

## Ontology Class Hierarchy



## Description:

In this class diagram, most of these classes are supposed to be inferred. This was done to increase the semantic expressivity of the data (And allow SWRL rules usage later on).

Below were the requirements for the Ontology file, I'll list each one of them and highlight where they are fulfilled with a justified reason:

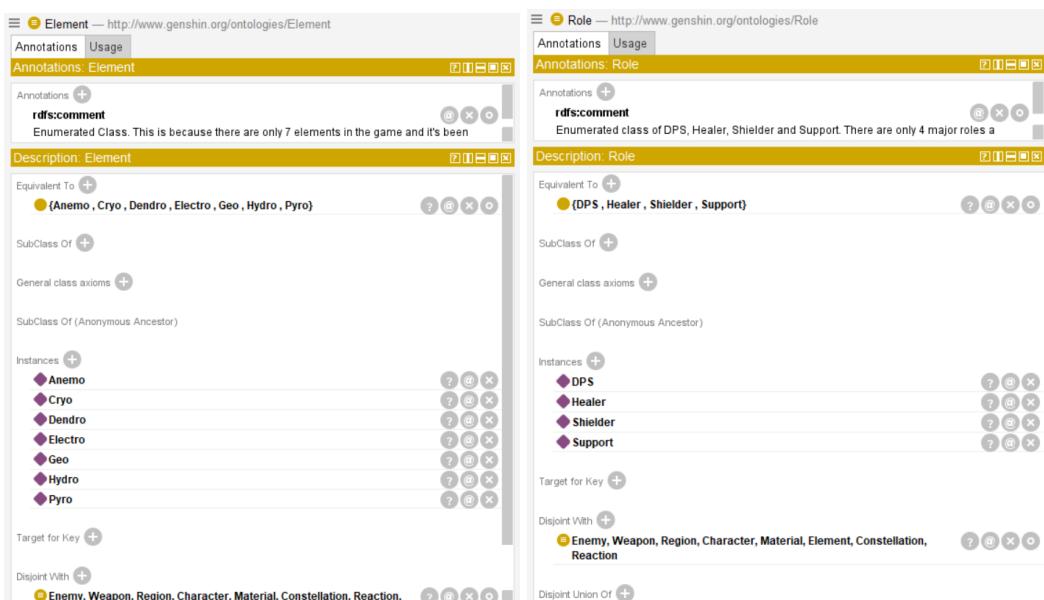
## Ontology 11 Requirements

### 1. At least one class defined as enumeration of its individuals:

In my ontology, four classes were defined as enumerations, Element, Role, EnemyType and WeaponType.

- Elements contain 7 individuals (Pyro, Cryo, Anemo, Geo, Dendro, Electro, Hydro)
- Role contains 4 individuals (DPS, Healer, Shielder, Support).
- EnemyType contains 4 individuals (Normal, Elite, Boss, Weekly Boss)
- WeaponType contains 5 individuals (Spear, Bow, Claymore, Polearm, Catalyst)

It's heavily implied that there will be 7 elements only throughout the entire story and there are 4 major roles in the game currently. This sits well with the game's design since you can have a maximum of 4 characters in your party. EnemyType and WeaponType may increase in the future but it's been 3 years since the game's launch and these have stayed the same.



## 2. At least one class defined using property cardinality restrictions:

All reactions in Genshin happen between at-least 2 elements (and advanced reactions occurring between more than 2 elements. Therefore, cardinality restrictions are placed where each individual needs to be related to ‘requiresElement’ to 2 or more than 2 Elements respectively.

The screenshot shows two ontology classes defined in Protégé:

**Reaction** (<http://www.genshin.org/ontologies/Reaction>)

- Annotations:** rdfs:comment: Reactions are well....reactions caused when two elements combine, creating a new effect. You
- Description:** Reaction
- Equivalent To:** requiresElement min 2 Element
- SubClass Of:** Reaction
- General class axioms:** None
- SubClass Of (Anonymous Ancestor):** None
- Instances:**
  - Burning
  - Electro-Charged
  - Frozen
  - Melt
  - Overloaded
  - Quicken
  - Superconduct
  - Vaporize

**AdvancedReaction** (<http://www.genshin.org/ontologies/AdvancedReaction>)

- Annotations:** rdfs:comment: Those specific reactions where more than 2 elements can participate in
- Description:** AdvancedReaction
- Equivalent To:** requiresElement min 3 Element
- SubClass Of:** Reaction
- General class axioms:** None
- SubClass Of (Anonymous Ancestor):** Reaction
- Instances:**
  - Burgeon
  - Crystallize
  - Hyperbloom
  - Swirl

### 3. At least one class defined using property range restriction

Almost all of my character classes are defined using this method. GeoCharacter, AnemoCharacter, HydroCharacter, CryoCharacter, PyroCharacter, DendroCharacter, GeoCharacter needs to have an individual which is related to “elementAlignment” property on a specific element (e.g. CryoCharacter individual needs to have elementAlignment to cryo), BowUser, ClaymoreUser, PolearmUser, CatalystUser and SwordUser are related in a similar manner.

The image shows two side-by-side ontology editor interfaces. The left interface is for 'CryoCharacter' and the right is for 'PolearmUser'. Both interfaces have tabs for 'Annotations' and 'Usage'. The 'Annotations' tab is active for both. The 'Description' section for CryoCharacter contains the comment: 'A character which has a Cryo Vision'. The 'Equivalent To' section lists 'Character' and '(elementAlignment value Cryo)'. The 'SubClass Of' section lists 'VisionHolder'. The 'General class axioms' section lists 'AnemoCharacter or CryoCharacter or DendroCharacter or ElectroCharacter or GeoCharacter or HydroCharacter or PyroCharacter'. The 'Instances' section lists characters: Aloy, Charlotte, Chongyun, Diona, Eula, Fremenit, Ganyu, Kaeya, Candace, Cyuno, Hu\_Tao, Mika, Raiden\_Shogun, Rosaria, Shenhe, Thoma, and Xiangling. The right interface for 'PolearmUser' has a similar structure. It lists 'Character' and '(usesWeaponType value "Polearm")' in the 'Equivalent To' section, and 'WeaponHolder' in the 'SubClass Of' section. The 'General class axioms' section lists 'BowUser or CatalystUser or ClaymoreUser or PolearmUser or SwordUser'. The 'Instances' section lists characters: BowUser, CatalystUser, ClaymoreUser, PolearmUser, and SwordUser.

### 4. At least one class defined as a union of classes

Vision holder class is a union of property restricted class of characters of each element.

Similarly, Weapon holder is a union of property restricted class of character of each weapon type:

The image shows two side-by-side ontology editor interfaces. The left interface is for 'VisionHolder' and the right is for 'WeaponHolder'. Both interfaces have tabs for 'Annotations' and 'Usage'. The 'Annotations' tab is active for both. The 'Description' section for VisionHolder contains the comment: 'Any character which has a vision (visions are basically elemental affinities given to characters)'. The 'Equivalent To' section lists 'Character' and '(elementAlignment value AnemoCharacter or CryoCharacter or DendroCharacter or ElectroCharacter or GeoCharacter or HydroCharacter or PyroCharacter)'. The right interface for 'WeaponHolder' has a similar structure. It lists 'Character' and '(usesWeaponType value "BowUser or CatalystUser or ClaymoreUser or PolearmUser or SwordUser")' in the 'Equivalent To' section, and 'Character' in the 'SubClass Of' section. The 'General class axioms' section lists 'BowUser or CatalystUser or ClaymoreUser or PolearmUser or SwordUser'. The 'Instances' section lists characters: BowUser, CatalystUser, ClaymoreUser, PolearmUser, and SwordUser.

Materials class is also a union of CharacterUpgradeMaterials and WeaponUpgradeMaterials.

Similarly, Weapon class is a union of all people weapon type classes:

The screenshot shows two ontology classes side-by-side. On the left is the 'Material' class, which has annotations for 'rdfs:comment' stating 'Materials are resources dropped by enemies. Used to level characters or weapons'. It also lists 'CharacterUpgradeMaterial or WeaponUpgradeMaterial' as its equivalents. On the right is the 'Weapon' class, which has annotations for 'rdfs:comment' stating 'Weapons wielded by characters'. It lists 'Bow or Catalyst or Claymore or Polearm or Sword' as its equivalents. Both classes have tabs for 'Annotations' and 'Usage' at the top.

## 5. At least one class defined as an intersection of classes

PlayableCharacters is one such class where if a character is both a vision holder and wields a weapon, they are deemed as playable characters. (Since all playable characters in Genshin hold a vision and are able to wield a weapon)

The screenshot shows the 'PlayableCharacter' class. It has annotations for 'rdfs:comment' stating 'Every playable character in genshin has a vision and has some preferred weapon they like'. It lists 'VisionHolder and WeaponHolder' as its equivalents. In the 'SubClass Of' section, it shows 'Character', 'VisionHolder', and 'WeaponHolder' as its subclasses. It also lists 'AnemoCharacter or CryoCharacter or DendroCharacter or ElectroCharacter or GeoCharacter or HydroCharacter or PyroCharacter' and 'BowUser or CatalystUser or ClaymoreUser or PolearmUser or SwordUser' as its subclasses of anonymous ancestors.

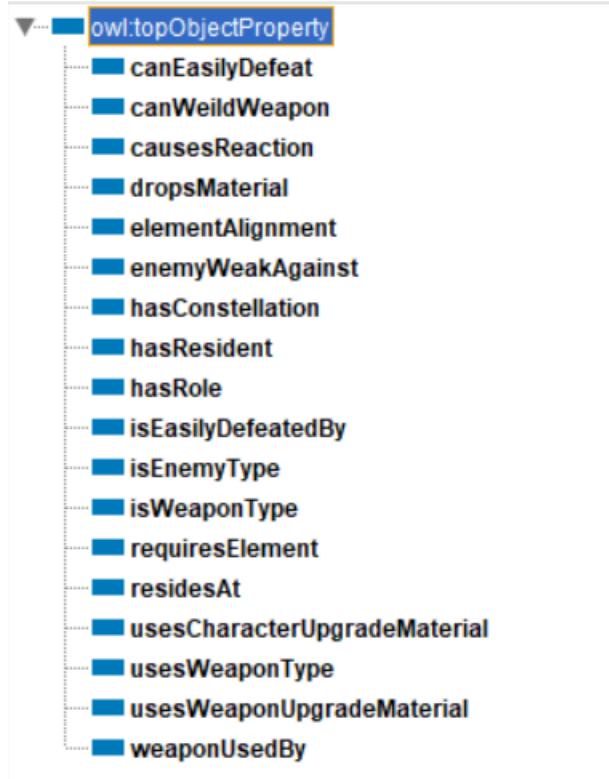
## 6. At least one class defined as a complement of classes

Here, we have other materials defined as those materials which are not character upgrade materials nor weapon upgrade materials:

The screenshot shows the 'OtherMaterial' class. It has annotations for 'rdfs:comment' stating 'Any material that isn't a weapon upgrade material nor a character upgrade material'. It lists 'Material and (not (CharacterUpgradeMaterial and WeaponUpgradeMaterial))' as its equivalents. It has tabs for 'Annotations' and 'Usage' at the top.

## 7. At least 7 object properties in total

Here you go :D. There's annotations for each object property explaining it too in the RDF/XML file.



## 8. At least one object property should be functional

ElementAlignment is a functional property. It's made functional because a character cannot have more than 1 element. If they do, it should mean that the two element instances are the same individuals. This is a fact imposed by the story of Genshin for all of its characters. Even the archons have only 1 element aligned to them.

Annotations: elementAlignment

rdfs:comment

Any character having an element. It's made functional because a character cannot have more than 1 element. If they do, it should mean that the two element instances are the same individuals.

Char Description: elementAlignment

Functional

## 9. At least one object property should be inverse functional

hasConstellation is an inverse functional property. Every character has their own unique constellation in the game (To be more in-depth without spoiling, it represents ‘them’ in the ‘sky’). Therefore, a constellation cannot belong to more than 1 character. If that case ever happens, then both character individuals are to be inferred as the same individual.

The screenshot shows the 'Annotations' tab for the 'hasConstellation' property. The 'rdfs:comment' field contains: 'Any character that has a constellation. This is mentioned in the database and has been made inverse functional because a constellation cannot belong to more than 1 character. If that case ever happens, then both character individuals are the same individual.' The 'Description' tab shows the following restrictions:

- Equivalent To: None
- SubProperty Of: None
- Inverse Of: None
- Domains (intersection): Character
- Ranges (intersection): Constellation

## 10. At least 3 object properties should have some range restrictions

Here you go! I'll show 4 of them but a lot of my object properties have range and domains restrictions:

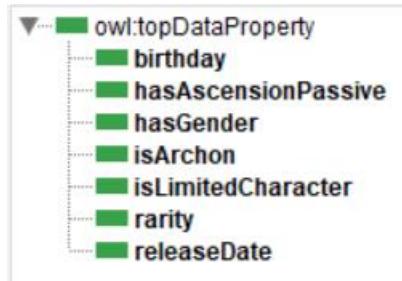
The screenshots show four properties with their respective range restrictions:

- canEasilyDefeat**: Range is Enemy.
- canWeildWeapon**: Range is Weapon.
- causesReaction**: Range is Reaction.
- dropsMaterial**: Range is Material.

Each screenshot displays the 'Annotations' and 'Description' tabs with their respective details and restrictions.

## 11. At least 7 datatype properties in total

Here you go!



# Mapping

Here are the 4 databases I used to map into my RDF/XML ontology:

Character Table (Displaying only 5 rows)

	character_name	rarity	region	vision	weapon_type	constellation	birthday	ascension_boss	limited	ascension	role	gender
0	Albedo	5	Mondstadt	Geo	Sword	Princeps_Cretaceus	2023-09-13	Basalt_Pillar	True	Geo_DMG_Bonus	Support	Male
1	Alhaitham	5	Sumeru	Dendro	Sword	Vultur_Volans	2023-02-11	Pseudo-Stamens	True	Dendro_DMG_Bonus	DPS	Male
2	Aloy	5	Mondstadt	Cryo	Bow	Nora_Fortis	2023-04-04	Crystalline_Bloom	True	Cryo_DMG_Bonus	Support	Female
3	Amber	4	Mondstadt	Pyro	Bow	Lepus	2023-08-10	Everflame_Seed	False	ATK	Support	Female
4	Arataki_Itto	5	Inazuma	Geo	Claymore	Taurus_Iracundus	2023-06-01	Riftborn_Regalia	True	CRIT_Rate	DPS	Male

Reaction Table (Displaying only 5 rows)

	Element	causesReaction
0	Pyro	Overloaded
1	Electro	Overloaded
2	Pyro	Melt
3	Cryo	Melt
4	Pyro	Vaporize

Weapons Table (Displaying only 5 rows)

	Name	Type	Rarity	UpgradeMaterial
0	Wolf's_Gravestone	Claymore	5	Tusk_of_the_Wolf
1	Skyward_Harp	Bow	5	Luminous_Sands_from_Guyun
2	Primordial_Jade_Winged-Spear	Polearm	5	Jade_Wing
3	Aquila_Favonia	Sword	5	Tusk_of_the_Wolf
4	Lost_Prayer_to_the_Sacred_Winds	Catalyst	5	Tile_of_Decarabian's_Tower

Enemies Table (Displaying only 5 rows)

	Name	Element	WeakAgainst	EnemyType	DropsMaterial
0	Hilichurl	Anemo	Electro	Normal	Firm_Arrowhead
1	Anemo_Samachurl	Anemo	Electro	Elite	Divining_Scroll
2	Hydro_Samachurl	Hydro	Pyro	Elite	Divining_Scroll
3	Cryo_Samachurl	Cryo	Pyro	Elite	Divining_Scroll
4	Mitachurl	Anemo	Electro	Elite	Firm_Arrowhead

First I parsed my ontology using RDFLib library, and then, it was just a matter of adding the right triples into the graph. Here's an example of what it looks like for Characters:

```
for index, row in df.iterrows():
    name = URIRef(gRes+row['character_name'])
    constellation = URIRef(gRes+row['constellation'])
    characterMaterial = URIRef(gRes+row['ascension_boss'])
    role = URIRef(gRes+row['role'])

    g.add((name, RDF.type, URIRef(gOnto + "Character")))
    g.add((name, RDF.type, OWL.NamedIndividual))
    g.add((name, FOAF.name, Literal(row['character_name'])))
    g.add((name, RDFS.label, Literal(row['character_name']), datatype = XSD.string))

    g.add((constellation, RDF.type, URIRef(gOnto + "Constellation")))
    g.add((constellation, RDF.type, OWL.NamedIndividual))
    g.add((constellation, RDFS.label, Literal(row['constellation']), datatype = XSD.string))

    g.add((characterMaterial, RDF.type, URIRef(gOnto + "Material")))
    g.add((characterMaterial, RDF.type, OWL.NamedIndividual))
    g.add((characterMaterial, RDFS.label, Literal(row['ascension_boss']), datatype = XSD.string))

    g.add((role, RDF.type, URIRef(gOnto + "Role")))
    g.add((role, RDF.type, OWL.NamedIndividual))
    g.add((role, RDFS.label, Literal(row['role']), datatype = XSD.string))

    g.add((name, URIRef(gOnto + "rarity"), Literal(row['rarity']), datatype = XSD.integer))
    g.add((name, URIRef(gOnto + "residesAt"), URIRef(gRes + row['region'])))
    g.add((name, URIRef(gOnto + "elementAlignment"), URIRef(gRes + row['vision'])))
    g.add((name, URIRef(gOnto + "usesWeaponType"), Literal(row['weapon_type']), datatype= XSD.string))
    g.add((name, URIRef(gOnto + "hasConstellation"), constellation))
    g.add((name, URIRef(gOnto + "birthday"), Literal(row['birthday']), datatype= XSD.dateTime))
    g.add((name, URIRef(gOnto + "usesCharacterUpgradeMaterial"), characterMaterial))
    g.add((name, URIRef(gOnto + "isLimitedCharacter"), Literal(row['limited']), datatype = XSD.boolean))
    g.add((name, URIRef(gOnto + "hasAscensionPassive"), Literal(row['ascension']), datatype = XSD.string))
    g.add((name, URIRef(gOnto + "hasRole"), role))
    g.add((name, URIRef(gOnto + "hasGender"), Literal(row['gender']), datatype = XSD.string))
```

After every time I added a table, I saved the updated ontology into a new file:

```
1 g.serialize(destination="afterAddingCharacters.rdf", format="application/rdf+xml")
```

## Linking

Genshin Impact characters from Wikidata were linked with my characters in my ontology.

Unfortunately, Wikidata didn't contain all of the characters so only some characters could be linked to there.

### Linking with Wikidata

```
: 1 # Keqing
 2 character = URIRef(gRes + 'Keqing')
 3 wikidataLink = URIRef(wd + 'Q107287676')
 4 g.add((character, OWL.sameAs, wikidataLink))
 5
 6 # Yun Jin
 7 character = URIRef(gRes + 'Yun_Jin')
 8 wikidataLink = URIRef(wd + 'Q110653384')
 9 g.add((character, OWL.sameAs, wikidataLink))
10
11 # Shenhe
12 character = URIRef(gRes + 'Shenhe')
13 wikidataLink = URIRef(wd + 'Q110677319')
14 g.add((character, OWL.sameAs, wikidataLink))
15
16 # Ganyu
17 character = URIRef(gRes + 'Ganyu')
18 wikidataLink = URIRef(wd + 'Q107618618')
19 g.add((character, OWL.sameAs, wikidataLink))
20
21 # Diluc
22 character = URIRef(gRes + 'Diluc')
23 wikidataLink = URIRef(wd + 'Q107263978')
24 g.add((character, OWL.sameAs, wikidataLink))
25
26 # Hu Tao
27 character = URIRef(gRes + 'Barbara')
28 wikidataLink = URIRef(wd + 'Q110911467')
29 g.add((character, OWL.sameAs, wikidataLink))
30
31 # Yanfei
32 character = URIRef(gRes + 'Yanfei')
33 wikidataLink = URIRef(wd + 'Q110965539')
34 g.add((character, OWL.sameAs, wikidataLink))
35
36 # Amber
37 character = URIRef(gRes + 'Amber')
38 wikidataLink = URIRef(wd + 'Q111008775')
39 g.add((character, OWL.sameAs, wikidataLink))
40
```

As such, 30 characters were successfully linked with Wikidata

# Reasoning

## SWRL Rules

### Rule 1-5:

I wanted to make some inferences that if there is a character belonging to BowUser defined class, then it should be able to wield every single instance of Bow class. Unfortunately, OWL didn't have the expressivity to allow this so I had to use some SWRL rules to make these inferences possible. Here's how the above situation looks like as a SWRL rule:

```
genshinOntology:Bow(?b) ^ genshinOntology:BowUser(?a)  
-> genshinOntology:canWeildWeapon(?a, ?b)
```

As such, 5 rules were made like this, each one trying to make inferences between specific weapon types and their respective holders.

### Rule 6:

Another place SWRL rule was implemented was to answer one of my competency question: "What characters can easily defeat X enemy?"

I had the weakness element of the enemy and each character's element alignment. All I had to do in SWRL was to make such question's relation a possibility. Here's how it looks like:

```
genshinOntology:elementAlignment(?character, ?element) ^ genshinOntology:enemyWeakAgainst(?enemy, ?element) ^  
genshinOntology:Character(?character) ^ genshinOntology:Enemy(?enemy) ->  
genshinOntology:canEasilyDefeat(?character, ?enemy)
```

It's a bit long, I know. Class type checks had to be in place so I wouldn't get any information where "Enemy" is weak against "Enemy". Since enemy shouldn't be the domain of this property.

## Reasoning

After running the reasoner, we notice a lot of stuff happening. Here are all the places where the reasoner inferred something (SWRL rule reasonings are at the bottom):

The screenshot shows the 'Annotations' tab for the 'AnemoCharacter' ontology. It contains the following information:

- rdfs:comment:** A character which has an Anemo Vision.
- Description:** AnemoCharacter
- Instances:** A list of characters: Aether, Faruzan, Jean, Kaedehara\_Kazuha, Lumine, Lynette, Sayu, Shikanoin\_Heizou, Sucrose, Venti, Wanderer, Xiao. Each character name is preceded by a purple diamond icon.
- Each character entry has three small circular icons to its right: a question mark, an '@' symbol, and an 'X' symbol.

For element specific character classes, it perfectly reasoned which characters should belong into which elemental class

The screenshot shows the 'Annotations' tab for the 'BowUser' ontology. It contains the following information:

- rdfs:comment:** A character which uses a Bow.
- Description:** BowUser
- Instances:** A list of characters: Aloy, Amber, Collei, Diona, Faruzan, Fischl, Ganyu, Gorou, Kujou\_Sara, Lyney, Tartaglia, Tighnari, Venti, Yelan, Yoimiya. Each character name is preceded by a purple diamond icon.
- Each character entry has three small circular icons to its right: a question mark, an '@' symbol, and an 'X' symbol.

For weapon specific character classes, it perfectly reasoned on which characters wield a bow and thus, are now individuals of BowUser

The screenshot shows the Protege ontology editor interface for the `DamageDealer` class. The main pane displays the following details:

- Annotations:** `rdfs:comment` - Prioritizes in dealing damage to the enemy.
- Description:** DamageDealer
- Equivalent To:** PlayableCharacter and (hasRole value DPS)
- SubClass Of:** PlayableCharacter
- General class axioms:** None listed.
- SubClass Of (Anonymous Ancestor):**
  - AnemoCharacter or CryoCharacter or DendroCharacter or ElectroCharacter or GeoCharacter or HydroCharacter or PyroCharacter
  - BowUser or CatalystUser or ClaymoreUser or PolarmUser or SwordUser
  - VisionHolder and WeaponHolder
- Instances:**
  - Aether
  - Alhautham
  - Arataki\_Ito
  - Cyno
  - Diluc

According to the dataset, it also reasoned perfectly on which characters are classified as damage dealers

The screenshot shows the Protege ontology editor interface for the `Boss` class. The main pane displays the following details:

- Annotations:** `rdfs:comment` - Very tough enemies (They even have a health bar shown on the screen).
- Description:** Boss
- Equivalent To:** Enemy and (enemyType value "Boss")
- SubClass Of:** Enemy
- General class axioms:** None listed.
- SubClass Of (Anonymous Ancestor):**
  - enemyType some xsd:string
- Instances:**
  - Anemo\_Hypostasis
  - Cryo\_Regisvine
  - Electro\_Hypostasis
  - Geo\_Hypostasis
  - Pyro\_Regisvine

It also reasoned perfectly on which enemies should be considered as individuals of boss class

**Annotations:** Weekly\_Boss

**rdfs:comment**  
They are bosses but are related to the story. You can claim rewards from them once per week,

**Description:** Weekly\_Boss

**Equivalent To**

- Enemy  
and (enemyType value "Weekly\_Boss")

**SubClass Of**

- Boss

**General class axioms**

**SubClass Of (Anonymous Ancestor)**

- enemyType some xsd:string
- Enemy  
and (enemyType value "Boss")

**Instances**

- Andrius
- Azhdaha
- Childe
- Dvalin

Similarly, it reasoned perfectly on which enemies are of weekly boss types too :D

**Annotations:** CharacterUpgradeMaterial

**rdfs:comment**  
A material which is used to upgrade characters

**Description:** CharacterUpgradeMaterial

**Equivalent To**

**SubClass Of**

- Material

**General class axioms**

**SubClass Of (Anonymous Ancestor)**

- CharacterUpgradeMaterial or WeaponUpgradeMaterial

**Instances**

- Artificed\_Spare\_Clockwork\_Component\_-Coppelia
- Artificed\_Spare\_Clockwork\_Component\_-Coppelius
- Basalt\_Pillar
- Cleansing\_Heart
- Crystalline\_Bloom
- Dew\_of\_Reputation
- Dragonheir's\_False\_Fin
- Emperor's\_Resolution
- Everflame\_Seed
- Evergloom\_Ring
- Fontemer\_Unhorn

From the materials and characters using those materials, it inferred which materials belong under the CharacterUpgradeMaterials class

**Annotations:** `rdfs:comment`  
A material which is used to upgrade weapons

**Description:** WeaponUpgradeMaterial

**Equivalent To:** Material

**General class axioms:**

**SubClass Of (Anonymous Ancestor):** CharacterUpgradeMaterial or WeaponUpgradeMaterial

**Instances:**

- Agnidus\_Agate\_Sliver
- Basalt\_Pillar
- Broken\_Fang
- Crystalline\_Bloom
- Fragment\_of\_Decarabian's\_Epic
- Guyun\_Stone\_Forest
- Heavy\_Horn
- Jade\_Wing
- Luminous\_Sands\_from\_Guyun
- Lustrous\_Stone\_from\_Guyun
- Mist\_Veiled\_Lead\_Elixir

From the materials and weapons using those materials, it inferred which materials belong under the `WeaponUpgradeMaterials` class

**Annotations:** `rdfs:comment`  
Reactions are well...reactions caused when two elements combine, creating a new effect. You

**Description:** Reaction

**Equivalent To:** requiresElement min 2 Element

**SubClass Of:**

**General class axioms:**

**SubClass Of (Anonymous Ancestor):**

**Instances:**

- Burning
- Electro-Charged
- Frozen
- Melt
- Overloaded
- Quicken
- Superconduct
- Vaporize

It also inferred correctly on Reactions

**Annotations:** AdvancedReaction

**rdfs:comment**  
Those specific reactions where more than 2 elements can participate in

**Description:** AdvancedReaction

**Equivalent To**

**SubClass Of**

**General class axioms**

**SubClass Of (Anonymous Ancestor)**

**Instances**

- Burgeon
- Crystallize
- Hyperbloom
- Swirl

As well as advanced reactions :)

**Annotations:** Region

**rdfs:comment**  
Countries where characters usually reside in

**Description:** Region

**Equivalent To**

**SubClass Of**

**General class axioms**

**SubClass Of (Anonymous Ancestor)**

**Instances**

- Fontaine
- Inazuma
- Liyue
- Mondstadt
- Snezhnaya
- Sumeru

It also inferred the regions automatically

**Bow — http://www.genshin.org/ontologies/Bow**

**Annotations:** Bow

**rdfs:comment**  
Represents any bow in the game

**Description: Bow**

**Equivalent To**  
**Weapon**  
and (isWeaponType value "Bow")

**SubClass Of**  
**Weapon**

**General class axioms**

**SubClass Of (Anonymous Ancestor)**  
**Bow or Catalyst or Claymore or Polearm or Sword**  
**Bow or Catalyst or Claymore or Polearm or Sword**

**Instances**

- Compound\_Bow
- Favonius\_Warbow
- Rust
- Skyward\_Harp
- The\_Stringless
- The\_Viridescent\_Hunt

**Sword — http://www.genshin.org/ontologies/Sword**

**Annotations:** Sword

**rdfs:comment**  
Represents any sword in the game

**Description: Sword**

**Equivalent To**  
**Weapon**  
and (isWeaponType value "Sword")

**SubClass Of**  
**Weapon**

**General class axioms**

**SubClass Of (Anonymous Ancestor)**  
**Bow or Catalyst or Claymore or Polearm or Sword**  
**Bow or Catalyst or Claymore or Polearm or Sword**

**Instances**

- Aquila\_Favonia
- Lion's\_Roar
- Prototype\_Rancour
- Sacrificial\_Sword
- Sword\_of\_Descension
- The\_Black\_Sword

It also correctly inferred on weapons being classified into their respective weapon categories!

## SWRL Reasoning

Description: Anemo_Hypostasis		Property assertions: Anemo_Hypostasis	
Types +			
Enemy	?	?	?
Boss	@	?	@
Same Individual As +			
Different Individuals +			
Description: Barbara		Property assertions: Barbara	
Types +		Object property assertions +	
Character	?	hasRole Healer	?
CatalystUser	@	hasConstellation Crater	@
Healer	?	usesCharacterUpgradeMaterial Cleansing_Heart	?
HydroCharacter	@	residesAt Mondstadt	@
Same Individual As +		elementAlignment Hydro	@
Different Individuals +		canEasilyDefeat Large_Electro_Slime	@
		canEasilyDefeat Electro_Hypostasis	@
		canEasilyDefeat Electro_Slime	@
		canEasilyDefeat Dendro_Slime	@
		canEasilyDefeat Electro_Abyss_Mage	@
		canWeildWeapon Solar_Pearl	@
		canWeildWeapon Mappa_Mare	@
		canWeildWeapon Wine_and_Song	@
		canWeildWeapon Lost_Prayer_to_the_Sacred_Winds	@
		canWeildWeapon Widsith	@
Description: Aquila_Favonia		Property assertions: Aquila_Favonia	
Types +		Object property assertions +	
Weapon	?	usesWeaponUpgradeMaterial Tusk_of_the_Wolf	?
Sword	@	weaponUsedBy Furina	@
Same Individual As +		weaponUsedBy Kamisato_Ayato	@
Different Individuals +		weaponUsedBy Layla	@
		weaponUsedBy Kirara	@
		weaponUsedBy Xingqiu	@
		weaponUsedBy Nilou	@
		weaponUsedBy Lumine	@
		weaponUsedBy Kaeya	@
		weaponUsedBy Alhaitham	@
		weaponUsedBy Kuki_Shinobu	@
		weaponUsedBy Jean	@
		weaponUsedBy Keqing	@
		weaponUsedBy Aether	@

As we can see, SWRL rules allowed such automatic inferences to happen between individuals. This wasn't possible in OWL because automating relations between instances of 2 different classes (like a dense neural network connection between 2 layers) wasn't possible with traditional owl expressivity.

**Note:** After mapping and linking, we saved everything into an RDF+XML file called the ‘afterAddingEverything.rdf’. Then we loaded our RDF+XML file into Protege and then we started the reasoner. After triples were reasoned, we saved the file into another RDF+XML file called ‘inferredKnowledgeAdded.rdf’.

## Linked Data Reasoning

The screenshot shows the Protege reasoning interface with two main panels: 'Types' and 'Object property assertions'.

**Types Panel:**

- Types: PolearmUser, PyroCharacter, Shielder
- Same Individual As: Thoma
- Different Individuals: (empty)

**Object property assertions Panel:**

- canEasilyDefeat Cryo\_Abyss\_Mage
- canEasilyDefeat Cryo\_Regisvine
- canEasilyDefeat Cryo\_Slime
- canEasilyDefeat Large\_Cryo\_Slime
- canEasilyDefeat Cryo\_Samachurl
- canEasilyDefeat Andrius
- canEasilyDefeat Hydro\_Samachurl
- canWeildWeapon Crescent\_Pike
- canWeildWeapon Dragonl's\_Bane
- canWeildWeapon Prototype\_Starglitter
- canWeildWeapon Primordial\_Jade\_Winged-Spear
- canWeildWeapon Royal\_Spear
- canWeildWeapon Deathmatch
- elementAlignment Pyro
- hasConstellation Rubrum\_Scutum

Here you can see the Wikidata entry for Thoma was successfully linked with our Thoma character in the ontology file.

# Publishing

## GraphDB

We published the data into GraphDB and Fuseki to create a SPARQL endpoint for our ontology. Here is how it looks like:

### Active repository

The screenshot shows the GraphDB Active repository interface. At the top, it says "Local" and "GenshinImpact · Genshin Impact Knowledge Graph (after)". Below that, it displays the following statistics:

- total statements **10,995**
- 7,910 explicit
- 3,085 inferred
- 1.39 expansion ratio

At the bottom, there are two buttons: "Import RDF data" and "Export RDF data".

The screenshot shows the GraphDB SPARQL Query & Update interface. The query entered is:

```
PREFIX ontology: <http://www.genshin.org/ontology/>
PREFIX wdt: <http://www.wikidata.org/prop/direct/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT ?characterName ?voiceActor ?voiceActorName
WHERE {
  ?character a ontology:Character;
    rdfs:label ?characterName.
  SERVICE <https://query.wikidata.org/sparql> {
    ?character wdt:P728 ?voiceActor.
    ?voiceActor rdfs:label ?voiceActorName.
    FILTER(lang(?voiceActorName) = "en")
  }
}
```

The results table shows the following data:

characterName	voiceActor	voiceActorName
1 "Shenhe"	http://www.wikidata.org/entity/Q4956	"Ayako Kawasumi"@en
2 "Fischl"	http://www.wikidata.org/entity/Q44552	"Maaya Uchida"@en
3 "Raiden_Shogun"	http://www.wikidata.org/entity/Q49552	"Miyuki Sawashiro"@en
4 "Lisa"	http://www.wikidata.org/entity/Q237371	"Rie Tanaka"@en
5 "Thoma"	http://www.wikidata.org/entity/Q361574	"Masakazu Morita"@en

## Fuseki

The screenshot shows the Apache Jena Fuseki interface at [localhost:3030/](http://localhost:3030/). The main header includes the Apache Jena logo, navigation links for datasets, manage, and help, and a server status indicator. Below the header, the title "Apache Jena Fuseki" and version information "Version 4.10.0, Uptime 2m 01s" are displayed. A search bar and a "Clear" button are also present. The main content area shows a table with one row for the dataset "/inferredKnowledgeAdded". The table has columns for "name" and "actions". The "actions" column contains four buttons: "query", "add data", "edit", and "info". Below the table is a page navigation bar with buttons for "«", "‹", "1", "›", and "»".

The screenshot shows the Apache Jena Fuseki SPARQL endpoint interface at [localhost:3030/#/dataset/inferredKnowledgeAdded/query](http://localhost:3030/#/dataset/inferredKnowledgeAdded/query). The top navigation bar includes the Apache Jena logo, a search bar, and various icons. The main area is divided into two sections: "Selection of triples" and "Selection of classes". The "Selection of triples" section contains a SPARQL query editor with the following code:

```
1: PREFIX rdfs: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2: PREFIX owl: <http://www.w3.org/2000/01/rdf-schema#>
3: SELECT * WHERE {
4: ?sub ?pred ?obj .
5: } LIMIT 10
```

The "Content Type (SELECT)" dropdown is set to "JSON", and the "Content Type (GRAPH)" dropdown is set to "Turtle". Below the query editor is a table titled "Table" showing the results of the query. The table has columns "sub", "pred", and "obj". There are 10 rows of results, each corresponding to a triple from the query. The last few rows show inferred data from SWRL rules.

sub	pred	obj
1 < <a href="http://www.genshin.org/ontology/">http://www.genshin.org/ontology/</a> >	< <a href="http://www.w3.org/1999/02/22-rdf-syntax-ns#owl#Ontology">http://www.w3.org/1999/02/22-rdf-syntax-ns#owl#Ontology</a> >	< <a href="http://www.w3.org/2002/07/owl#Ontology">http://www.w3.org/2002/07/owl#Ontology</a> >
2 < <a href="http://swrl.stanford.edu/ontologies/3.3/swrl.owl#isRule">http://swrl.stanford.edu/ontologies/3.3/swrl.owl#isRule</a> >	< <a href="http://www.w3.org/1999/02/22-rdf-syntax-ns#owl#AnnotationProperty">http://www.w3.org/1999/02/22-rdf-syntax-ns#owl#AnnotationProperty</a> >	< <a href="http://www.w3.org/2002/07/owl#AnnotationProperty">http://www.w3.org/2002/07/owl#AnnotationProperty</a> >
3 < <a href="http://xmlns.com/foaf/0.1/name">http://xmlns.com/foaf/0.1/name</a> >	< <a href="http://www.w3.org/1999/02/22-rdf-syntax-ns#owl#AnnotationProperty">http://www.w3.org/1999/02/22-rdf-syntax-ns#owl#AnnotationProperty</a> >	< <a href="http://www.w3.org/2002/07/owl#AnnotationProperty">http://www.w3.org/2002/07/owl#AnnotationProperty</a> >
4 < <a href="http://www.genshin.org/ontologies/canEasilyDefeat">http://www.genshin.org/ontologies/canEasilyDefeat</a> >	< <a href="http://www.w3.org/1999/02/22-rdf-syntax-ns#owl#ObjectProperty">http://www.w3.org/1999/02/22-rdf-syntax-ns#owl#ObjectProperty</a> >	< <a href="http://www.w3.org/2002/07/owl#ObjectProperty">http://www.w3.org/2002/07/owl#ObjectProperty</a> >
5 < <a href="http://www.genshin.org/ontologies/canEasilyDefeat">http://www.genshin.org/ontologies/canEasilyDefeat</a> >	< <a href="http://www.w3.org/2002/07/owl#inverseOf">http://www.w3.org/2002/07/owl#inverseOf</a> >	< <a href="http://www.genshin.org/ontologies/isEasilyDefeatedBy">http://www.genshin.org/ontologies/isEasilyDefeatedBy</a> >
6 < <a href="http://www.genshin.org/ontologies/canEasilyDefeat">http://www.genshin.org/ontologies/canEasilyDefeat</a> >	< <a href="http://www.w3.org/2000/01/rdf-schema#dou">http://www.w3.org/2000/01/rdf-schema#dou</a> >	< <a href="http://www.genshin.org/ontologies/Character">http://www.genshin.org/ontologies/Character</a> >
7 < <a href="http://www.genshin.org/ontologies/canEasilyDefeat">http://www.genshin.org/ontologies/canEasilyDefeat</a> >	< <a href="http://www.w3.org/2000/01/rdf-schema#fra">http://www.w3.org/2000/01/rdf-schema#fra</a> >	< <a href="http://www.genshin.org/ontologies/enemy">http://www.genshin.org/ontologies/enemy</a> >
8 < <a href="http://www.genshin.org/ontologies/canEasilyDefeat">http://www.genshin.org/ontologies/canEasilyDefeat</a> >	< <a href="http://www.w3.org/2000/01/rdf-schema#co">http://www.w3.org/2000/01/rdf-schema#co</a> >	Inferred from SWRL rules. Only given to those characters which can easily defeat an enemy given that the enemy is weak to them.
9 < <a href="http://www.genshin.org/ontologies/isEasilyDefeatedBy">http://www.genshin.org/ontologies/isEasilyDefeatedBy</a> >	< <a href="http://www.w3.org/1999/02/22-rdf-syntax-ns#owl#ObjectProperty">http://www.w3.org/1999/02/22-rdf-syntax-ns#owl#ObjectProperty</a> >	< <a href="http://www.w3.org/2002/07/owl#ObjectProperty">http://www.w3.org/2002/07/owl#ObjectProperty</a> >
10 < <a href="http://www.genshin.org/ontologies/isEasilyDefeatedBy">http://www.genshin.org/ontologies/isEasilyDefeatedBy</a> >	< <a href="http://www.w3.org/2000/01/rdf-schema#co">http://www.w3.org/2000/01/rdf-schema#co</a> >	SWRL Rule. Pair all enemies which are easily defeated by a character

## Validation Using SPARQL Endpoint

Here are the 9 ( + 1 bonus ) competency questions we expected our graph to answer. I added the bonus in during deliverable 4 because I was able to do Federated Queries using my graph

### Q1. What materials do I need to level up X character? E.g Kaeya

The screenshot shows a SPARQL endpoint interface. On the left, there is a sidebar with ten competency tabs labeled Competency 1 through Competency 10. Below these tabs is the SPARQL query:

```
+ 1 PREFIX ontology: <http://www.genshin.org/ontology/>
  2 PREFIX resource: <http://www.genshin.org/resource/>
  3 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
  4 PREFIX foaf: <http://xmlns.com/foaf/0.1/>
  5
  6 # Q1. What material do I need to level up X character? E.g Kaeya
  7 SELECT ?character ?characterName ?materialName {
  8   ?material rdfs:label ?materialName;
  9   a ontology:Material.
 10  ?character ontology:usesCharacterUpgradeMaterial ?material;
 11  a ontology:Character;
 12  foaf:name ?characterName.
 13  FILTER(?character = resource:Kaeya).
 14 }
 15
 16 }
```

On the right, the results are displayed in a table:

character	characterName	materialName
1 http://www.genshin.org/resource/Kaeya	"Kaeya"	"Hoarfrost_Core"

### Q2. What materials do I need to level up X weapon? E.g Favonius Warbow

The screenshot shows a SPARQL endpoint interface. On the left, there is a sidebar with ten competency tabs labeled Competency 1 through Competency 10. Below these tabs is the SPARQL query:

```
+ 1 PREFIX ontology: <http://www.genshin.org/ontology/>
  2 PREFIX resource: <http://www.genshin.org/resource/>
  3 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
  4 PREFIX foaf: <http://xmlns.com/foaf/0.1/>
  5
  6 # Q2. What materials do I need to level up X weapon? E.g Favonius Warbow
  7 SELECT ?weapon ?weaponName ?materialName {
  8   ?material rdfs:label ?materialName;
  9   a ontology:Material.
 10  ?weapon ontology:usesWeaponUpgradeMaterial ?material;
 11  a ontology:Weapon;
 12  foaf:name ?weaponName.
 13  FILTER(?weapon = resource:Favonius_Warbow).
 14 }
 15
 16 }
```

On the right, the results are displayed in a table:

weapon	weaponName	label
1 http://www.genshin.org/resource/Favonius_Warbow	"Favonius_Warbow"	"Basalt_Pillar"

### Q3. List all characters of X element. E.g Hydro

The screenshot shows a SPARQL query interface with the following details:

- Query Editor:** Contains a SPARQL query to list characters of the Hydro element.
- Results Table:** Shows 15 rows of data where each row maps a character to their name and the element they belong to.
- Buttons:** Includes standard table operations like copy, paste, and refresh, along with download and run buttons.

	character	characterName	element
1	http://www.genshin.org/resource/Furina	"Furina"	http://www.genshin.org/resource/Hydro
2	http://www.genshin.org/resource/Kamisato_Ayato	"Kamisato_Ayato"	http://www.genshin.org/resource/Hydro
3	http://www.genshin.org/resource/Nilou	"Nilou"	http://www.genshin.org/resource/Hydro
4	http://www.genshin.org/resource/Xingqiu	"Xingqiu"	http://www.genshin.org/resource/Hydro
5	http://www.genshin.org/resource/Barbara	"Barbara"	http://www.genshin.org/resource/Hydro
6	https://www.wikidata.org/wiki/Q110911467	"Barbara"	http://www.genshin.org/resource/Hydro
7	http://www.genshin.org/resource/Candace	"Candace"	http://www.genshin.org/resource/Hydro
8	https://www.wikidata.org/wiki/Q121919991	"Candace"	http://www.genshin.org/resource/Hydro
9	http://www.genshin.org/resource/Tartaglia	"Tartaglia"	http://www.genshin.org/resource/Hydro
10	http://www.genshin.org/resource/Yelan	"Yelan"	http://www.genshin.org/resource/Hydro
11	https://www.wikidata.org/wiki/Q117195714	"Yelan"	http://www.genshin.org/resource/Hydro
12	http://www.genshin.org/resource/Mona	"Mona"	http://www.genshin.org/resource/Hydro

### Q4. Which element is involved in the most reactions?

The screenshot shows a SPARQL query interface with the following details:

- Query Editor:** Contains a SPARQL query to find the element with the highest reaction count.
- Results Table:** Shows 1 row of data indicating that the element 'Electro' is involved in the most reactions.
- Buttons:** Includes standard table operations like copy, paste, and refresh, along with download and run buttons.

	element	reactionCount
1	http://www.genshin.org/resource/Electro	"7"^^xsd:integer

### Q5. Which element has the most healers?

The screenshot shows a SPARQL query interface with the following details:

- Query Editor:** Contains a SPARQL query to find the element with the highest healer count.
- Results Table:** Shows 1 row of data indicating that the element 'Cryo' has the most healers.
- Buttons:** Includes standard table operations like copy, paste, and refresh, along with download and run buttons.

	element	healerCount
1	http://www.genshin.org/resource/Cryo	"4"^^xsd:integer

## Q6. List me all the weapons a specific character can use? E.g Kaeya

Competency 1 × Competency 2 × Competency 3 × Competency 4 × Competency 5 ×

Competency 6 × Competency 7 × Competency 8 × Competency 9 × Competency 10 ×

```

1 PREFIX ontology: <http://www.genshin.org/ontology/>
2 PREFIX resource: <http://www.genshin.org/resource/>
3 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
4
5 # Q6. List me all the weapons a specific character can use? E.g Kaeya
6 SELECT ?character ?characterName ?weapon ?weaponName
7 WHERE {
8   ?character a ontology:Character;
9   rdfs:label ?characterName;
10  ontology:canUseWeapon ?weapon .
11  ?weapon a ontology:Weapon;
12  rdfs:label ?weaponName .
13  FILTER(?character = resource:Kaeya).
14 }

```

Table Raw Response Pivot Table Google Chart Download as

Filter query results □ Showing results from 1 to 6 of 6. Query took 0.1s, yesterday at 21:02.

character	characterName	weapon	weaponName
http://www.genshin.org/resource/Kaeya	"Kaeya"	http://www.genshin.org/resource/Aquila_Favonia	"Aquila_Favonia"
http://www.genshin.org/resource/Kaeya	"Kaeya"	http://www.genshin.org/resource/Prototype_Rancour	"Prototype_Rancour"
http://www.genshin.org/resource/Kaeya	"Kaeya"	http://www.genshin.org/resource/Sacrificial_Sword	"Sacrificial_Sword"
http://www.genshin.org/resource/Kaeya	"Kaeya"	http://www.genshin.org/resource/Sword_of_Descension	"Sword_of_Descension"
http://www.genshin.org/resource/Kaeya	"Kaeya"	http://www.genshin.org/resource/The_Black_Sword	"The_Black_Sword"
http://www.genshin.org/resource/Kaeya	"Kaeya"	http://www.genshin.org/resource/Lions_Roar	"Lions_Roar"

## Q7. What characters are strong against X boss? E.g Pyro Regisvine

Competency 1 × Competency 2 × Competency 3 × Competency 4 × Competency 5 ×

Competency 6 × Competency 7 × Competency 8 × Competency 9 × Competency 10 ×

```

1 PREFIX ontology: <http://www.genshin.org/ontology/>
2 PREFIX resource: <http://www.genshin.org/resource/>
3 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
4
5 # Q7. What characters are strong against X boss? E.g Pyro Regisvine
6 SELECT DISTINCT ?characterName ?boss
7 WHERE {
8   ?character a ontology:Character;
9   rdfs:label ?characterName;
10  ontology:canEasilyDefeat ?boss .
11  ?boss a ontology:Boss .
12  FILTER(?boss = resource:Pyro_Regisvine).
13 }

```

Table Raw Response Pivot Table Google Chart Download as

Filter query results □ Showing results from 1 to 15 of 15. Query took 0.1s, yesterday at 21:02.

characterName	boss
"Aloy"	http://www.genshin.org/resource/Pyro_Regisvine
"Kaeya"	http://www.genshin.org/resource/Pyro_Regisvine
"Kamisato_Ayaka"	http://www.genshin.org/resource/Pyro_Regisvine
"Layla"	http://www.genshin.org/resource/Pyro_Regisvine
"Qiqi"	http://www.genshin.org/resource/Pyro_Regisvine
"Charlotte"	http://www.genshin.org/resource/Pyro_Regisvine
"Chongyun"	http://www.genshin.org/resource/Pyro_Regisvine
"Diona"	http://www.genshin.org/resource/Pyro_Regisvine
"Eula"	http://www.genshin.org/resource/Pyro_Regisvine
"Freminet"	http://www.genshin.org/resource/Pyro_Regisvine
"Ganyu"	http://www.genshin.org/resource/Pyro_Regisvine
"Mika"	http://www.genshin.org/resource/Pyro_Regisvine
"Rosaria"	http://www.genshin.org/resource/Pyro_Regisvine
"Shenhe"	http://www.genshin.org/resource/Pyro_Regisvine
"Wriothesley"	http://www.genshin.org/resource/Pyro_Regisvine

## Q8: What two characters can be used to cause a Vaporize reaction?

Competency 1 × Competency 2 × Competency 3 × Competency 4 × Competency 5 ×

Competency 6 × Competency 7 × Competency 8 × Competency 9 × Competency 10 ×

```

1 PREFIX ontology: <http://www.genshin.org/ontology/>
2 PREFIX resource: <http://www.genshin.org/resource/>
3 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
4 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
5
6 #Q8: What two characters can be used to cause a Vaporize reaction?
7
8 SELECT ?name1 ?name2 ?reaction
9 WHERE {
10   ?char1 rdf:type ontology:Character;
11   rdfs:label ?name1;
12   ontology:elementAlignment ?element1 .
13   ?char2 rdf:type ontology:Character;
14   rdfs:label ?name2;
15   ontology:elementAlignment ?element2 .
16   ?reaction ontology:requiresElement ?element1;
17   ontology:requiresElement ?element2 .
18   FILTER (?element1 != ?element2) .
19   FILTER (?reaction = resource:Vaporize) .
20 }
21

```

**Run**

Filter query results ▾ Showing results from 1 to 540 of 540. Query took 0.1s, yesterday at 21:04.

	name1	name2	reaction
1	'Furina'	'Amber'	http://www.genshin.org/resource/Vaporize
2	'Furina'	'Amber'	http://www.genshin.org/resource/Vaporize
3	'Furina'	'Bennett'	http://www.genshin.org/resource/Vaporize
4	'Furina'	'Dehya'	http://www.genshin.org/resource/Vaporize
5	'Furina'	'Diluc'	http://www.genshin.org/resource/Vaporize
6	'Furina'	'Diluc'	http://www.genshin.org/resource/Vaporize
7	'Furina'	'Hu_Tao'	http://www.genshin.org/resource/Vaporize
8	'Furina'	'Klee'	http://www.genshin.org/resource/Vaporize
9	'Furina'	'Lyney'	http://www.genshin.org/resource/Vaporize
10	'Furina'	'Thoma'	http://www.genshin.org/resource/Vaporize
11	'Furina'	'Thoma'	http://www.genshin.org/resource/Vaporize
12	'Furina'	'Xiangling'	http://www.genshin.org/resource/Vaporize

## Q9: List me a team of 4 characters, each with a different role, which can cause a hyperbloom reaction together

Competency 1 × Competency 2 × Competency 3 × Competency 4 × Competency 5 ×

Competency 6 × Competency 7 × Competency 8 × Competency 9 × Competency 10 ×

```

1 PREFIX ontology: <http://www.genshin.org/ontology/>
2 PREFIX resource: <http://www.genshin.org/resource/>
3 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
4 #Q9: List me a team of 4 characters, each with a different role, which
5 can cause a hyperbloom reaction together
6 SELECT DISTINCT ?nameLabel1 ?nameLabel2 ?nameLabel3 ?nameLabel4
7 WHERE {
8   ?name1 a ontology:PlayableCharacter;
9   ontology:hasRole ?role1;
10  rdfs:label ?nameLabel1;
11  ontology:elementAlignment ?elem1.
12  ?name2 a ontology:PlayableCharacter;
13  ontology:hasRole ?role2;
14  rdfs:label ?nameLabel2;
15  ontology:elementAlignment ?elem2.
16  ?name3 a ontology:PlayableCharacter;
17  ontology:hasRole ?role3;
18  rdfs:label ?nameLabel3;
19  ontology:elementAlignment ?elem3.
20  ?name4 a ontology:PlayableCharacter;
21  ontology:hasRole ?role4;
22  rdfs:label ?nameLabel4;
23  ontology:elementAlignment ?elem4.
24  ?reaction ontology:requiresElement ?elem1, ?elem2, ?elem3, ?elem4.
25  FILTER (?reaction = resource:Hyperbloom)
26  FILTER (?name1 != ?name2 && ?name1 != ?name3 && ?name1 != ?name4 &&
27  ?name2 != ?name3 && ?name2 != ?name4 && ?name3 != ?name4)
28  FILTER (?role1 != ?role2 && ?role1 != ?role3 && ?role1 != ?role4 &&
29  ?role2 != ?role3 && ?role2 != ?role4 && ?role3 != ?role4)
30  FILTER ((?elem1 != ?elem2 && ?elem1 != ?elem3 && ?elem1 != ?elem4) ||
31  (?elem1 != ?elem2 && ?elem1 != ?elem4 && ?elem2 != ?elem4) ||
32  (?elem1 != ?elem3 && ?elem1 != ?elem4 && ?elem3 != ?elem4) ||
33  (?elem2 != ?elem3 && ?elem2 != ?elem4 && ?elem2 != ?elem3))

```

**Run**

Filter query results ▾ Showing results from 1 to 1,000 of 8,712. Query took 0.4s, moments ago.

	nameLabel1	nameLabel2	nameLabel3	nameLabel4
1	'Alhaitham'	'Baizhu'	'Beidou'	'Barbara'
2	'Alhaitham'	'Baizhu'	'Beidou'	'Sangonomiya_Kokomi'
3	'Alhaitham'	'Baizhu'	'Dori'	'Barbara'
4	'Alhaitham'	'Baizhu'	'Dori'	'Sangonomiya_Kokomi'
5	'Alhaitham'	'Baizhu'	'Kujou_Sara'	'Barbara'
6	'Alhaitham'	'Baizhu'	'Kujou_Sara'	'Sangonomiya_Kokomi'
7	'Alhaitham'	'Baizhu'	'Kukl_Shinobu'	'Furina'
8	'Alhaitham'	'Baizhu'	'Kukl_Shinobu'	'Xingqiu'
9	'Alhaitham'	'Baizhu'	'Kukl_Shinobu'	'Candace'
10	'Alhaitham'	'Baizhu'	'Kukl_Shinobu'	'Mona'
11	'Alhaitham'	'Baizhu'	'Lisa'	'Barbara'
12	'Alhaitham'	'Baizhu'	'Lisa'	'Sangonomiya_Kokomi'
13	'Kirara'	'Baizhu'	'Beidou'	'Kamisato_Ayato'
14	'Kirara'	'Baizhu'	'Beidou'	'Nilou'
15	'Kirara'	'Baizhu'	'Beidou'	'Tartaglia'
16	'Kirara'	'Baizhu'	'Beidou'	'Yelan'
17	'Kirara'	'Baizhu'	'Beidou'	'Neuvillette'
18	'Kirara'	'Baizhu'	'Cyno'	'Furina'
19	'Kirara'	'Baizhu'	'Cyno'	'Xingqiu'

## Q10 (Bonus Federated Query): Get the voice actor names from Wikidata for characters.

Competency 1 × Competency 2 × Competency 3 × Competency 4 × Competency 5 ×

Competency 6 × Competency 7 × Competency 8 × Competency 9 × Competency 10 × ⊕

```

1 PREFIX ontology: <http://www.genshin.org/ontology/>
2 PREFIX wdt: <http://www.wikidata.org/prop/direct/>
3 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
4
5 # Q10: Get the voice actor names from Wikidata for Characters.
6 SELECT ?characterName ?voiceActor ?voiceActorName
7 WHERE {
8   ?character a ontology:Character;
9     rdfs:label ?characterName.
10  SERVICE <https://query.wikidata.org/sparql> {
11    ?character wdt:P725 ?voiceActor.
12    ?voiceActor rdfs:label ?voiceActorName.
13    FILTER(LANG(?voiceActorName) = "en") .
14  }
15
16
17
18 }
```

Table Raw Response Pivot Table Google Chart Download as

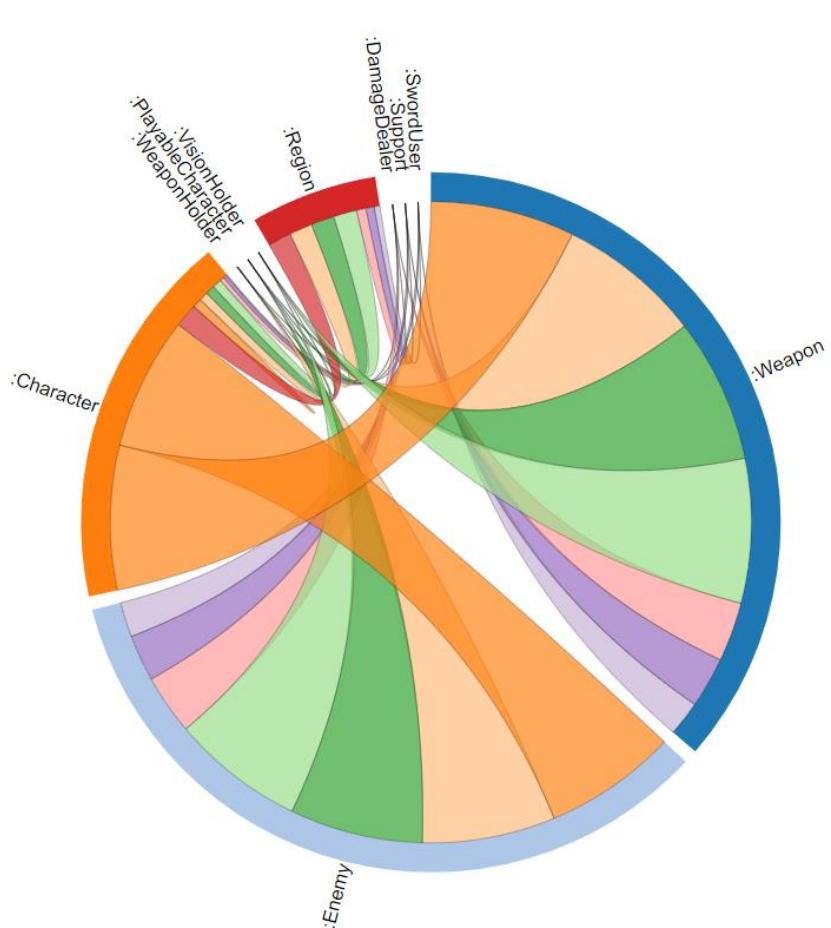
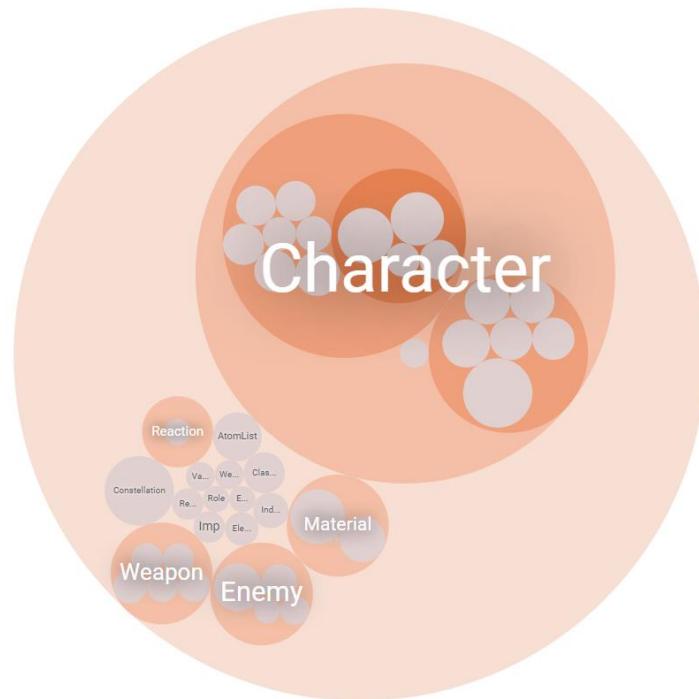
Filter query results Showing results from 1 to 60 of 60. Query took 36s, moments ago.

characterName	voiceActor	voiceActorName
"Fischl"	http://www.wikidata.org/entity/Q44552	"Maaya Uchida"@en
"Raiden_Shogun"	http://www.wikidata.org/entity/Q49552	"Miyuki Sawashiro"@en
"Shenhe"	http://www.wikidata.org/entity/Q49566	"Ayako Kawasumi"@en
"Lisa"	http://www.wikidata.org/entity/Q23731	"Rie Tanaka"@en
"Thoma"	http://www.wikidata.org/entity/Q961574	"Masakazu Morita"@en
"Keqing"	http://www.wikidata.org/entity/Q665398	"Eri Kitamura"@en
"Yae_Miko"	http://www.wikidata.org/entity/Q871452	"Ayane Sakura"@en
"Xiao"	http://www.wikidata.org/entity/Q957795	"Yoshitsugu Matsukawa"@en
"Qiqi"	http://www.wikidata.org/entity/Q1022023	"Yukari Tamura"@en
"Nahida"	http://www.wikidata.org/entity/Q1022023	"Yukari Tamura"@en
"Yoimiya"	http://www.wikidata.org/entity/Q1153353	"Kana Ueda"@en
"Diluc"	http://www.wikidata.org/entity/Q1139912	"Kensho Ono"@en

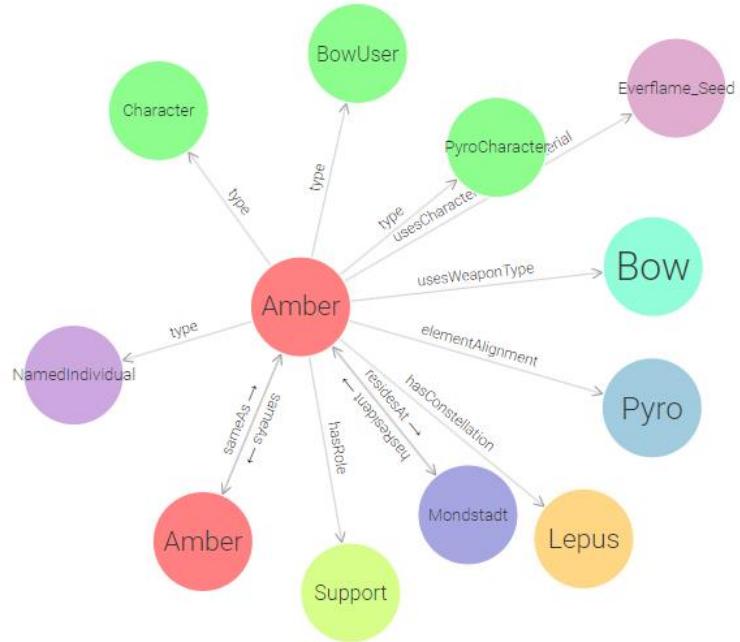
Run

Press Alt+Enter to autocomplete

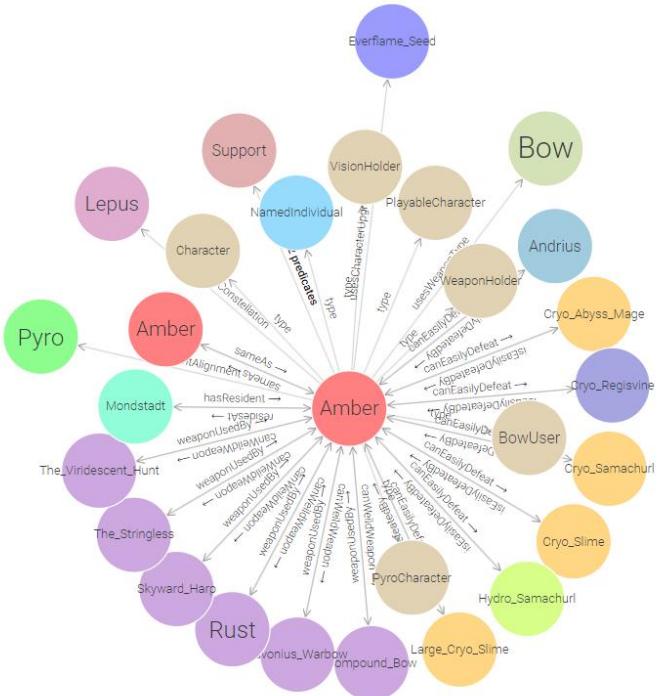
## Visualizations

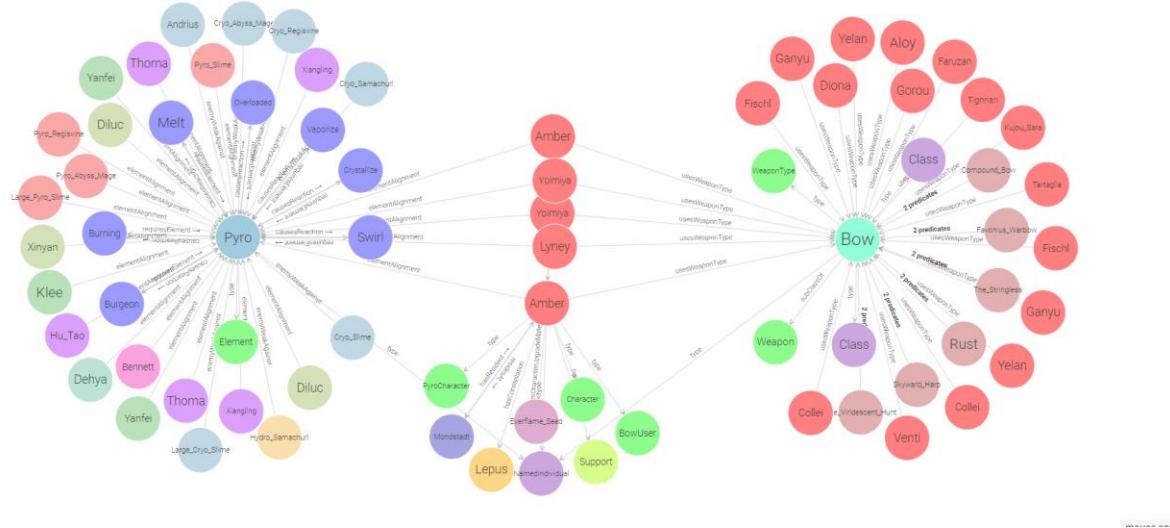


## Before adding Inference

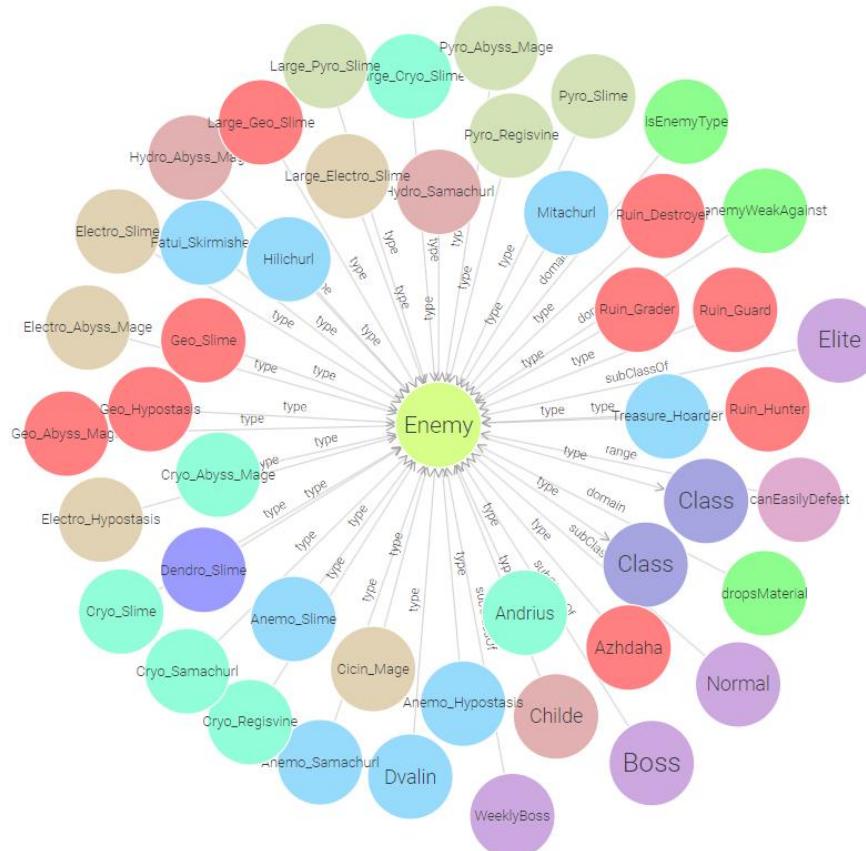


After adding Inference (The second Amber is its Wikidata counterpart, I used owl:sameAs for it)

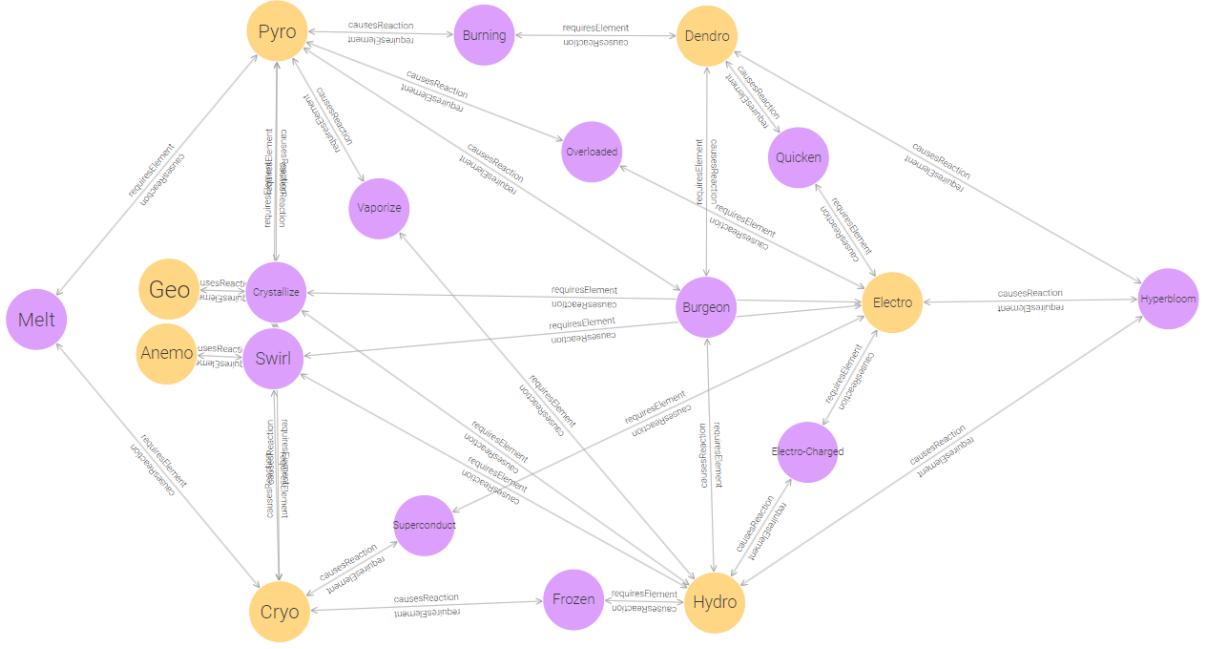




Over here, we can see Amber, Yoimiya and Lyney being Bow character who have a Pyro vision/



Here we can see the individuals related to `Enemy` class, the subclasses and domain/ranges for object properties where `Enemy` class is used in



These are the relations between Elements and Reactions

## Reflect

This was honestly a fun project with a lot of learning involved. Perhaps the most important thing we learned was how to properly represent data in a meaningful way. There were a lot of times and a lot of wasted hours where we had to redesign our ontology because the relations we tried to achieve between individuals just weren't making any sense. But at the end, we managed to create a pretty accurate representation of how the game works in the form of a graph.

Converting unstructured data like this to linked data (especially when SWRL rules were used) did have a lot of benefits. Whenever we played the game and a new boss/weekly boss would come out, we would try to figure out which characters would be good against it and whenever a new character was added, we would try to figure out new teams we could make using that character. Thanks to the inferencing power of SWRL rules and OWL itself -we were able to automate this questions and stored all of the direct information necessary to answer it in our graph :).

Overall there is still a lot left in this game's graph development. If we can create a 1 to 1 model of the game using these technologies then theorycrafting team combinations and finding the most optimal teams to beat spiral abyss would be just automated within these graphs. In the future, I'll probably revisit this a few more times, (I'll need to convert this to an LPG first though) because this was more of a passion project than a project being done for a course.

Thank you so much for allowing our team to choose this topic, I know you must have been a bit worried because we even mentioned in the beginning that we have 0 tabular data to work on this project idea but when there's creative freedom given, there will always be a way.

Farewell