

**BENAZIR BHUTTO SHAHEED UNIVERSITY LYARI KARACHI**

**DEPARTMENT OF COMPUTER SCIENCE**

**REPORT ON FOOD ONTOLGOY**

**SUBMITTED BY:**

**NABEEL**

**604BCS/18-S/9**

**7th SEMESTER**

**SECTION ‘B’**

**9th BATCH**

**SUBMITTED TO:**

**MA’AM AMBREEN**

**Contents**

1. Tool Used
2. Topic Inspiration
3. Background knowledge of topic
4. Previously work done on selected topic
5. Introduction
6. Phases
7. Class
   1. Adding Class Tab ………………………………………………………… 04
   2. Adding Class Views ……………………………………………………… 04
   3. Adding Classes …………………………………………………………… 04
   4. Adding Sub Class ………………………………………………………… 05
   5. Classes and Sub classes …………………………………………………... 06
8. Data Property

2.1) Adding Data property Tab ………………………………………………... 07

2.2) Adding Data property views ……………………………………………… 07

2.3) Adding Data property …………………………………………………….. 07

2.4) Domain Range to Data property………………………………………….. 08

3) Object Property

3.1) Adding Object property Tab ……………………………………………… 08

3.2) Adding object property views …………………………………………….. 09

3.3) Adding object property …………………………………………………… 09

3.4) Domain Range to object property ………………………………………… 09

4)Object sub properties ………………………………………………………………. 10

5) Properties(Inverse, Transitive, Functional, Symmetric etc) ………………………... 10

6) Equivalent To

6.1) Equivalent Class …………………………………………………………... 11

6.2) Equivalent Object property ………………………………………………... 12

6.3) Equivalent Data Property ………………………………………………….. 12

**Contents**

7) Disjoint

7.1) Disjoint Class ……………………………………………………………… 13

7.2) Disjoint object property …………………………………………………… 13

7.3) Disjoint data property ……………………………………………………... 14

8) Individuals

8.1) Creating Individuals ………………………………………………………. 14

8.2) Individuals ………………………………………………………………… 15

9) Object and Data property assertion

9.1) Object property assertion ………………………………………………….. 15

9.2) Data property assertion ……………………………………………………. 15

10) Negative Object and Data property assertion

10.1) Negative Object property assertion ……………………………………….. 16

10.2) Negative Data property assertion …………………………………………. 16

11) Inverse of

11.1) Is\_a Inverse of Has & Has\_a ………………………………………………. 17

11.2) Has\_a Inverse of Has & Is\_a ………………………………………………. 17

12) Same Individuals as different individuals as

12.1) Same individuals as ………………………………………………………... 18

12.2) Different individuals as ……………………………………………………. 18

13) Cardinality

13.1) Setting up Cardinality ……………………………………………………… 19

13.2) Cardinality …………………………………………………………………. 19

14) Allvaluesfrom Somevaluesfrom Hasvalue …………………………………………..

15) SPARQL Queries …….………………………………………………………………

7) Conclusion

**Table of Figures**

1. Class
   1. Adding Class Tab …………………………………………………………… 04
   2. Adding Class Views ………………………………………………………… 04
   3. Adding Classes ……………………………………………………………… 04
   4. Adding Sub Class …………………………………………………………… 05
   5. Classes and Sub classes ……………………………………………………... 06
2. Data Property

2.1) Adding Data property Tab …………………………………………………... 07

2.2) Adding Data property views ………………………………………………… 07

2.3) Adding Data property ……………………………………………………….. 07

2.4) Domain Range to Data property…………………………………………….. 08

3) Object Property

3.1) Adding Object property Tab ………………………………………………… 08

3.2) Adding object property views ……………………………………………….. 09

3.3) Adding object property ……………………………………………………… 09

3.4) Domain Range to object property …………………………………………… 09

4)Object sub properties …………………………………………………………………. 10

5) Properties(Inverse, Transitive, Functional, Symmetric etc) …………………………... 10

6) Equivalent To

6.1) Equivalent Class ……………………………………………………………... 11

6.2) Equivalent Object property …………………………………………………... 12

6.3) Equivalent Data Property …………………………………………………….. 12

7) Disjoint

7.1) Disjoint Class ………………………………………………………………… 13

7.2) Disjoint object property ……………………………………………………… 13

7.3) Disjoint data property ………………………………………………………... 14

8) Individuals

8.1) Creating Individuals …………………………………………………………. 14

8.2) Individuals …………………………………………………………………… 15

**Table of Figures**

9) Object and Data property assertion

9.1) Object property assertion …………………………………………………….. 15

9.2) Data property assertion ………………………………………………………. 15

10) Negative Object and Data property assertion

10.1) Negative Object property assertion ………………………………………….. 16

10.2) Negative Data property assertion ……………………………………………. 16

11) Inverse of

11.1) Is\_a Inverse of Has & Has\_a …………………………………………………. 17

11.2) Has\_a Inverse of Has & Is\_a …………………………………………………. 17

12) Same Individuals as different individuals as

12.1) Same individuals as …………………………………………………………... 18

12.2) Different individuals as ………………………………………………………. 18

13) Cardinality

13.1) Setting up Cardinality ……………………………………………………… 19

13.2) Cardinality …………………………………………………………………. 19

14) Allvaluesfrom Somevaluesfrom Hasvalue …………………………………………..

**Tool Used**

I have used protégé, which is an open source ontology editor and knowledge management system use to create an intelligent system. Protégé is supported by strong developer community. It can be used to create both simple and complex ontology.

**Topic Inspiration**

As we can see that now a days there are many diseases caused by eating unhealthy foods and not taking of calories that human body requires daily. Therefore, I have developed an intelligent system by creating ontology that can help person in their diets and can help them to prevent their self by taking a healthy food.

**Background knowledge of Topic**

More than 3,000 mixtures are being added to prepared food, with various consequences for the food: to add tone, settle, texturize, safeguard, improve, thicken, add flavor, mellow, emulsify, etc. As indicated by World Health Organization, governments have recently centered around enactment to lessen such fixings or mixtures in made food sources as they may have results causing wellbeing dangers like coronary illness, disease, diabetes, allergens, and weight. By overseeing what and the amount to eat just as what not to eat, we can expand a patient's life quality through evasion of unfortunate fixings. Keen e-smart health frameworks with amazing information bases can give ideas of proper food varieties to people. Cutting edge savvy knowledgebase frameworks won't just incorporate conventional syntactic-based pursuit, which restricts the utility of the list items, yet will likewise give semantics to rich looking.

**Previously work done on selected Topic**

PIPS (Personalized Information Platform for Health and Life Services) is an E-wellbeing Coordinated task subsidized by the European Commission under the Framework 6 call, that meansto make novel medical care conveyance models by building a climate for Wellbeing, and Knowledge Services Support. This climate coordinates various advancements to empower medical services experts to gain admittance to applicable, refreshed clinical information, and European residents to pick better ways of life. The undertaking intends to unite medical care providers,

residents, public associations, food/drug industry and administrations, specialists, and wellbeing related approach producers to make a dynamic information climate. This powerful climate expands on conventional and new methodologies for taking care of information from current clinical practice, proof based medication, and divergent information sources from wellbeing/nourishment spaces.

**Introduction**

In this new era, we can see that everyday the food quality is getting low. Due to which if a person consumed an unhealthy food, he or she can suffer different diseases such as allergies, blood pressure problem, sugar problem and many other diseases. As, this is the new of era of technology where everything is getting automated, I have tried to create an automated system which can help person to safe itself before consuming food by checking the food ingredients if it is healthy or not for his body.

The cosmology in the proposed framework depicts wellbeing hazard gatherings, undesirable fixings or mixtures in food sources, fixings inadmissible for those in the danger gatherings, fixings with results relying upon patient qualities, etc. Hence, through the SW approach including cosmology utilization, specialized individual wellbeing frameworks depend on ontological information the board, which is effectively extensible to permit reception of extra e-wellbeing apparatuses. The philosophy is divided among individual wellbeing administrations and e-wellbeing instruments that give interoperation indicated by utilizing OWL.

**Phase 01: Classes Subclasses**

* 1. **Adding Class Tab**

Go to Window option, select tabs and then check the class option.

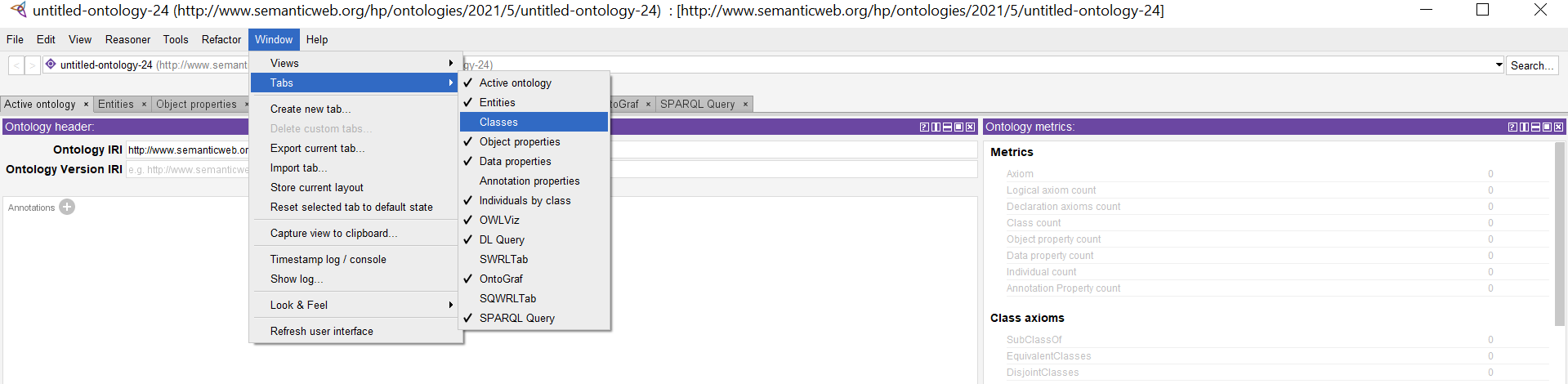


Fig 1.1 Adding class Tab

* 1. **Adding Class Views**

Go to Window option, select views and then click on class views from listed options.

Graphical user interface, application

Description automatically generated

Fig 1.2 Class Views

* 1. **Adding Classes**

Right click on owl:Thing, select option add a class.

Graphical user interface, text

Description automatically generated

Fig 1.3 Adding Classes

* 1. **Adding Subclass**

Right click on the parent class and then select add subclass

**Graphical user interface, text, application, email

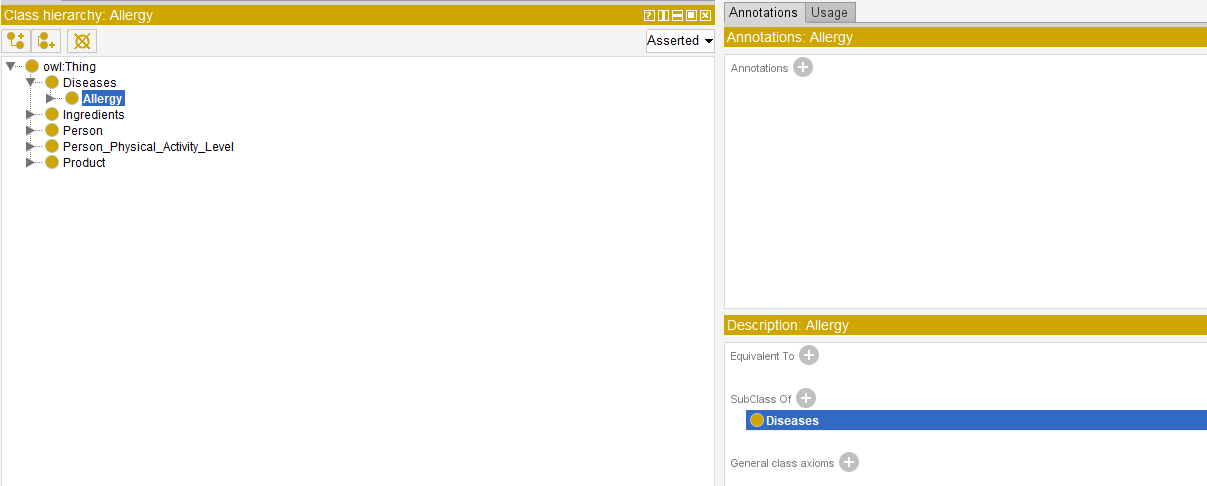
Description automatically generated** 

Fig 1.4 Adding Subclass of Allergy

* 1. **Classes and Subclasses.**

Here is the screenshot of all the classes and subclasses that I have created for food ontology.

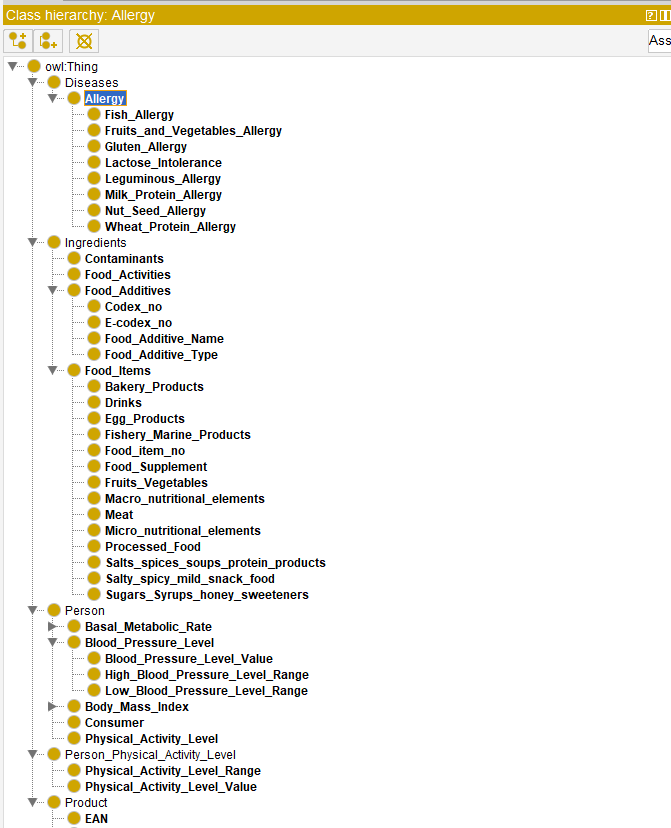


Fig 1.5 Classes and Subclasses

**Phase 02: Data property**

**2.1 Adding Data Properties Tab.**

Go to Window option, select tabs and then check the data properties option.

Graphical user interface, text, application, Word

Description automatically generated

Fig 2.1 Adding Data Properties Tab.

**2.2 Adding Data Properties Views.**

Go to Window option, select views and then click on data property views from listed options.

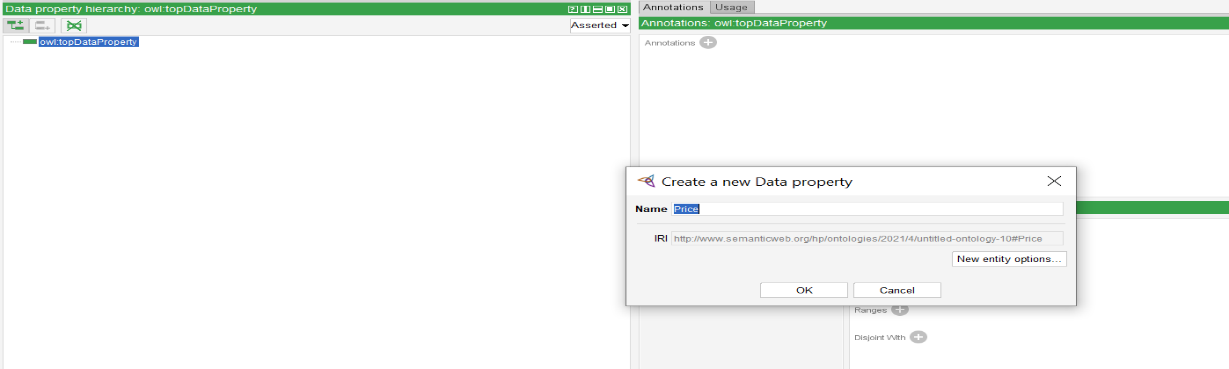
Graphical user interface, application

Description automatically generated

Fig 2.2 Adding Data Properties Views.

**2.3 Adding Data Property.**

Right click on owl:topDataProperty and select add data property.



2.3 Adding Data Property.

**2.4 Domain Ranges to Data Property.**

Click on domain to add a domain for a data property and select data type as a range.

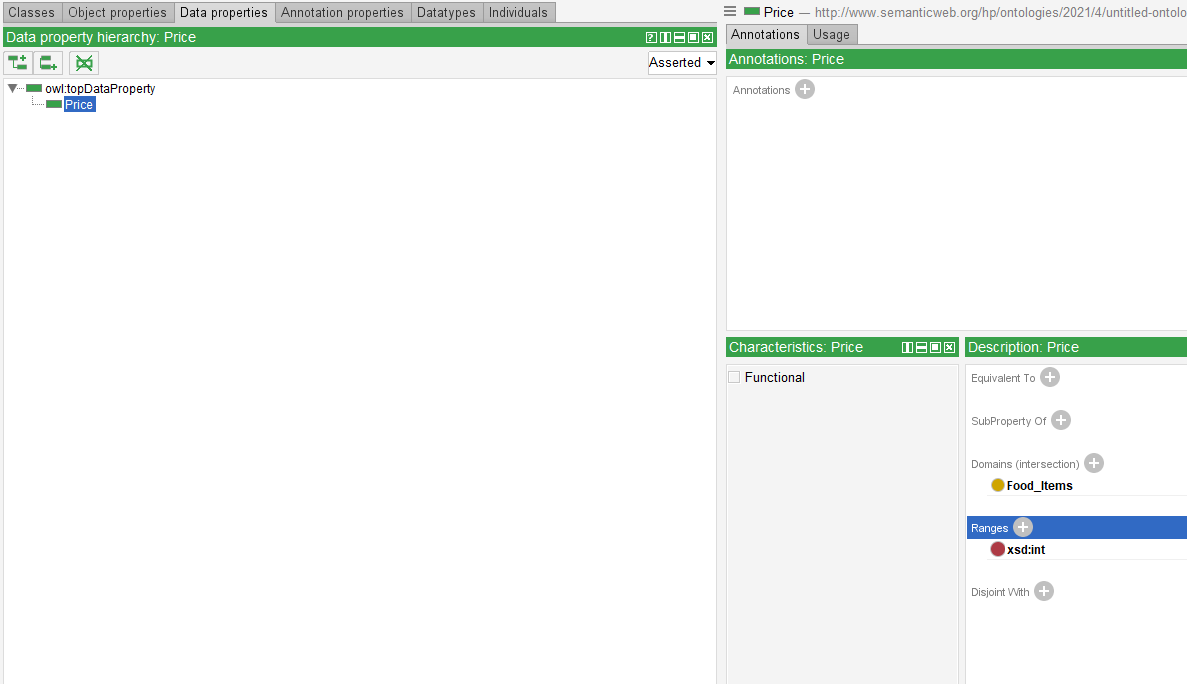


Fig 2.4 Domain Ranges to Data Property.

**Phase 03: Object property.**

**3.1 Adding Object Properties Tab.**

Go to Window option, select tabs and then check the object properties option

Graphical user interface, text, application, Word

Description automatically generated

Fig 3.1 Adding Object Properties Tab.

**3.2 Adding Object Properties Views.**

Go to Window option, select views and then click on object property views from listed options.

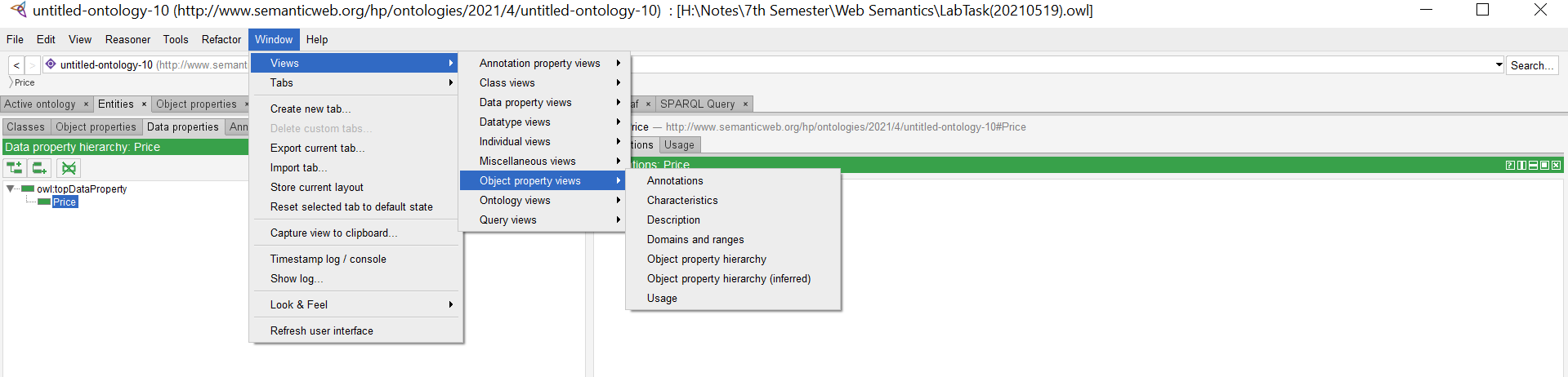


Fig 3.2 Adding Object Properties Views.

**3.3 Adding Object Properties.**

Right click on owl:topObjectProperty and select add object property

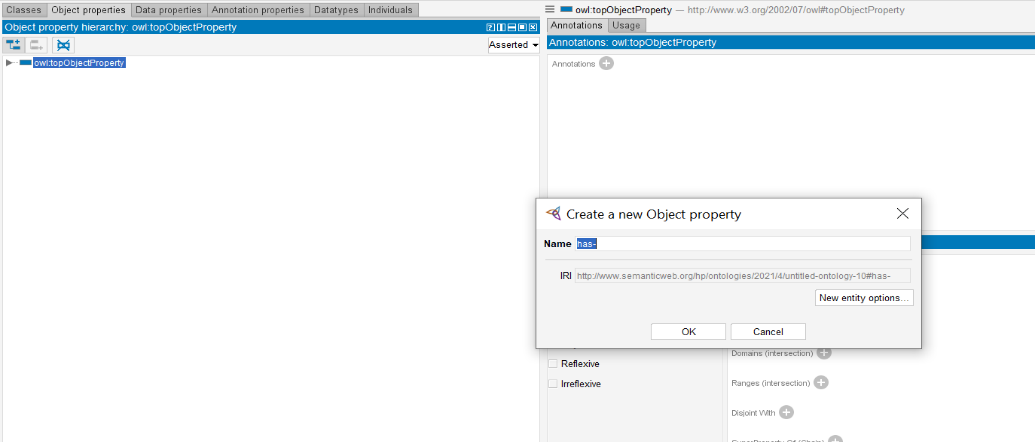


Fig 3.3 Adding Object Properties.

**3.4 Domain Range to Object Property.**

Click on domain to add a domain for a object property and add range for a object property.

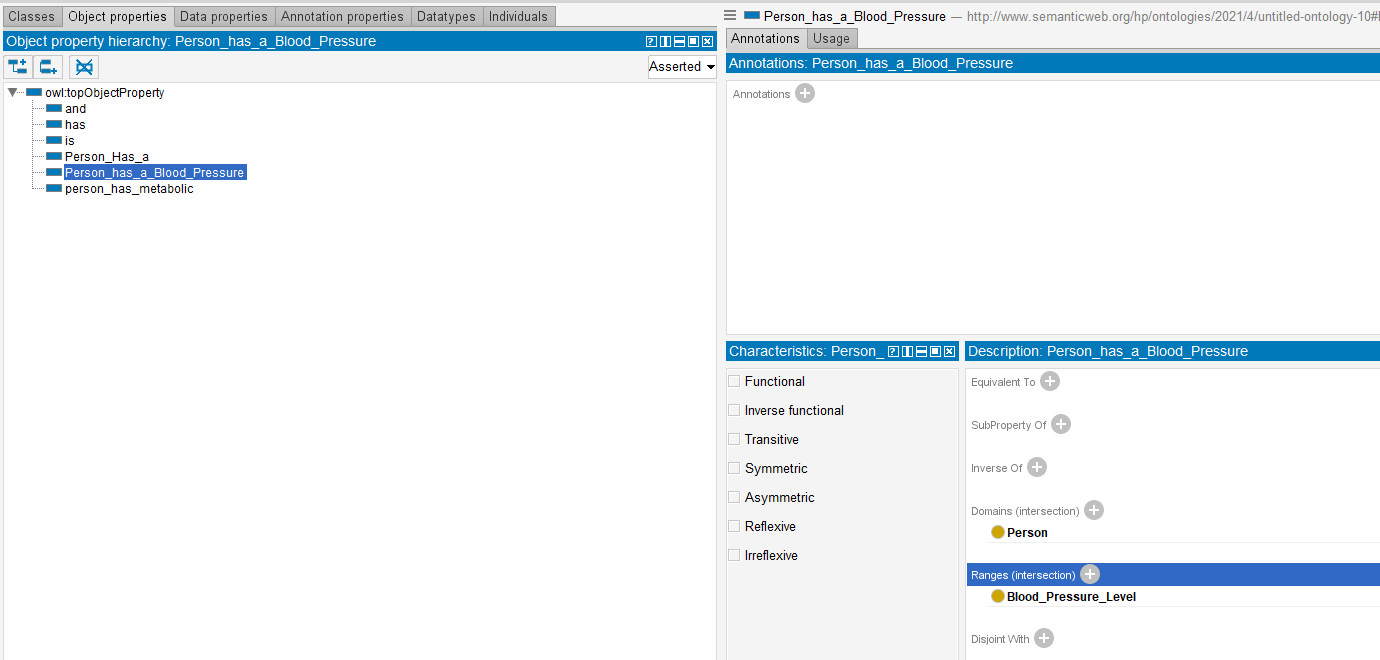


Fig 3.4 Domain Range to Object Property.

**Phase 04: Object sub properties**

**4.1 Object Sub Properties**

The object properties which we add under any object property are known as object sub property.

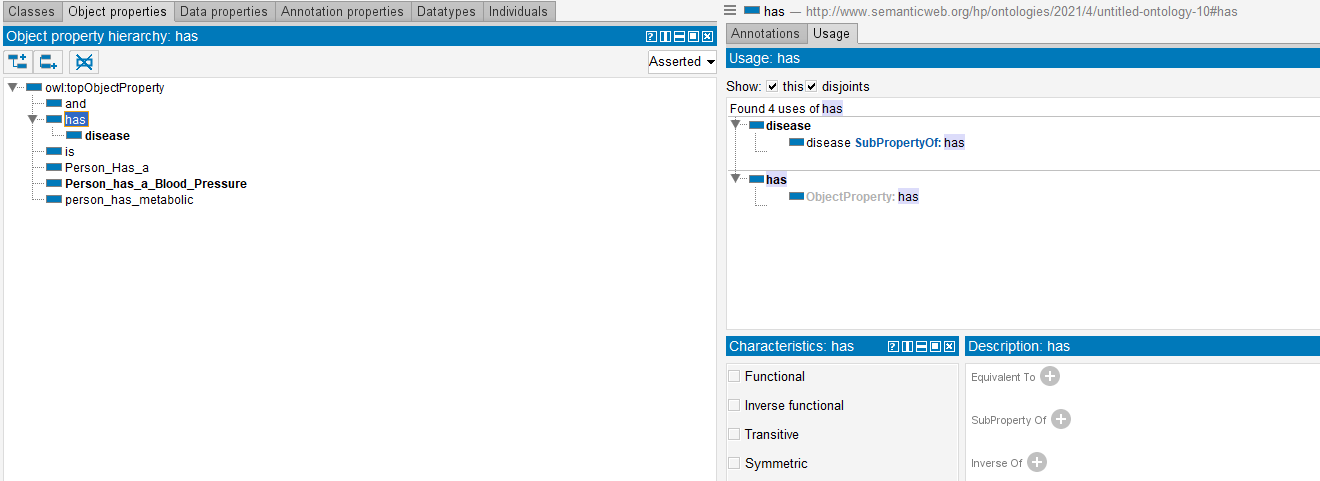


Fig 4.1 Object Sub Properties

**Phase 05: Properties (inverse, transitive, functional, symmetric etc.)**

* **Functional** - asserts that the selected property is Functional. Intuitively, this means that for any given individual, the property can have at most one value. In other words, there can be at most one out going relationship along the property for that individual. Note that, if multiple individuals are specified as values for the property then these values will be inferred to denote the same object.
* **Inverse Functional** - asserts that the selected property is Inverse Functional. Intuitively, this means the inverse property of the selected property (whether it explicitly declared or not) is Functional. In other words, there can be at most one incoming relationship along the property for that individual. Note that, if multiple individuals are specified as incoming values for the property then these values will be inferred to denote the same object.
* **Transitive** - asserts that the selected property is Transitive. Intuitively, this means that if individual x is related to individual y, and individual y is related to individual z, then individual x will be related to individual z. In other words, a single “hop” is implied over a chain of two along a given property if that property is transitive.
* **Symmetric** - asserts that the selected property is Symmetric. Intuitively, this means that the property has itself as an inverse, so if individual x is related to individual y then individual y must also be related to individual x along the same property.

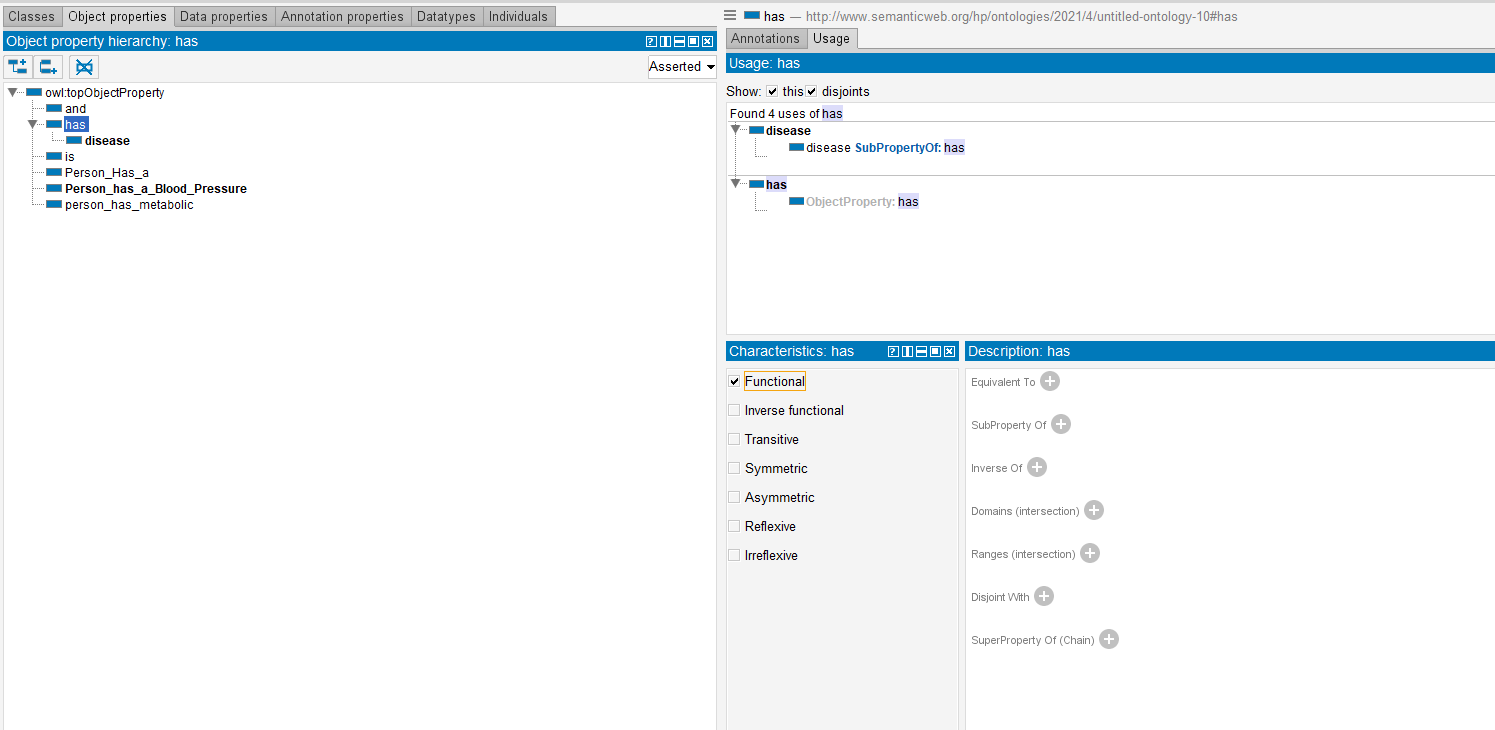


Fig 5.1 Properties.

**Phase 06: Equivalent to**

**6.1 Equivalent Classes**

The classes which are equal in properties and behaviors are known as equivalent class.

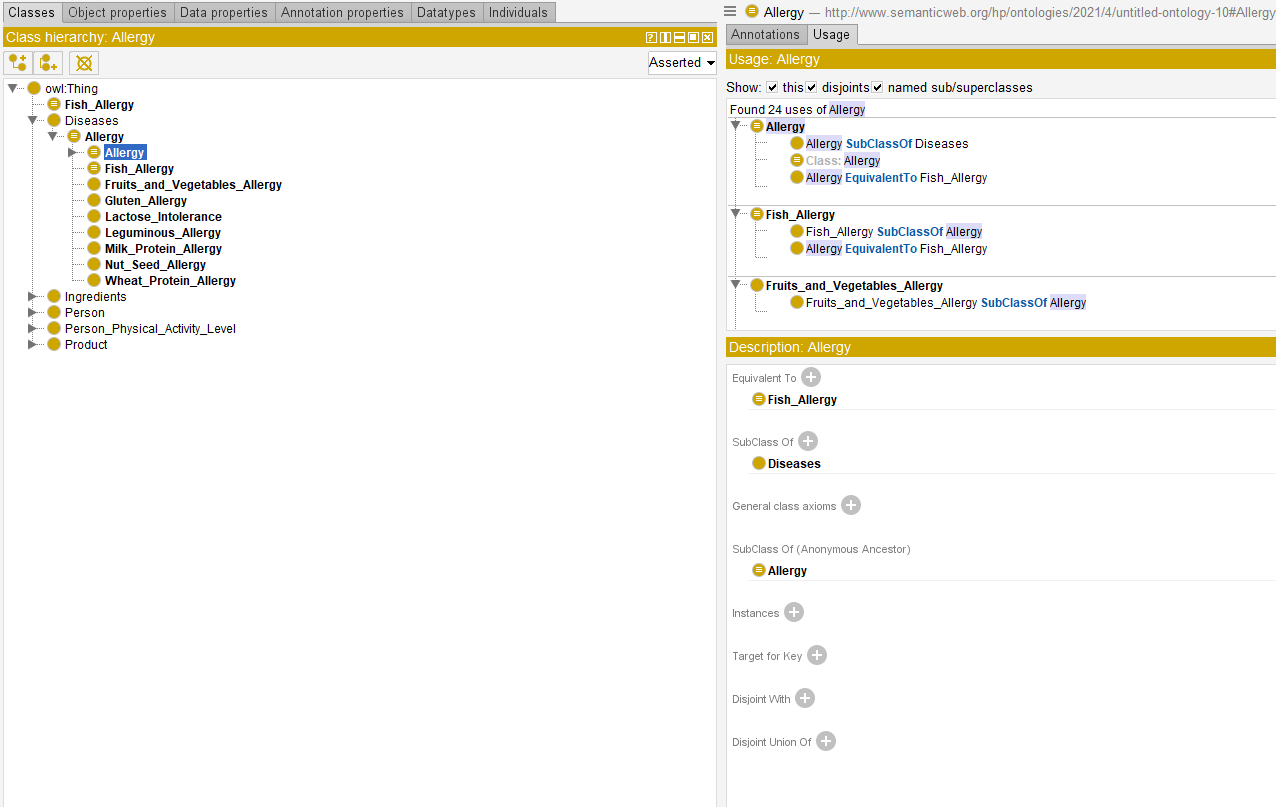


Fig 6.1 Equivalent Classes

**6.2 Equivalent Object Property**

The object properties which are equal in working are known as equivalent object property.

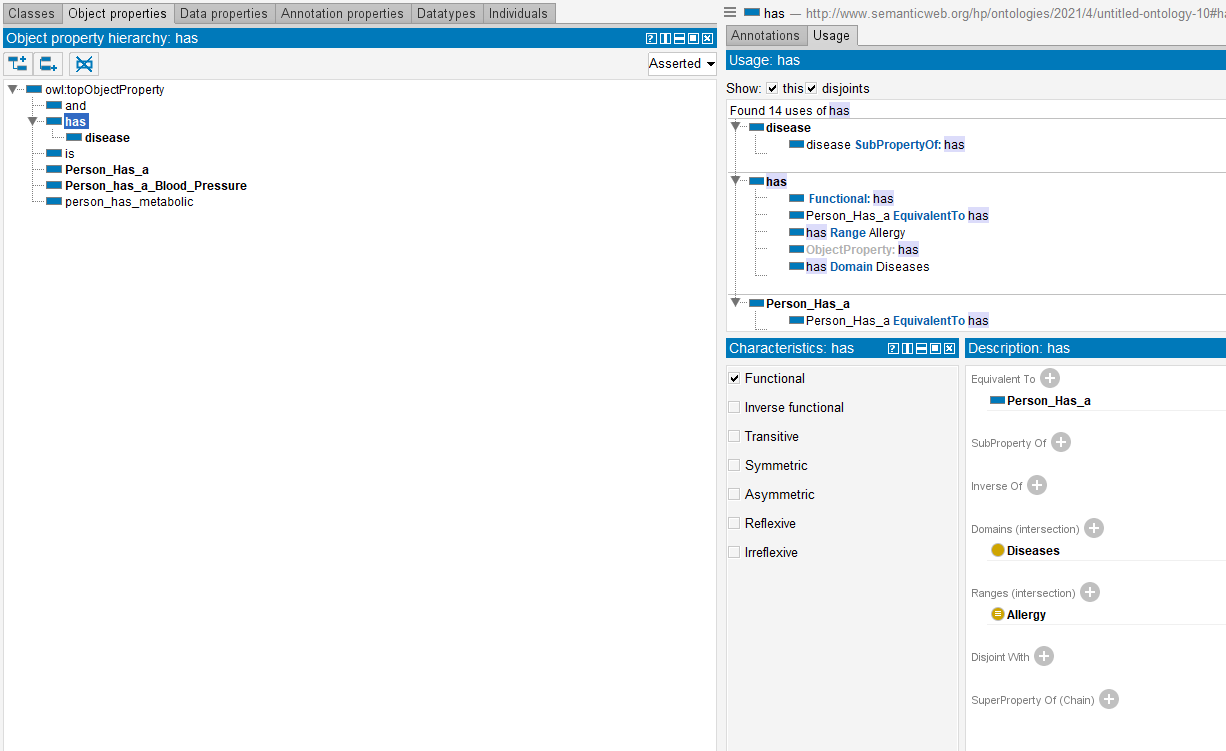


Fig 6.2 Equivalent Object Property

**6.3 Equivalent Data Property**

The data properties which are equal in behaviors and meanings are known as equivalent data property.

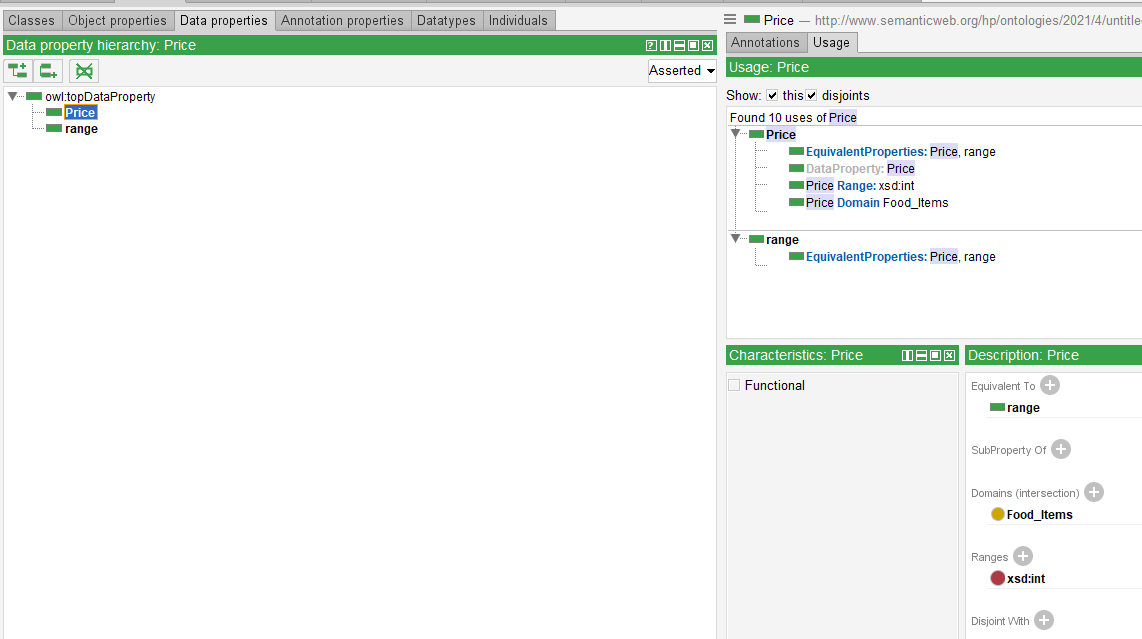


Fig 6.3 Equivalent Data Property

**Phase 07: Disjoint**

**7.1 Disjoint Classes**

The classes which are meaningless to each other are known as disjoint classes.

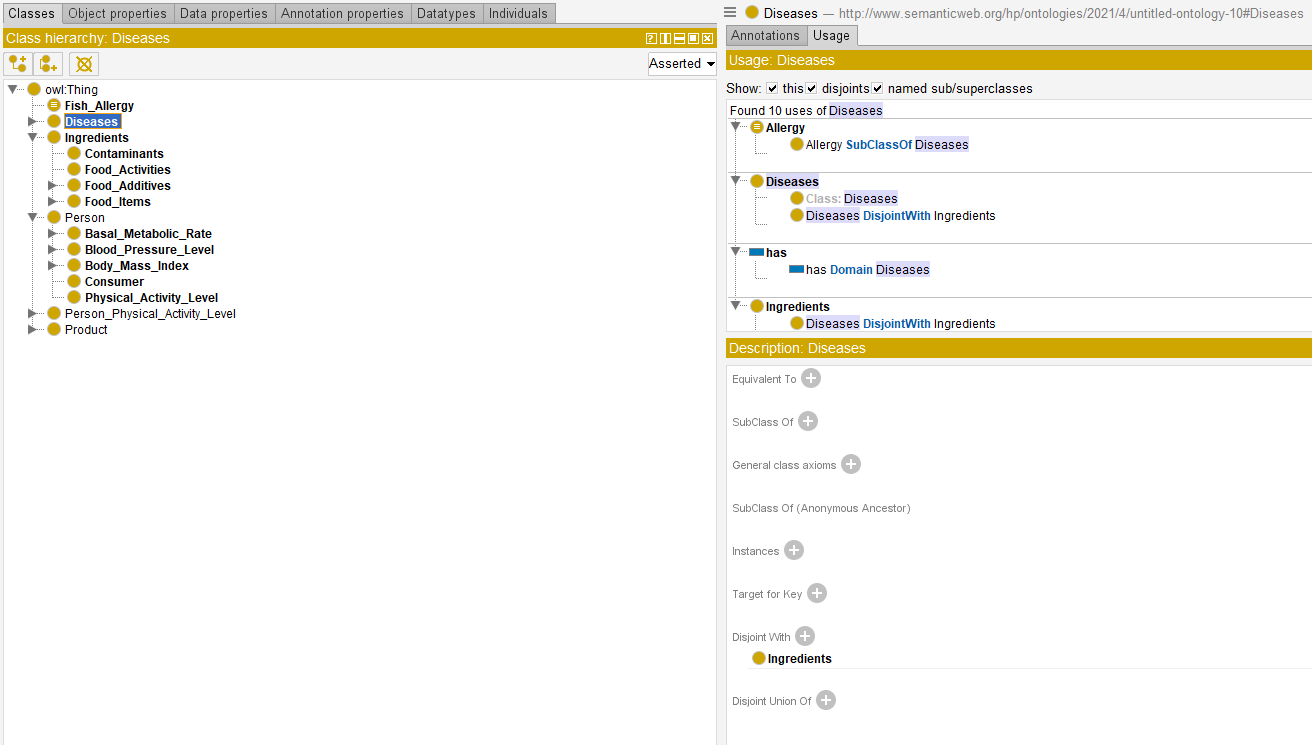


Fig 7.1 Disjoint Classes

**7.2 Disjoint Object Properties**

The object properties which are meaningless to each other are known as disjoint object properties.

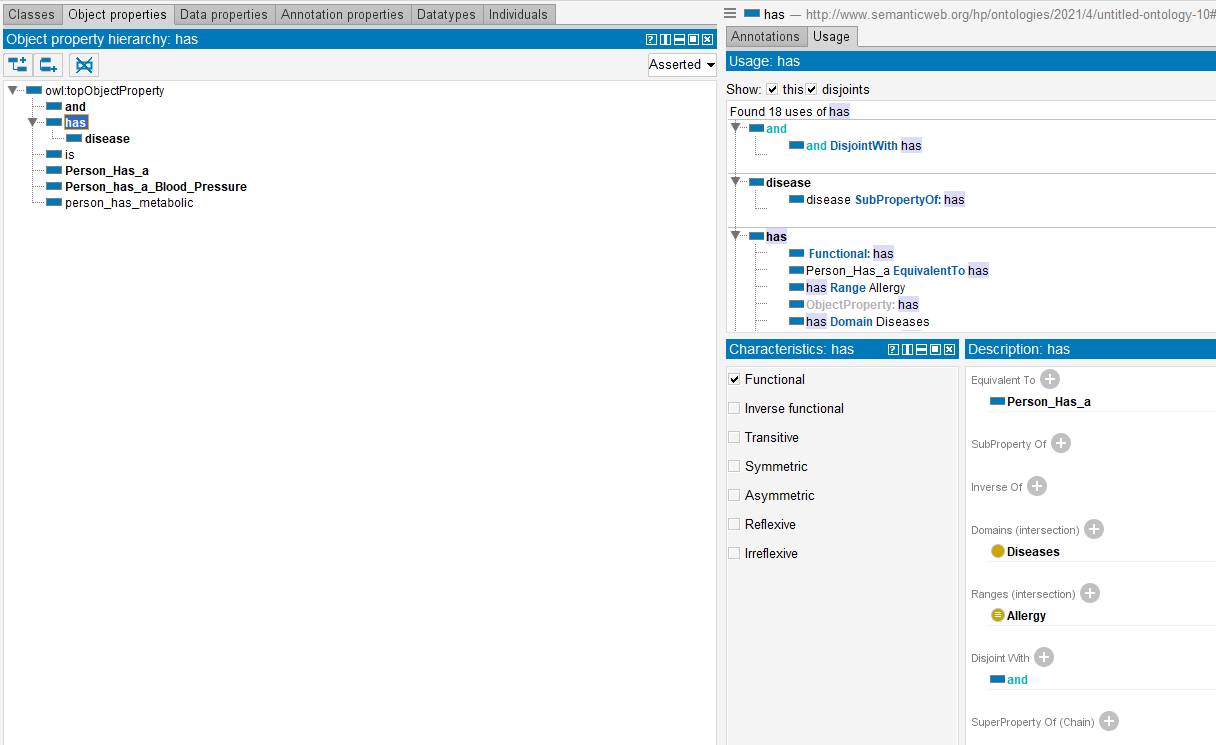


Fig 7.2 Disjoint Object Properties

**7.3 Disjoint Data Properties**

The data properties which are meaningless to each other are known as disjoint data properties.

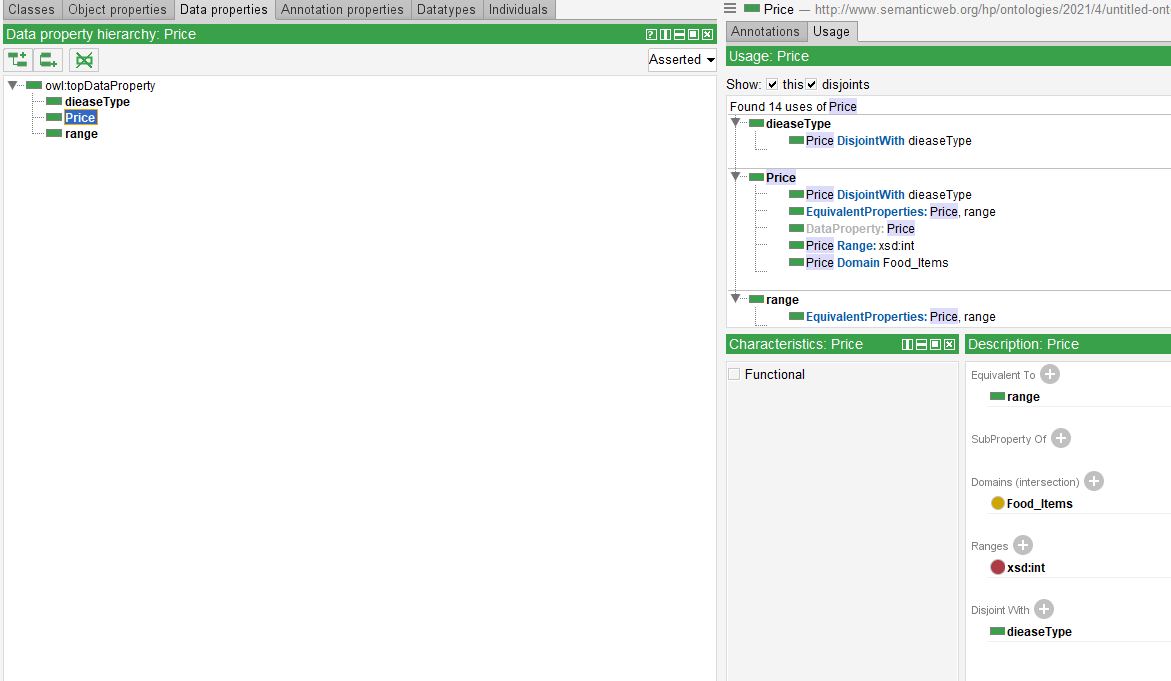


Fig 7.3 Disjoint Data Properties

**Phase 08: Individual**

**8.1 Creating Individual.**

To create indviduals click on to add class as a individual.

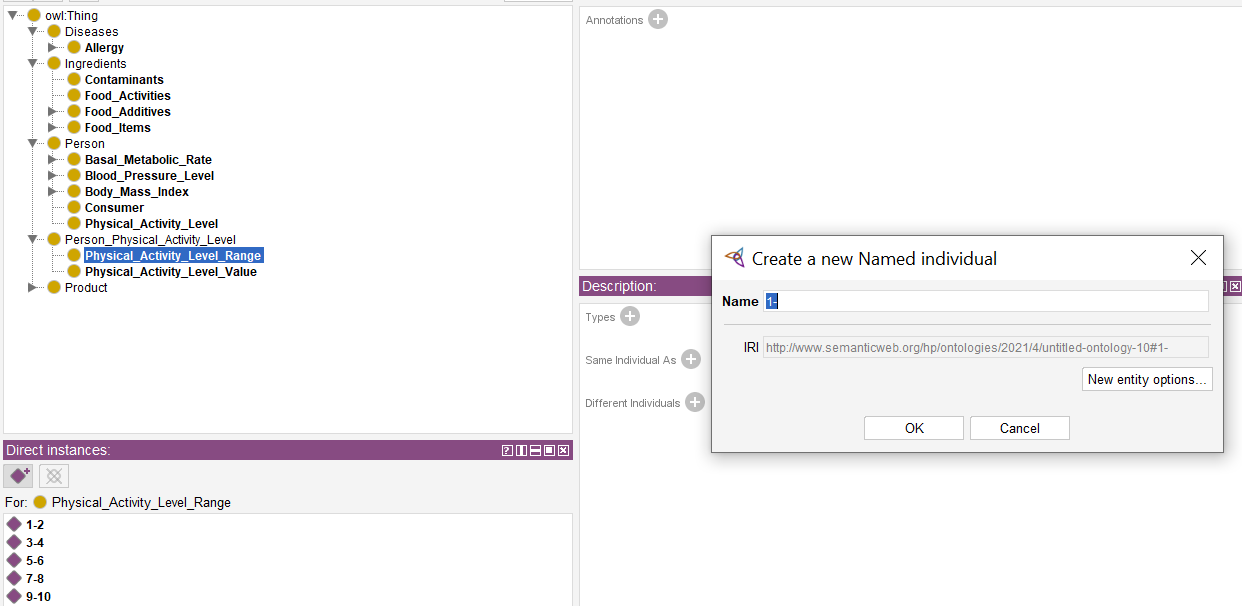


Fig 8.1 Creating Individual

**8.2 Individuals.**

Individuals are known as the instances of classes.

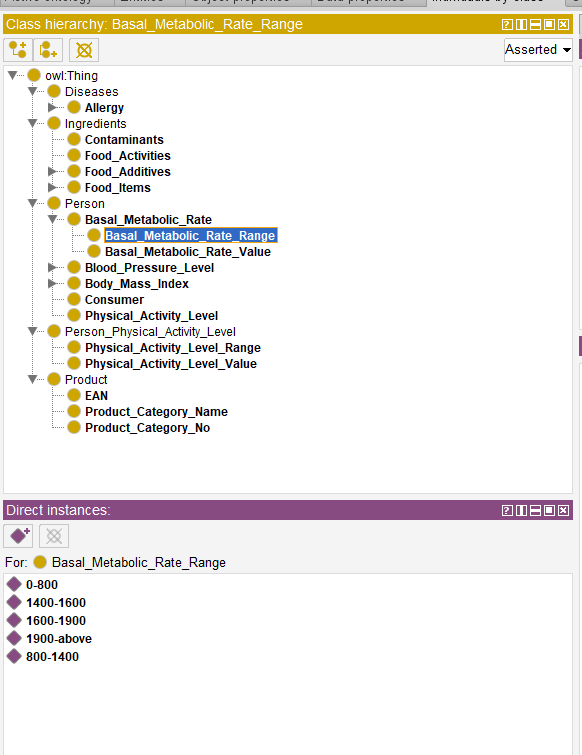


Fig 8.2 Individuals

**Phase 09: Object and data property assertion**

**9.1 Object Property Assertion**

We can add object property to individual by selecting object property assertion.

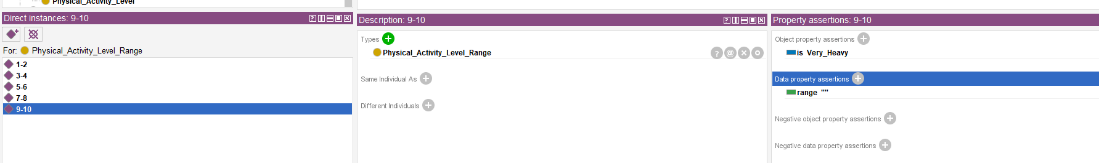


Fig 9.1 Object Property Assertion.

**9.2 Data Property Assertion**

We can add data property to individual by selecting data property assertion.

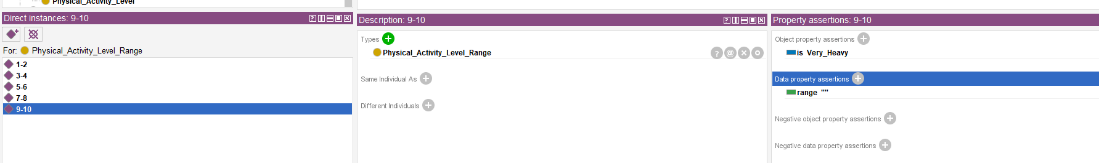


Fig 9.2 Data Property Assertion.

**Phase 10: Negative object and data property assertion**

**10.1 Negative Object Property Assertion**

The negative object property assertion is that which is not equal to the actual purpose of individual.

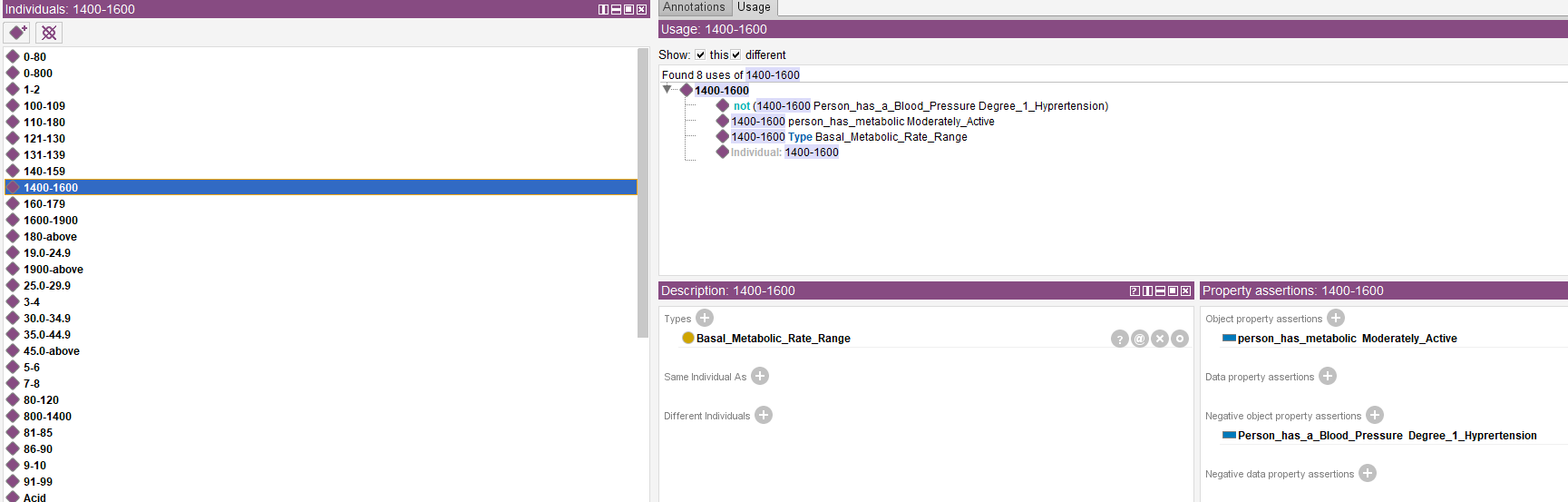


Fig 10.1 Negative Object Property Assertion.

**10.2 Negative Data Property Assertion**

The negative data property assertion is that which is not equal to the actual purpose of individual.

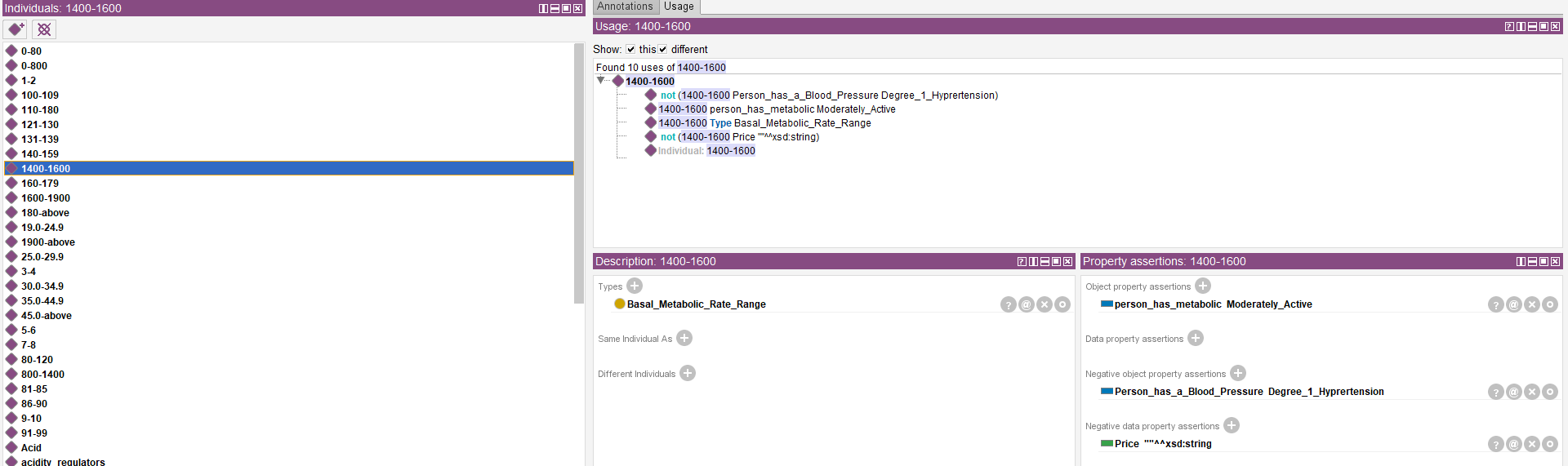


Fig 10.2 Negative Data Property Assertion.

**Phase 11: Inverse of**

**11.1 Is\_a Inverse of Has & Has\_a**

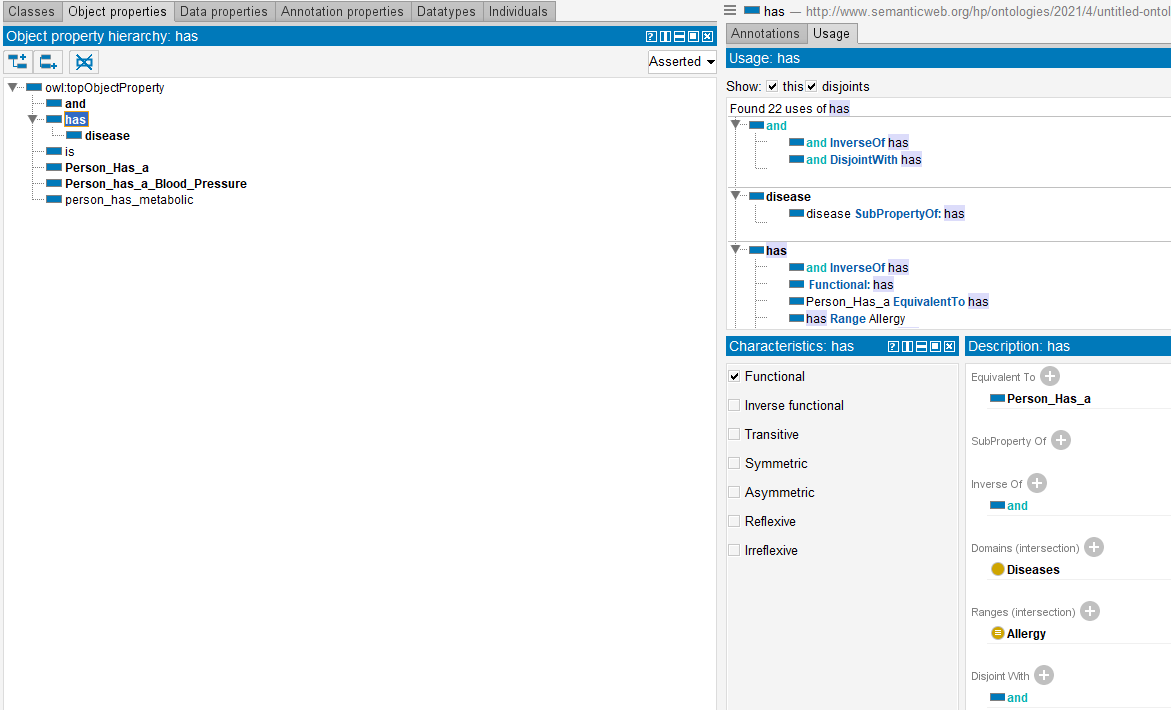


Fig 11.1 Is\_a Inverse of Has & Has\_a.

**11.2 Has\_a Inverse of Has & Is\_a**

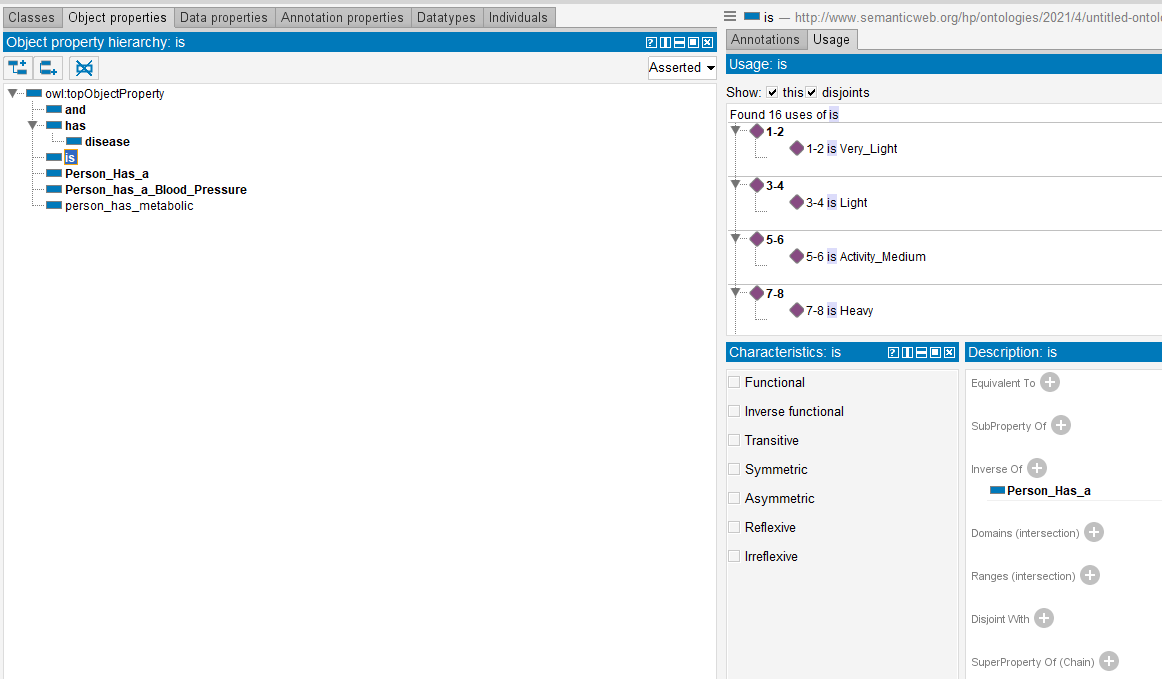


Fig 11.2 Has\_a Inverse of Has & Is\_a.

**Phase 12: Same individual as Different individual as**

**12.1 Same Individual as**

The individuals which are same in meaning are known as same individuals as.

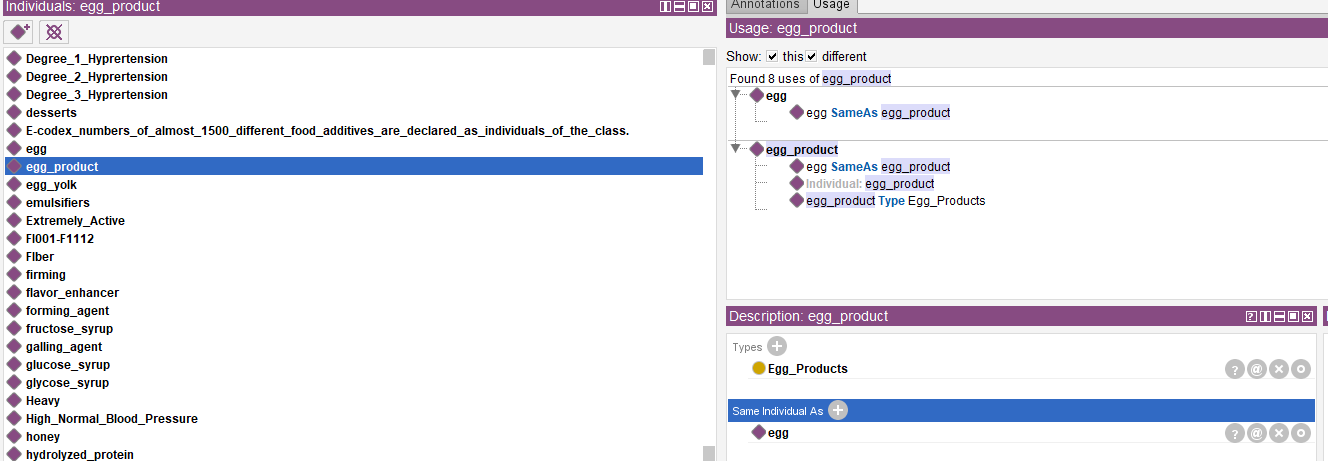


Fig 12.1 Same Individual as.

**12.2 Different Individual as**

The individuals which are different in meaning are known as different individuals as.

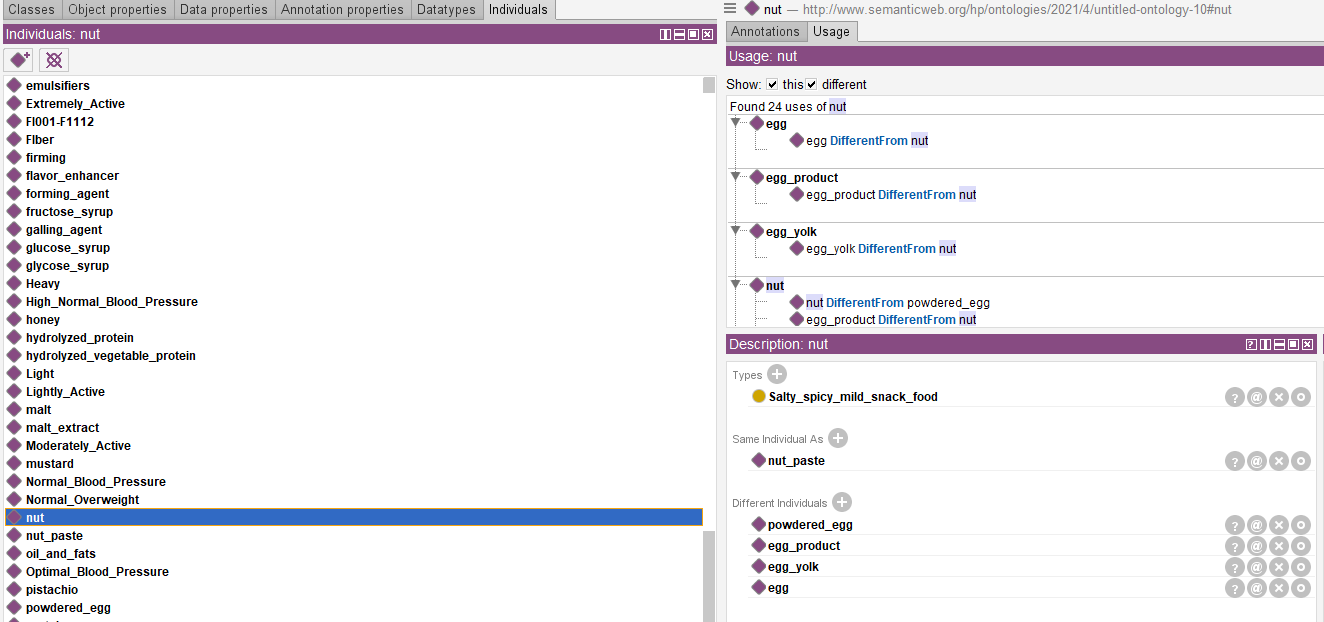


Fig 12.2 Different Individual as.

**Phase 13: Cardinality**

**13.1 Setting up Cardinality**

Cardinality is used to add some restriction on a class.

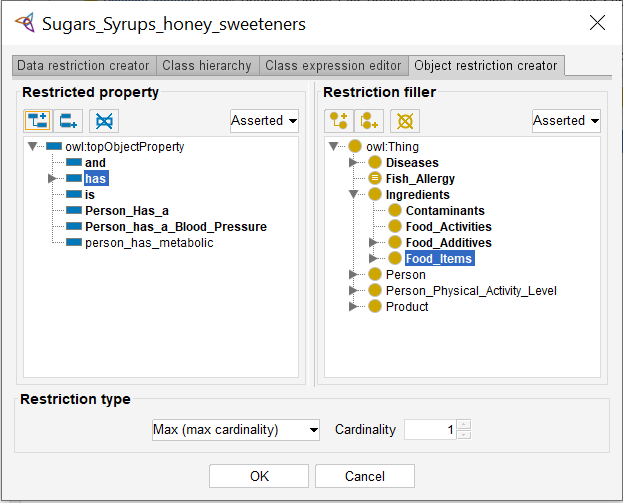


Fig 13.1 Setting up Cardinality

**13.2 Cardinality**

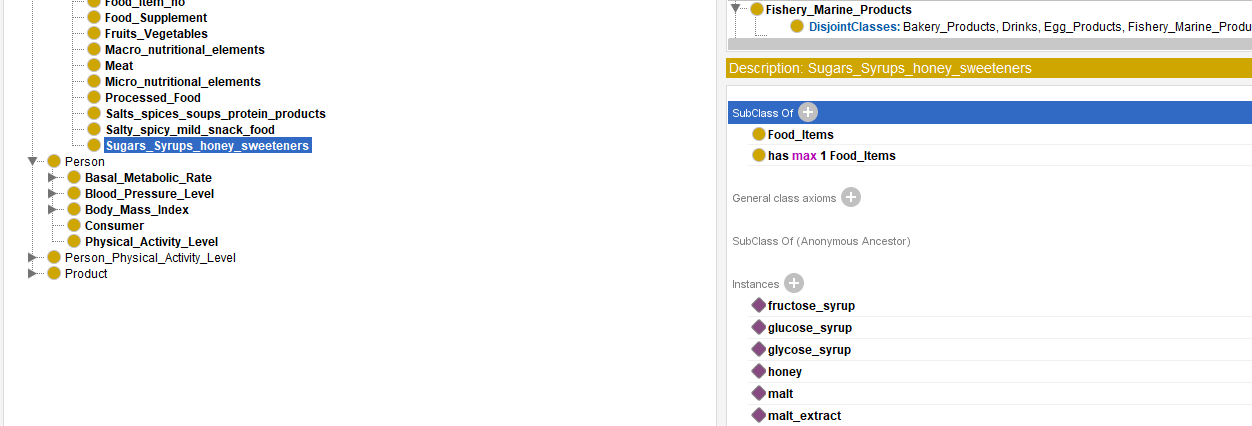


Fig 13.2 Cardinality

**Phase 14: Allvaluesfrom Somevaluesfrom Hasvalue**

**Phase 15: Queries**

These are the some SPARQL queries which I have used to fetch data from food ontology.

1. Select? Diseases? Thing

Where (? Diseases rdfs:subClassOf ?Thing)

1. Select? Allergy? Diseases

Where (? Allergy rdfs:subClassOf ?Diseases)

1. Select? Fish Allergy? Diseases

Where (? Fish Allergy rdfs:subClassOf ?Allergy)

1. Select? Fish Allergy? Diseases

Where (? Fish Allergy rdfs:subClassOf ?Allergy)

1. Select? Fish Allergy? Diseases

Where (? Fruits and Vegetables Allergy rdfs:subClassOf ?Allergy)

1. Select? Fish Allergy? Diseases

Where (? Gluten Allergy rdfs:subClassOf ?Allergy)

1. Select? Fish Allergy? Diseases

Where (? Fish Allergy rdfs:subClassOf ?Allergy)

1. Select? Wheat Protein Allergy? Diseases

Where (? Wheat Protein Allergy rdfs:subClassOf ?Allergy)

1. Select? Ingredients? Thing

Where (? Ingredients rdfs:subClassOf ?Thing)

1. Select? Contaminants? Ingredients

Where (? Contaminants rdfs:subClassOf ?Ingredients)

1. Select? Food Additives? Ingredients

Where (? Fish Food Additives rdfs:subClassOf ?Ingredients)

1. Select? Codex No? Food Additives

Where (? Codex No rdfs:subClassOf ?Food Additives)

1. Select? Food Items? Ingredients

Where (? Food Items rdfs:subClassOf ?Ingredients)

1. Select? Bakery Products? Food Items

Where (? Bakery Products rdfs:subClassOf ?Food Items)

1. Select? Drinks? Food Items

Where (? Drinks rdfs:subClassOf ?Food Items)

1. Select? Food Supplements? Food Items

Where (? Food Supplements rdfs:subClassOf ?Food Items)

1. Select? Processed Food? Food Items

Where (? Processed Food rdfs:subClassOf ?Food Items)

1. Select? Blood Pressure Level Range? Blood Pressure Level

Where (? Blood Pressure Level Range rdfs:subClassOf ? Blood Pressure Level)

1. Select? Blood Pressure Level? Person

Where (? Blood Pressure Level rdfs:subClassOf ?Person)

1. Select? Body Mass Index Range? Body Mass Index

Where (? Body Mass Index Range rdfs:subClassOf ? Body Mass Index)

**Conclusion**

The framework is particularly intended for looking at bundled food items on market retires and proposing the chose item's propriety to food purchasers as indicated by their medical issue or prejudices. The framework utilizes its own metaphysics information base that includes four subsections: individual, sickness, item, and food fixings/compounds. The fixings/compounds include nourishing data about business items from the food field and substance point of view. This framework cosmology information base is utilized to divide information between versatile savvy gadgets of food purchasers and the item data set.