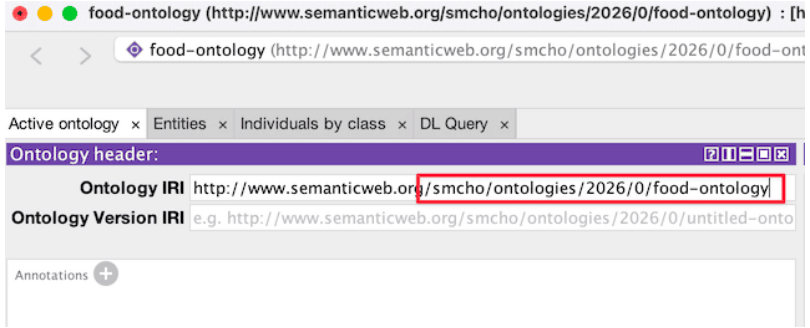
1. Protégé

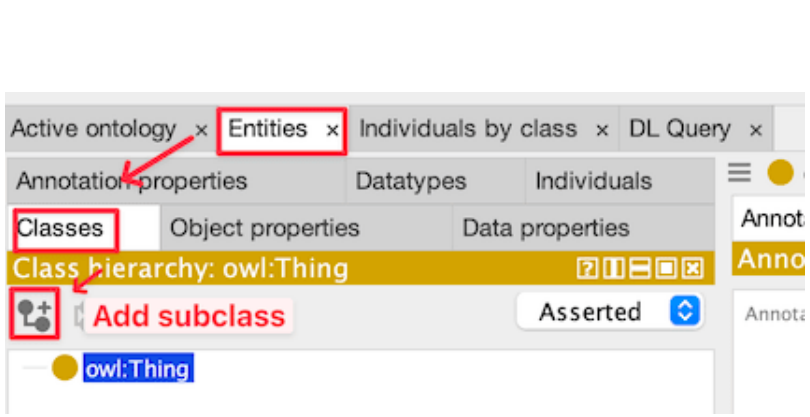
Create a new ontology project

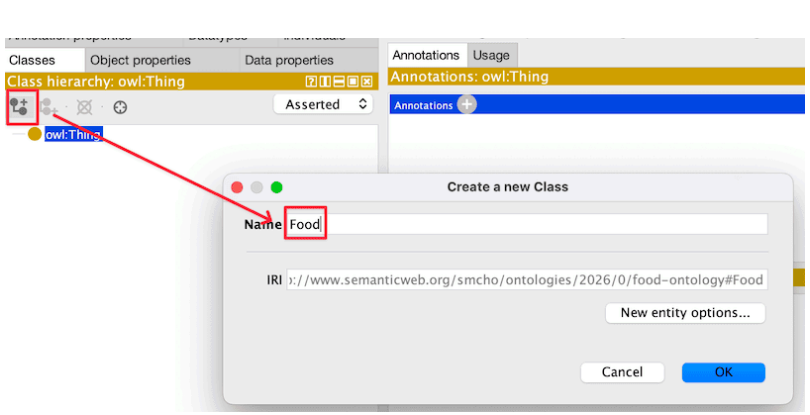
File -> New

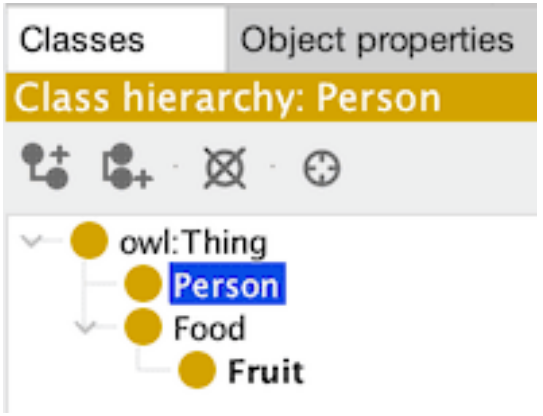


- Ontology project is created & file is loaded
- You are ready to start modeling
- Change the IRI (Identifier) of the ontology

Create Classes

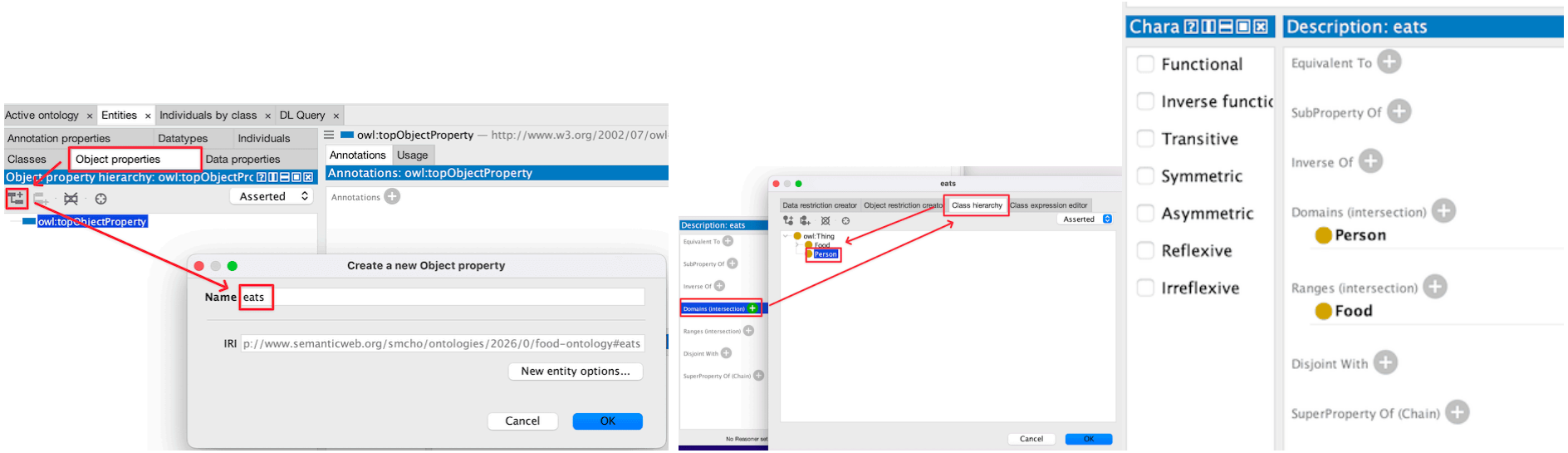






Create Relationships (Properties)

Makes the relationship that: Person (Domain) eats Food (Range)



Up to this point, we have created the following ontology:

1. semantic rule: eats (Person -> Food)

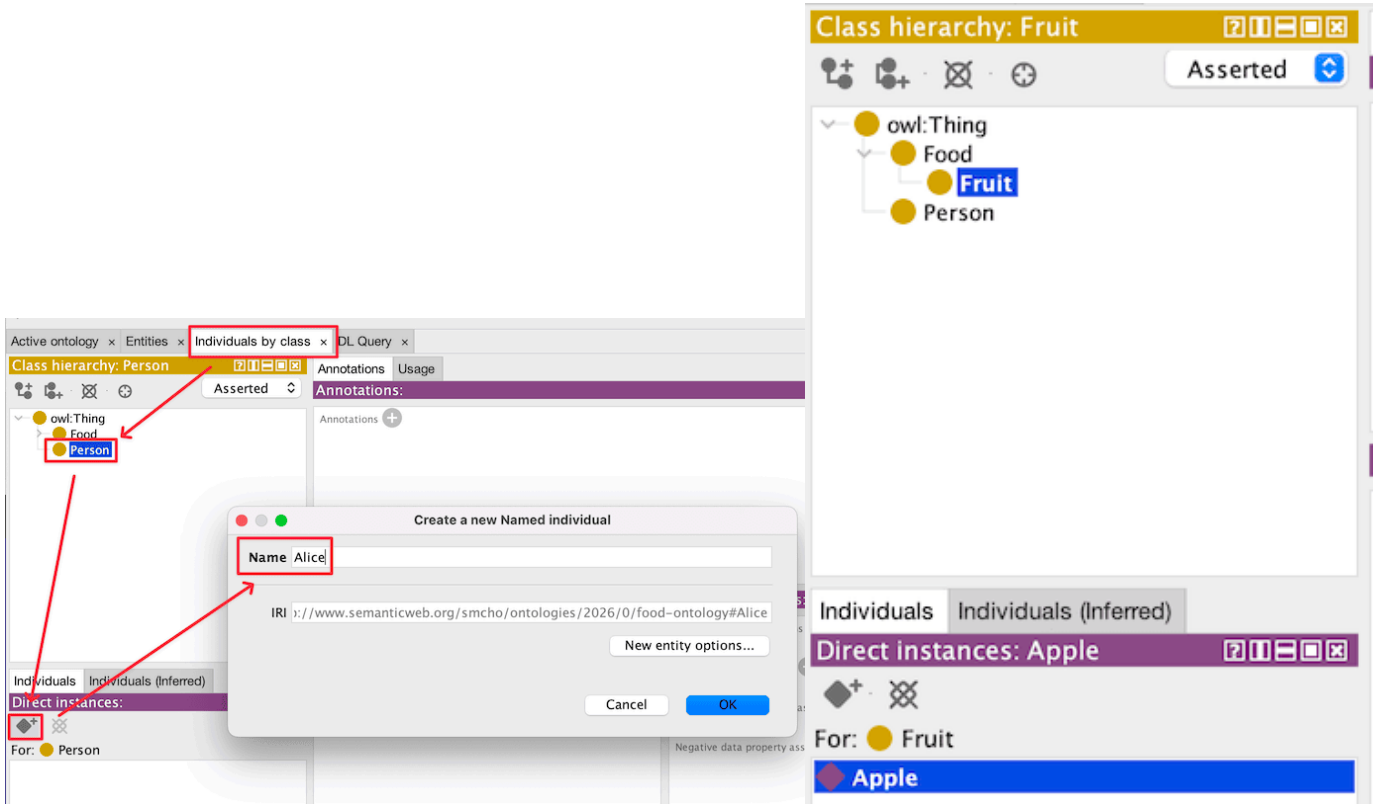
- Automatic type inference
- Data validation
- Logical reasoning

2. If we assert: `Alice eats Apple`

Progege can infer that:

```
Alice is a Person
Apple is Food
```

Create Individuals (Instances)

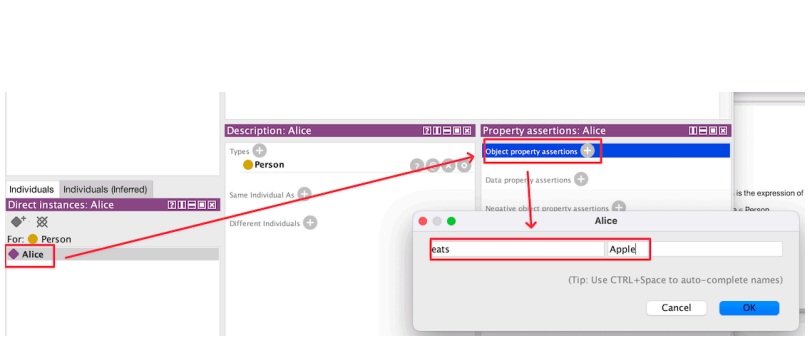


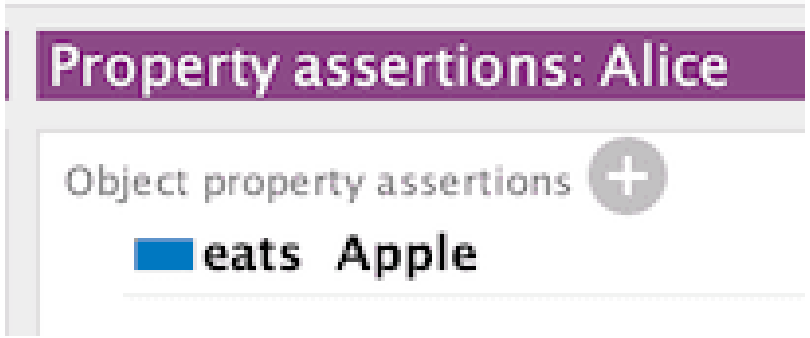
This is the expression of the individuals we just created:

$\text{Alice} \in \text{Person}$

$\text{Apple} \in \text{Fruit}$

Assert Relationships between Individuals





- Assert that Alice eats Apple

  -Assert that Alice eats Apple

We just created a semantic triple:

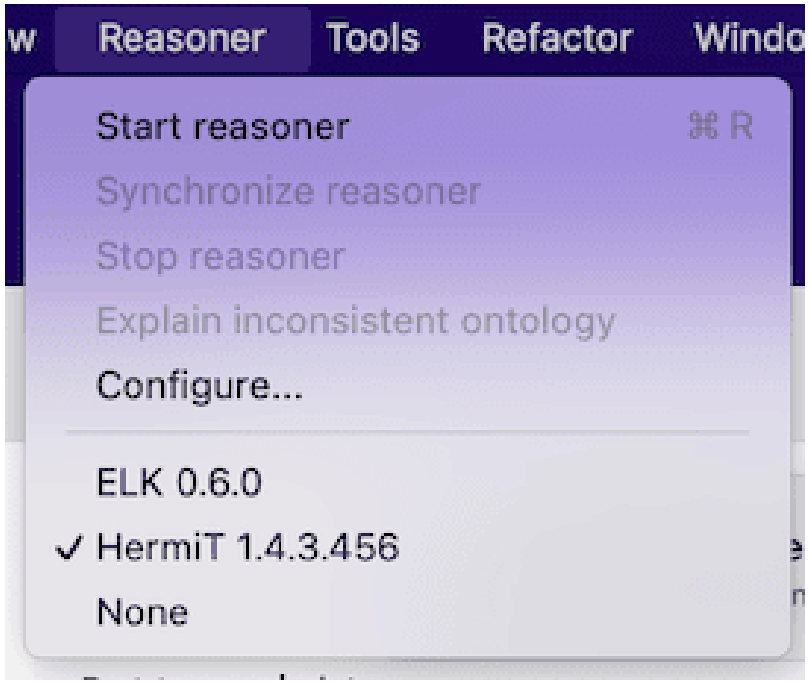
```
Subject   Predicate   Object
Alice  —  eats   →   Apple
```

- Now we have:
  - Alice $\in$ Person
  - Apple $\in$ Food
  - eats(Alice, Apple)

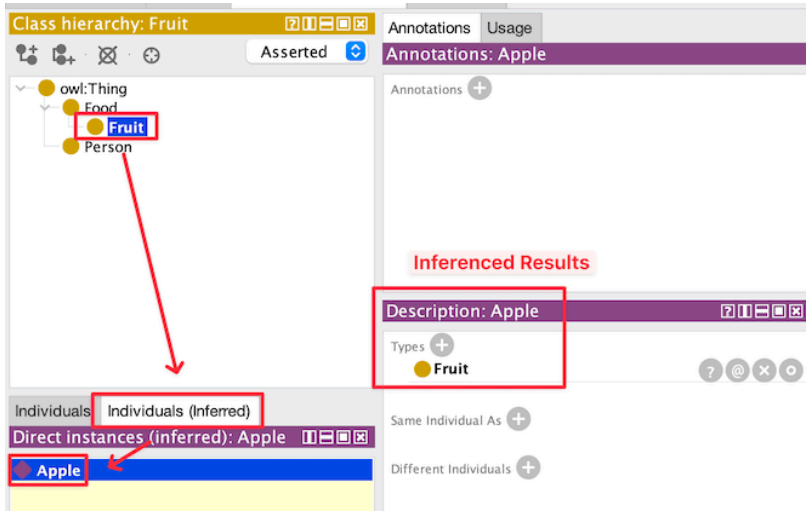
Run the Reasonor - Automatic Inference in Protege

Make sure this is checked at the bottom right corner:

- Click on the "Reasoner" menu
- Select "Start Reasoner"



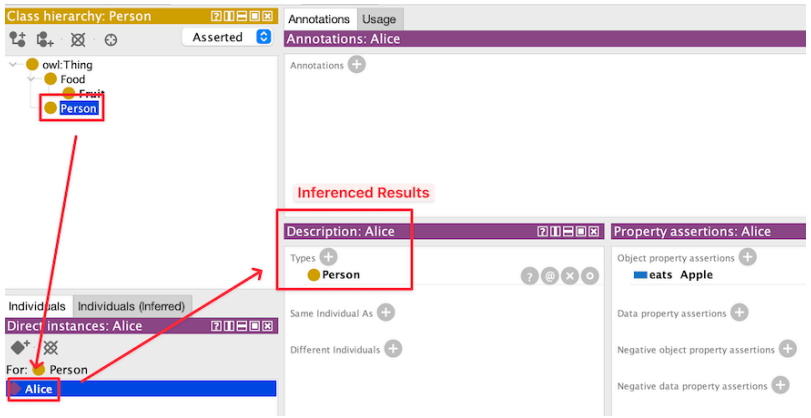
Verify Instance Inference



Even if you never manually assigned Food.

Protege infers that Apple is Food because Alice (a Person) eats Apple.

```
Fruit ⊆ Food
Apple ∈ Fruit
⇒ Apple ∈ Food
```

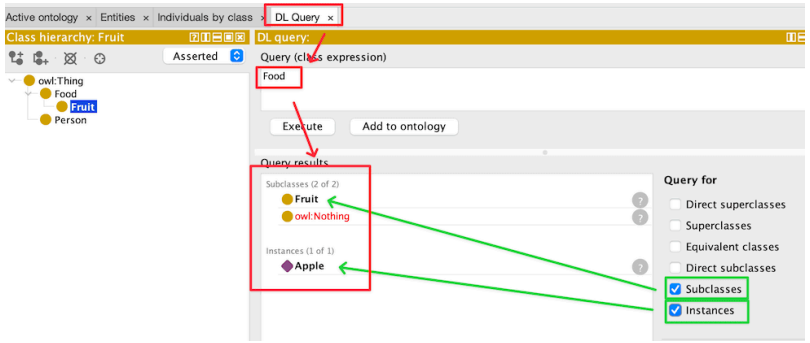


Even if you never manually assigned Person.

Protege infers that Alice is Person because Alice eats Apple (a Food).

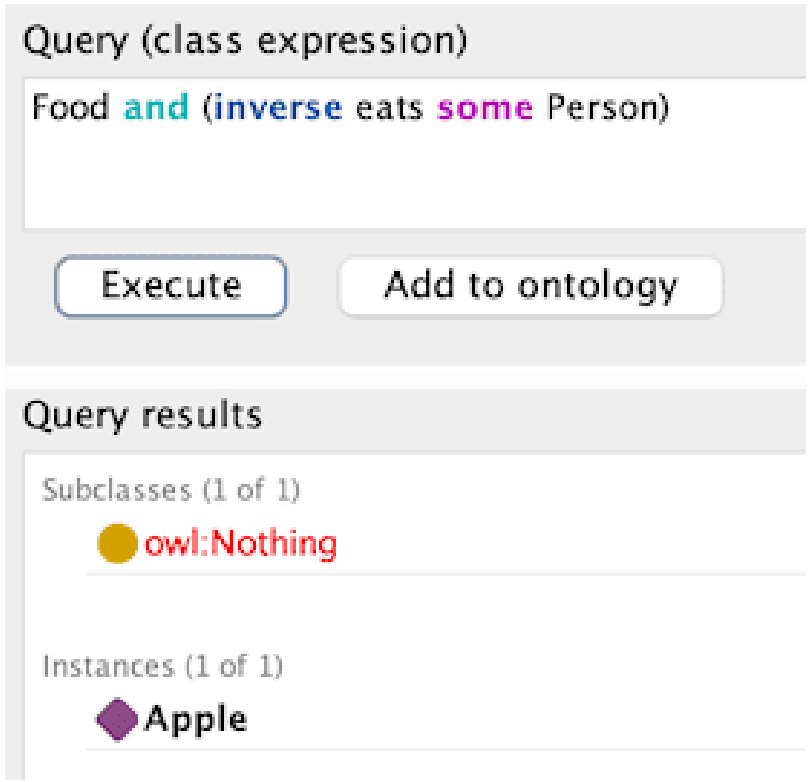
```
Domain(eats) = Person
Alice eats Apple
⇒ Alice ∈ Person
```

DL Query (Semantic Querying)



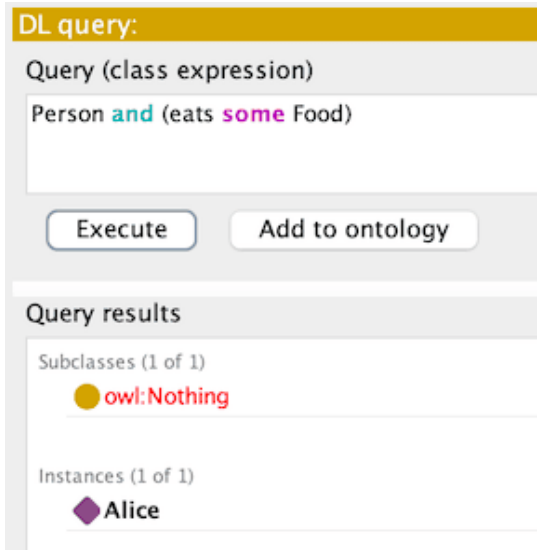
Entering "Food" in the DL Query tab and clicking "Execute" will return all individuals inferred to be of type Food.

$$\texttt{Fruit} \subseteq \texttt{Food}$$
$$\Rightarrow \texttt{Apple} \in \texttt{Food}$$




```
Food and (inverse eats some Person)
```

```
Alice eats Apple
⇒ Apple is eaten by a Person
⇒ Apple ∈ Food
```




```
Person and (eats some Fruit)
```

```
Fruit ⊆ Food
Alice eats Apple
⇒ Alice eats some Fruit
⇒ Alice ∈ Person
```

Interpretation

This is not explicitly asserted:

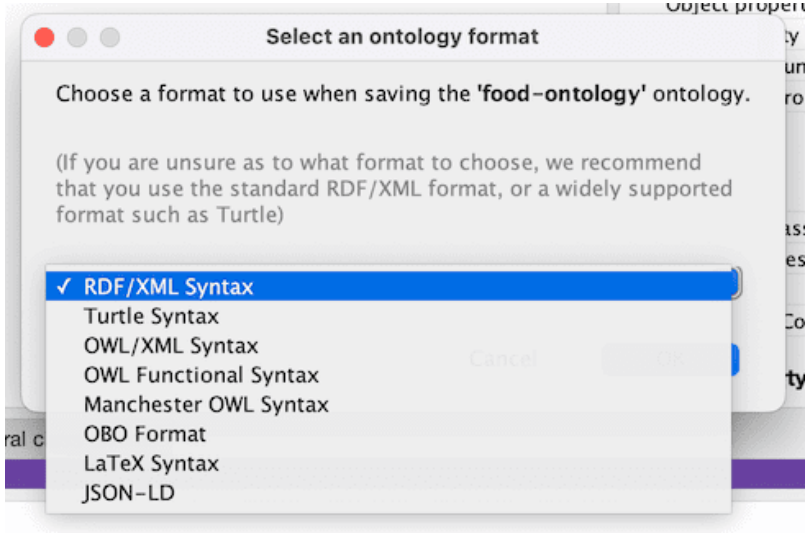
But the system figured it out automatically.

```
Apple is Food.
```

That's what makes ontologies powerful!

- Knowledge graphs
- AI reasoning
- Semantic search
- Rule-based systems
- RAG grounding

Save ontology file



- Save & Load as RDF/XML format file

What is saved? Persistent Knowledge

ontology assertsions are saved:

```
Fruit ⊆ Food
Alice eats Apple
Apple ∈ Fruit
Domain(eats) = Person
Range(eats) = Food
```

What is NOT saved?

Reasoner inferences are NOT saved:

```
Apple ∈ Food
Alice ∈ Person
```

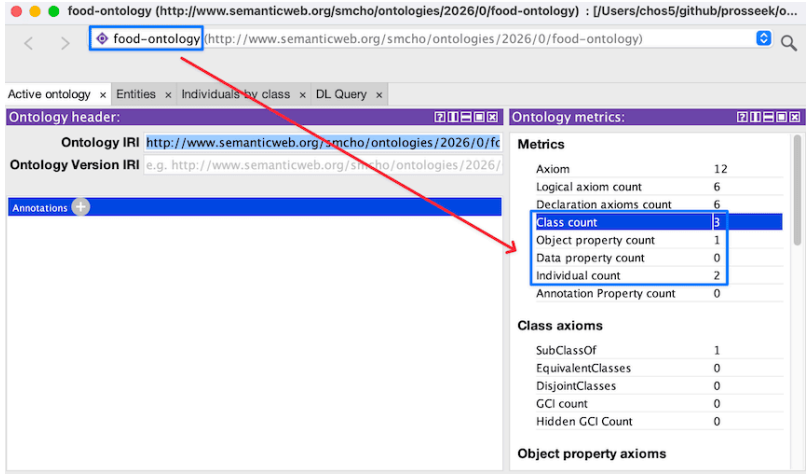
DL Queries are NOT saved.

```
Food and (inverse eats some Person)
```

- This is "Query", not Ontology Assertion.
- Inferences are not saved, only explicit assertions are saved.
- When you close and reopen the ontology file, the inferred types will be lost.

Load ontology file

- Open the saved RDF/XML file in Protégé
  - Choose File -> Open
  - Select the saved RDF/XML file



- All the explicit assertions are loaded
- Run the Reasoner again to re-infer types

File Formats

- Turtle file: `.ttl`
- RDF/XML file: `.owl` or `.rdf`
- OWL Functional Syntax: `.ofn`
- OWL/XML Syntax: `.owlxml`
- Manchester Syntax: `.omn`
- JSON-LD: `.jsonld`

Install Plugins

1. File -> Check for plugins...
2. Select and install desired plugins (e.g., SHACL, SPARQL Query)
3. Restart Protégé to activate plugins
4. Window -> Tabs -> Select installed plugin tab