Ontologies in an Automated Workflow Part 2

John Beverley

Assistant Professor, *University at Buffalo*
Co-Director, National Center for Ontological Research
Affiliate Faculty, *Institute of Artificial Intelligence and Data Science*

Outline

- Murder Mystery Revisited
- SPARQL & DBPedia
- CCO Quality Control SPARQL Checks

Outline

- Murder Mystery Revisited

- SPARQL & DBPedia

- CCO Quality Control SPARQL Checks

A Murder Mystery

A Psychologist has been Murdered…

1. Suspects: Werner, Mark, Neil, and Barry
2. Mark plays violin
3. Neil, Mark, and Barry daydream
4. Barry cannot play music, but reads and solves problems
5. Werner never daydreams
6. Mark doesn’t problem solve
7. Werner plays trumpet and problem solves
8. Neil cannot play music and cannot read
9. The murderer daydreams, is either a musician or literate, and solves problems

A Psychologist has been Murdered…

1. Suspects: Werner, Mark, Neil, and Barry
2. Mark plays violin
3. Neil, Mark, and Barry daydream
4. Barry cannot play music, but reads and solves problems
5. Werner never daydreams
6. Mark doesn’t problem solve
7. Werner plays trumpet and problem solves
8. Neil cannot play music and cannot read
9. The murderer daydreams, is either a musician or literate, and solves problems

Represent these facts in Protégé.

Create a ‘Murderer’ class that is equivalent to line 9.

If done correctly, the Protege reasoner will return exactly one member of this class.

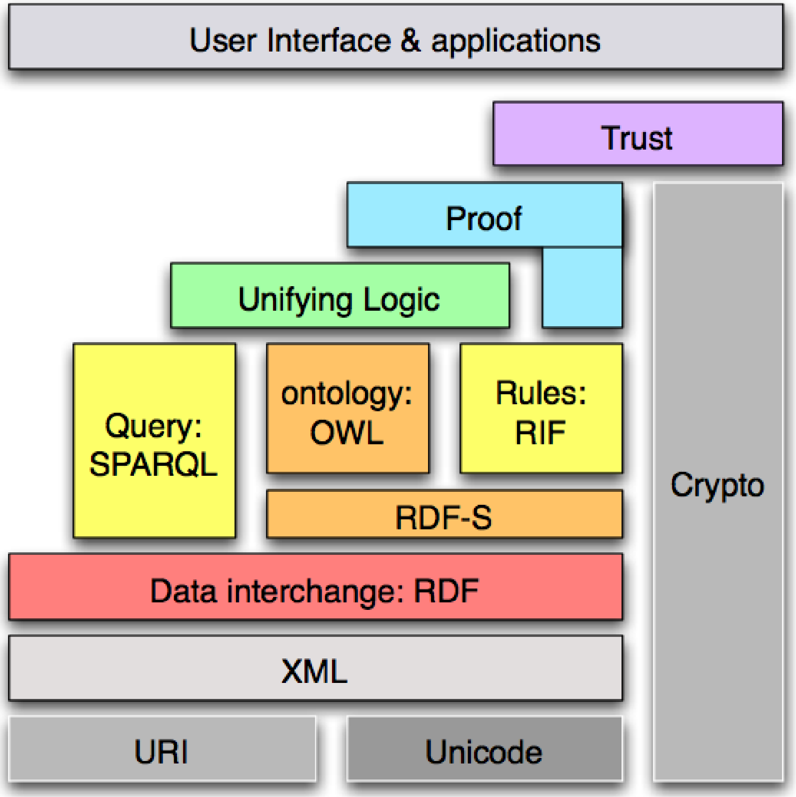
Outline

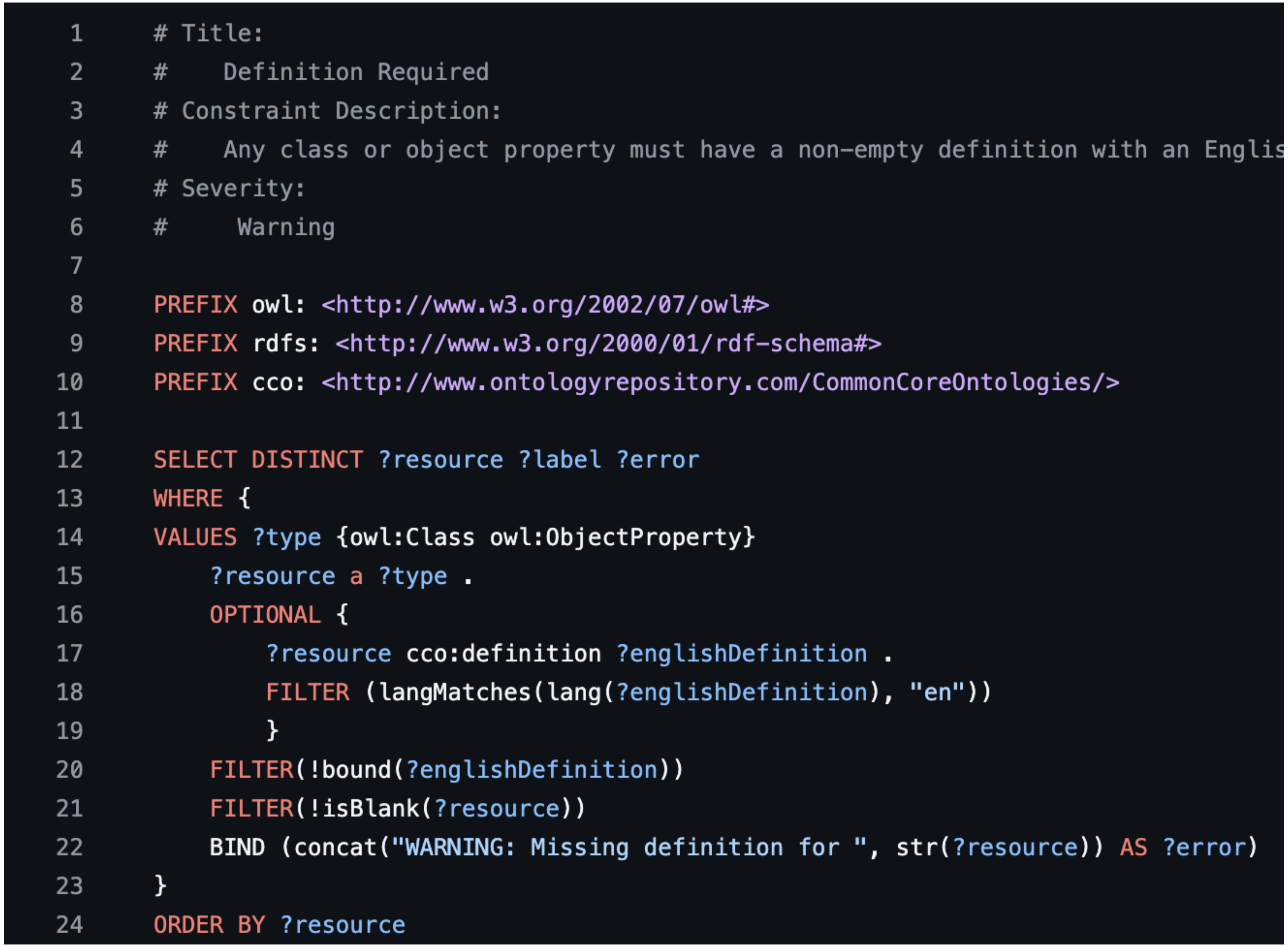
- Murder Mystery Revisited
- SPARQL & DBPedia
- CCO Quality Control SPARQL Checks

Semantic Web Stack

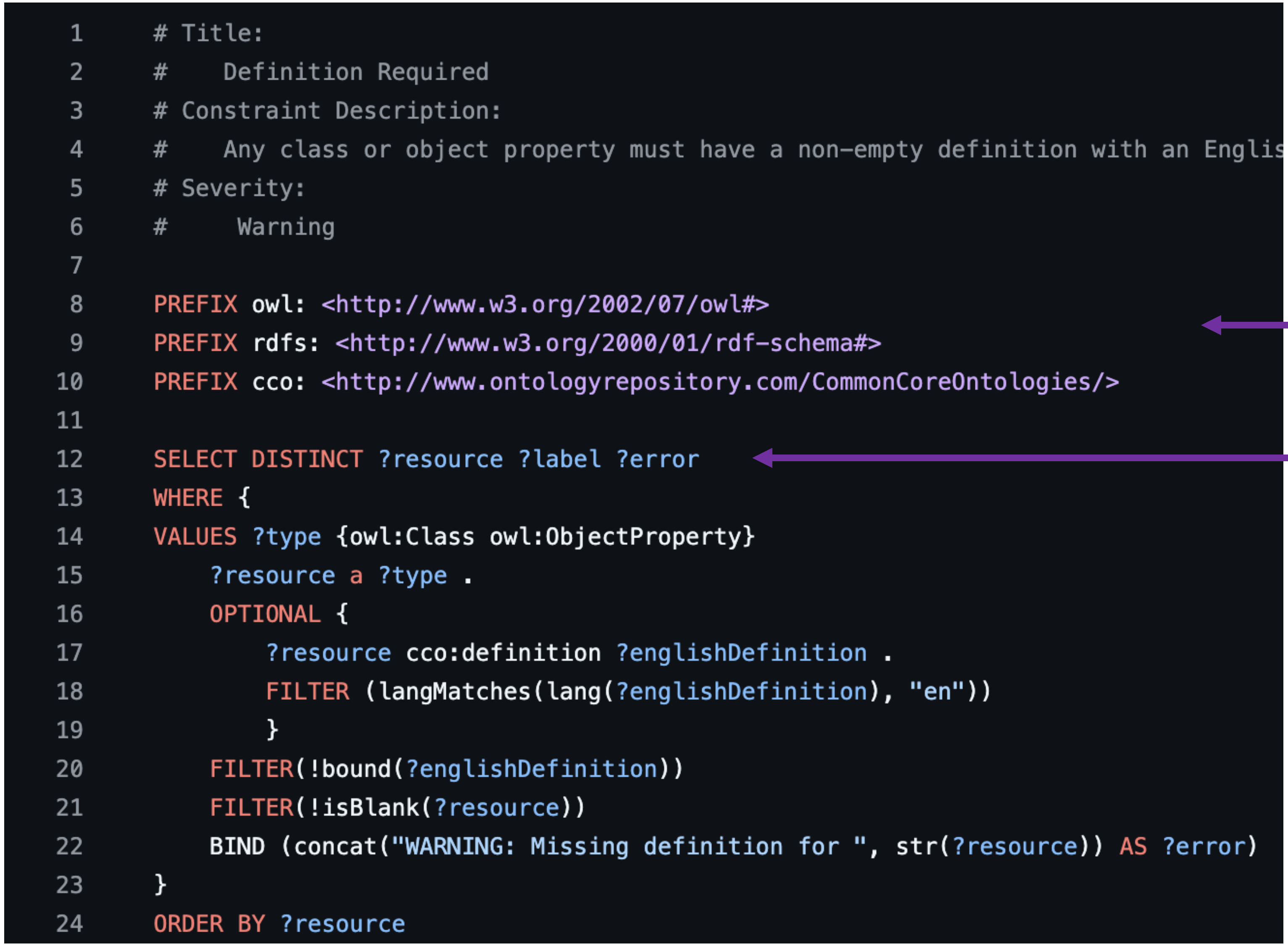
- "SPARQL" stands for:
  - **S**PARQL **P**rotocol
  - **A**nd **R**DF
  - **Q**uery **L**anguage

- SPARQL is a:
  - Core semantic web technology
  - *Query language* for RDF
  - A *protocol* for transmitting queries over HTTP





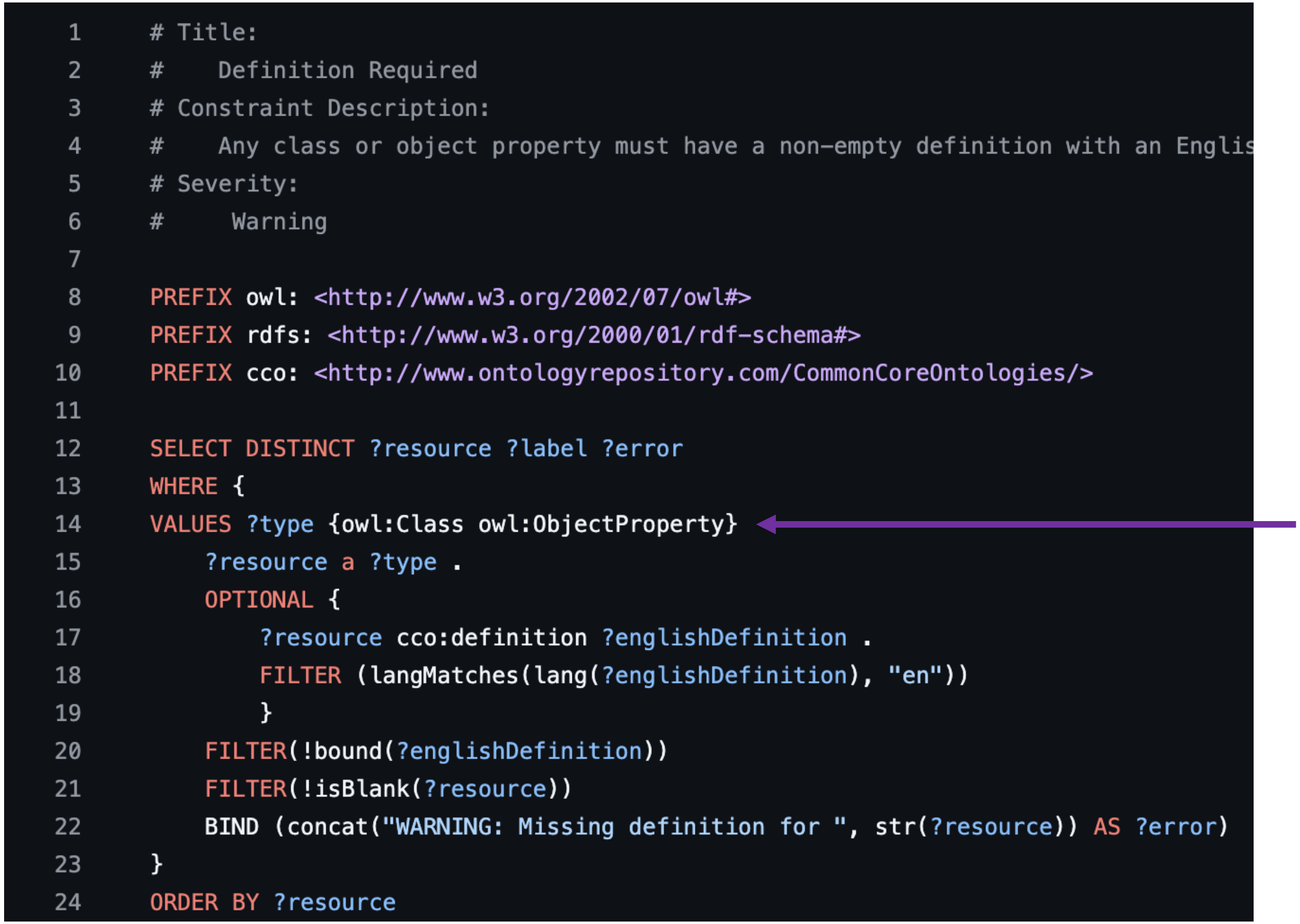
```
# Title:
#    Definition Required
# Constraint Description:
#    Any class or object property must have a non-empty definition with an Englis
# Severity:
#     Warning

PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX cco: <http://www.ontologyrepository.com/CommonCoreOntologies/>

SELECT DISTINCT ?resource ?label ?error
WHERE {
VALUES ?type {owl:Class owl:ObjectProperty}
    ?resource a ?type .
    OPTIONAL {
        ?resource cco:definition ?englishDefinition .
        FILTER (langMatches(lang(?englishDefinition), "en"))
        }
    FILTER(!bound(?englishDefinition))
    FILTER(!isBlank(?resource))
    BIND (concat("WARNING: Missing definition for ", str(?resource)) AS ?error)
}
ORDER BY ?resource
```



```
# Title:
#    Definition Required
# Constraint Description:
#    Any class or object property must have a non-empty definition with an Englis
# Severity:
#     Warning

PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX cco: <http://www.ontologyrepository.com/CommonCoreOntologies/>

SELECT DISTINCT ?resource ?label ?error
WHERE {
VALUES ?type {owl:Class owl:ObjectProperty}
    ?resource a ?type .
    OPTIONAL {
        ?resource cco:definition ?englishDefinition .
        FILTER (langMatches(lang(?englishDefinition), "en"))
        }
    FILTER(!bound(?englishDefinition))
    FILTER(!isBlank(?resource))
    BIND (concat("WARNING: Missing definition for ", str(?resource)) AS ?error)
}
ORDER BY ?resource
```

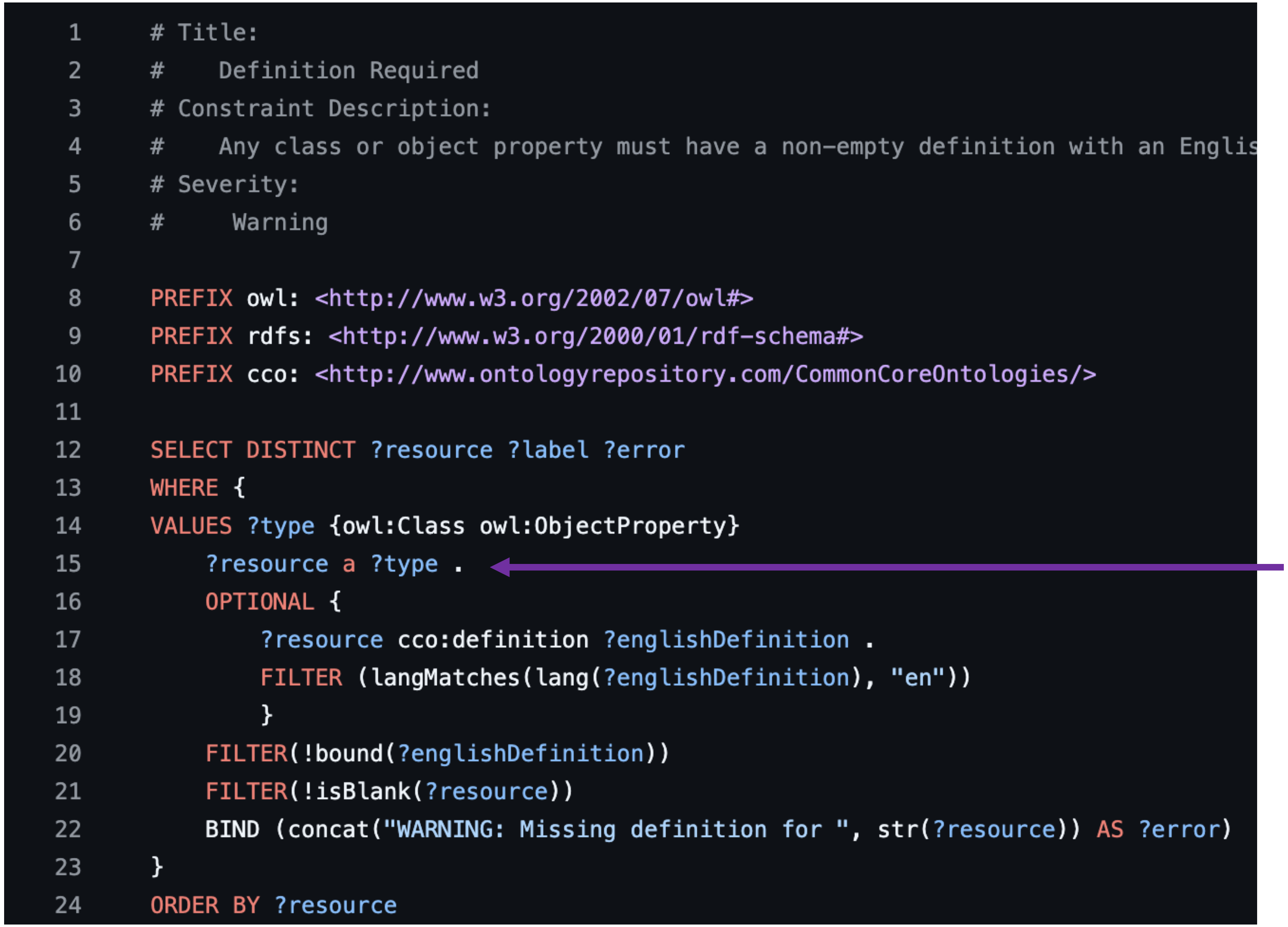
Declare the namespace

Declare return variables



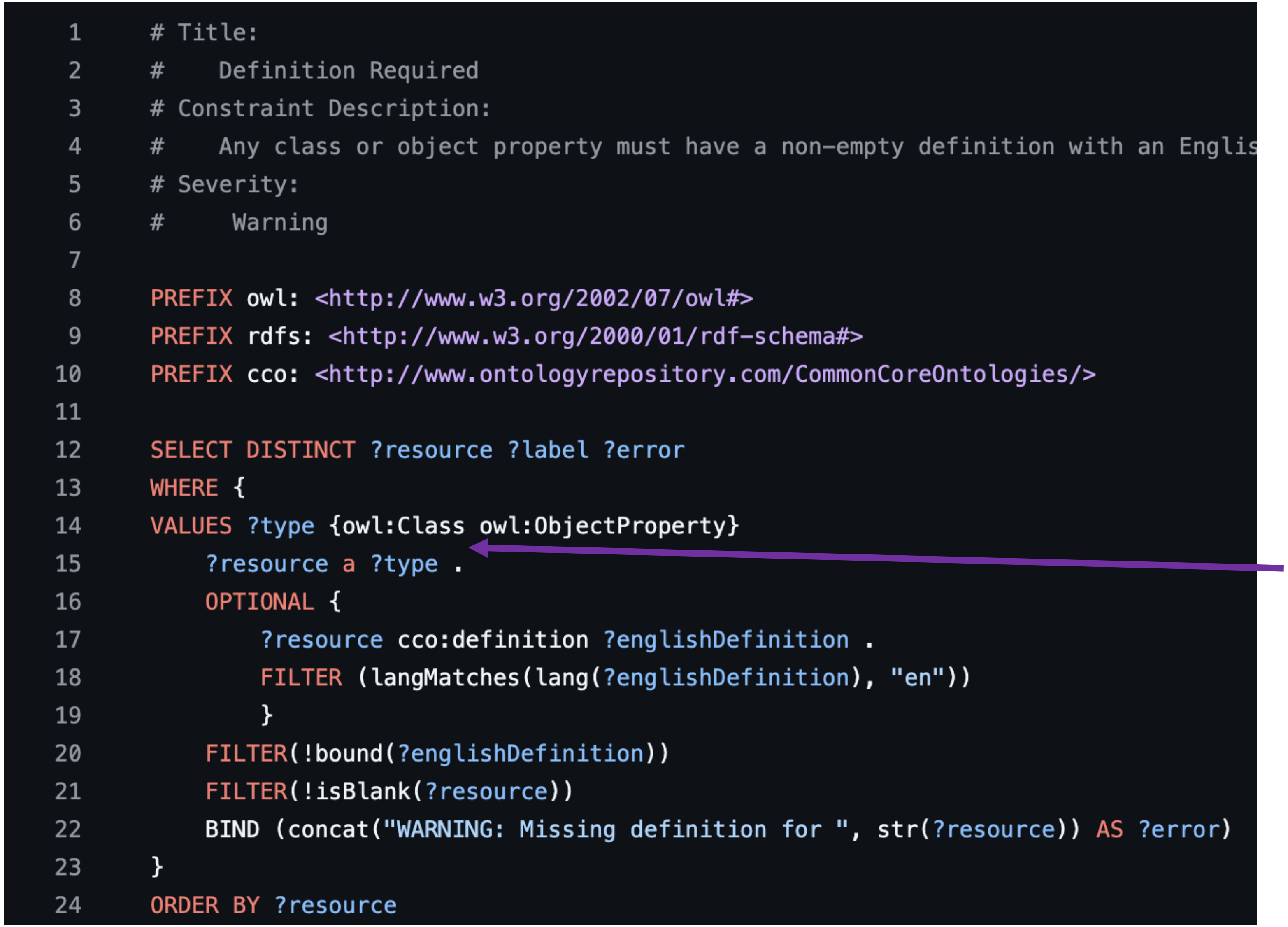
```
# Title:
#    Definition Required
# Constraint Description:
#    Any class or object property must have a non-empty definition with an Englis
# Severity:
#     Warning

PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX cco: <http://www.ontologyrepository.com/CommonCoreOntologies/>

SELECT DISTINCT ?resource ?label ?error
WHERE {
VALUES ?type {owl:Class owl:ObjectProperty}
    ?resource a ?type .
    OPTIONAL {
        ?resource cco:definition ?englishDefinition .
        FILTER (langMatches(lang(?englishDefinition), "en"))
        }
    FILTER(!bound(?englishDefinition))
    FILTER(!isBlank(?resource))
    BIND (concat("WARNING: Missing definition for ", str(?resource)) AS ?error)
}
ORDER BY ?resource
```

VALUES type means that “type” is a variable ranging over everything in the brackets



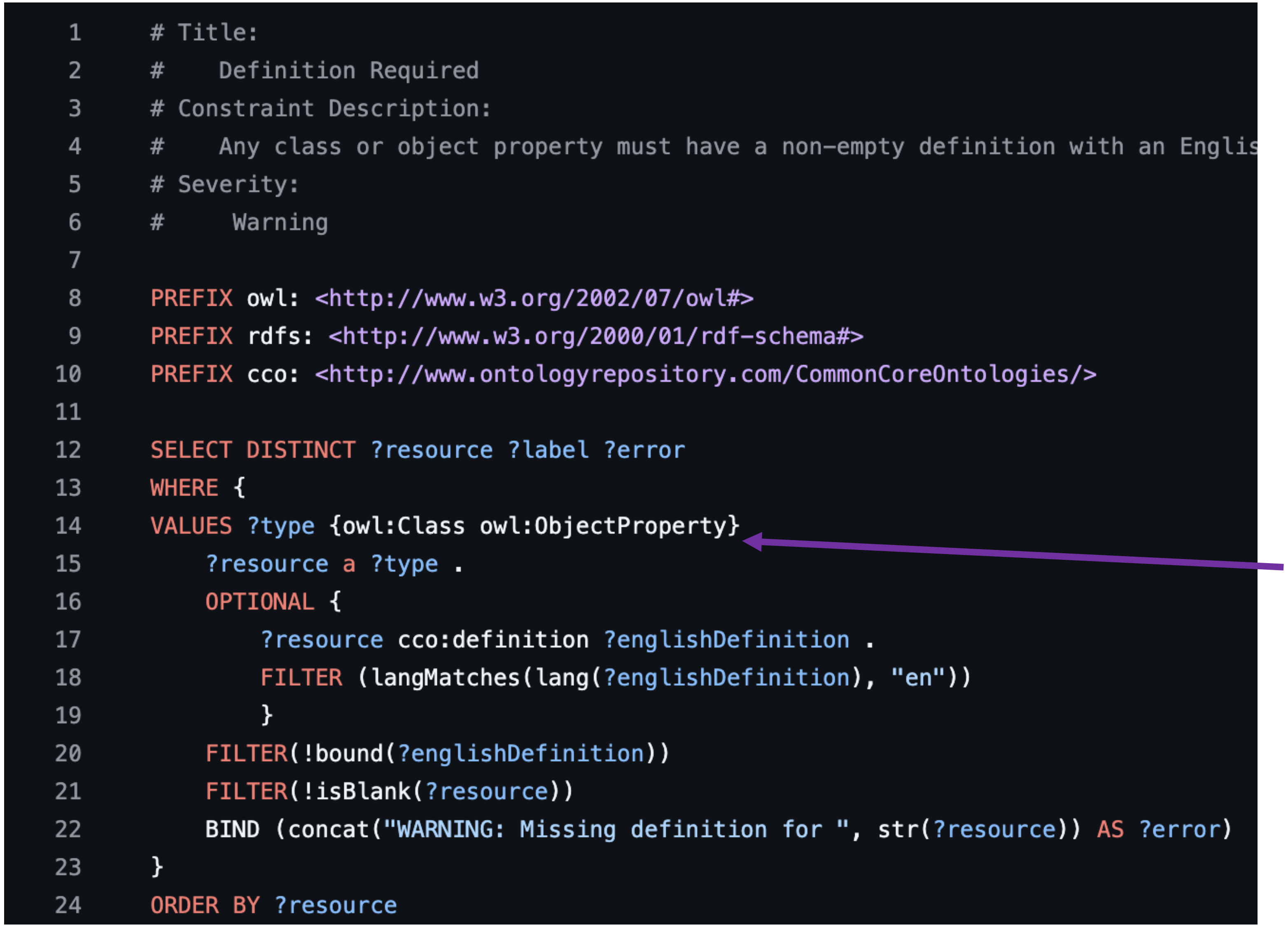
```
# Title:
#    Definition Required
# Constraint Description:
#    Any class or object property must have a non-empty definition with an Englis
# Severity:
#     Warning

PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX cco: <http://www.ontologyrepository.com/CommonCoreOntologies/>

SELECT DISTINCT ?resource ?label ?error
WHERE {
VALUES ?type {owl:Class owl:ObjectProperty}
    ?resource a ?type .
    OPTIONAL {
        ?resource cco:definition ?englishDefinition .
        FILTER (langMatches(lang(?englishDefinition), "en"))
        }
    FILTER(!bound(?englishDefinition))
    FILTER(!isBlank(?resource))
    BIND (concat("WARNING: Missing definition for ", str(?resource)) AS ?error)
}
ORDER BY ?resource
```

That is, for any resource that is…



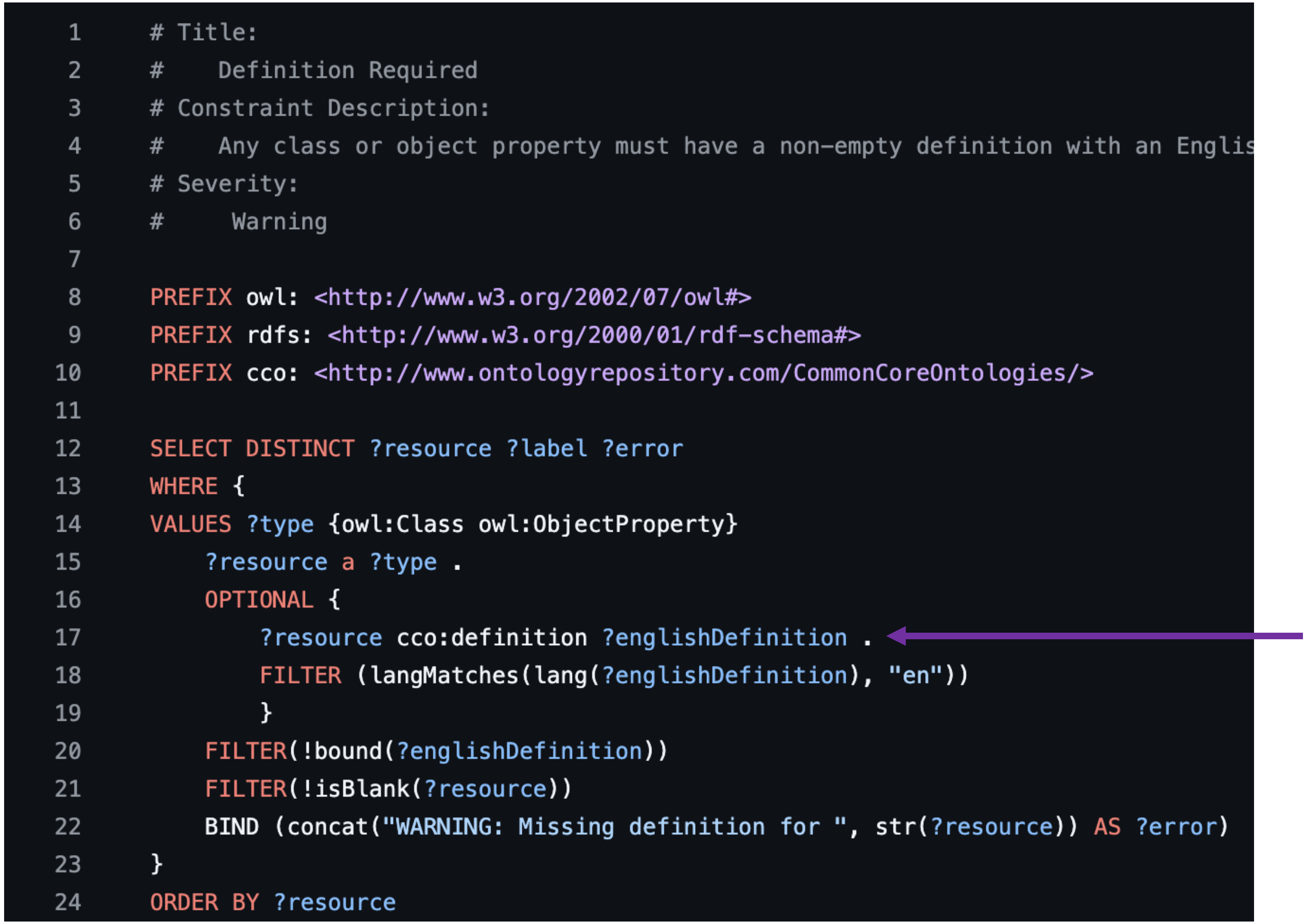
```
# Title:
#    Definition Required
# Constraint Description:
#    Any class or object property must have a non-empty definition with an Englis
# Severity:
#     Warning

PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX cco: <http://www.ontologyrepository.com/CommonCoreOntologies/>

SELECT DISTINCT ?resource ?label ?error
WHERE {
VALUES ?type {owl:Class owl:ObjectProperty}
    ?resource a ?type .
    OPTIONAL {
        ?resource cco:definition ?englishDefinition .
        FILTER (langMatches(lang(?englishDefinition), "en"))
        }
    FILTER(!bound(?englishDefinition))
    FILTER(!isBlank(?resource))
    BIND (concat("WARNING: Missing definition for ", str(?resource)) AS ?error)
}
ORDER BY ?resource
```

…an owl:Class…



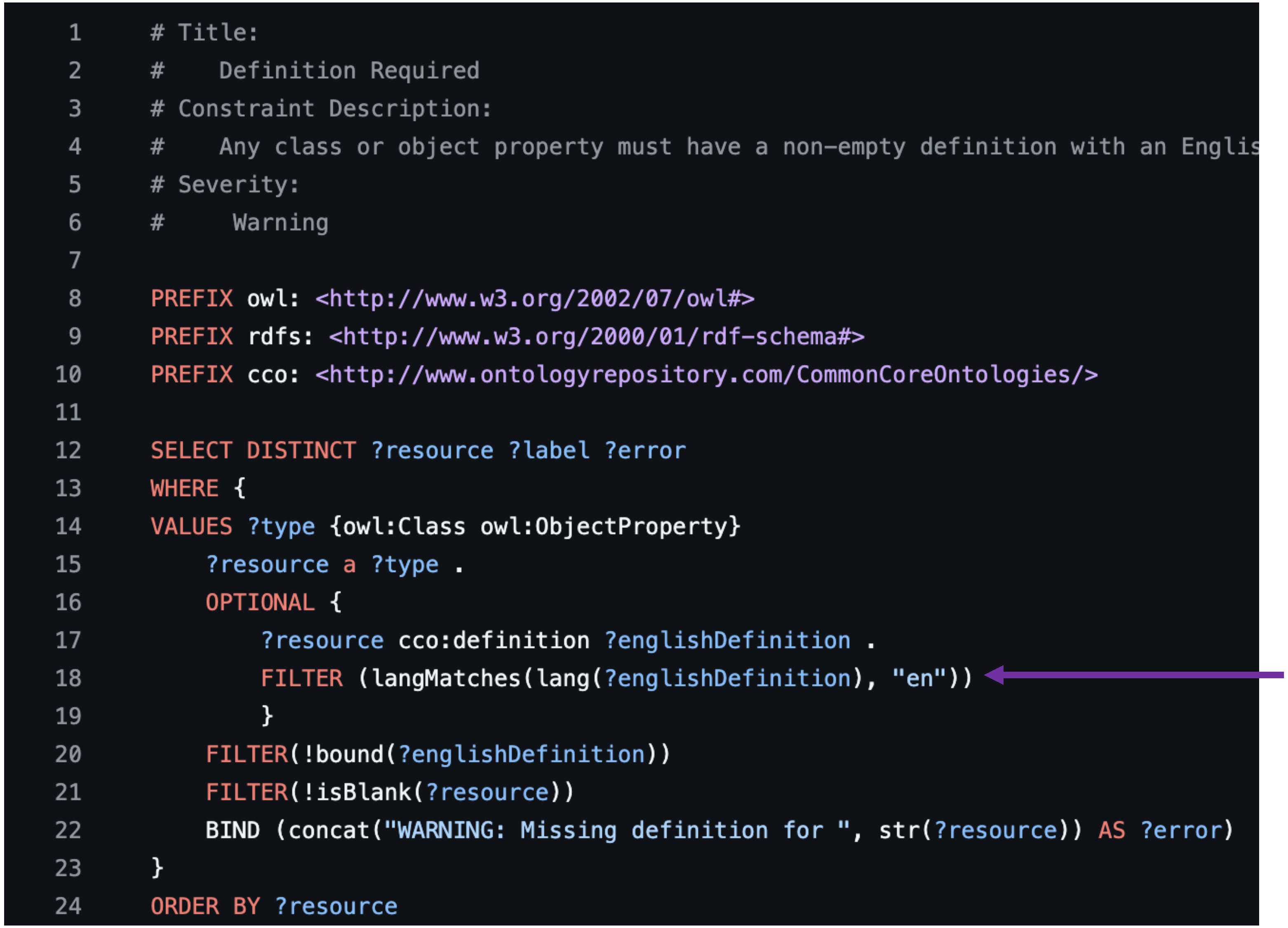
```
# Title:
#    Definition Required
# Constraint Description:
#    Any class or object property must have a non-empty definition with an Englis
# Severity:
#     Warning

PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX cco: <http://www.ontologyrepository.com/CommonCoreOntologies/>

SELECT DISTINCT ?resource ?label ?error
WHERE {
VALUES ?type {owl:Class owl:ObjectProperty}
    ?resource a ?type .
    OPTIONAL {
        ?resource cco:definition ?englishDefinition .
        FILTER (langMatches(lang(?englishDefinition), "en"))
        }
    FILTER(!bound(?englishDefinition))
    FILTER(!isBlank(?resource))
    BIND (concat("WARNING: Missing definition for ", str(?resource)) AS ?error)
}
ORDER BY ?resource
```

…or an owl:objectProperty



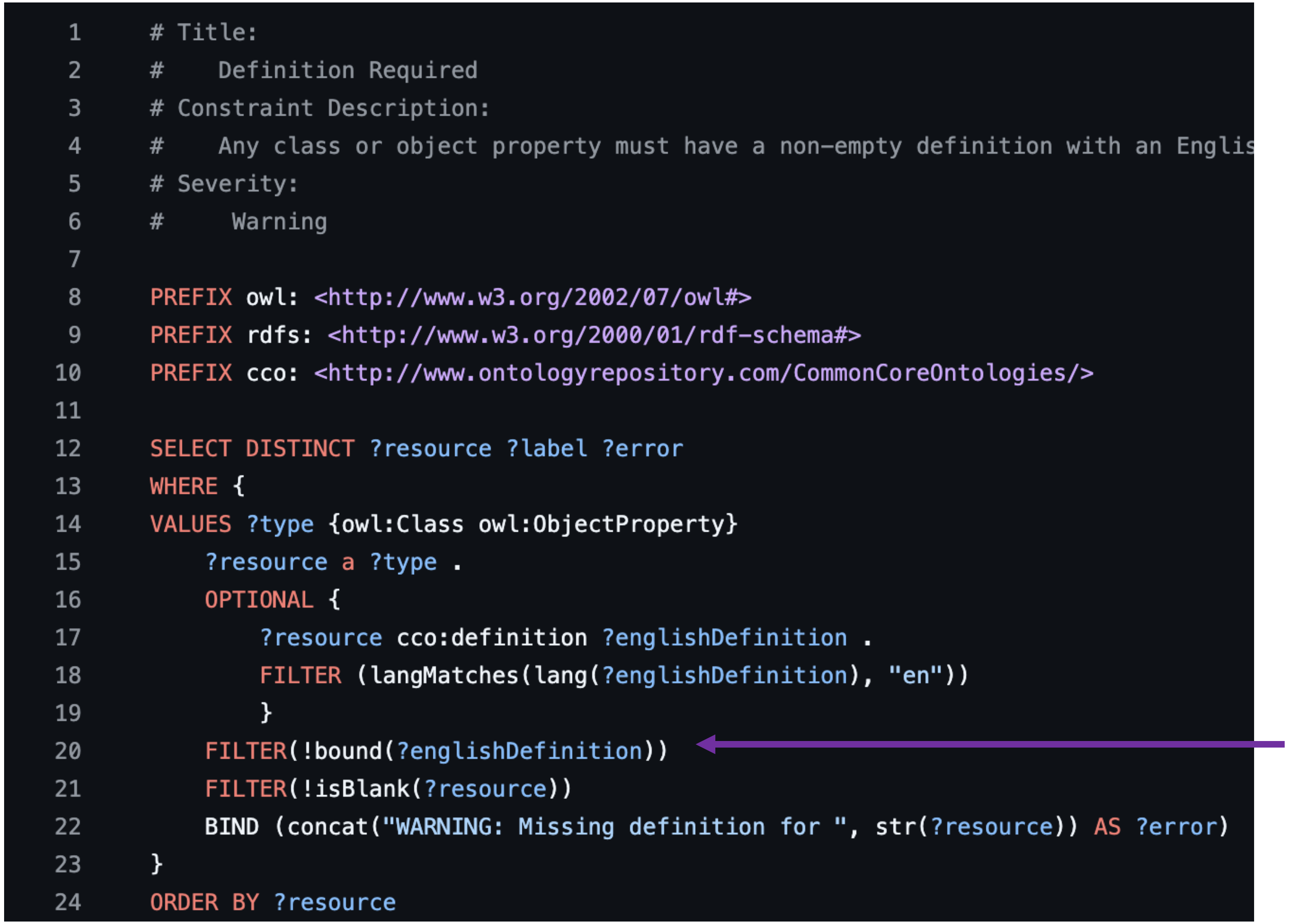
```sparql
# Title:
#    Definition Required
# Constraint Description:
#    Any class or object property must have a non-empty definition with an Englis
# Severity:
#     Warning

PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX cco: <http://www.ontologyrepository.com/CommonCoreOntologies/>

SELECT DISTINCT ?resource ?label ?error
WHERE {
VALUES ?type {owl:Class owl:ObjectProperty}
    ?resource a ?type .
    OPTIONAL {
        ?resource cco:definition ?englishDefinition .
        FILTER (langMatches(lang(?englishDefinition), "en"))
        }
    FILTER(!bound(?englishDefinition))
    FILTER(!isBlank(?resource))
    BIND (concat("WARNING: Missing definition for ", str(?resource)) AS ?error)
}
ORDER BY ?resource
```

Return resource even if the definition is missing



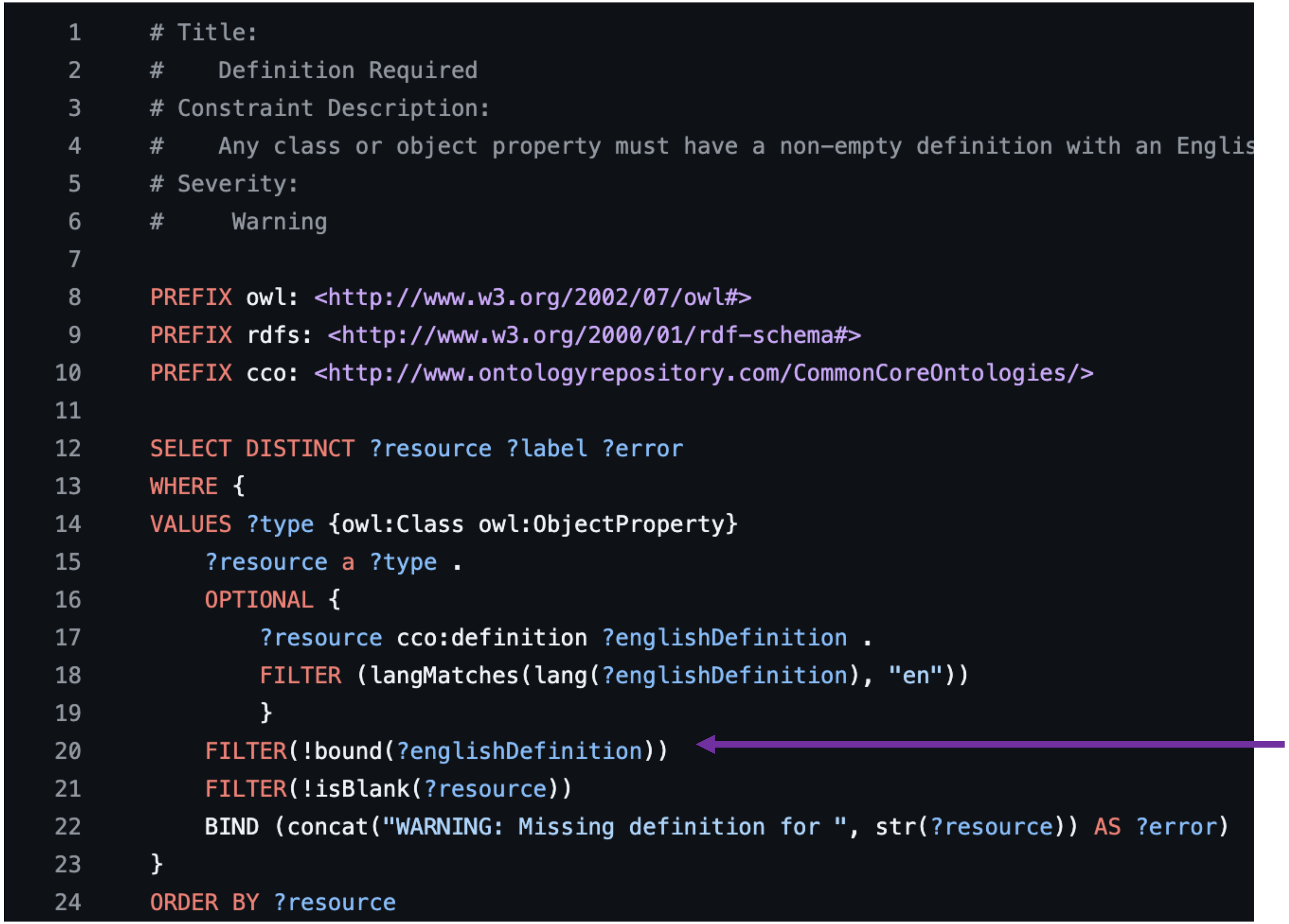
```
# Title:
#    Definition Required
# Constraint Description:
#    Any class or object property must have a non-empty definition with an Englis
# Severity:
#     Warning

PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX cco: <http://www.ontologyrepository.com/CommonCoreOntologies/>

SELECT DISTINCT ?resource ?label ?error
WHERE {
VALUES ?type {owl:Class owl:ObjectProperty}
    ?resource a ?type .
    OPTIONAL {
        ?resource cco:definition ?englishDefinition .
        FILTER (langMatches(lang(?englishDefinition), "en"))
        }
    FILTER(!bound(?englishDefinition))
    FILTER(!isBlank(?resource))
    BIND (concat("WARNING: Missing definition for ", str(?resource)) AS ?error)
}
ORDER BY ?resource
```

...or is missing an xsd English language tag



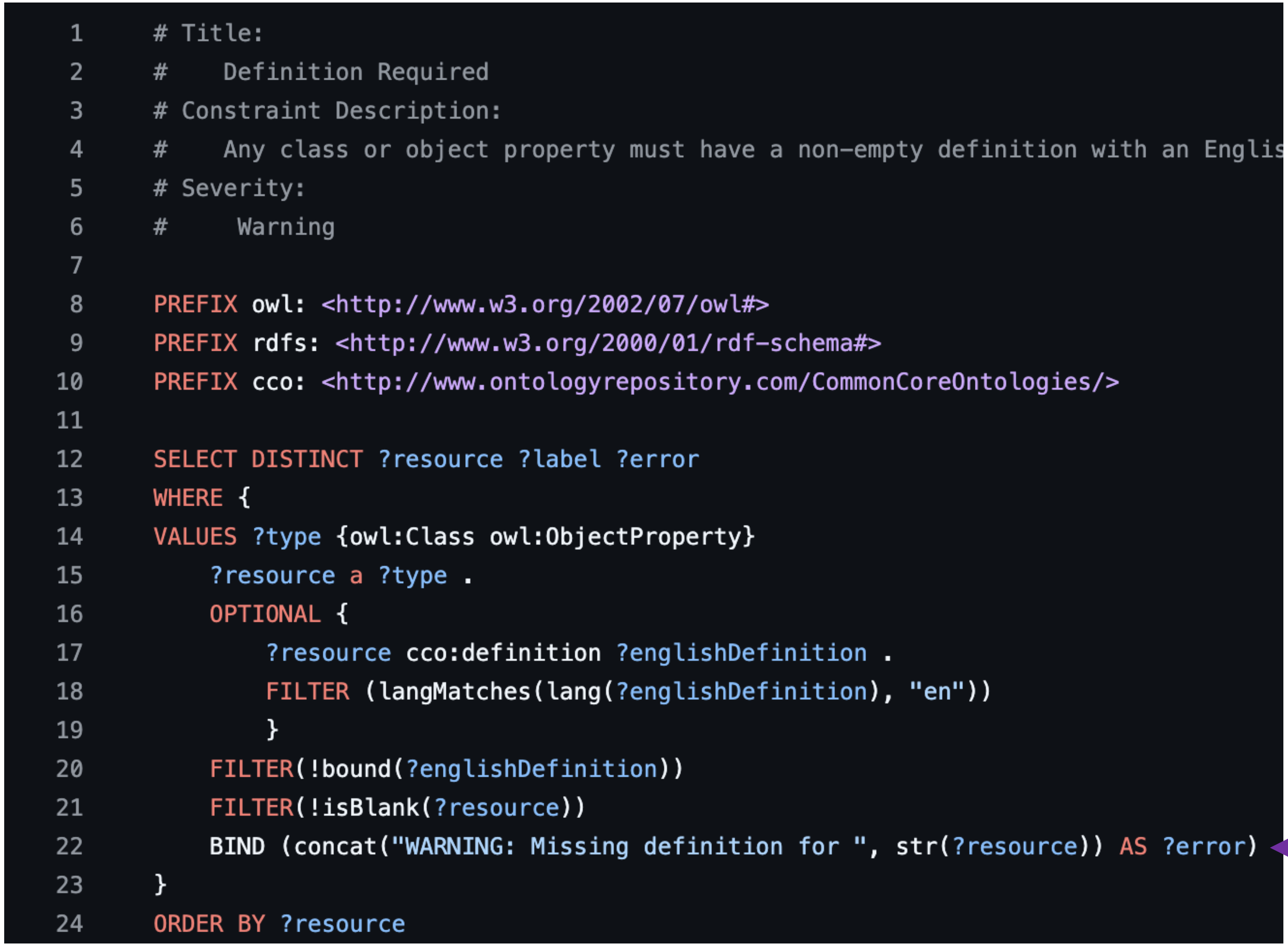
```
# Title:
#    Definition Required
# Constraint Description:
#    Any class or object property must have a non-empty definition with an Englis
# Severity:
#     Warning

PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX cco: <http://www.ontologyrepository.com/CommonCoreOntologies/>

SELECT DISTINCT ?resource ?label ?error
WHERE {
VALUES ?type {owl:Class owl:ObjectProperty}
    ?resource a ?type .
    OPTIONAL {
        ?resource cco:definition ?englishDefinition .
        FILTER (langMatches(lang(?englishDefinition), "en"))
        }
    FILTER(!bound(?englishDefinition))
    FILTER(!isBlank(?resource))
    BIND (concat("WARNING: Missing definition for ", str(?resource)) AS ?error)
}
ORDER BY ?resource
```

But indeed, keep only the results that don’t have a definition, i.e. ?englishDefinition is unbound because it’s empty



```
# Title:
#    Definition Required
# Constraint Description:
#    Any class or object property must have a non-empty definition with an Englis
# Severity:
#     Warning

PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX cco: <http://www.ontologyrepository.com/CommonCoreOntologies/>

SELECT DISTINCT ?resource ?label ?error
WHERE {
VALUES ?type {owl:Class owl:ObjectProperty}
    ?resource a ?type .
    OPTIONAL {
        ?resource cco:definition ?englishDefinition .
        FILTER (langMatches(lang(?englishDefinition), "en"))
        }
    FILTER(!bound(?englishDefinition))
    FILTER(!isBlank(?resource))
    BIND (concat("WARNING: Missing definition for ", str(?resource)) AS ?error)
}
ORDER BY ?resource
```

Ignore blank nodes



```
# Title:
#    Definition Required
# Constraint Description:
#    Any class or object property must have a non-empty definition with an Englis
# Severity:
#     Warning

PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX cco: <http://www.ontologyrepository.com/CommonCoreOntologies/>

SELECT DISTINCT ?resource ?label ?error
WHERE {
VALUES ?type {owl:Class owl:ObjectProperty}
    ?resource a ?type .
    OPTIONAL {
        ?resource cco:definition ?englishDefinition .
        FILTER (langMatches(lang(?englishDefinition), "en"))
        }
    FILTER(!bound(?englishDefinition))
    FILTER(!isBlank(?resource))
    BIND (concat("WARNING: Missing definition for ", str(?resource)) AS ?error)
}
ORDER BY ?resource
```

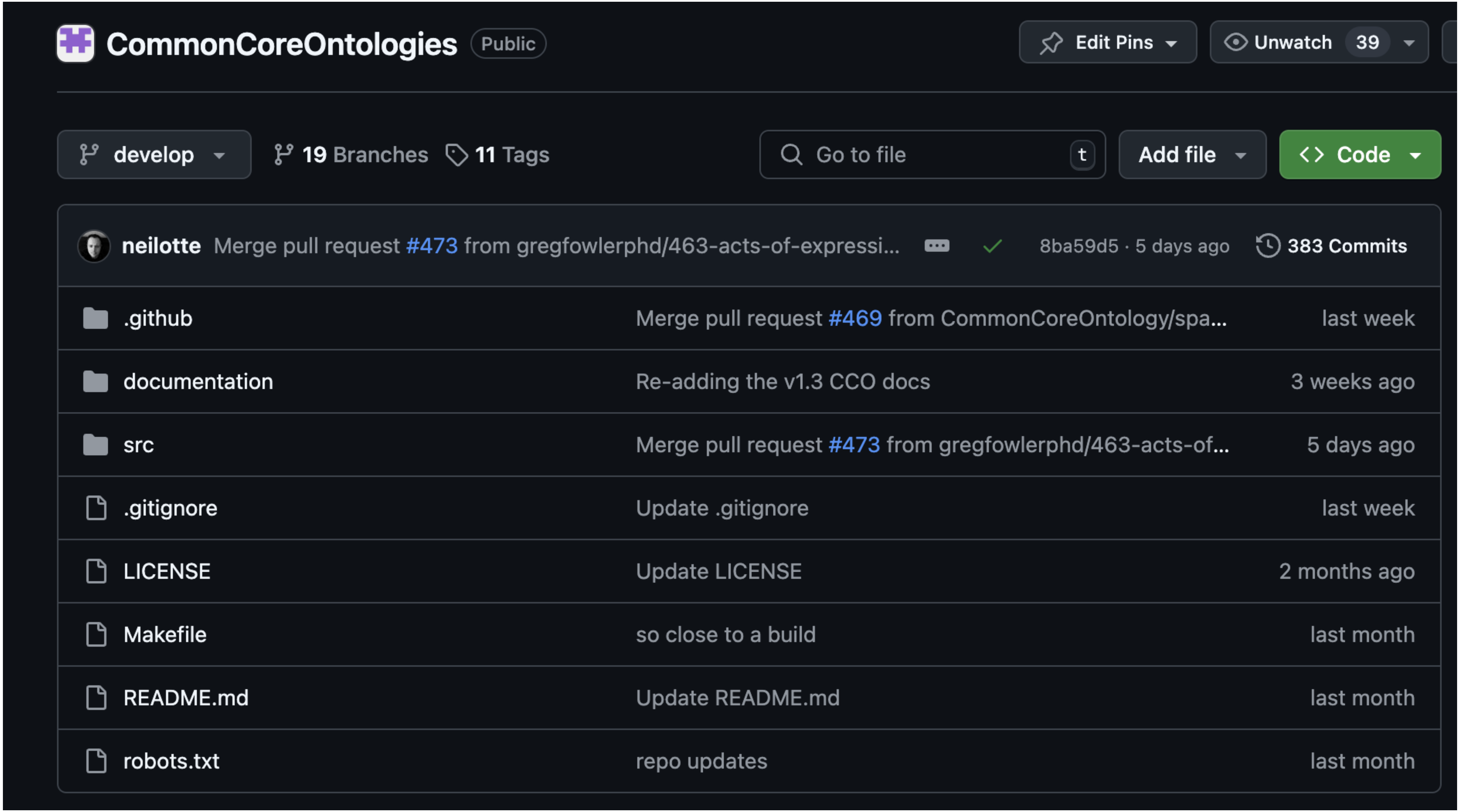
Format results with a description of the error

Dbpedia Challenge: https://dbpedia.org/sparql

1. What are the ten most populous cities in China?
2. What are the birthdates of every U.S. president who served in the 20th century?
3. What three cities in the U.S. have the highest percentage of foreign-born residents?
4. Which paintings by Vincent Van Gogh are currently located in museums in the U.S.?
5. How many rdfs labels include the string “lio”?

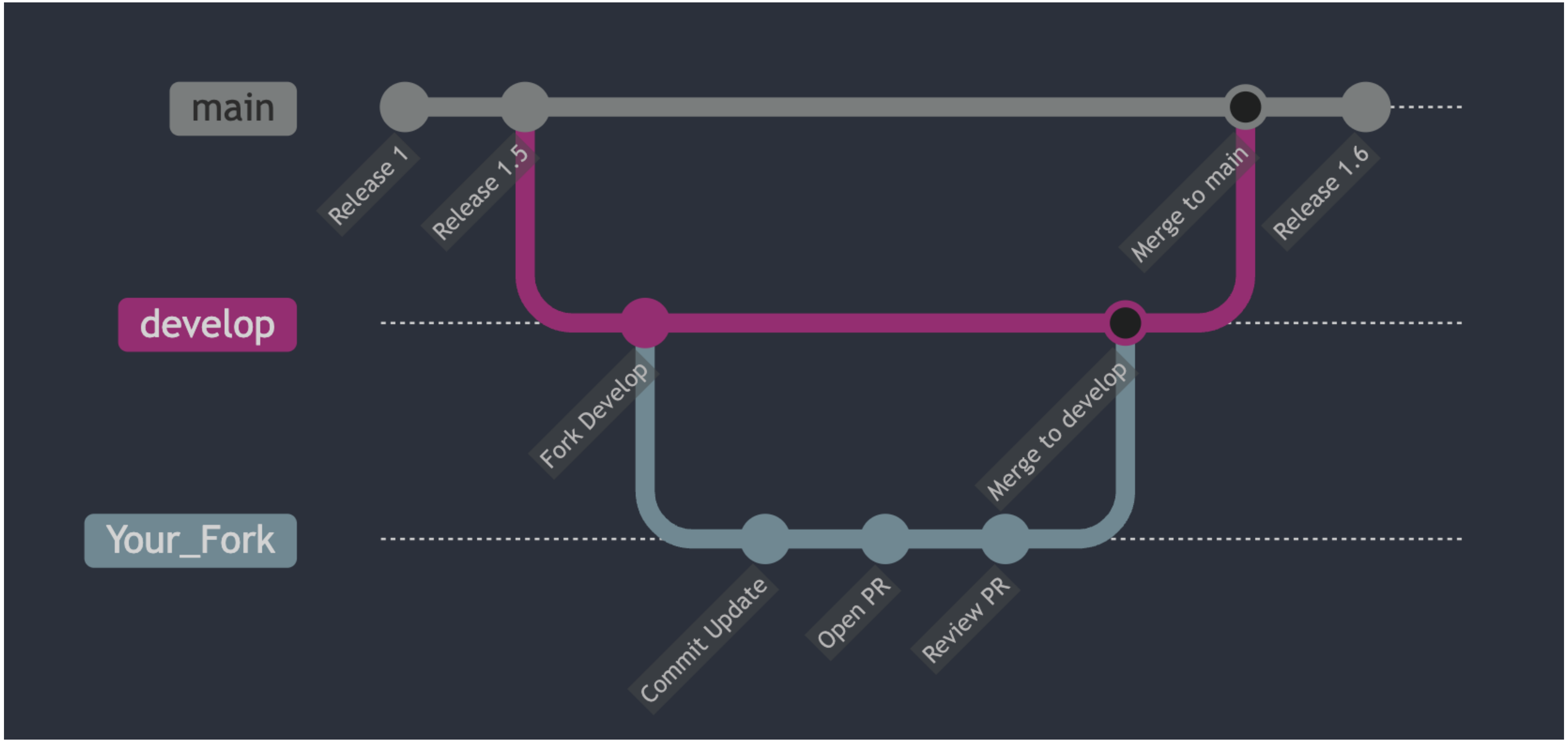
Outline

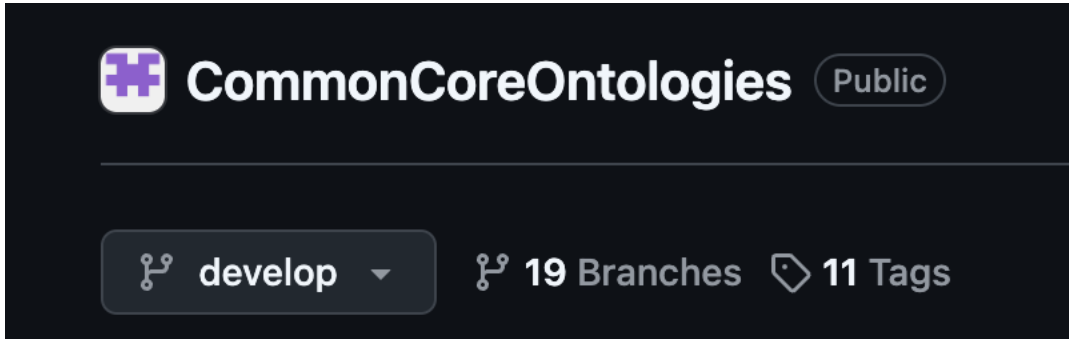
- Murder Mystery Revisited
- SPARQL & DBPedia
- CCO Quality Control SPARQL Checks



neilotte Merge pull request #473 from gregfowlerphd/463-acts-of-expressi... 8ba59d5 · 5 days ago 383 Commits

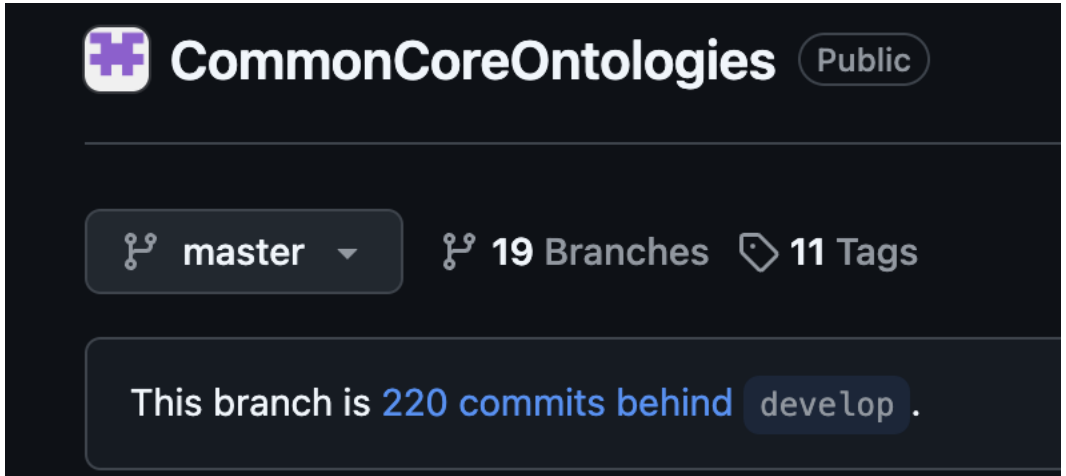
.github	Merge pull request #469 from CommonCoreOntology/spa...	last week
documentation	Re-adding the v1.3 CCO docs	3 weeks ago
src	Merge pull request #473 from gregfowlerphd/463-acts-of...	5 days ago
.gitignore	Update .gitignore	last week
LICENSE	Update LICENSE	2 months ago
Makefile	so close to a build	last month
README.md	Update README.md	last month
robots.txt	repo updates	last month





CommonCoreOntologies Public

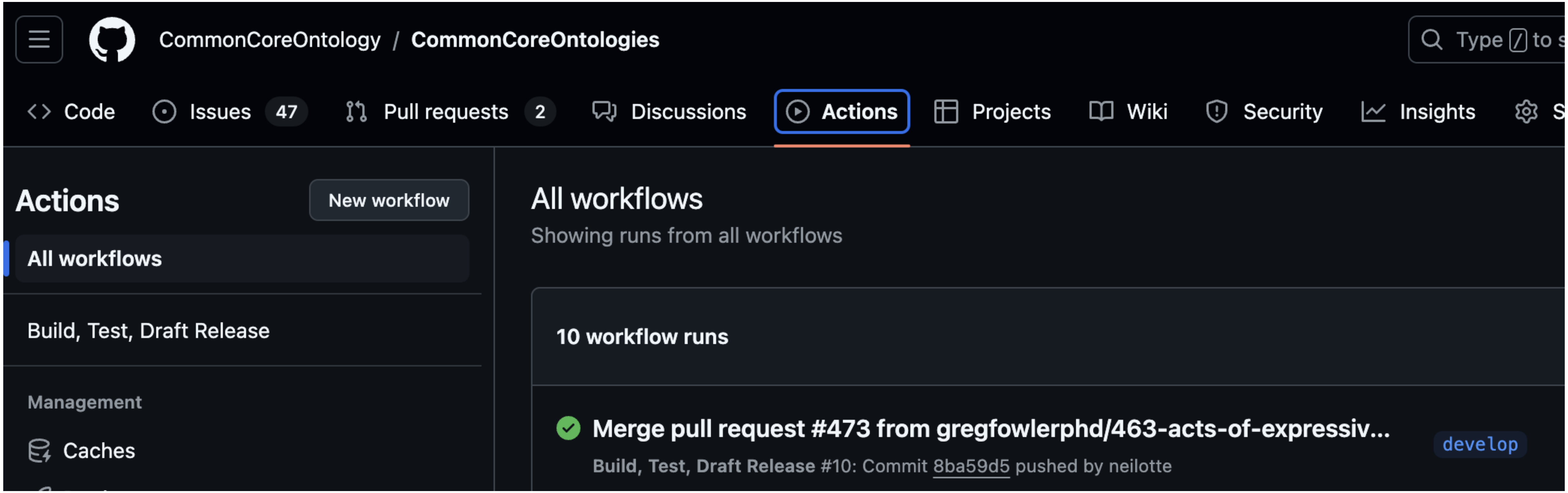
develop 19 Branches 11 Tags



CommonCoreOntologies Public

master 19 Branches 11 Tags

This branch is 220 commits behind `develop`.



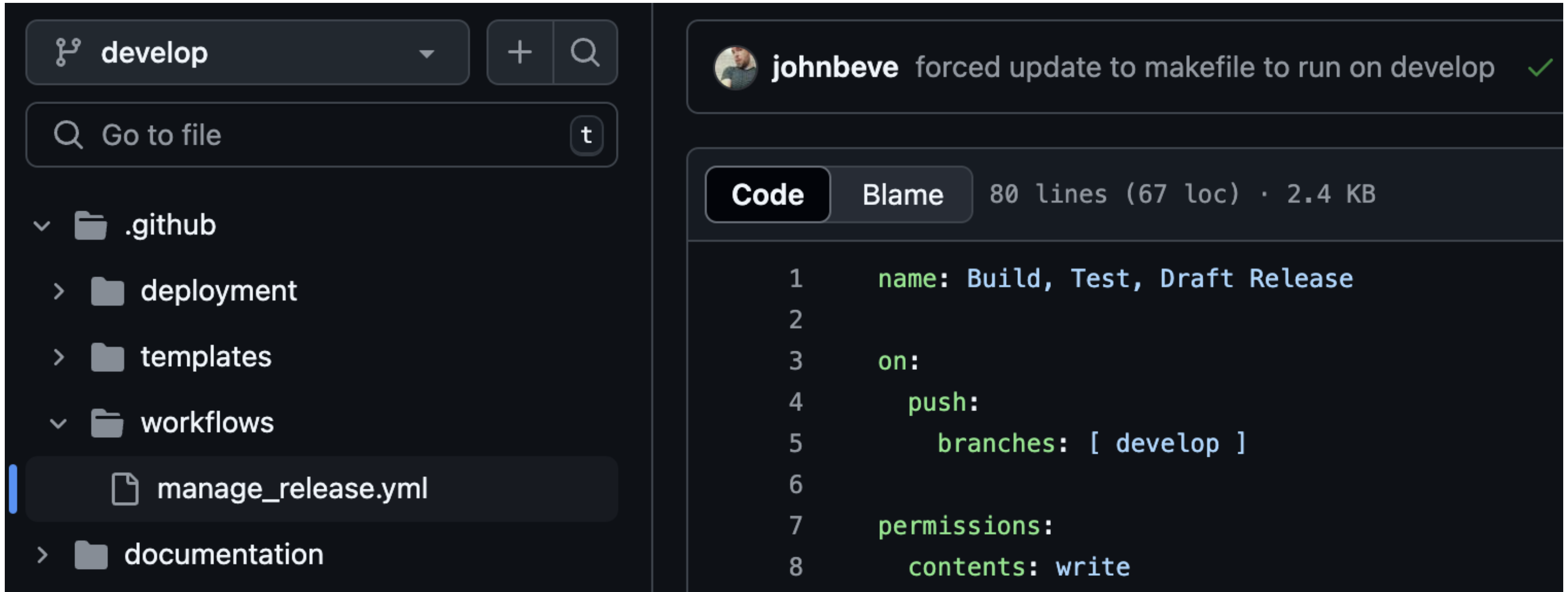
All workflows

Showing runs from all workflows

10 workflow runs

✓ Merge pull request #473 from gregfowlerphd/463-acts-of-expressiv... develop

Build, Test, Draft Release #10: Commit 8ba59d5 pushed by neilotte



johnbeve forced update to makefile to run on develop ✓

80 lines (67 loc) · 2.4 KB

```
name: Build, Test, Draft Release

on:
  push:
    branches: [ develop ]

permissions:
  contents: write
```

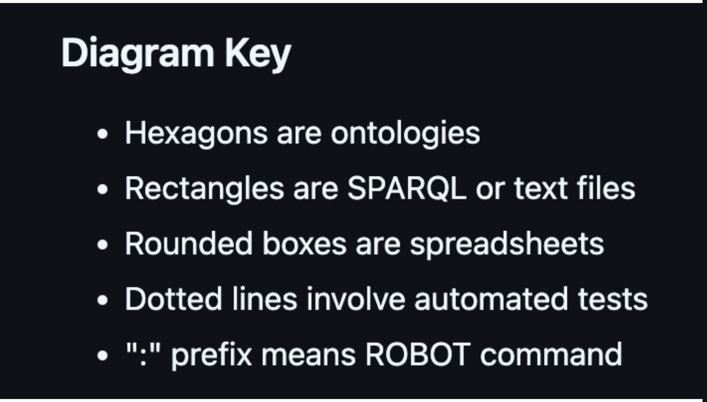
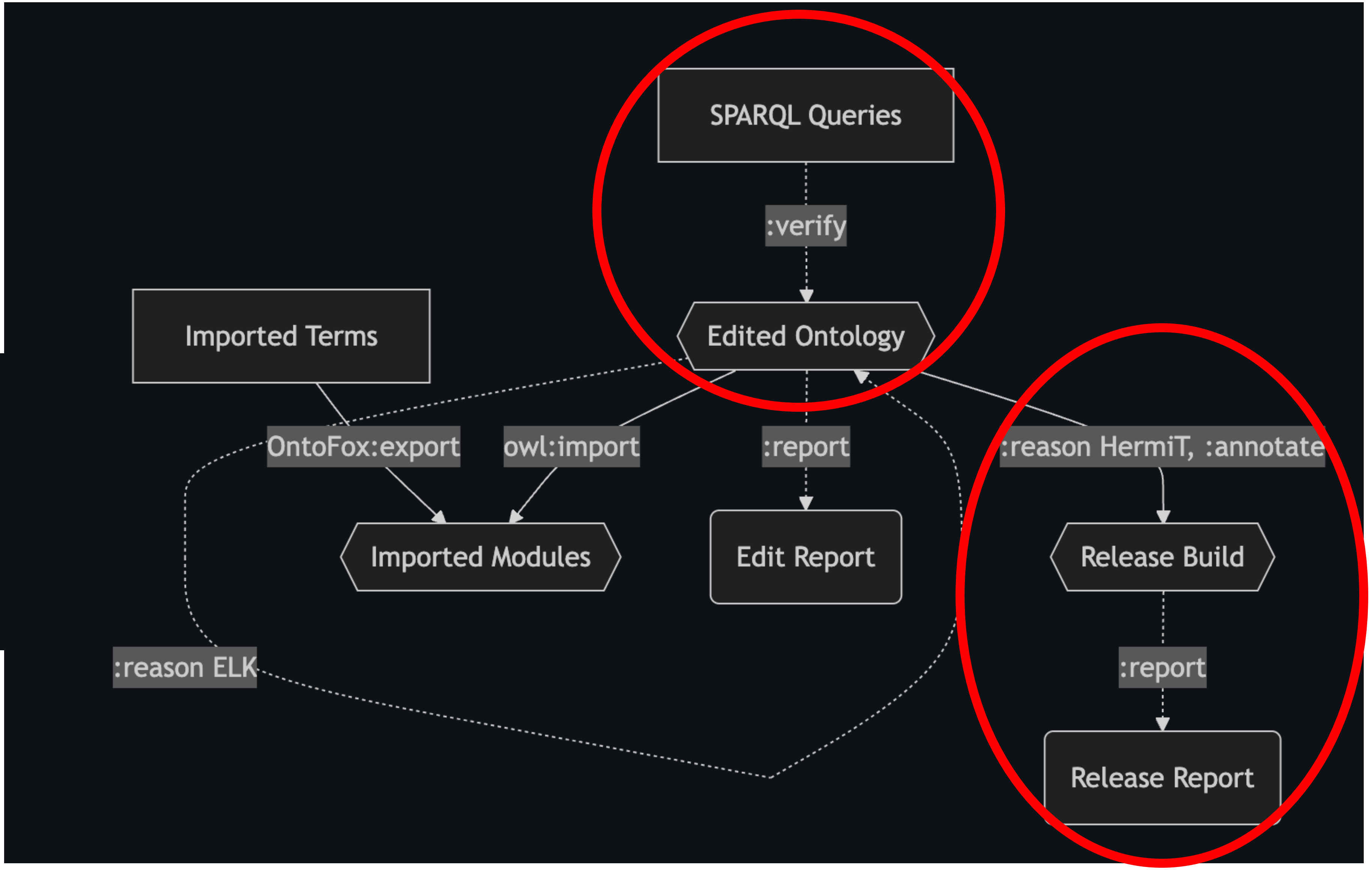
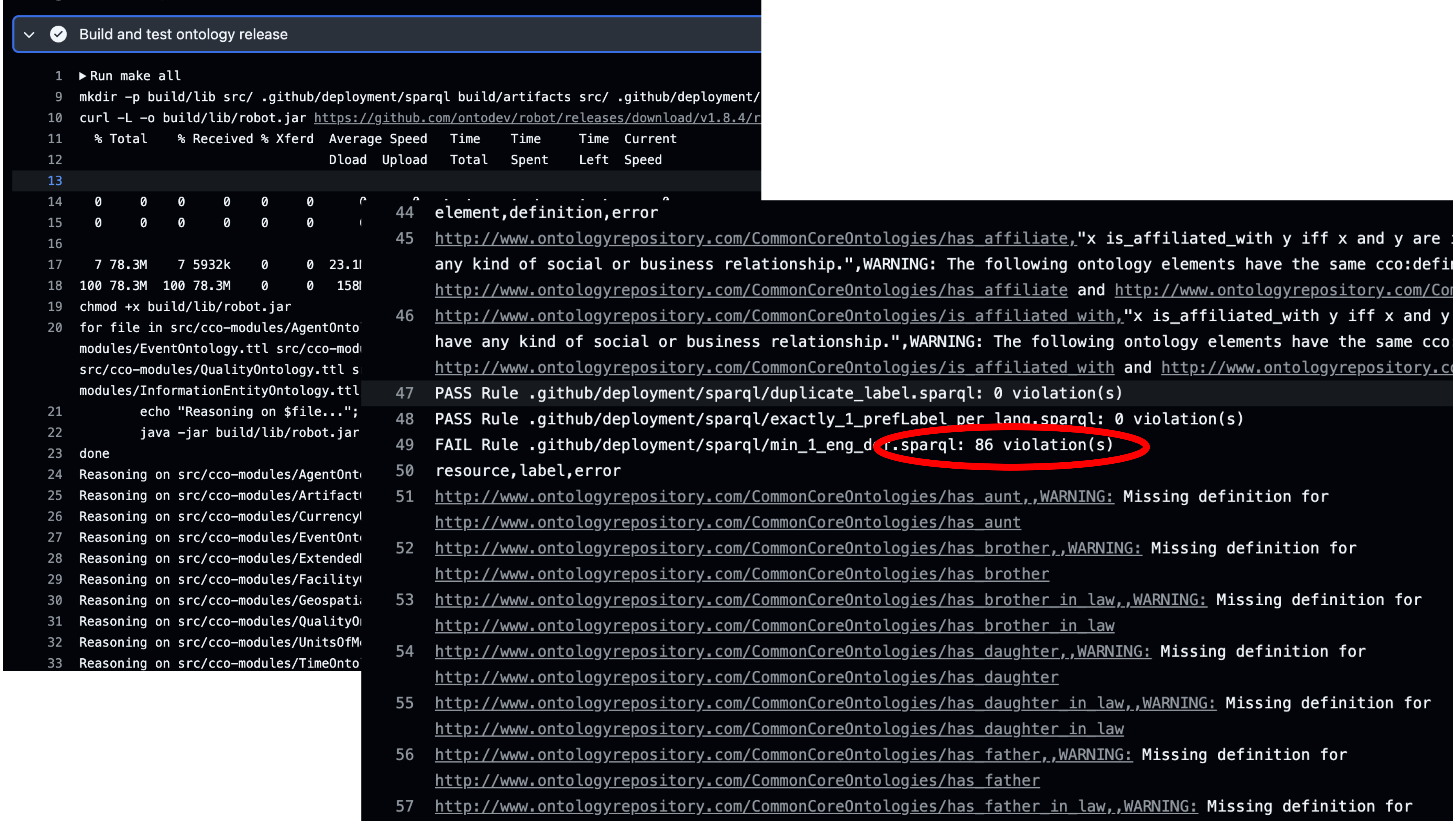


Diagram Key

- Hexagons are ontologies
- Rectangles are SPARQL or text files
- Rounded boxes are spreadsheets
- Dotted lines involve automated tests
- ":" prefix means ROBOT command



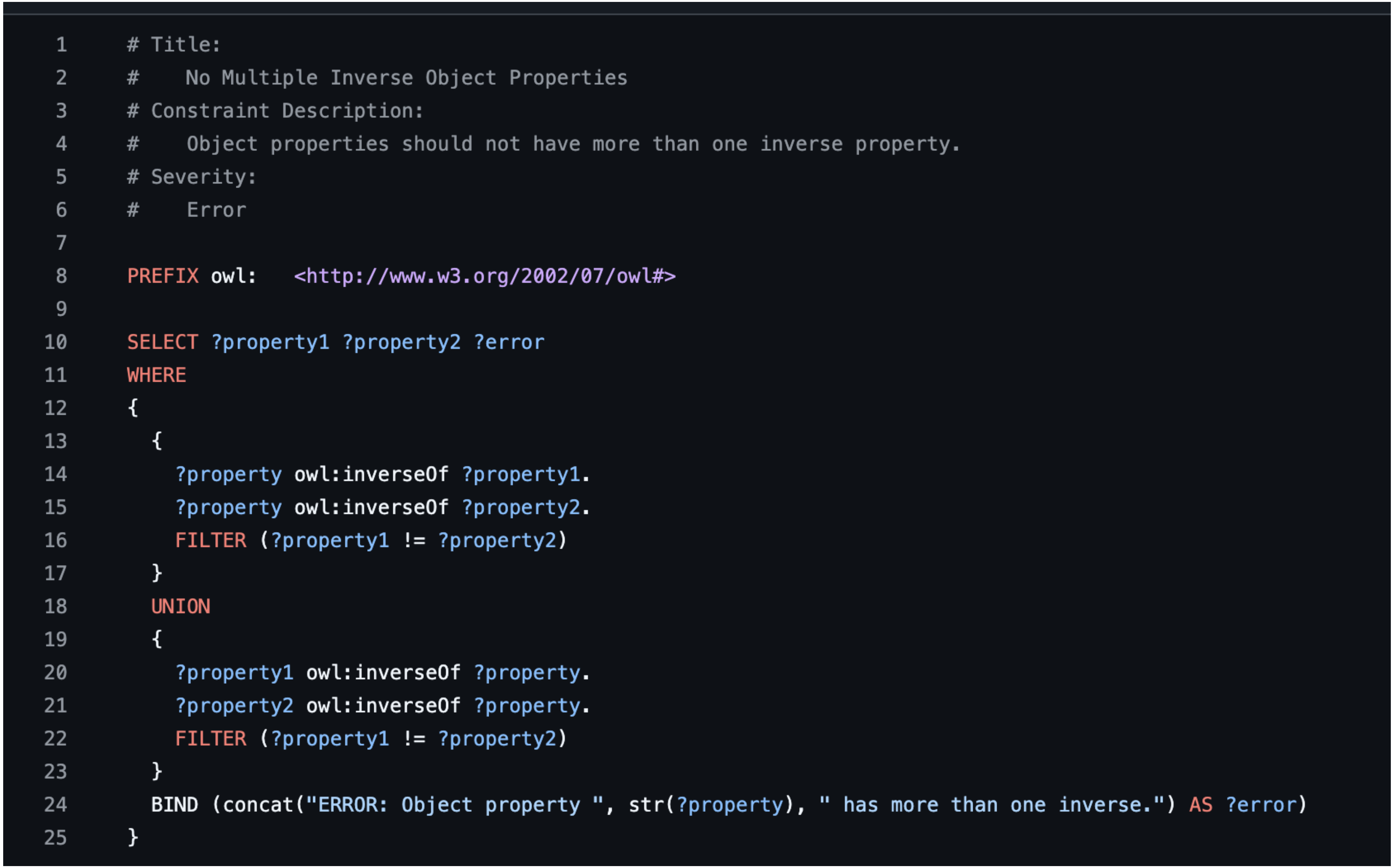
https://github.com/tmprd/ontology-pipeline/blob/master/docs/Architecture.md



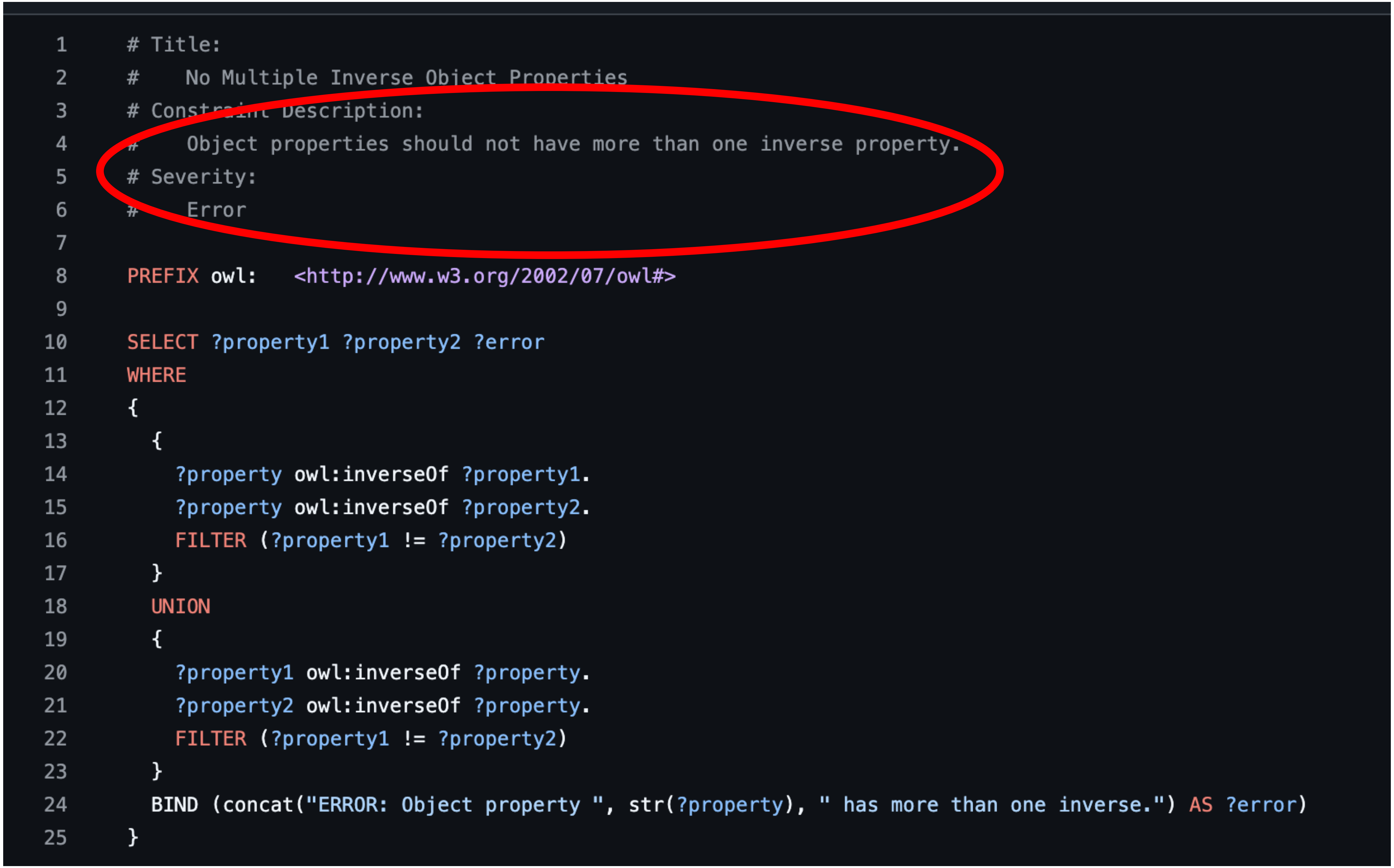
Build and test ontology release

```
▶ Run make all
mkdir -p build/lib src/ .github/deployment/sparql build/artifacts src/ .github/deployment/
curl -L -o build/lib/robot.jar https://github.com/ontodev/robot/releases/download/v1.8.4/r
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed

  0     0    0     0    0     0      (
  0     0    0     0    0     0      (

  7 78.3M    7 5932k    0     0  23.1M
100 78.3M  100 78.3M    0     0   158M
chmod +x build/lib/robot.jar
for file in src/cco-modules/AgentOnto
modules/EventOntology.ttl src/cco-modu
src/cco-modules/QualityOntology.ttl s
modules/InformationEntityOntology.ttl
        echo "Reasoning on $file...";
        java -jar build/lib/robot.jar
done
Reasoning on src/cco-modules/AgentOnt
Reasoning on src/cco-modules/Artifact
Reasoning on src/cco-modules/Currency
Reasoning on src/cco-modules/EventOnt
Reasoning on src/cco-modules/Extended
Reasoning on src/cco-modules/Facility
Reasoning on src/cco-modules/Geospati
Reasoning on src/cco-modules/QualityO
Reasoning on src/cco-modules/UnitsOfM
Reasoning on src/cco-modules/TimeOnto
```

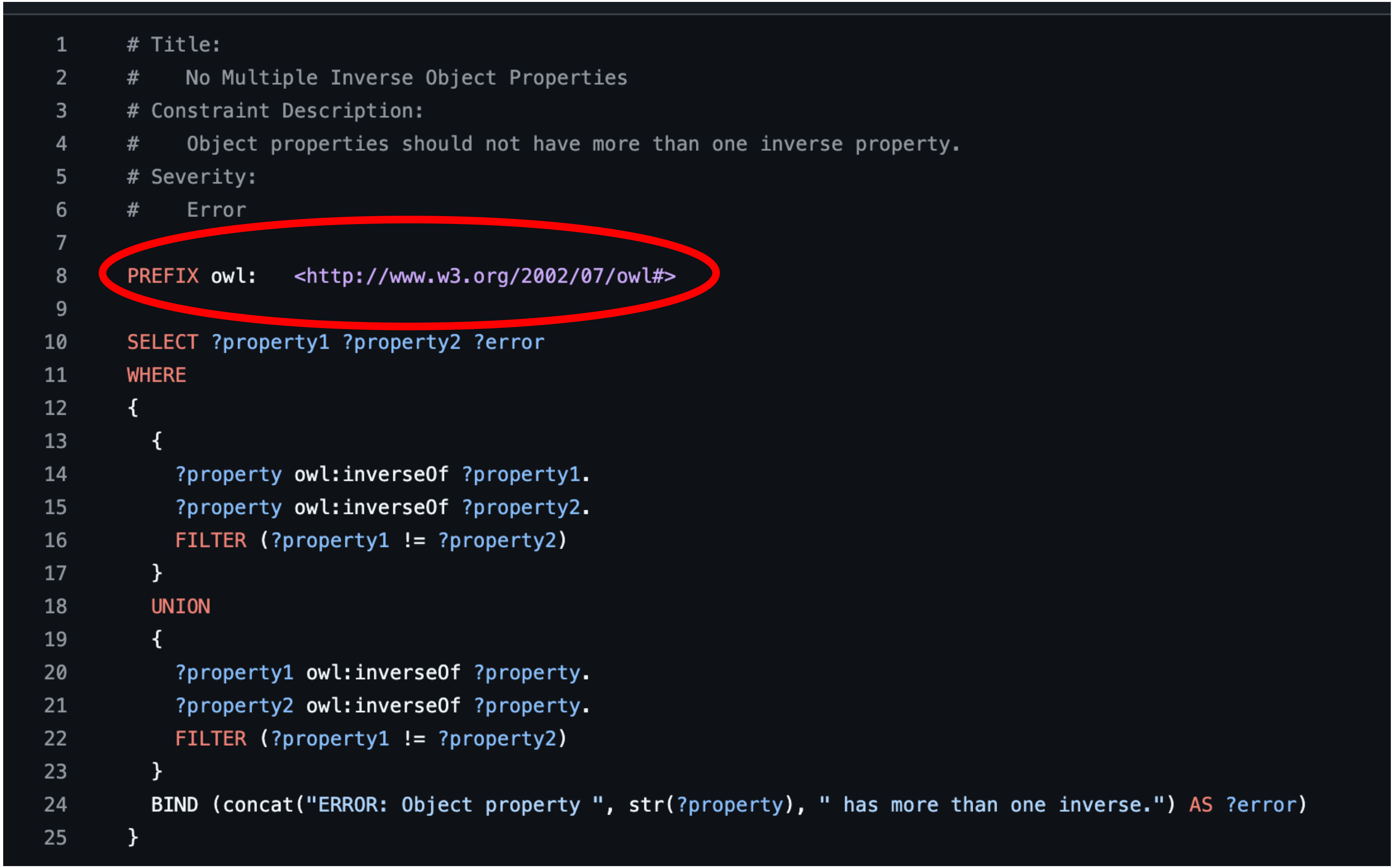
```
element,definition,error
http://www.ontologyrepository.com/CommonCoreOntologies/has_affiliate,"x is_affiliated_with y iff x and y are
any kind of social or business relationship.",WARNING: The following ontology elements have the same cco:defi
http://www.ontologyrepository.com/CommonCoreOntologies/has_affiliate and http://www.ontologyrepository.com/Co
http://www.ontologyrepository.com/CommonCoreOntologies/is_affiliated_with,"x is_affiliated_with y iff x and y
have any kind of social or business relationship.",WARNING: The following ontology elements have the same cco
http://www.ontologyrepository.com/CommonCoreOntologies/is_affiliated_with and http://www.ontologyrepository.c
PASS Rule .github/deployment/sparql/duplicate_label.sparql: 0 violation(s)
PASS Rule .github/deployment/sparql/exactly_1_prefLabel_per_lang.sparql: 0 violation(s)
FAIL Rule .github/deployment/sparql/min_1_eng_def.sparql: 86 violation(s)
resource,label,error
http://www.ontologyrepository.com/CommonCoreOntologies/has_aunt,,WARNING: Missing definition for
http://www.ontologyrepository.com/CommonCoreOntologies/has_aunt
http://www.ontologyrepository.com/CommonCoreOntologies/has_brother,,WARNING: Missing definition for
http://www.ontologyrepository.com/CommonCoreOntologies/has_brother
http://www.ontologyrepository.com/CommonCoreOntologies/has_brother_in_law,,WARNING: Missing definition for
http://www.ontologyrepository.com/CommonCoreOntologies/has_brother_in_law
http://www.ontologyrepository.com/CommonCoreOntologies/has_daughter,,WARNING: Missing definition for
http://www.ontologyrepository.com/CommonCoreOntologies/has_daughter
http://www.ontologyrepository.com/CommonCoreOntologies/has_daughter_in_law,,WARNING: Missing definition for
http://www.ontologyrepository.com/CommonCoreOntologies/has_daughter_in_law
http://www.ontologyrepository.com/CommonCoreOntologies/has_father,,WARNING: Missing definition for
http://www.ontologyrepository.com/CommonCoreOntologies/has_father
http://www.ontologyrepository.com/CommonCoreOntologies/has_father_in_law,,WARNING: Missing definition for
```



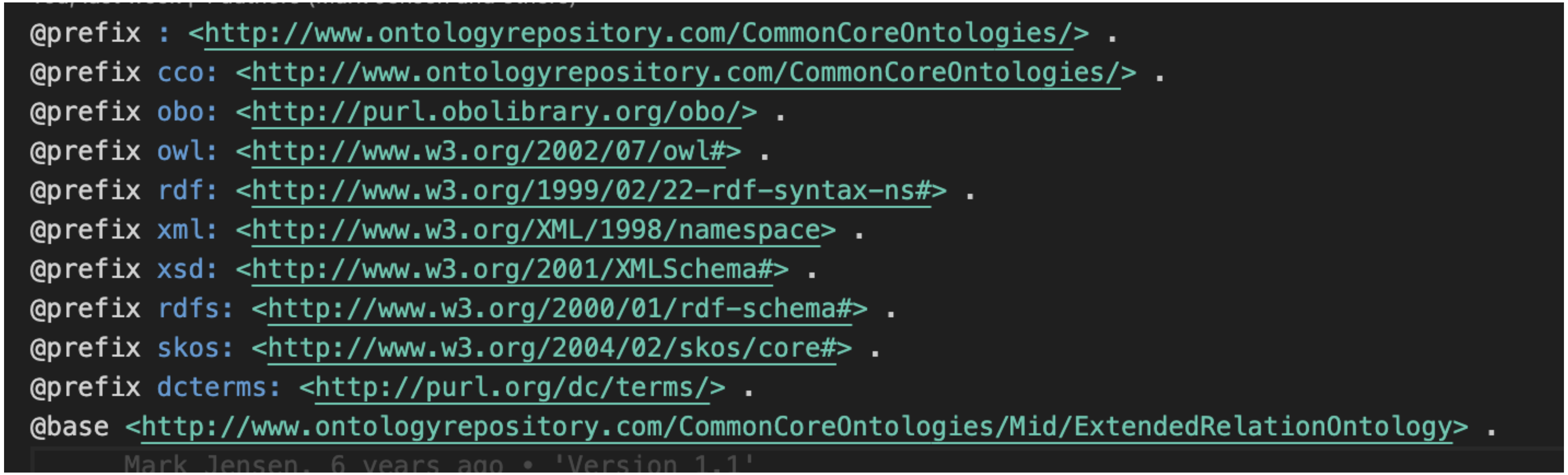
```
# Title:
#    No Multiple Inverse Object Properties
# Constraint Description:
#    Object properties should not have more than one inverse property.
# Severity:
#    Error

PREFIX owl:   <http://www.w3.org/2002/07/owl#>

SELECT ?property1 ?property2 ?error
WHERE
{
  {
    ?property owl:inverseOf ?property1.
    ?property owl:inverseOf ?property2.
    FILTER (?property1 != ?property2)
  }
  UNION
  {
    ?property1 owl:inverseOf ?property.
    ?property2 owl:inverseOf ?property.
    FILTER (?property1 != ?property2)
  }
  BIND (concat("ERROR: Object property ", str(?property), " has more than one inverse.") AS ?error)
}
```



```
# Title:
#    No Multiple Inverse Object Properties
# Constraint Description:
#    Object properties should not have more than one inverse property.
# Severity:
#    Error

PREFIX owl:   <http://www.w3.org/2002/07/owl#>

SELECT ?property1 ?property2 ?error
WHERE
{
  {
    ?property owl:inverseOf ?property1.
    ?property owl:inverseOf ?property2.
    FILTER (?property1 != ?property2)
  }
  UNION
  {
    ?property1 owl:inverseOf ?property.
    ?property2 owl:inverseOf ?property.
    FILTER (?property1 != ?property2)
  }
  BIND (concat("ERROR: Object property ", str(?property), " has more than one inverse.") AS ?error)
}
```

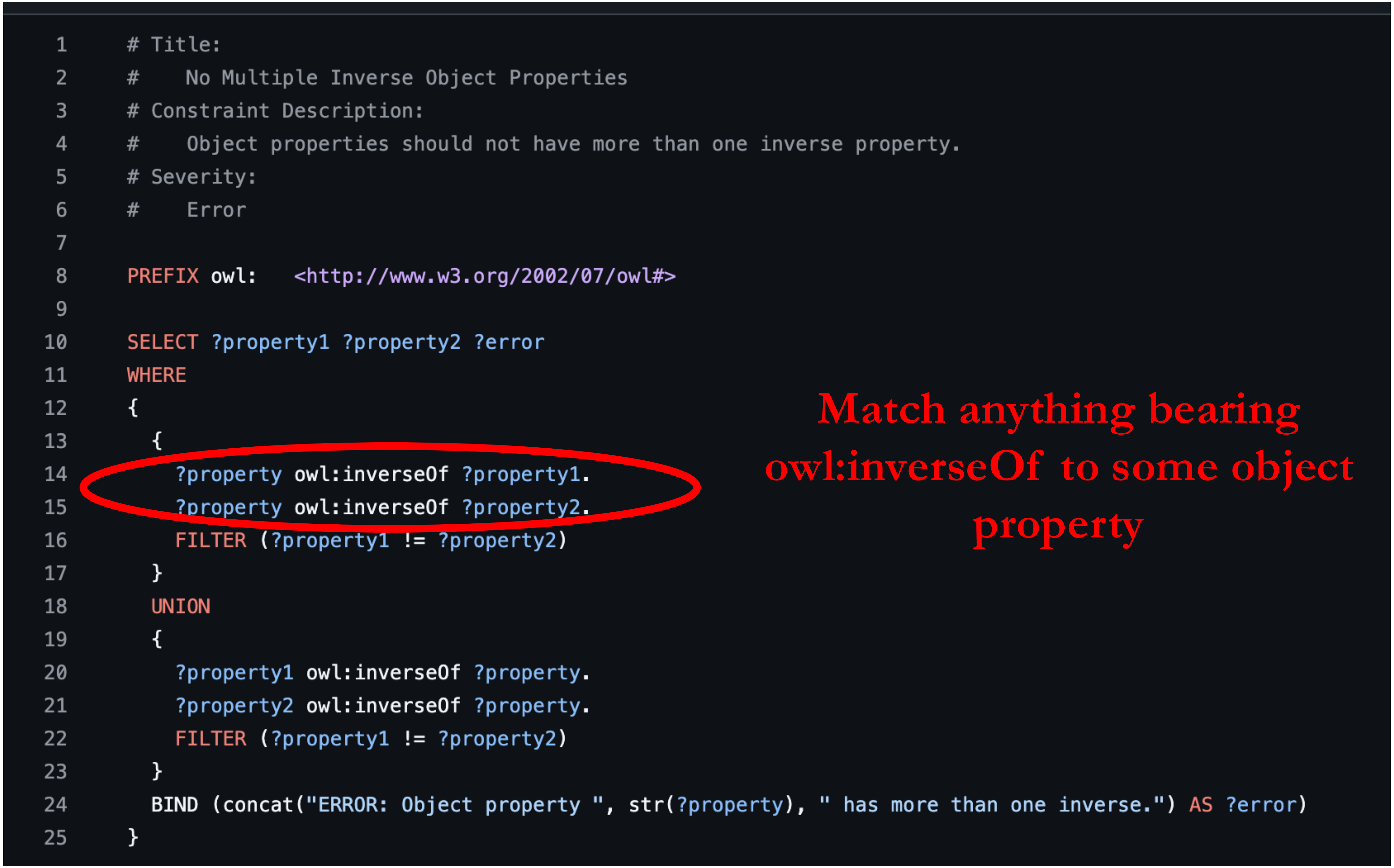


```
# Title:
#    No Multiple Inverse Object Properties
# Constraint Description:
#    Object properties should not have more than one inverse property.
# Severity:
#    Error

PREFIX owl:   <http://www.w3.org/2002/07/owl#>

SELECT ?property1 ?property2 ?error
WHERE
{
  {
    ?property owl:inverseOf ?property1.
    ?property owl:inverseOf ?property2.
    FILTER (?property1 != ?property2)
  }
  UNION
  {
    ?property1 owl:inverseOf ?property.
    ?property2 owl:inverseOf ?property.
    FILTER (?property1 != ?property2)
  }
  BIND (concat("ERROR: Object property ", str(?property), " has more than one inverse.") AS ?error)
}
```



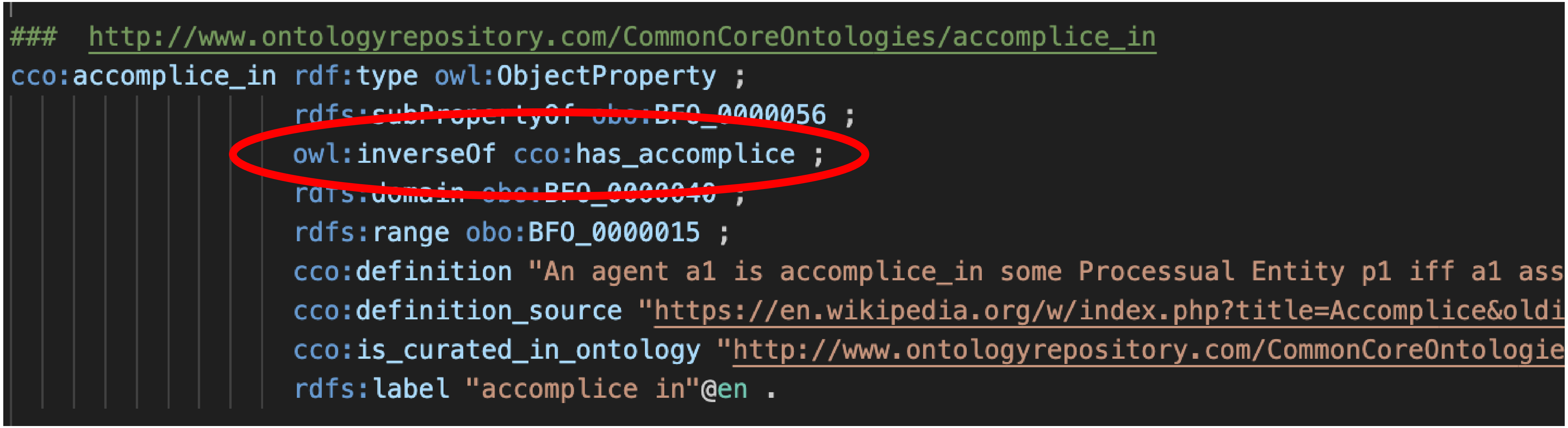
```
@prefix : <http://www.ontologyrepository.com/CommonCoreOntologies/> .
@prefix cco: <http://www.ontologyrepository.com/CommonCoreOntologies/> .
@prefix obo: <http://purl.obolibrary.org/obo/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@base <http://www.ontologyrepository.com/CommonCoreOntologies/Mid/ExtendedRelationOntology> .
      Mark Jensen, 6 years ago • 'Version 1.1'
```

List of prefixes used in the ontology file can be found at the top of the file



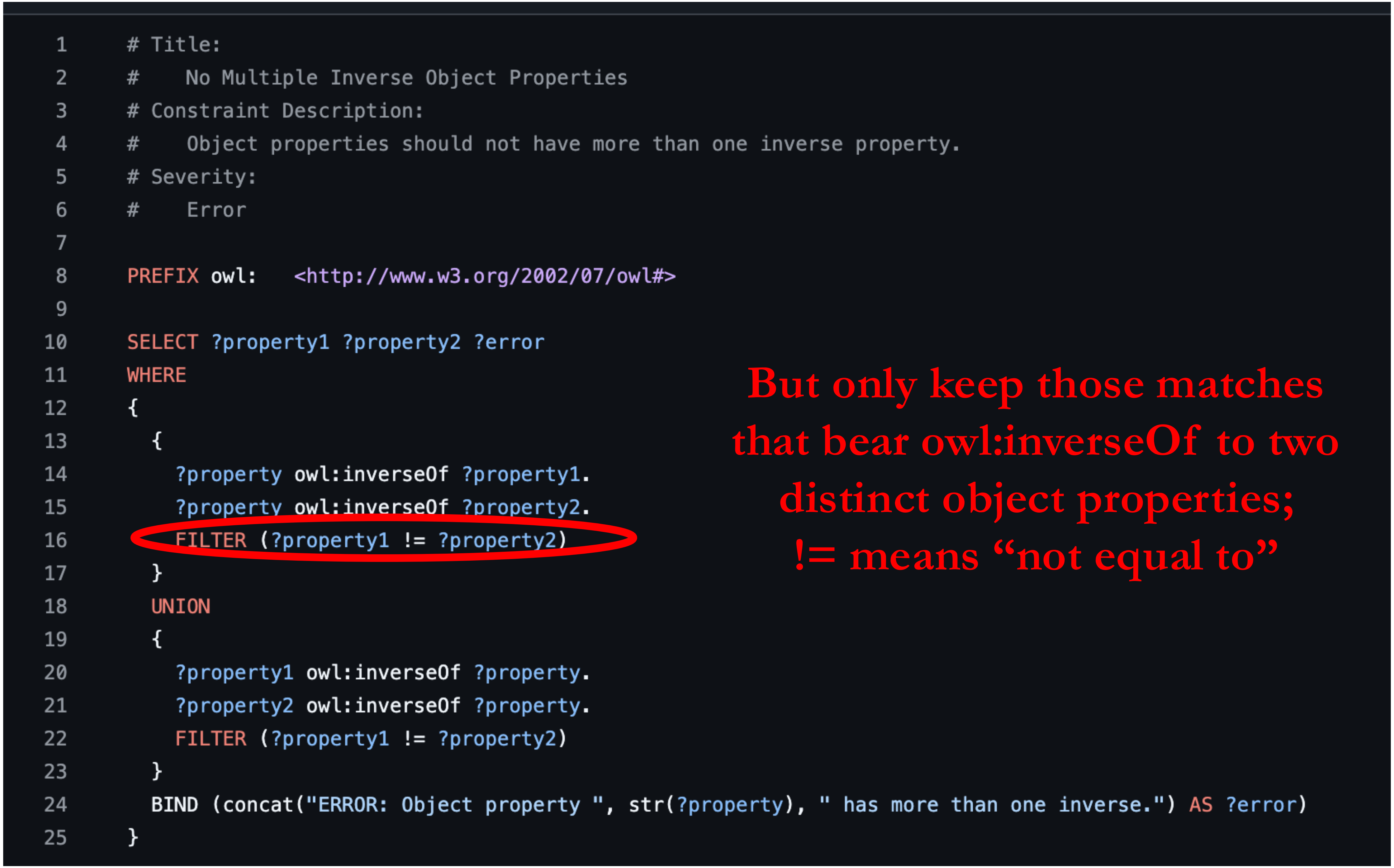
```
# Title:
#    No Multiple Inverse Object Properties
# Constraint Description:
#    Object properties should not have more than one inverse property.
# Severity:
#    Error

PREFIX owl:   <http://www.w3.org/2002/07/owl#>

SELECT ?property1 ?property2 ?error
WHERE
{
  {
    ?property owl:inverseOf ?property1.
    ?property owl:inverseOf ?property2.
    FILTER (?property1 != ?property2)
  }
  UNION
  {
    ?property1 owl:inverseOf ?property.
    ?property2 owl:inverseOf ?property.
    FILTER (?property1 != ?property2)
  }
  BIND (concat("ERROR: Object property ", str(?property), " has more than one inverse.") AS ?error)
}
```

Match anything bearing owl:inverseOf to some object property



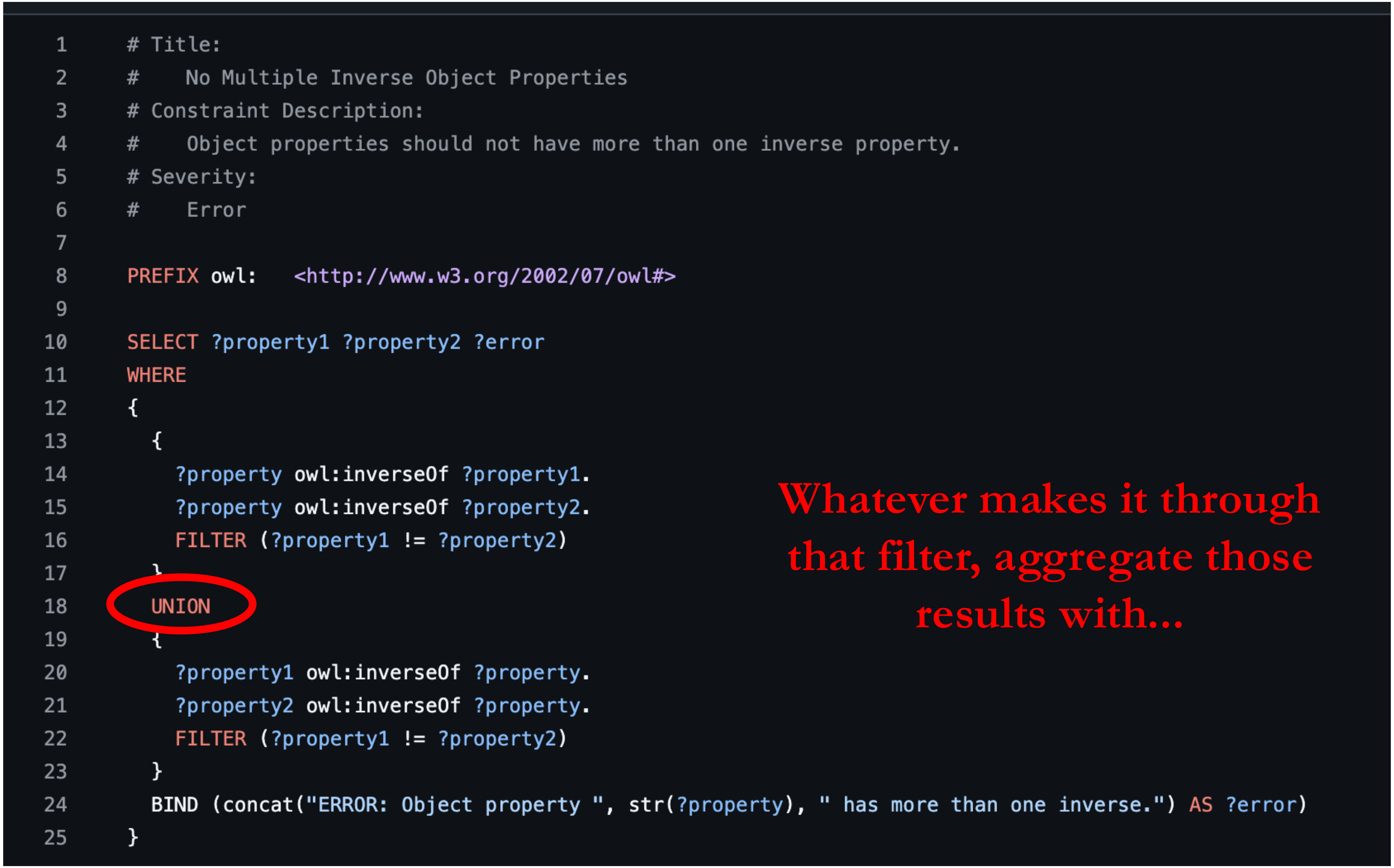
```
###  http://www.ontologyrepository.com/CommonCoreOntologies/accomplice_in
cco:accomplice_in rdf:type owl:ObjectProperty ;
                  rdfs:subPropertyOf obo:BFO_0000056 ;
                  owl:inverseOf cco:has_accomplice ;
                  rdfs:domain obo:BFO_0000040 ;
                  rdfs:range obo:BFO_0000015 ;
                  cco:definition "An agent a1 is accomplice_in some Processual Entity p1 iff a1 ass
                  cco:definition_source "https://en.wikipedia.org/w/index.php?title=Accomplice&oldi
                  cco:is_curated_in_ontology "http://www.ontologyrepository.com/CommonCoreOntologie
                  rdfs:label "accomplice in"@en .
```

For example, the object property cco:accomplice_in will match since it bears owl:inverseOf to cco:has_accomplice



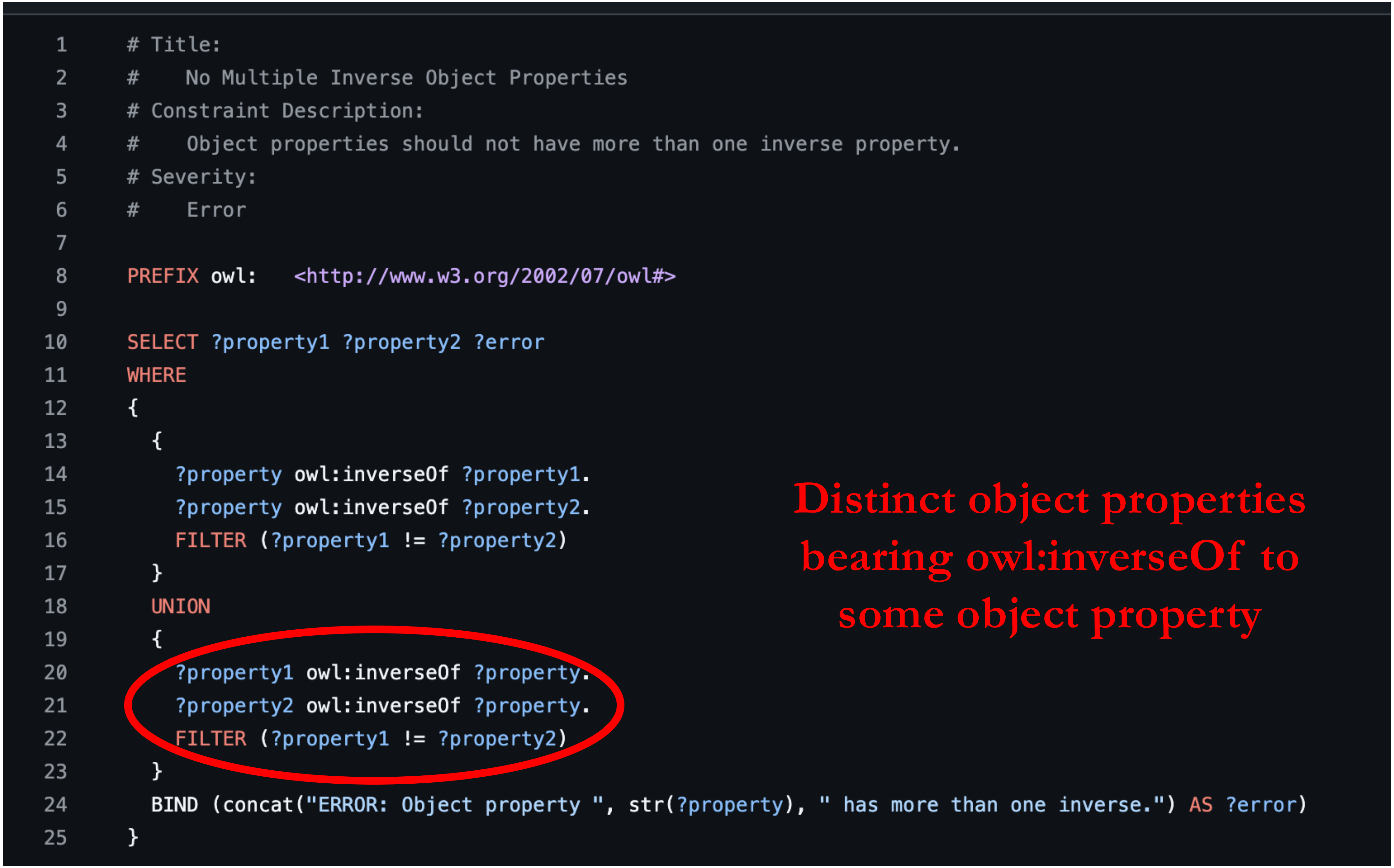
```
# Title:
#    No Multiple Inverse Object Properties
# Constraint Description:
#    Object properties should not have more than one inverse property.
# Severity:
#    Error

PREFIX owl:   <http://www.w3.org/2002/07/owl#>

SELECT ?property1 ?property2 ?error
WHERE
{
  {
    ?property owl:inverseOf ?property1.
    ?property owl:inverseOf ?property2.
    FILTER (?property1 != ?property2)
  }
  UNION
  {
    ?property1 owl:inverseOf ?property.
    ?property2 owl:inverseOf ?property.
    FILTER (?property1 != ?property2)
  }
  BIND (concat("ERROR: Object property ", str(?property), " has more than one inverse.") AS ?error)
}
```

But only keep those matches that bear owl:inverseOf to two distinct object properties; != means “not equal to”



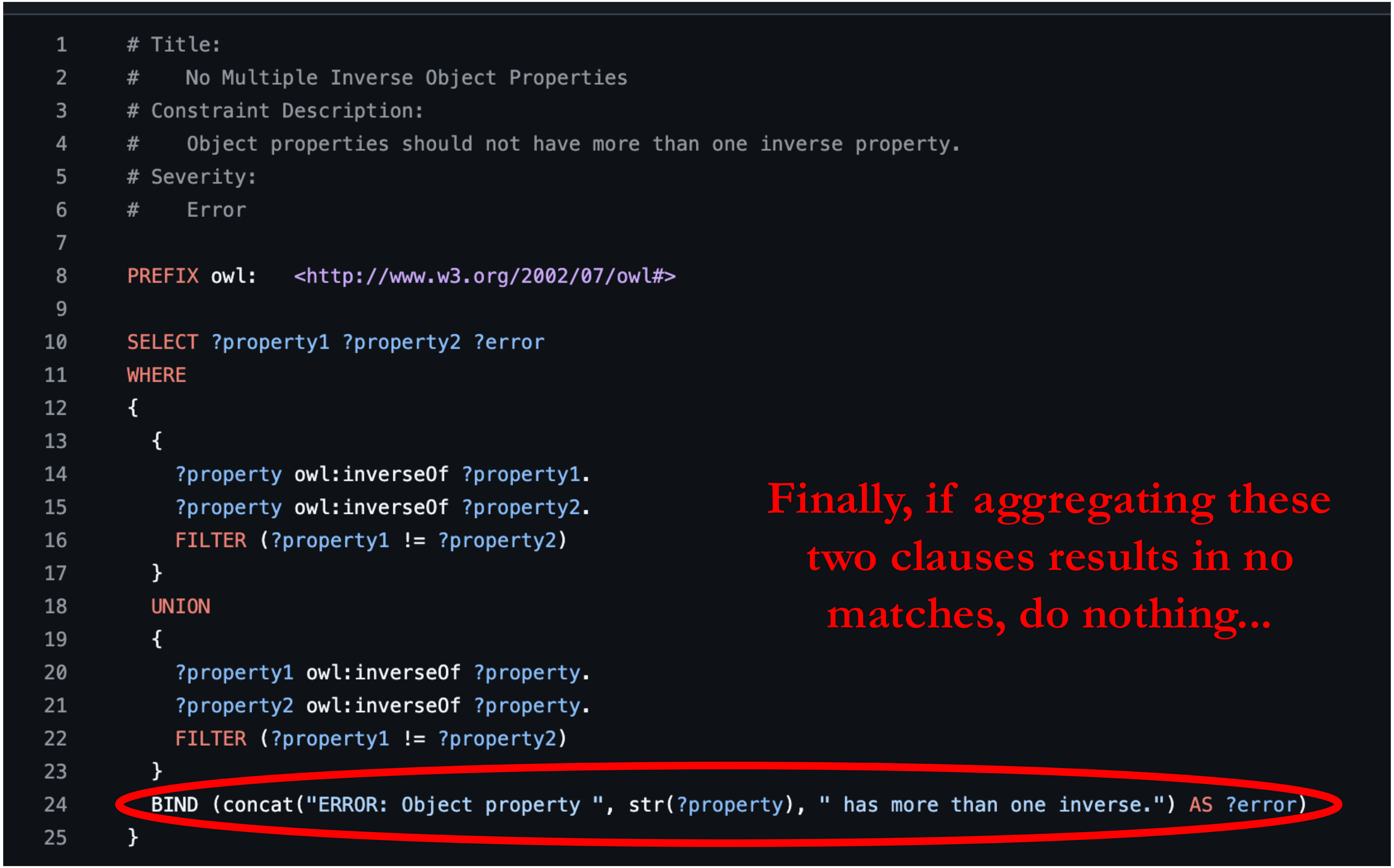
```
# Title:
#    No Multiple Inverse Object Properties
# Constraint Description:
#    Object properties should not have more than one inverse property.
# Severity:
#    Error

PREFIX owl:   <http://www.w3.org/2002/07/owl#>

SELECT ?property1 ?property2 ?error
WHERE
{
  {
    ?property owl:inverseOf ?property1.
    ?property owl:inverseOf ?property2.
    FILTER (?property1 != ?property2)
  }
  UNION
  {
    ?property1 owl:inverseOf ?property.
    ?property2 owl:inverseOf ?property.
    FILTER (?property1 != ?property2)
  }
  BIND (concat("ERROR: Object property ", str(?property), " has more than one inverse.") AS ?error)
}
```

Whatever makes it through that filter, aggregate those results with…



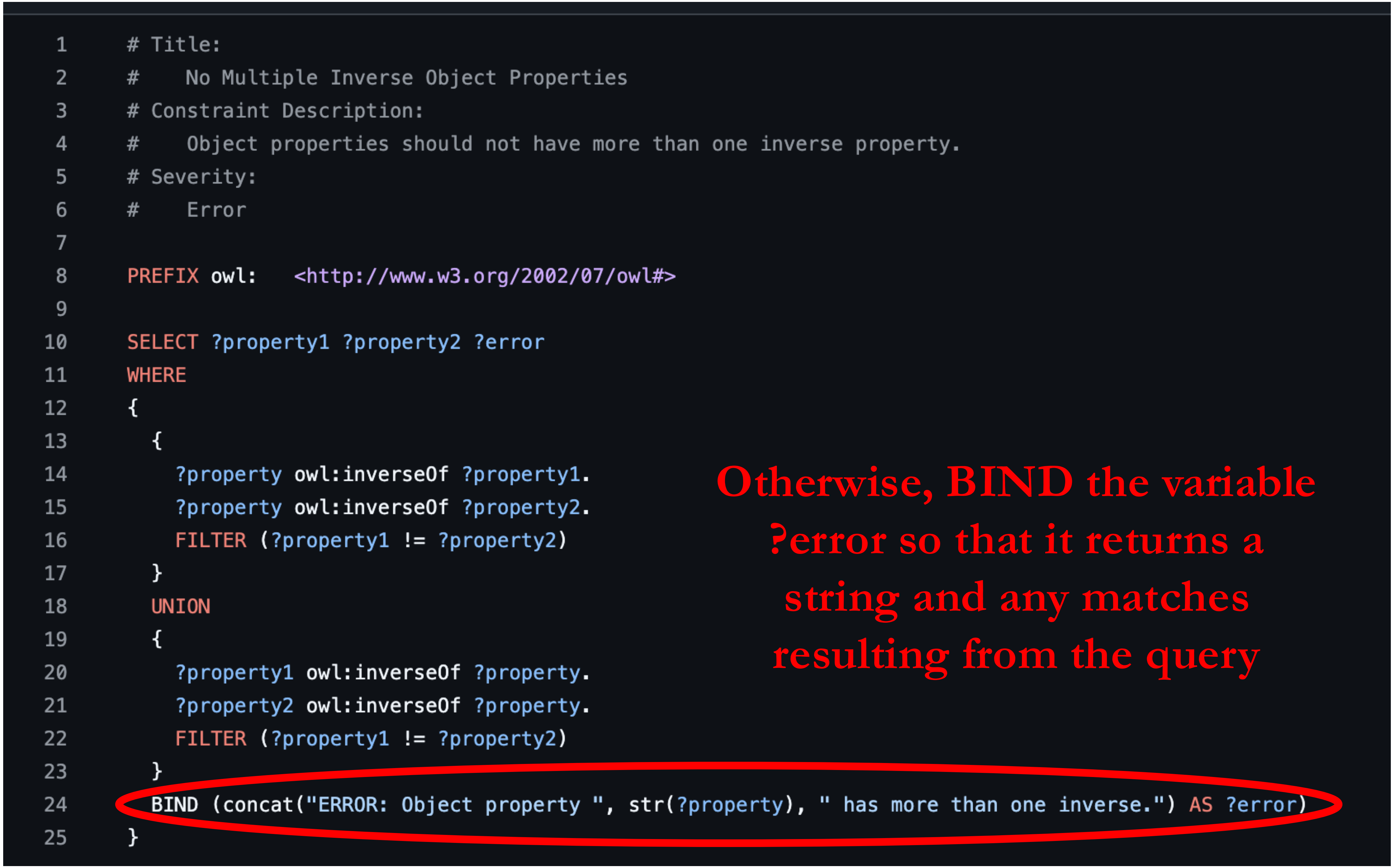
```sparql
# Title:
#    No Multiple Inverse Object Properties
# Constraint Description:
#    Object properties should not have more than one inverse property.
# Severity:
#    Error

PREFIX owl:   <http://www.w3.org/2002/07/owl#>

SELECT ?property1 ?property2 ?error
WHERE
{
  {
    ?property owl:inverseOf ?property1.
    ?property owl:inverseOf ?property2.
    FILTER (?property1 != ?property2)
  }
  UNION
  {
    ?property1 owl:inverseOf ?property.
    ?property2 owl:inverseOf ?property.
    FILTER (?property1 != ?property2)
  }
  BIND (concat("ERROR: Object property ", str(?property), " has more than one inverse.") AS ?error)
}
```

Distinct object properties bearing owl:inverseOf to some object property



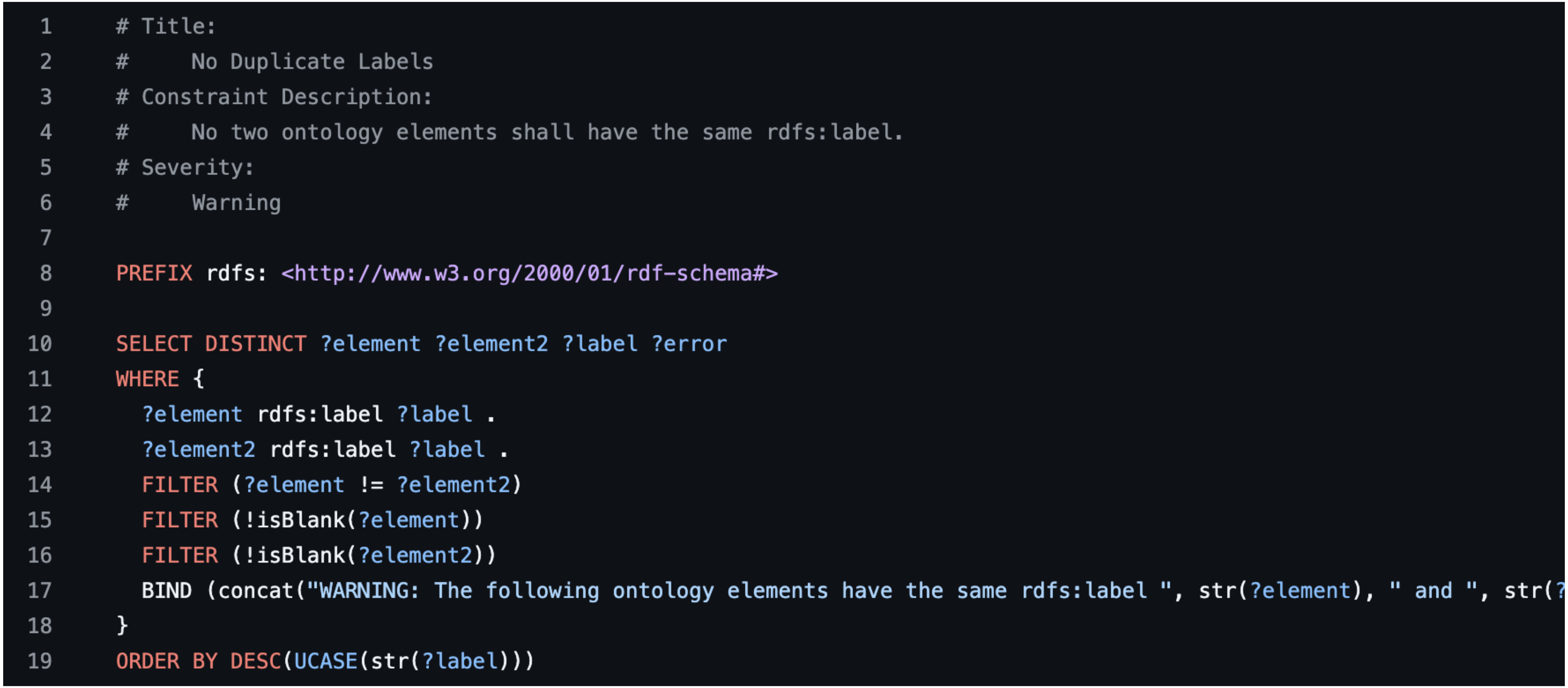
```
# Title:
#    No Multiple Inverse Object Properties
# Constraint Description:
#    Object properties should not have more than one inverse property.
# Severity:
#    Error

PREFIX owl:   <http://www.w3.org/2002/07/owl#>

SELECT ?property1 ?property2 ?error
WHERE
{
  {
    ?property owl:inverseOf ?property1.
    ?property owl:inverseOf ?property2.
    FILTER (?property1 != ?property2)
  }
  UNION
  {
    ?property1 owl:inverseOf ?property.
    ?property2 owl:inverseOf ?property.
    FILTER (?property1 != ?property2)
  }
  BIND (concat("ERROR: Object property ", str(?property), " has more than one inverse.") AS ?error)
}
```

Finally, if aggregating these two clauses results in no matches, do nothing...

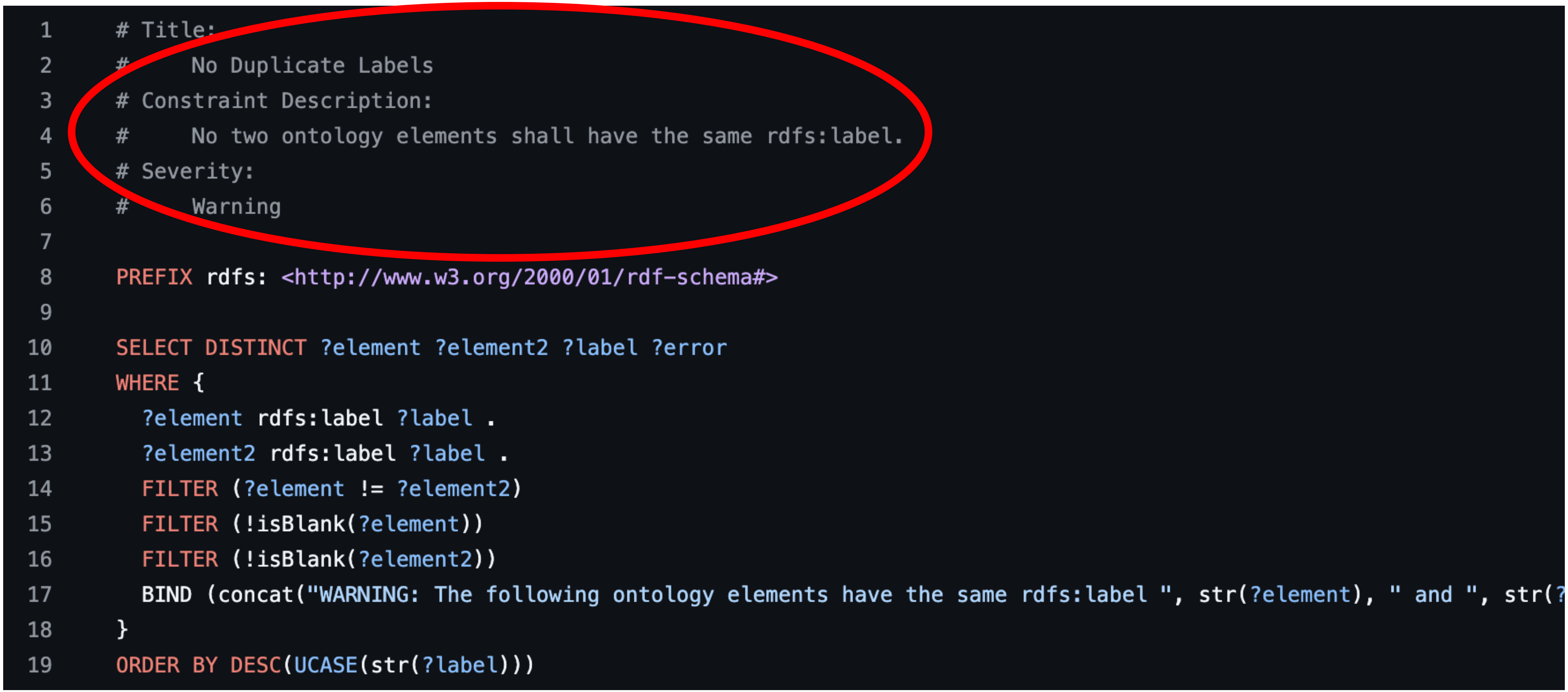


```
# Title:
#    No Multiple Inverse Object Properties
# Constraint Description:
#    Object properties should not have more than one inverse property.
# Severity:
#    Error

PREFIX owl:   <http://www.w3.org/2002/07/owl#>

SELECT ?property1 ?property2 ?error
WHERE
{
  {
    ?property owl:inverseOf ?property1.
    ?property owl:inverseOf ?property2.
    FILTER (?property1 != ?property2)
  }
  UNION
  {
    ?property1 owl:inverseOf ?property.
    ?property2 owl:inverseOf ?property.
    FILTER (?property1 != ?property2)
  }
  BIND (concat("ERROR: Object property ", str(?property), " has more than one inverse.") AS ?error)
}
```

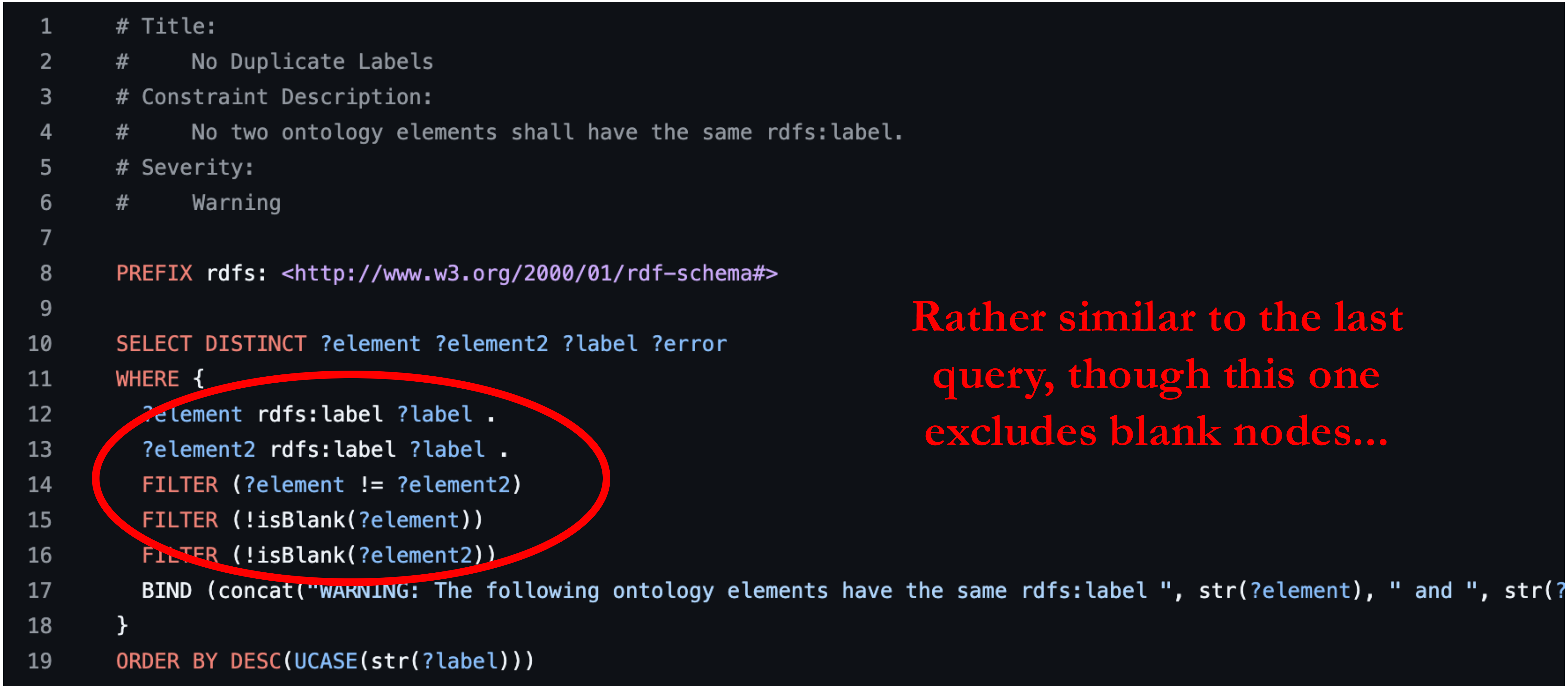
Otherwise, BIND the variable ?error so that it returns a string and any matches resulting from the query



```
# Title:
#     No Duplicate Labels
# Constraint Description:
#     No two ontology elements shall have the same rdfs:label.
# Severity:
#     Warning

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT DISTINCT ?element ?element2 ?label ?error
WHERE {
  ?element rdfs:label ?label .
  ?element2 rdfs:label ?label .
  FILTER (?element != ?element2)
  FILTER (!isBlank(?element))
  FILTER (!isBlank(?element2))
  BIND (concat("WARNING: The following ontology elements have the same rdfs:label ", str(?element), " and ", str(?
}
ORDER BY DESC(UCASE(str(?label)))
```

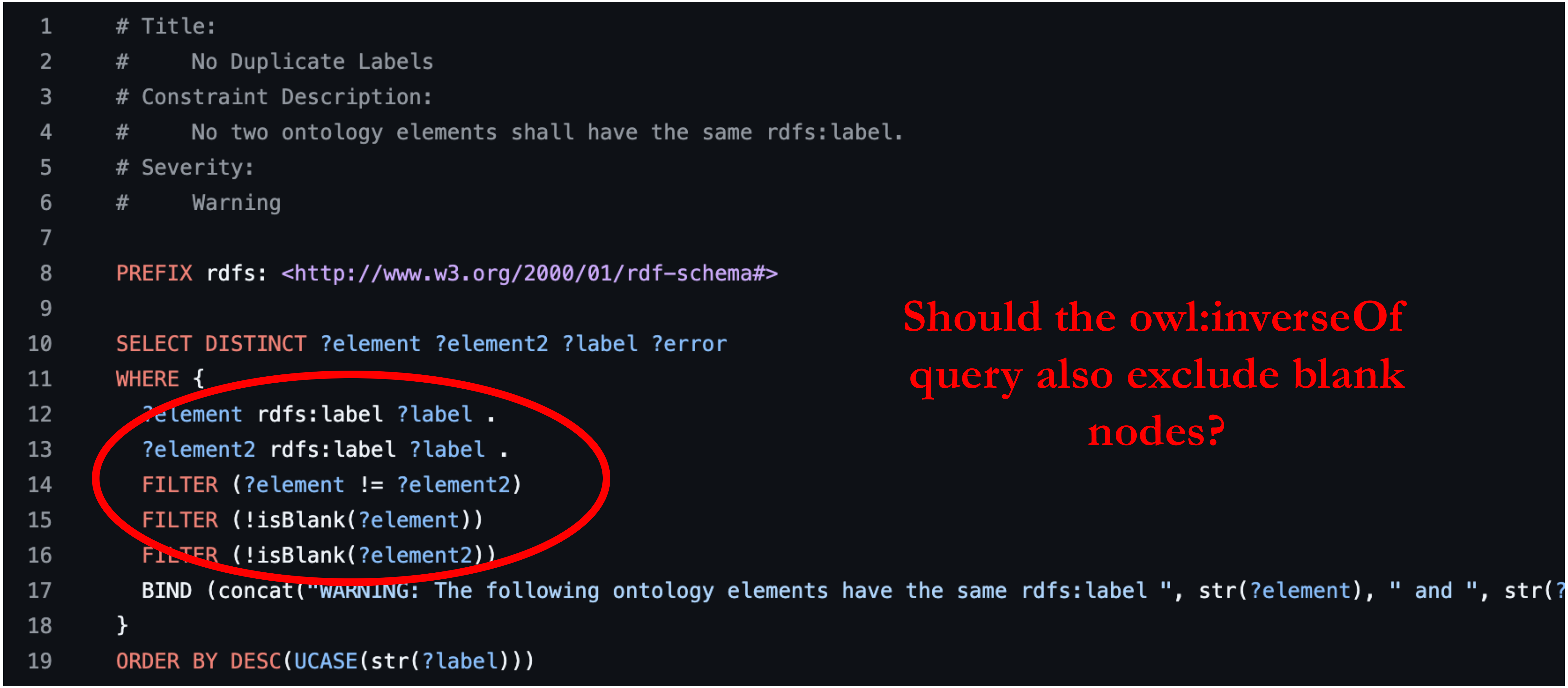


```
# Title:
#     No Duplicate Labels
# Constraint Description:
#     No two ontology elements shall have the same rdfs:label.
# Severity:
#     Warning

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT DISTINCT ?element ?element2 ?label ?error
WHERE {
  ?element rdfs:label ?label .
  ?element2 rdfs:label ?label .
  FILTER (?element != ?element2)
  FILTER (!isBlank(?element))
  FILTER (!isBlank(?element2))
  BIND (concat("WARNING: The following ontology elements have the same rdfs:label ", str(?element), " and ", str(?
}
ORDER BY DESC(UCASE(str(?label)))
```



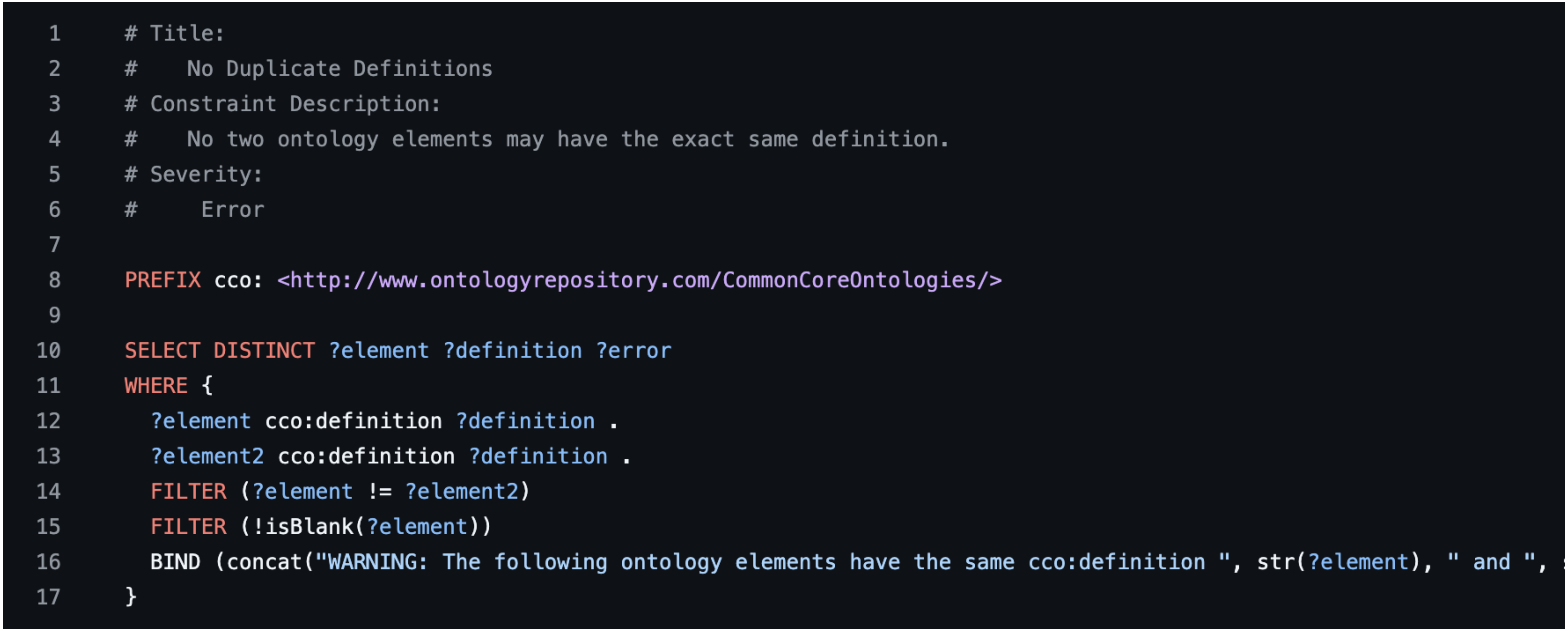
```
# Title:
#     No Duplicate Labels
# Constraint Description:
#     No two ontology elements shall have the same rdfs:label.
# Severity:
#     Warning

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT DISTINCT ?element ?element2 ?label ?error
WHERE {
  ?element rdfs:label ?label .
  ?element2 rdfs:label ?label .
  FILTER (?element != ?element2)
  FILTER (!isBlank(?element))
  FILTER (!isBlank(?element2))
  BIND (concat("WARNING: The following ontology elements have the same rdfs:label ", str(?element), " and ", str(?
}
ORDER BY DESC(UCASE(str(?label)))
```

Rather similar to the last query, though this one excludes blank nodes…

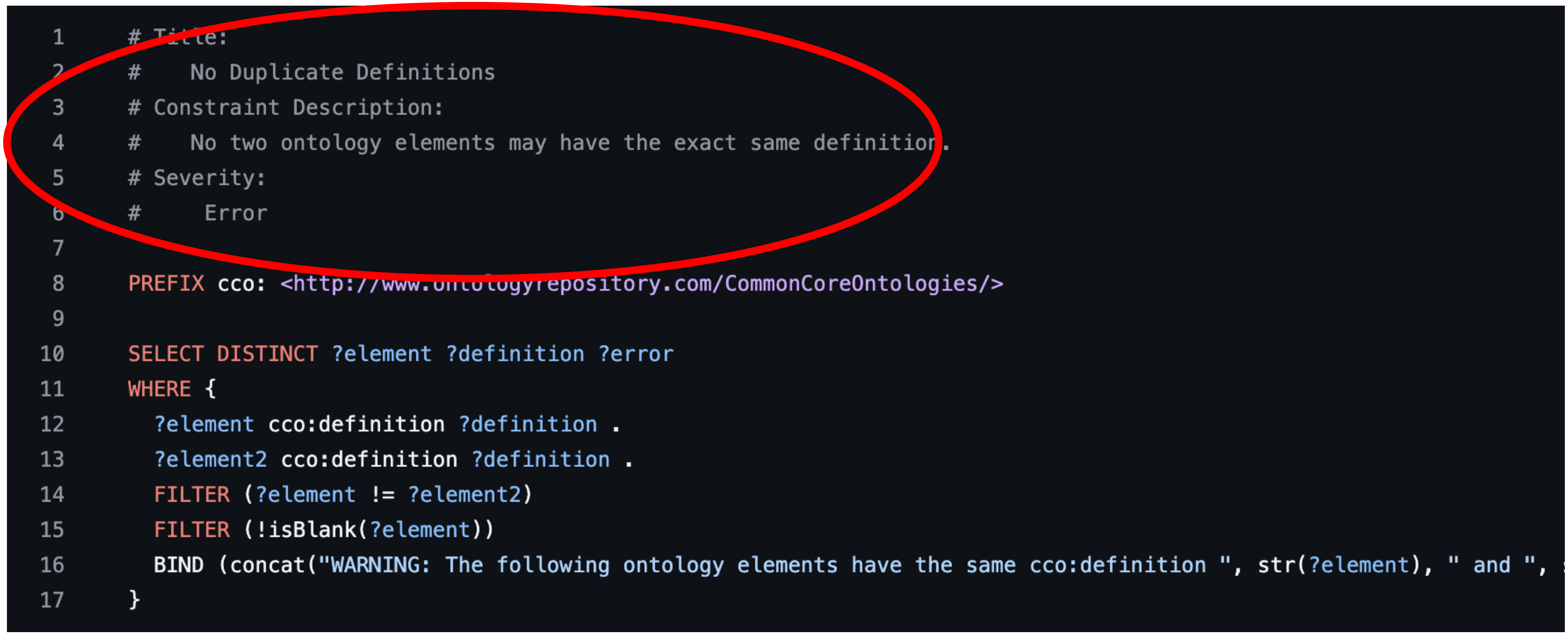


```
# Title:
#     No Duplicate Labels
# Constraint Description:
#     No two ontology elements shall have the same rdfs:label.
# Severity:
#     Warning

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT DISTINCT ?element ?element2 ?label ?error
WHERE {
  ?element rdfs:label ?label .
  ?element2 rdfs:label ?label .
  FILTER (?element != ?element2)
  FILTER (!isBlank(?element))
  FILTER (!isBlank(?element2))
  BIND (concat("WARNING: The following ontology elements have the same rdfs:label ", str(?element), " and ", str(?
}
ORDER BY DESC(UCASE(str(?label)))
```

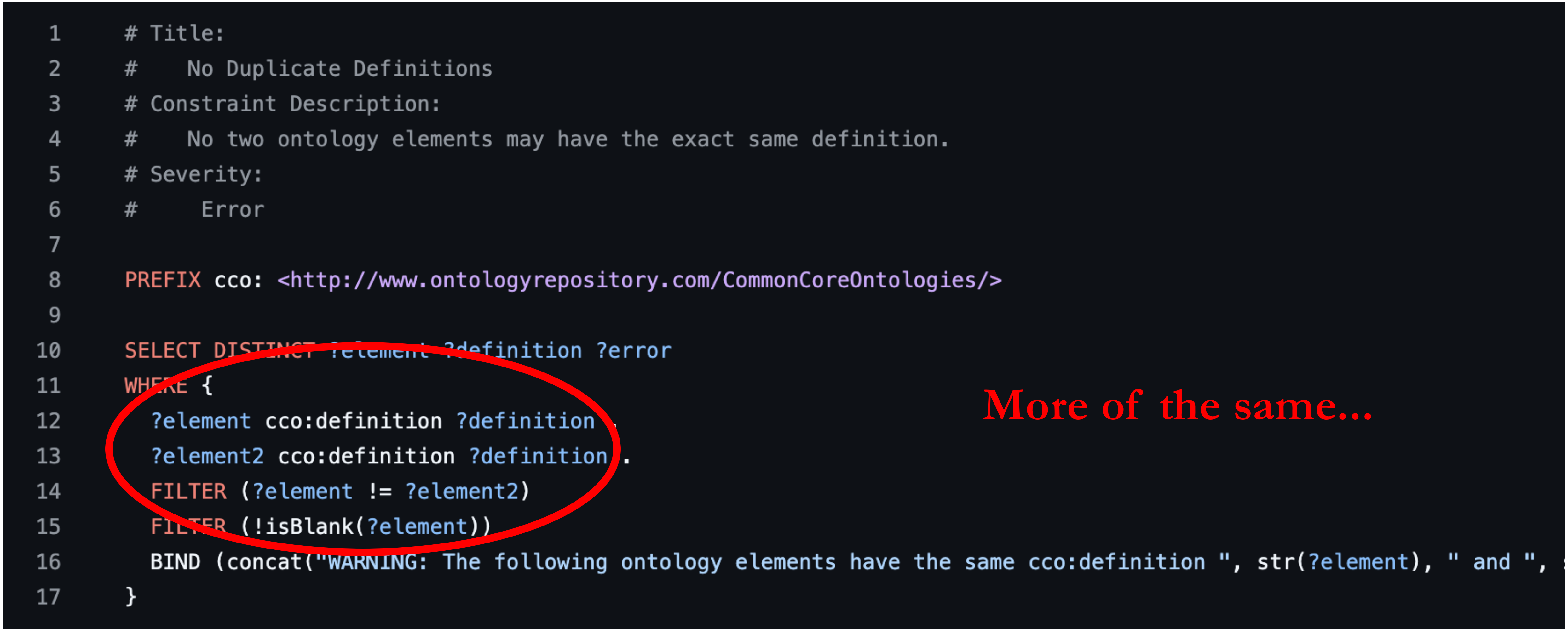
Should the owl:inverseOf query also exclude blank nodes?



```
# Title:
#    No Duplicate Definitions
# Constraint Description:
#    No two ontology elements may have the exact same definition.
# Severity:
#     Error

PREFIX cco: <http://www.ontologyrepository.com/CommonCoreOntologies/>

SELECT DISTINCT ?element ?definition ?error
WHERE {
  ?element cco:definition ?definition .
  ?element2 cco:definition ?definition .
  FILTER (?element != ?element2)
  FILTER (!isBlank(?element))
  BIND (concat("WARNING: The following ontology elements have the same cco:definition ", str(?element), " and ",
}
```

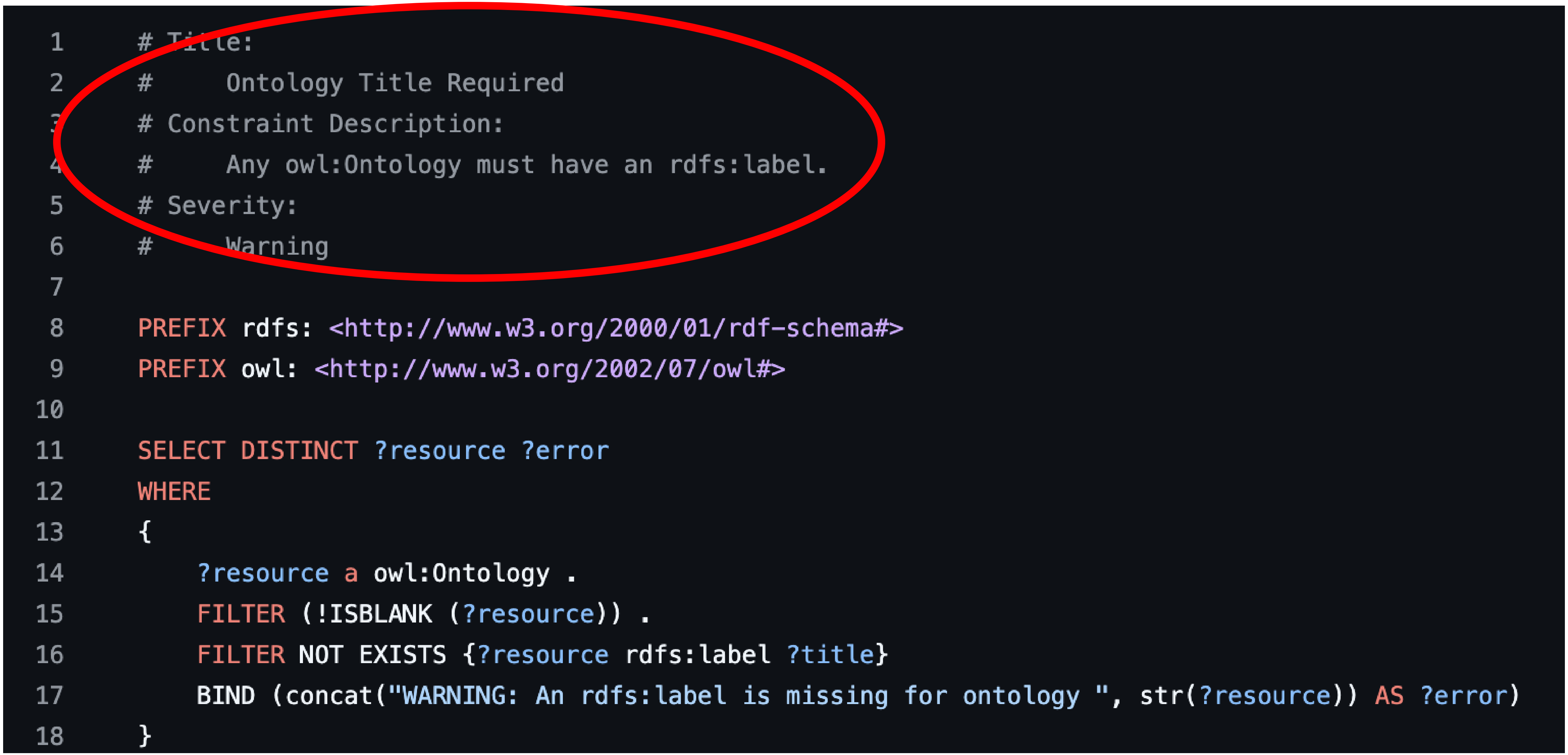


```
# Title:
#    No Duplicate Definitions
# Constraint Description:
#    No two ontology elements may have the exact same definition.
# Severity:
#     Error

PREFIX cco: <http://www.ontologyrepository.com/CommonCoreOntologies/>

SELECT DISTINCT ?element ?definition ?error
WHERE {
  ?element cco:definition ?definition .
  ?element2 cco:definition ?definition .
  FILTER (?element != ?element2)
  FILTER (!isBlank(?element))
  BIND (concat("WARNING: The following ontology elements have the same cco:definition ", str(?element), " and ",
}
```

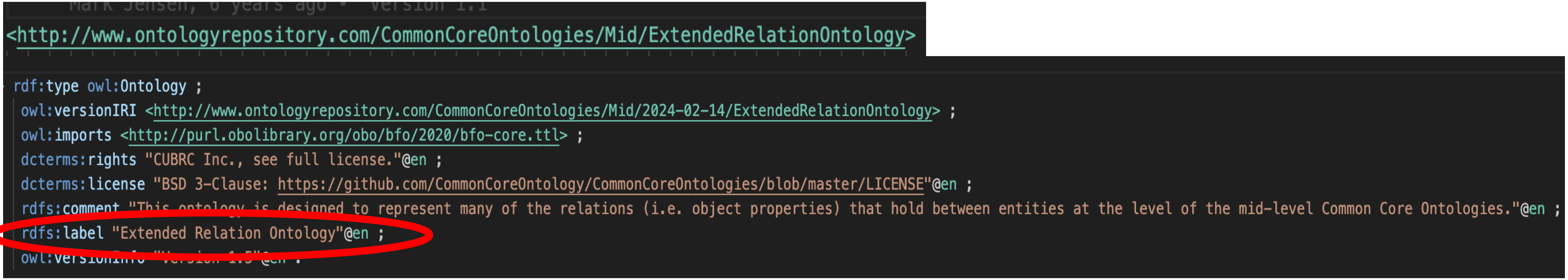


```
# Title:
#    No Duplicate Definitions
# Constraint Description:
#    No two ontology elements may have the exact same definition.
# Severity:
#     Error

PREFIX cco: <http://www.ontologyrepository.com/CommonCoreOntologies/>

SELECT DISTINCT ?element ?definition ?error
WHERE {
  ?element cco:definition ?definition .
  ?element2 cco:definition ?definition .
  FILTER (?element != ?element2)
  FILTER (!isBlank(?element))
  BIND (concat("WARNING: The following ontology elements have the same cco:definition ", str(?element), " and ",
}
```

More of the same...



```
# Title:
#     Ontology Title Required
# Constraint Description:
#     Any owl:Ontology must have an rdfs:label.
# Severity:
#     Warning

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>

SELECT DISTINCT ?resource ?error
WHERE
{
    ?resource a owl:Ontology .
    FILTER (!ISBLANK (?resource)) .
    FILTER NOT EXISTS {?resource rdfs:label ?title}
    BIND (concat("WARNING: An rdfs:label is missing for ontology ", str(?resource)) AS ?error)
}
```

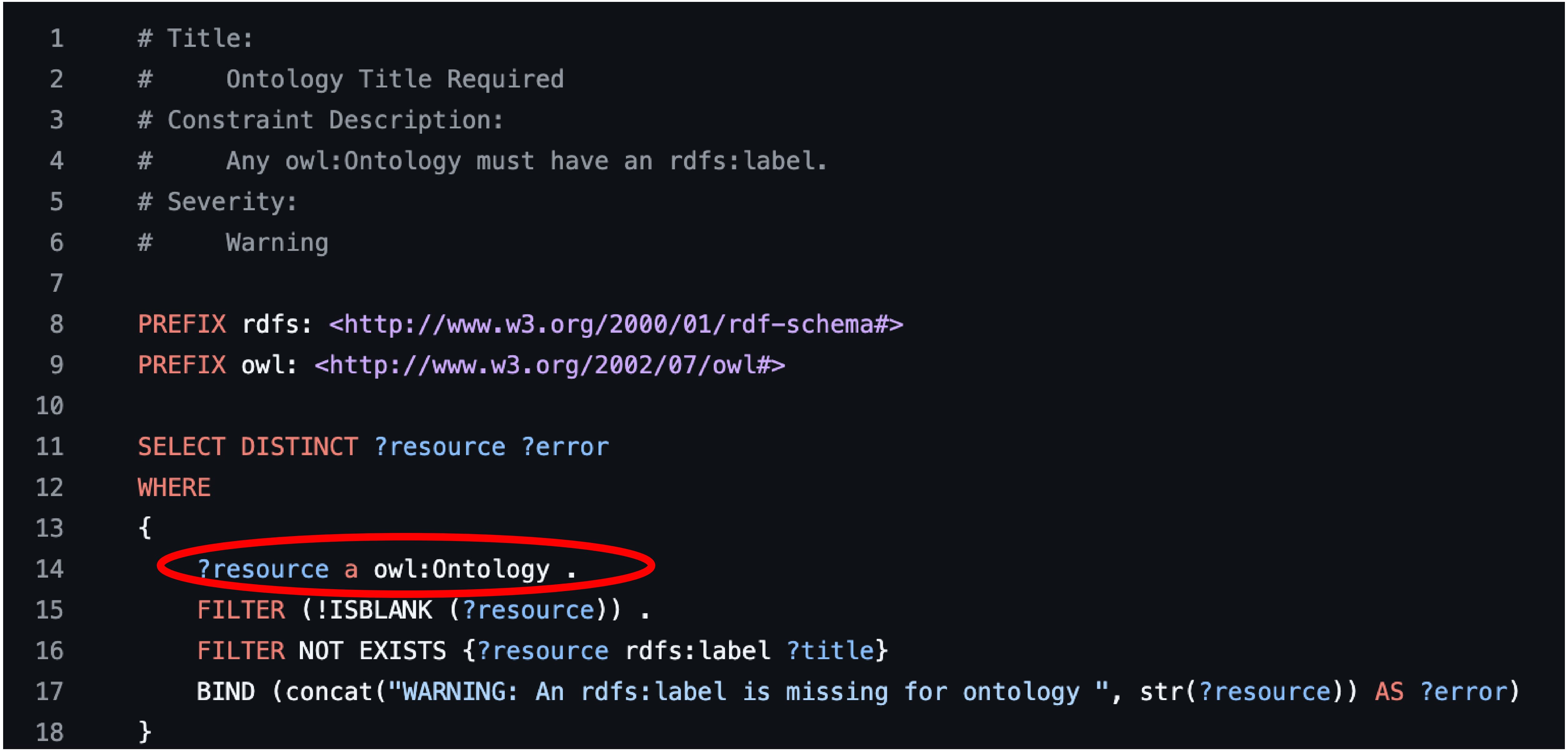


```
    Mark Jensen, 6 years ago •  Version 1.1
<http://www.ontologyrepository.com/CommonCoreOntologies/Mid/ExtendedRelationOntology>

 rdf:type owl:Ontology ;
  owl:versionIRI <http://www.ontologyrepository.com/CommonCoreOntologies/Mid/2024-02-14/ExtendedRelationOntology> ;
  owl:imports <http://purl.obolibrary.org/obo/bfo/2020/bfo-core.ttl> ;
  dcterms:rights "CUBRC Inc., see full license."@en ;
  dcterms:license "BSD 3-Clause: https://github.com/CommonCoreOntology/CommonCoreOntologies/blob/master/LICENSE"@en ;
  rdfs:comment "This ontology is designed to represent many of the relations (i.e. object properties) that hold between entities at the level of the mid-level Common Core Ontologies."@en ;
  rdfs:label "Extended Relation Ontology"@en ;
  owl:versionInfo "Version 1.5"@en .
```

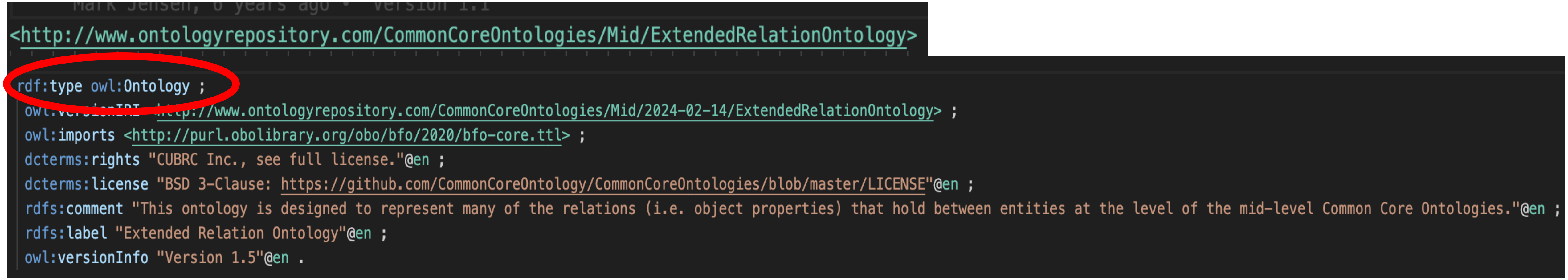
This is useful information when writing SPARQL queries against the ontology

This tells you, for example, how to extract the ontology label from the file

You’ll need a query matching the triple scheme: IRI rdfs:label ?o



```
# Title:
#     Ontology Title Required
# Constraint Description:
#     Any owl:Ontology must have an rdfs:label.
# Severity:
#     Warning

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>

SELECT DISTINCT ?resource ?error
WHERE
{
    ?resource a owl:Ontology .
    FILTER (!ISBLANK (?resource)) .
    FILTER NOT EXISTS {?resource rdfs:label ?title}
    BIND (concat("WARNING: An rdfs:label is missing for ontology ", str(?resource)) AS ?error)
}
```



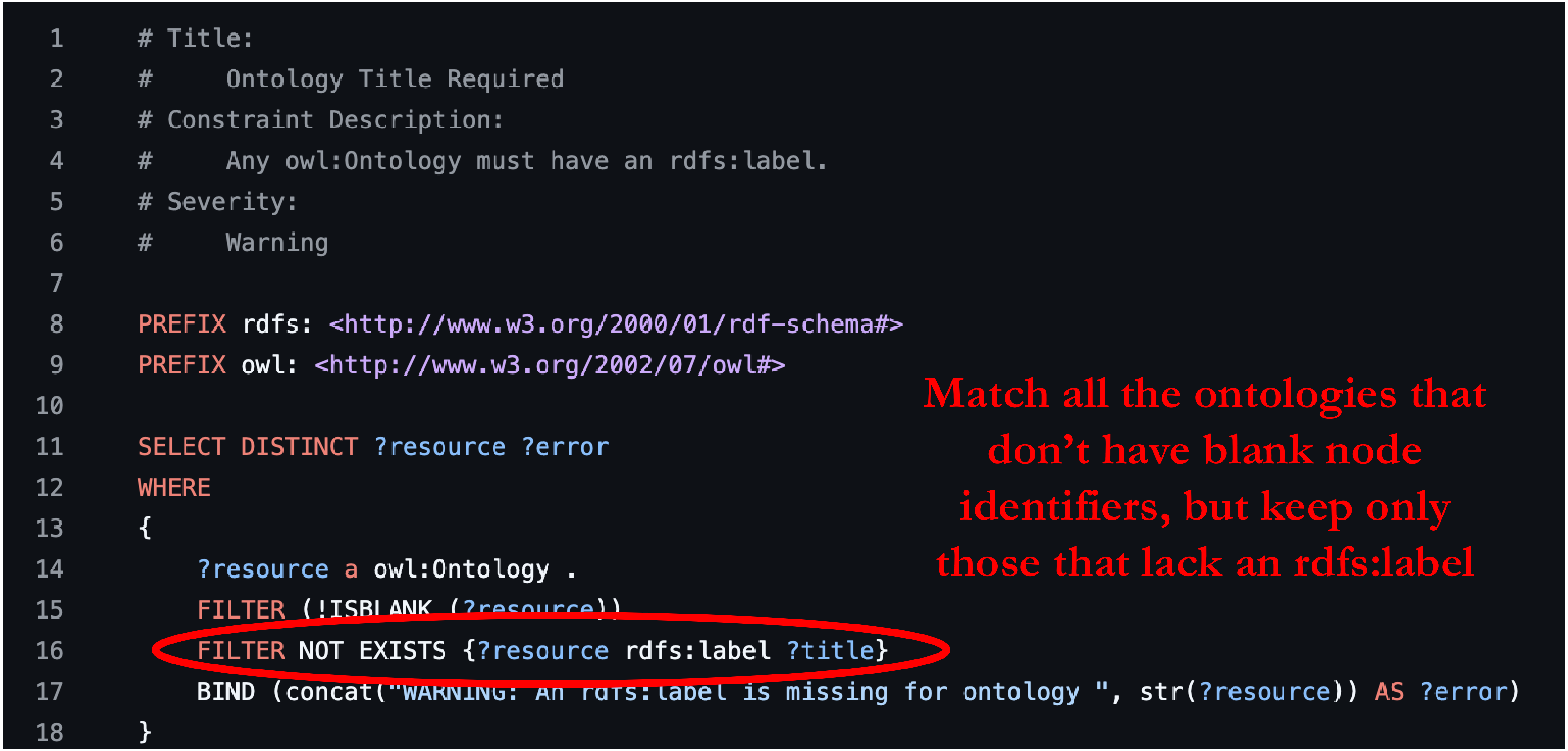
```
    Mark Jensen, 6 years ago • Version 1.1
<http://www.ontologyrepository.com/CommonCoreOntologies/Mid/ExtendedRelationOntology>

rdf:type owl:Ontology ;
 owl:versionIRI <http://www.ontologyrepository.com/CommonCoreOntologies/Mid/2024-02-14/ExtendedRelationOntology> ;
 owl:imports <http://purl.obolibrary.org/obo/bfo/2020/bfo-core.ttl> ;
 dcterms:rights "CUBRC Inc., see full license."@en ;
 dcterms:license "BSD 3-Clause: https://github.com/CommonCoreOntology/CommonCoreOntologies/blob/master/LICENSE"@en ;
 rdfs:comment "This ontology is designed to represent many of the relations (i.e. object properties) that hold between entities at the level of the mid-level Common Core Ontologies."@en ;
 rdfs:label "Extended Relation Ontology"@en ;
 owl:versionInfo "Version 1.5"@en .
```

rdf:type is sometimes abbreviated as “a”

?resource a owl:Ontology

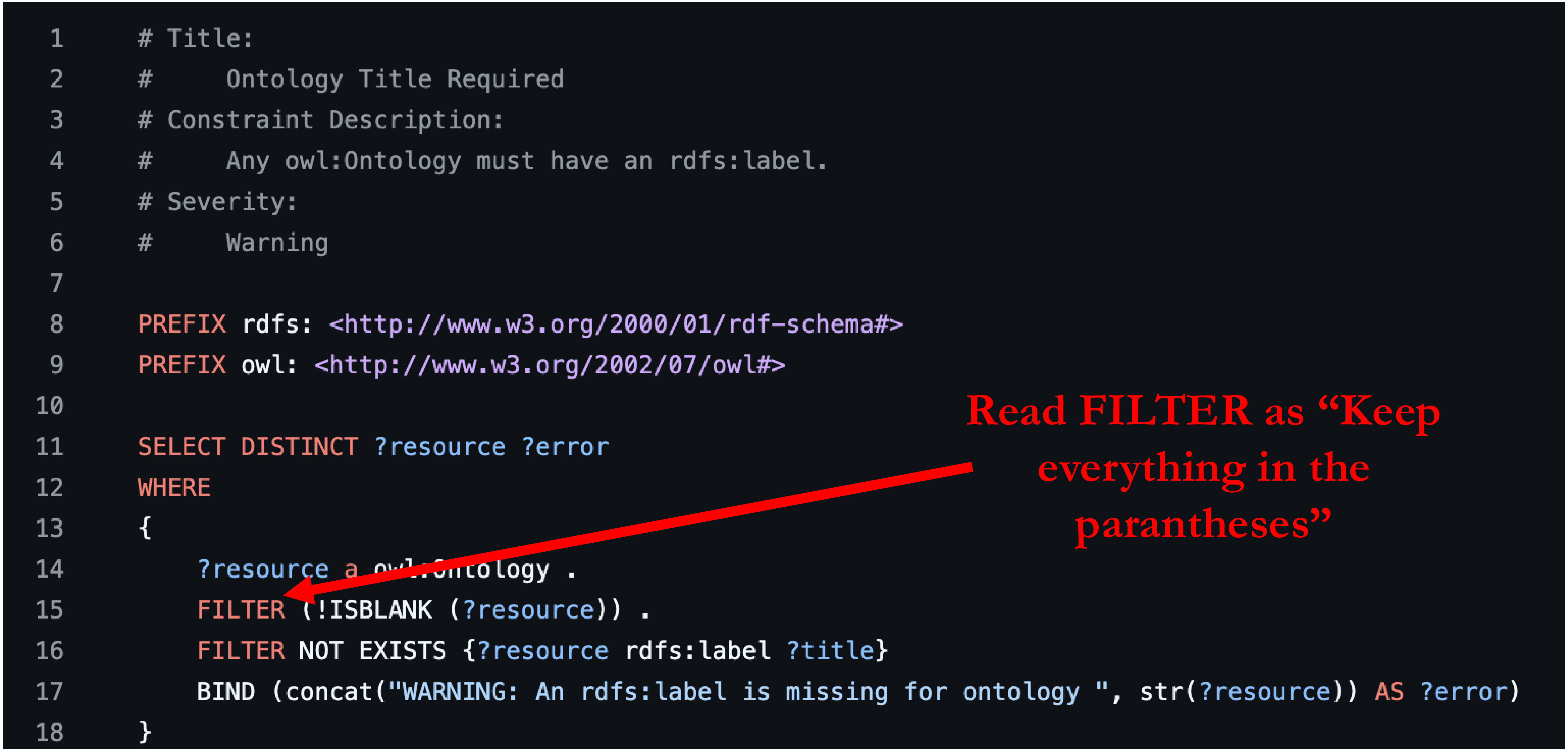
=

?resource rdf:type owl:Ontology



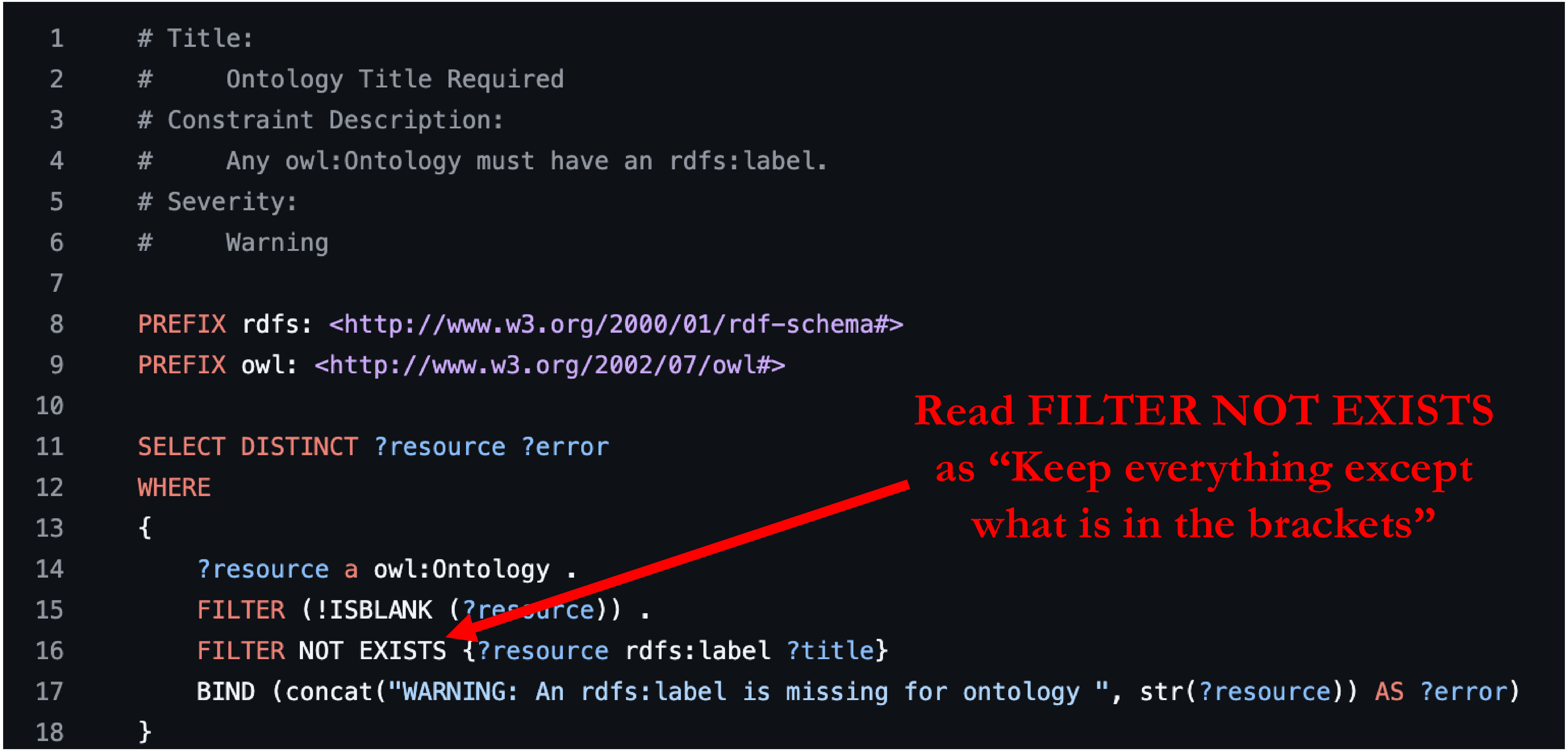
```
# Title:
#     Ontology Title Required
# Constraint Description:
#     Any owl:Ontology must have an rdfs:label.
# Severity:
#     Warning

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>

SELECT DISTINCT ?resource ?error
WHERE
{
    ?resource a owl:Ontology .
    FILTER (!ISBLANK (?resource))
    FILTER NOT EXISTS {?resource rdfs:label ?title}
    BIND (concat("WARNING: An rdfs:label is missing for ontology ", str(?resource)) AS ?error)
}
```

Match all the ontologies that don’t have blank node identifiers, but keep only those that lack an rdfs:label



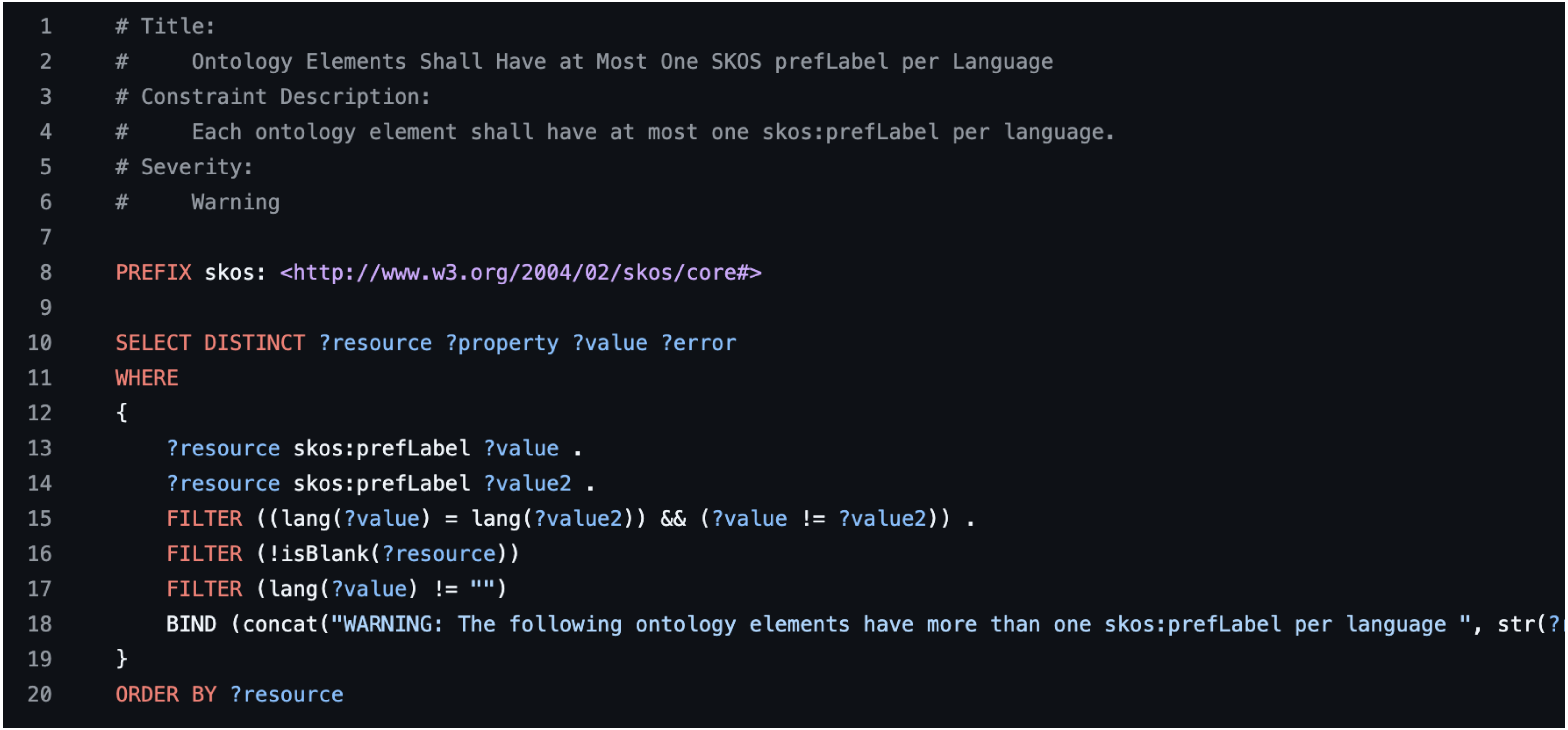
```
# Title:
#     Ontology Title Required
# Constraint Description:
#     Any owl:Ontology must have an rdfs:label.
# Severity:
#     Warning

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>

SELECT DISTINCT ?resource ?error
WHERE
{
    ?resource a owl:Ontology .
    FILTER (!ISBLANK (?resource)) .
    FILTER NOT EXISTS {?resource rdfs:label ?title}
    BIND (concat("WARNING: An rdfs:label is missing for ontology ", str(?resource)) AS ?error)
}
```

Read FILTER as "Keep everything in the parantheses"

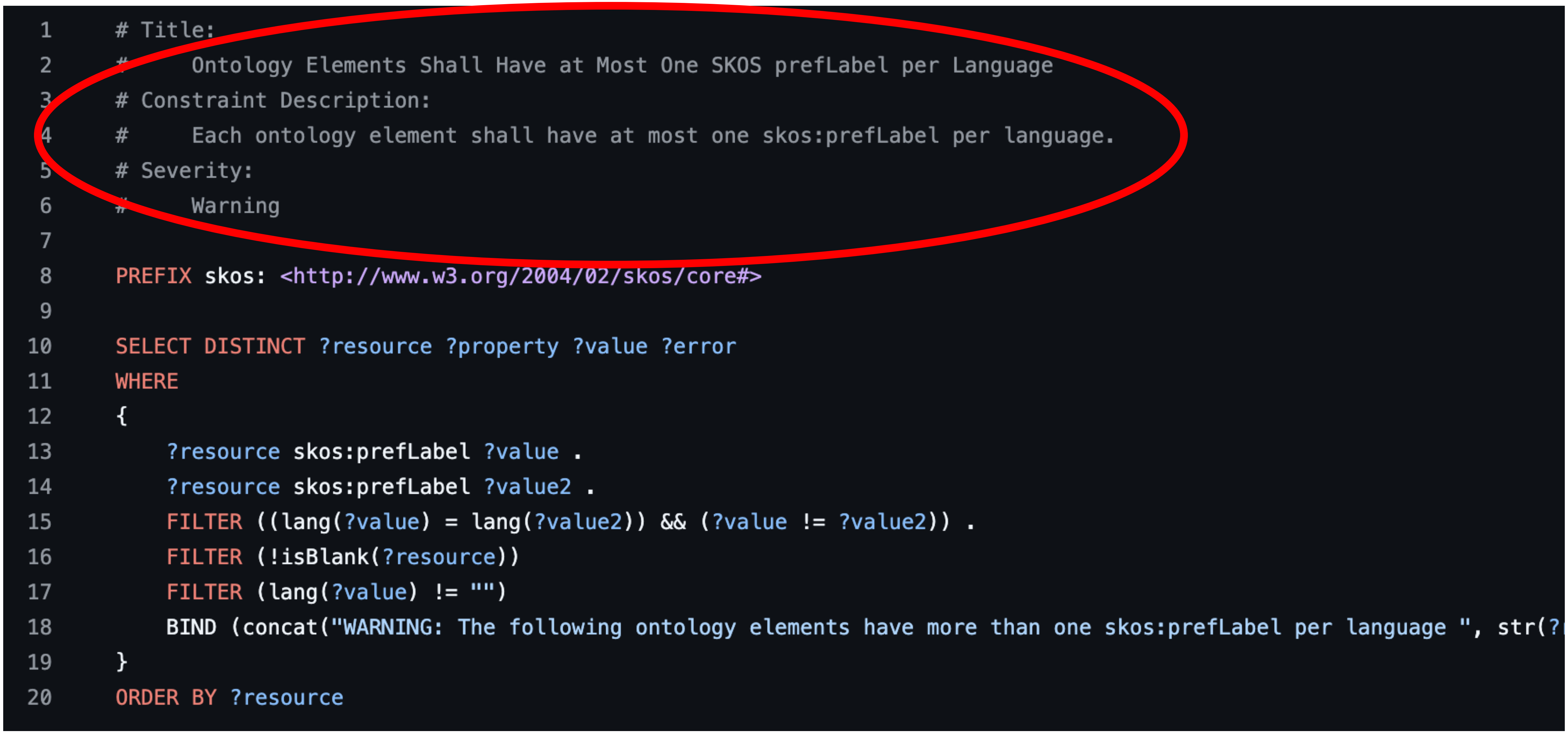


```
# Title:
#     Ontology Title Required
# Constraint Description:
#     Any owl:Ontology must have an rdfs:label.
# Severity:
#     Warning

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>

SELECT DISTINCT ?resource ?error
WHERE
{
    ?resource a owl:Ontology .
    FILTER (!ISBLANK (?resource)) .
    FILTER NOT EXISTS {?resource rdfs:label ?title}
    BIND (concat("WARNING: An rdfs:label is missing for ontology ", str(?resource)) AS ?error)
}
```

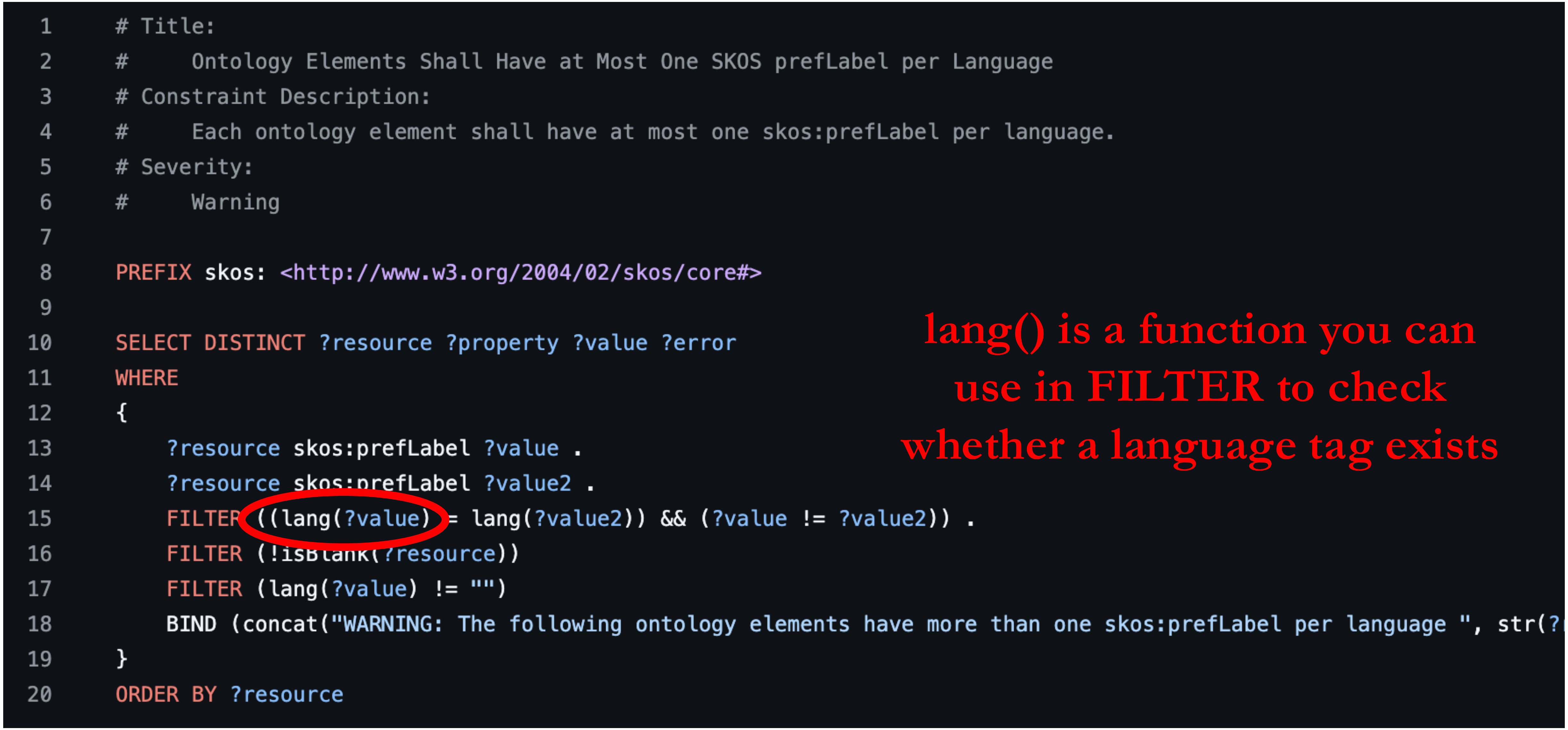
Read FILTER NOT EXISTS as “Keep everything except what is in the brackets”



```
# Title:
#     Ontology Elements Shall Have at Most One SKOS prefLabel per Language
# Constraint Description:
#     Each ontology element shall have at most one skos:prefLabel per language.
# Severity:
#     Warning

PREFIX skos: <http://www.w3.org/2004/02/skos/core#>

SELECT DISTINCT ?resource ?property ?value ?error
WHERE
{
    ?resource skos:prefLabel ?value .
    ?resource skos:prefLabel ?value2 .
    FILTER ((lang(?value) = lang(?value2)) && (?value != ?value2)) .
    FILTER (!isBlank(?resource))
    FILTER (lang(?value) != "")
    BIND (concat("WARNING: The following ontology elements have more than one skos:prefLabel per language ", str(?
}
ORDER BY ?resource
```

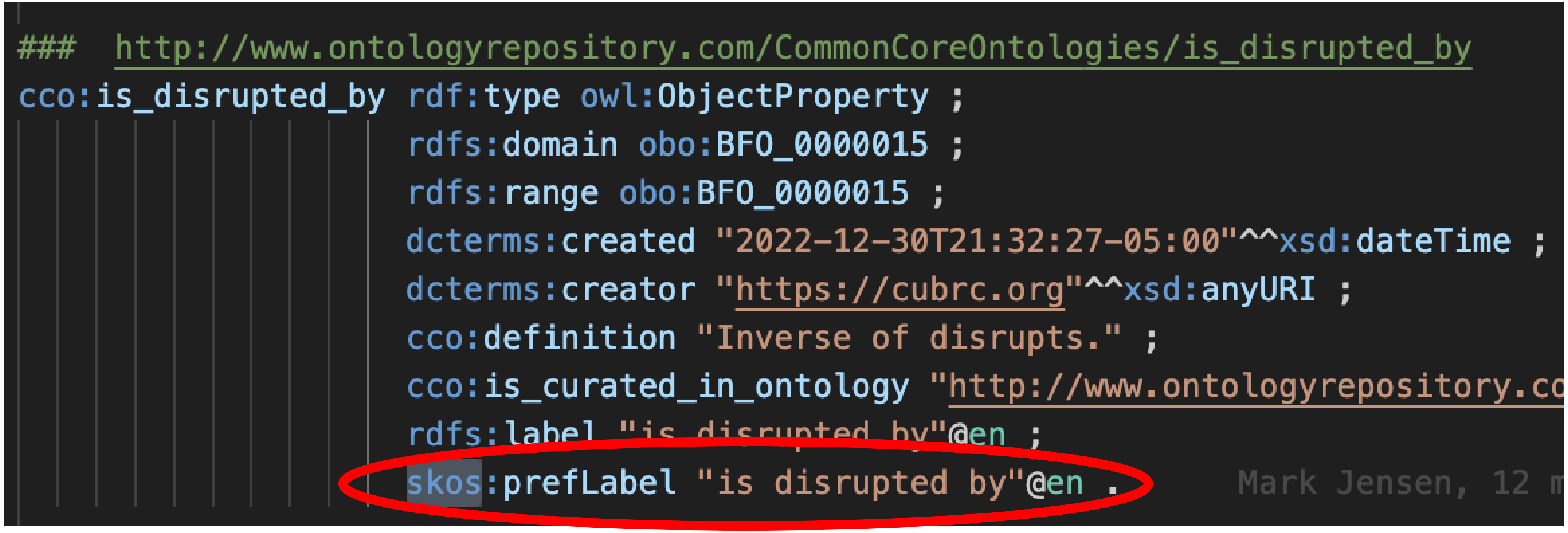


```
# Title:
#     Ontology Elements Shall Have at Most One SKOS prefLabel per Language
# Constraint Description:
#     Each ontology element shall have at most one skos:prefLabel per language.
# Severity:
#     Warning

PREFIX skos: <http://www.w3.org/2004/02/skos/core#>

SELECT DISTINCT ?resource ?property ?value ?error
WHERE
{
    ?resource skos:prefLabel ?value .
    ?resource skos:prefLabel ?value2 .
    FILTER ((lang(?value) = lang(?value2)) && (?value != ?value2)) .
    FILTER (!isBlank(?resource))
    FILTER (lang(?value) != "")
    BIND (concat("WARNING: The following ontology elements have more than one skos:prefLabel per language ", str(?
}
ORDER BY ?resource
```



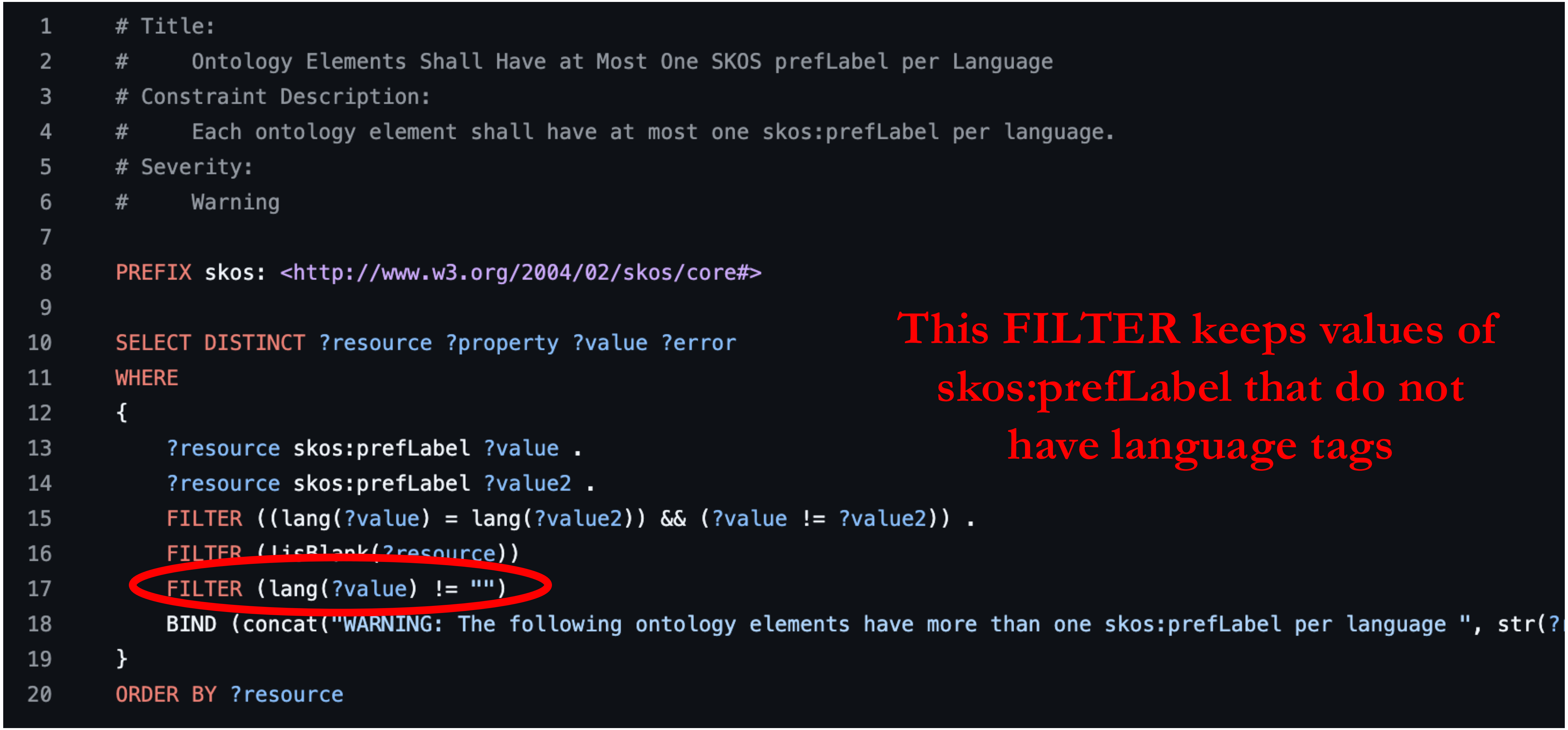
```
# Title:
#     Ontology Elements Shall Have at Most One SKOS prefLabel per Language
# Constraint Description:
#     Each ontology element shall have at most one skos:prefLabel per language.
# Severity:
#     Warning

PREFIX skos: <http://www.w3.org/2004/02/skos/core#>

SELECT DISTINCT ?resource ?property ?value ?error
WHERE
{
    ?resource skos:prefLabel ?value .
    ?resource skos:prefLabel ?value2 .
    FILTER ((lang(?value) = lang(?value2)) && (?value != ?value2)) .
    FILTER (!isBlank(?resource))
    FILTER (lang(?value) != "")
    BIND (concat("WARNING: The following ontology elements have more than one skos:prefLabel per language ", str(?
}
ORDER BY ?resource
```

lang() is a function you can use in FILTER to check whether a language tag exists



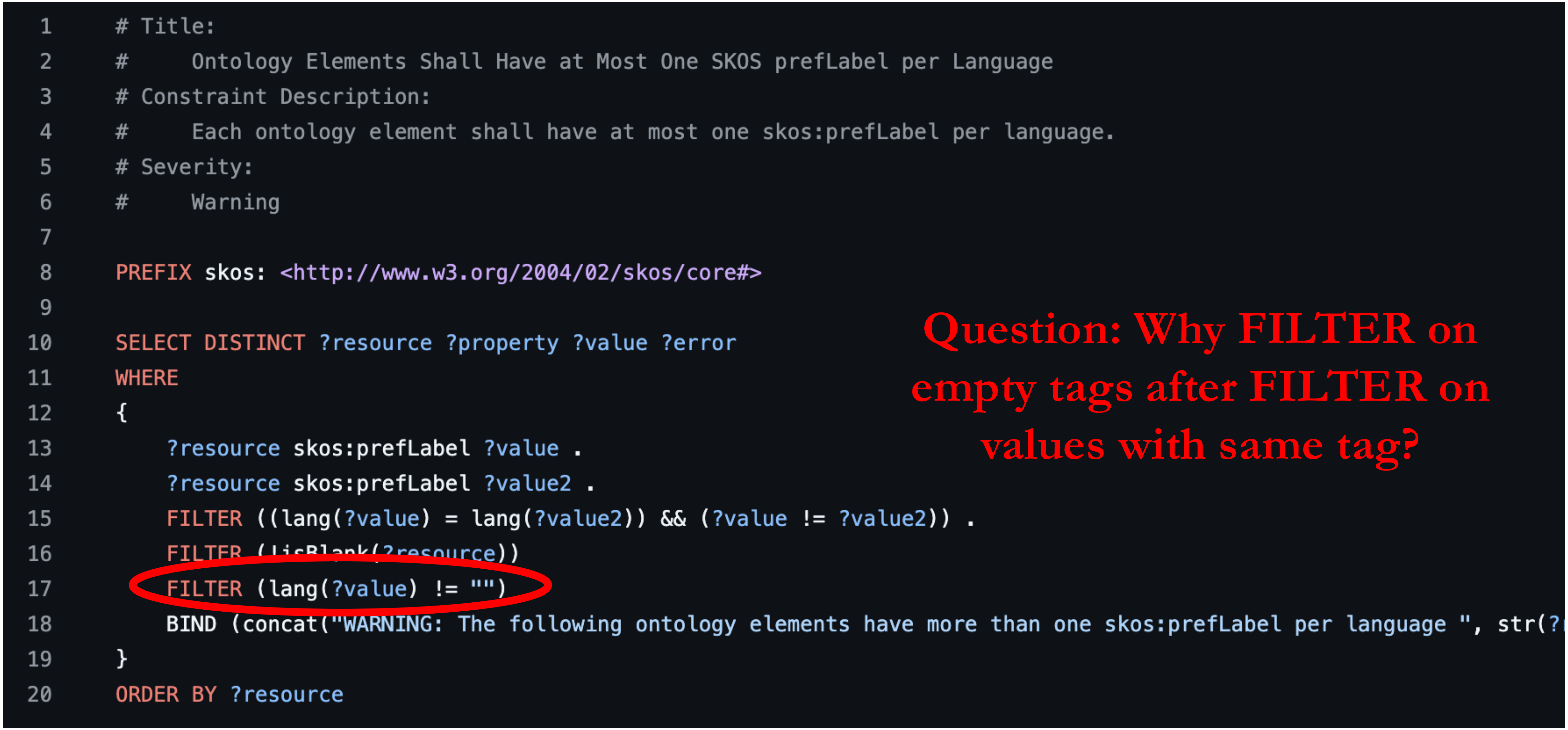
```
###  http://www.ontologyrepository.com/CommonCoreOntologies/is_disrupted_by
cco:is_disrupted_by rdf:type owl:ObjectProperty ;
                    rdfs:domain obo:BFO_0000015 ;
                    rdfs:range obo:BFO_0000015 ;
                    dcterms:created "2022-12-30T21:32:27-05:00"^^xsd:dateTime ;
                    dcterms:creator "https://cubrc.org"^^xsd:anyURI ;
                    cco:definition "Inverse of disrupts." ;
                    cco:is_curated_in_ontology "http://www.ontologyrepository.co
                    rdfs:label "is disrupted by"@en ;
                    skos:prefLabel "is disrupted by"@en .     Mark Jensen, 12 m
```

Language tags in Turtle follow quote marks and an ampersand, e.g. @en



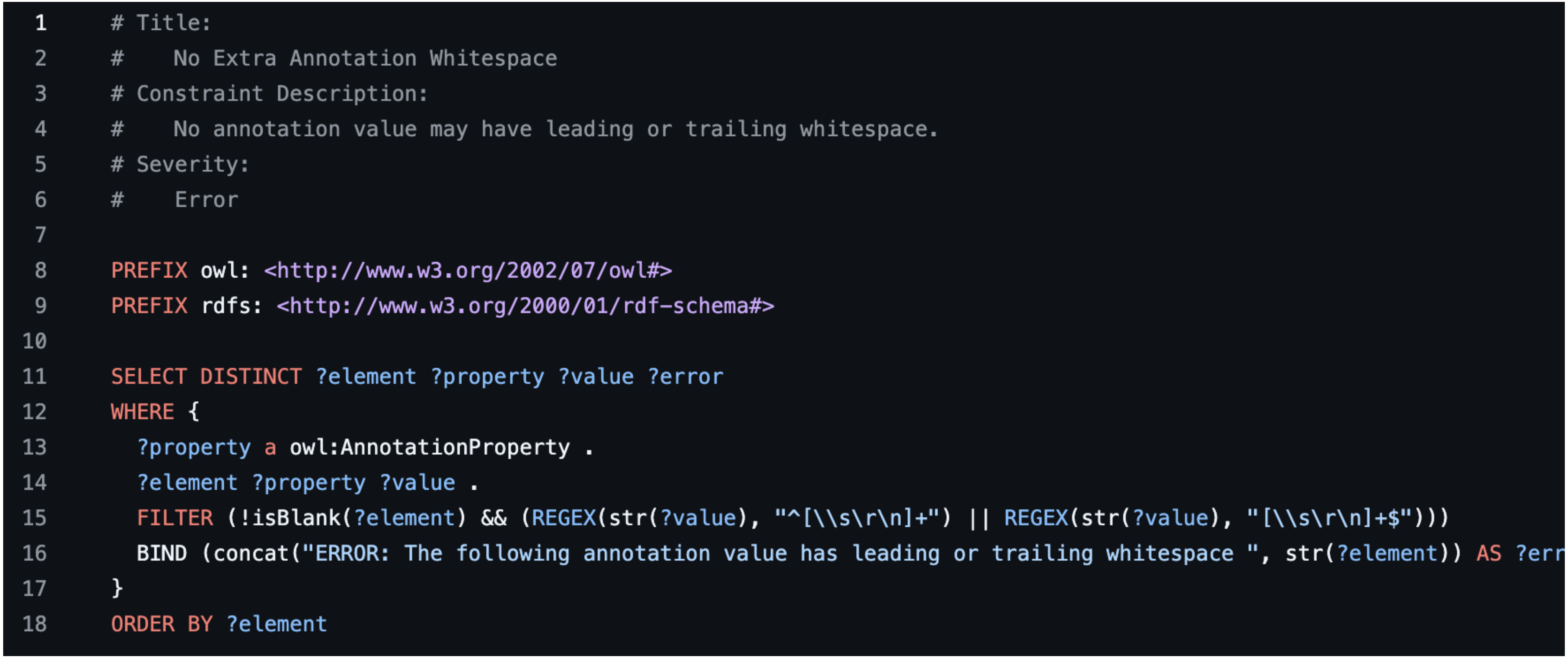
```
# Title:
#     Ontology Elements Shall Have at Most One SKOS prefLabel per Language
# Constraint Description:
#     Each ontology element shall have at most one skos:prefLabel per language.
# Severity:
#     Warning

PREFIX skos: <http://www.w3.org/2004/02/skos/core#>

SELECT DISTINCT ?resource ?property ?value ?error
WHERE
{
    ?resource skos:prefLabel ?value .
    ?resource skos:prefLabel ?value2 .
    FILTER ((lang(?value) = lang(?value2)) && (?value != ?value2)) .
    FILTER (!isBlank(?resource))
    FILTER (lang(?value) != "")
    BIND (concat("WARNING: The following ontology elements have more than one skos:prefLabel per language ", str(?
}
ORDER BY ?resource
```

This FILTER keeps values of skos:prefLabel that do not have language tags

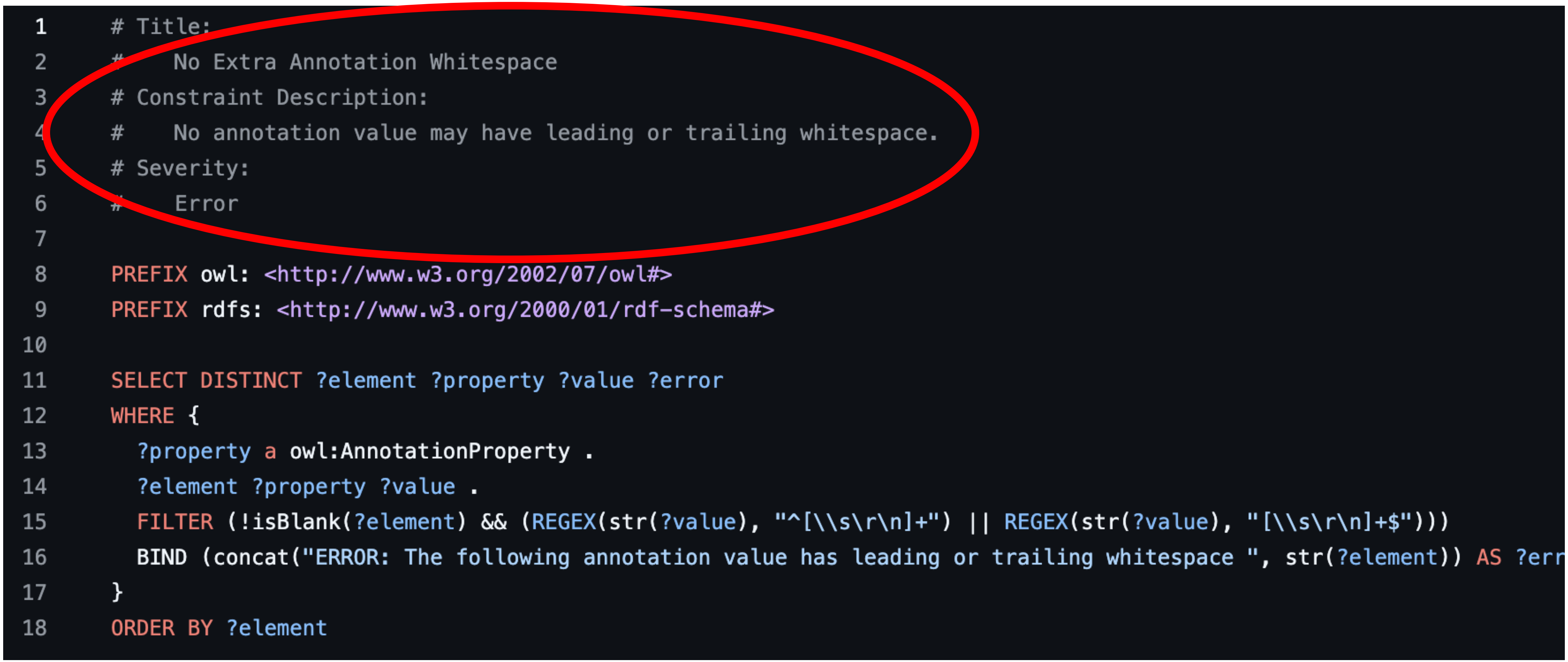


```
# Title:
#     Ontology Elements Shall Have at Most One SKOS prefLabel per Language
# Constraint Description:
#     Each ontology element shall have at most one skos:prefLabel per language.
# Severity:
#     Warning

PREFIX skos: <http://www.w3.org/2004/02/skos/core#>

SELECT DISTINCT ?resource ?property ?value ?error
WHERE
{
    ?resource skos:prefLabel ?value .
    ?resource skos:prefLabel ?value2 .
    FILTER ((lang(?value) = lang(?value2)) && (?value != ?value2)) .
    FILTER (!isBlank(?resource))
    FILTER (lang(?value) != "")
    BIND (concat("WARNING: The following ontology elements have more than one skos:prefLabel per language ", str(?
}
ORDER BY ?resource
```

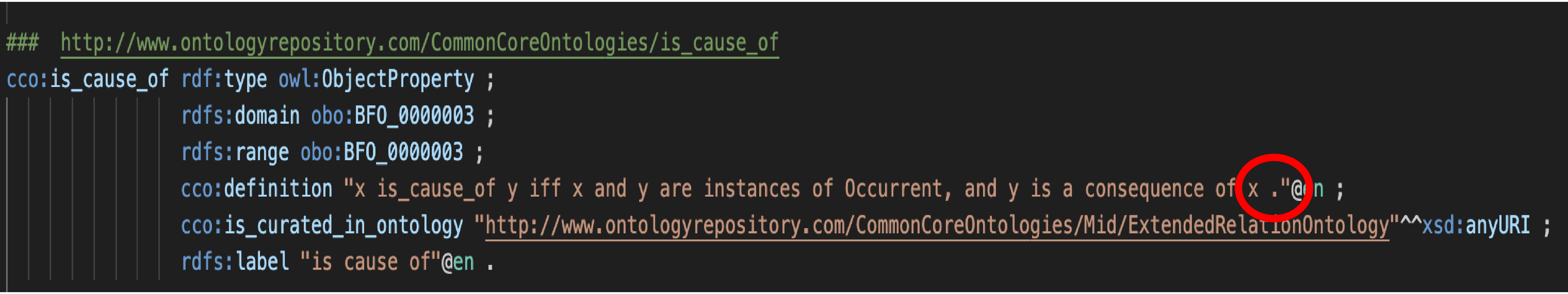
Question: Why FILTER on empty tags after FILTER on values with same tag?



```
# Title:
#    No Extra Annotation Whitespace
# Constraint Description:
#    No annotation value may have leading or trailing whitespace.
# Severity:
#    Error

PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT DISTINCT ?element ?property ?value ?error
WHERE {
  ?property a owl:AnnotationProperty .
  ?element ?property ?value .
  FILTER (!isBlank(?element) && (REGEX(str(?value), "^[\\s\r\n]+") || REGEX(str(?value), "[\\s\r\n]+$")))
  BIND (concat("ERROR: The following annotation value has leading or trailing whitespace ", str(?element)) AS ?err
}
ORDER BY ?element
```

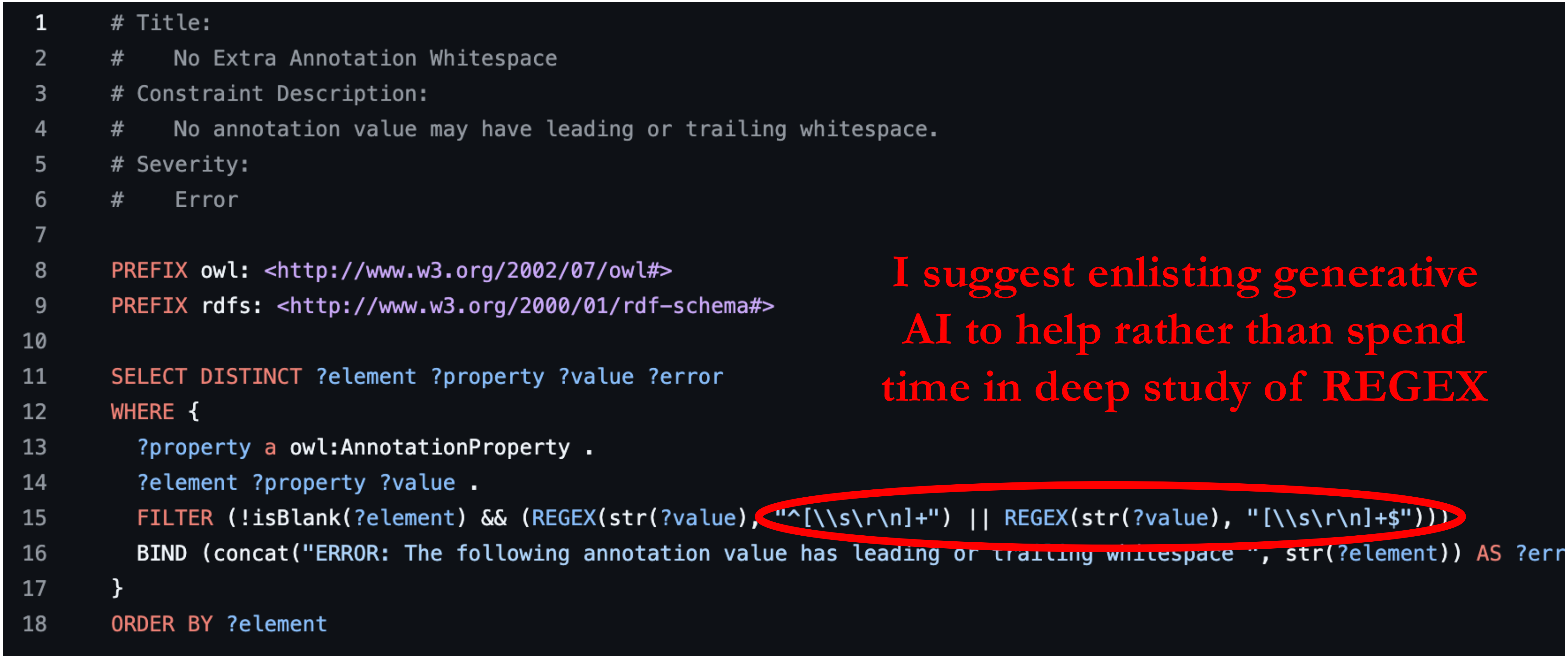


```
# Title:
#    No Extra Annotation Whitespace
# Constraint Description:
#    No annotation value may have leading or trailing whitespace.
# Severity:
#    Error

PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT DISTINCT ?element ?property ?value ?error
WHERE {
  ?property a owl:AnnotationProperty .
  ?element ?property ?value .
  FILTER (!isBlank(?element) && (REGEX(str(?value), "^[\\s\r\n]+") || REGEX(str(?value), "[\\s\r\n]+$")))
  BIND (concat("ERROR: The following annotation value has leading or trailing whitespace ", str(?element)) AS ?err
}
ORDER BY ?element
```



```
###  http://www.ontologyrepository.com/CommonCoreOntologies/is_cause_of
cco:is_cause_of rdf:type owl:ObjectProperty ;
                rdfs:domain obo:BFO_0000003 ;
                rdfs:range obo:BFO_0000003 ;
                cco:definition "x is_cause_of y iff x and y are instances of Occurrent, and y is a consequence of x ."@en ;
                cco:is_curated_in_ontology "http://www.ontologyrepository.com/CommonCoreOntologies/Mid/ExtendedRelationOntology"^^xsd:anyURI ;
                rdfs:label "is cause of"@en .
```

Use REGEX in FILTER to identify specific strings or whitespace in files

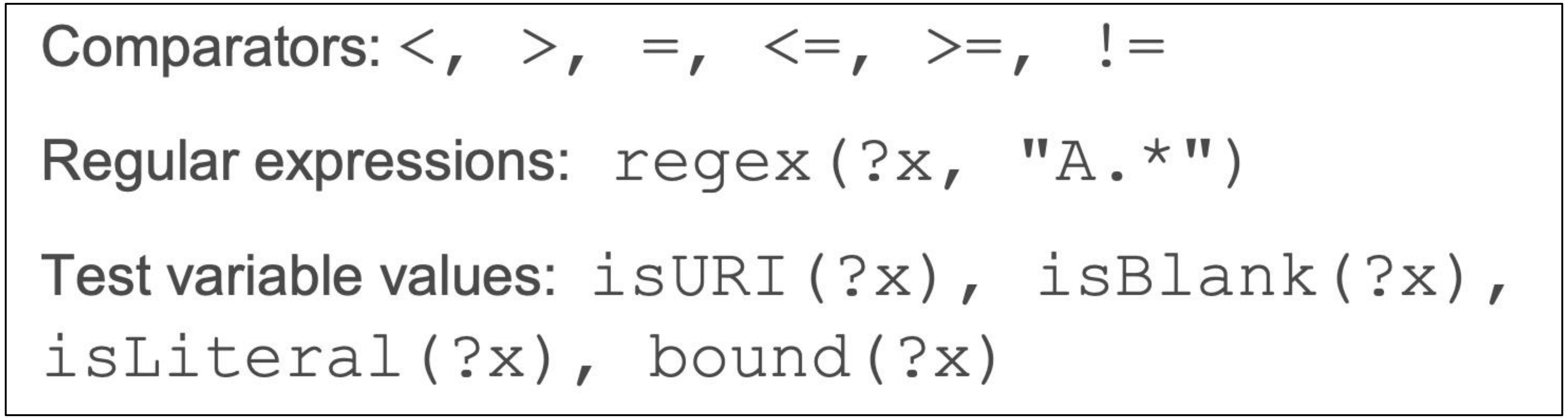


```
# Title:
#    No Extra Annotation Whitespace
# Constraint Description:
#    No annotation value may have leading or trailing whitespace.
# Severity:
#    Error

PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT DISTINCT ?element ?property ?value ?error
WHERE {
  ?property a owl:AnnotationProperty .
  ?element ?property ?value .
  FILTER (!isBlank(?element) && (REGEX(str(?value), "^[\\s\r\n]+") || REGEX(str(?value), "[\\s\r\n]+$")))
  BIND (concat("ERROR: The following annotation value has leading or trailing whitespace ", str(?element)) AS ?err
}
ORDER BY ?element
```

I suggest enlisting generative AI to help rather than spend time in deep study of REGEX

FILTER

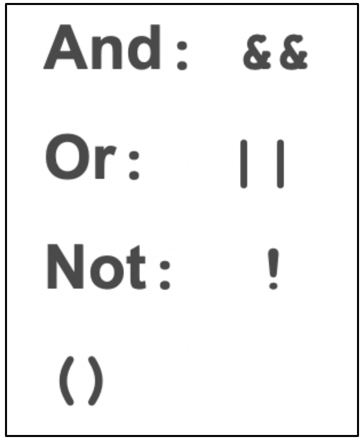
- FILTER functions include:



Comparators: <, >, =, <=, >=, !=

Regular expressions: regex(?x, "A.*")

Test variable values: isURI(?x), isBlank(?x), isLiteral(?x), bound(?x)

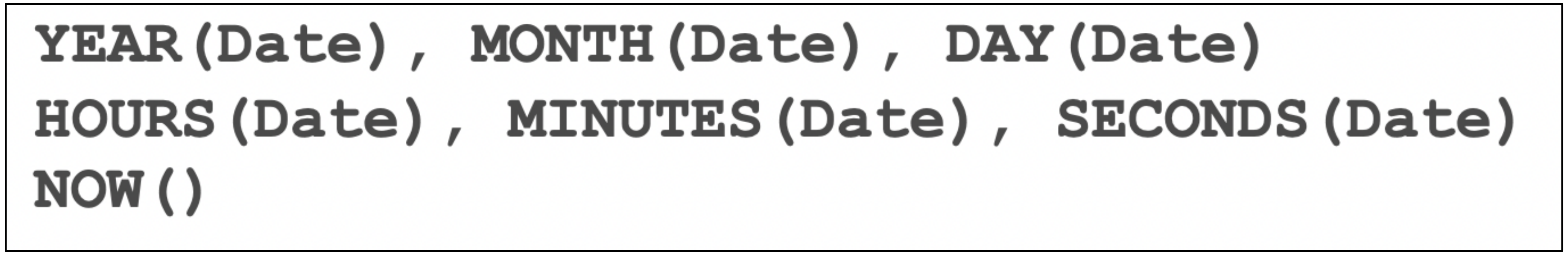


And: &&

Or: ||

Not: !

()

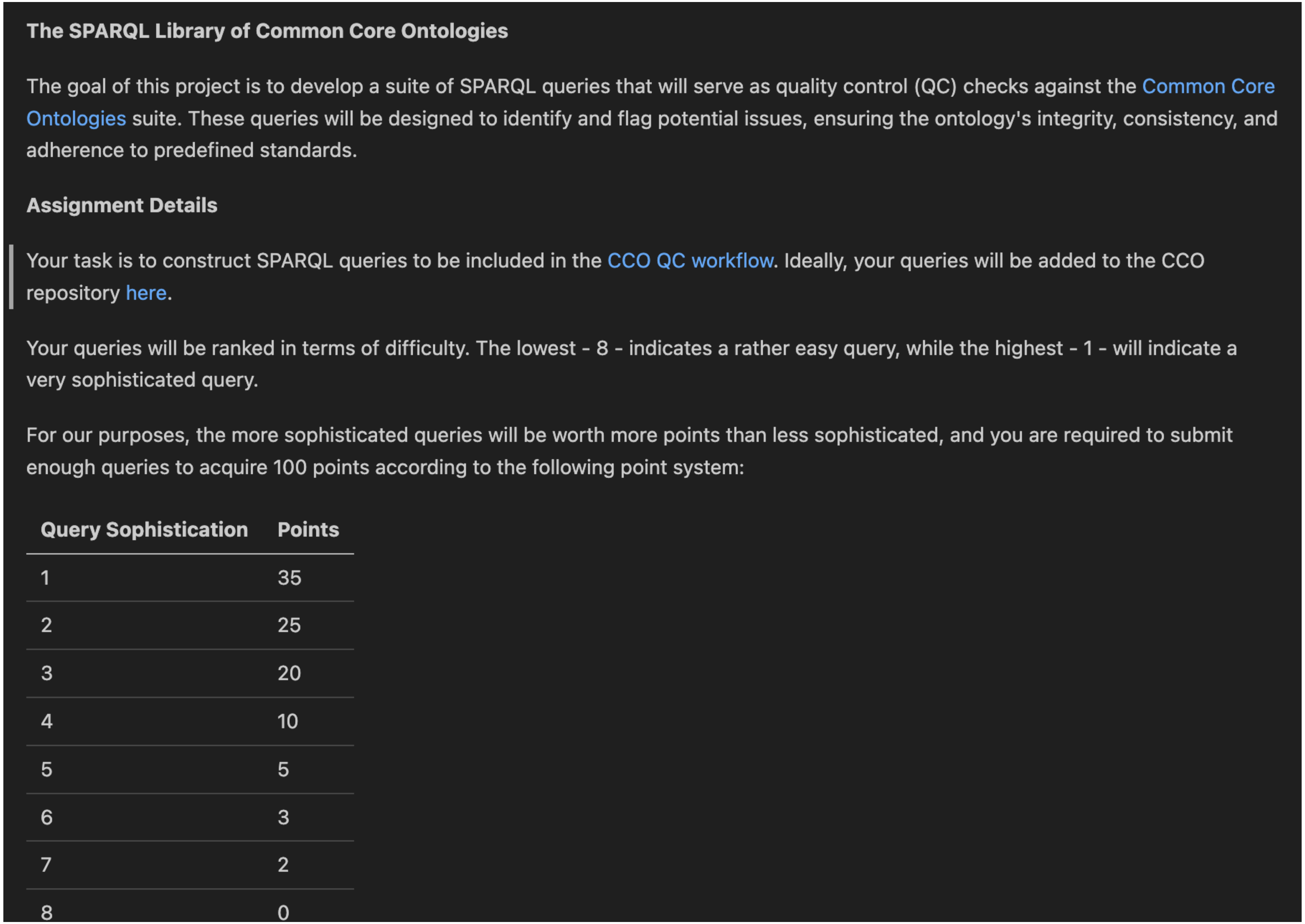


YEAR(Date), MONTH(Date), DAY(Date)
HOURS(Date), MINUTES(Date), SECONDS(Date)
NOW()

Quality Control

- As you investigate the CCO repository, you'll note there are not many SPARQL queries being run against builds

You will be providing new quality control SPARQL queries to be added to the CCO repository



The SPARQL Library of Common Core Ontologies

The goal of this project is to develop a suite of SPARQL queries that will serve as quality control (QC) checks against the Common Core Ontologies suite. These queries will be designed to identify and flag potential issues, ensuring the ontology's integrity, consistency, and adherence to predefined standards.

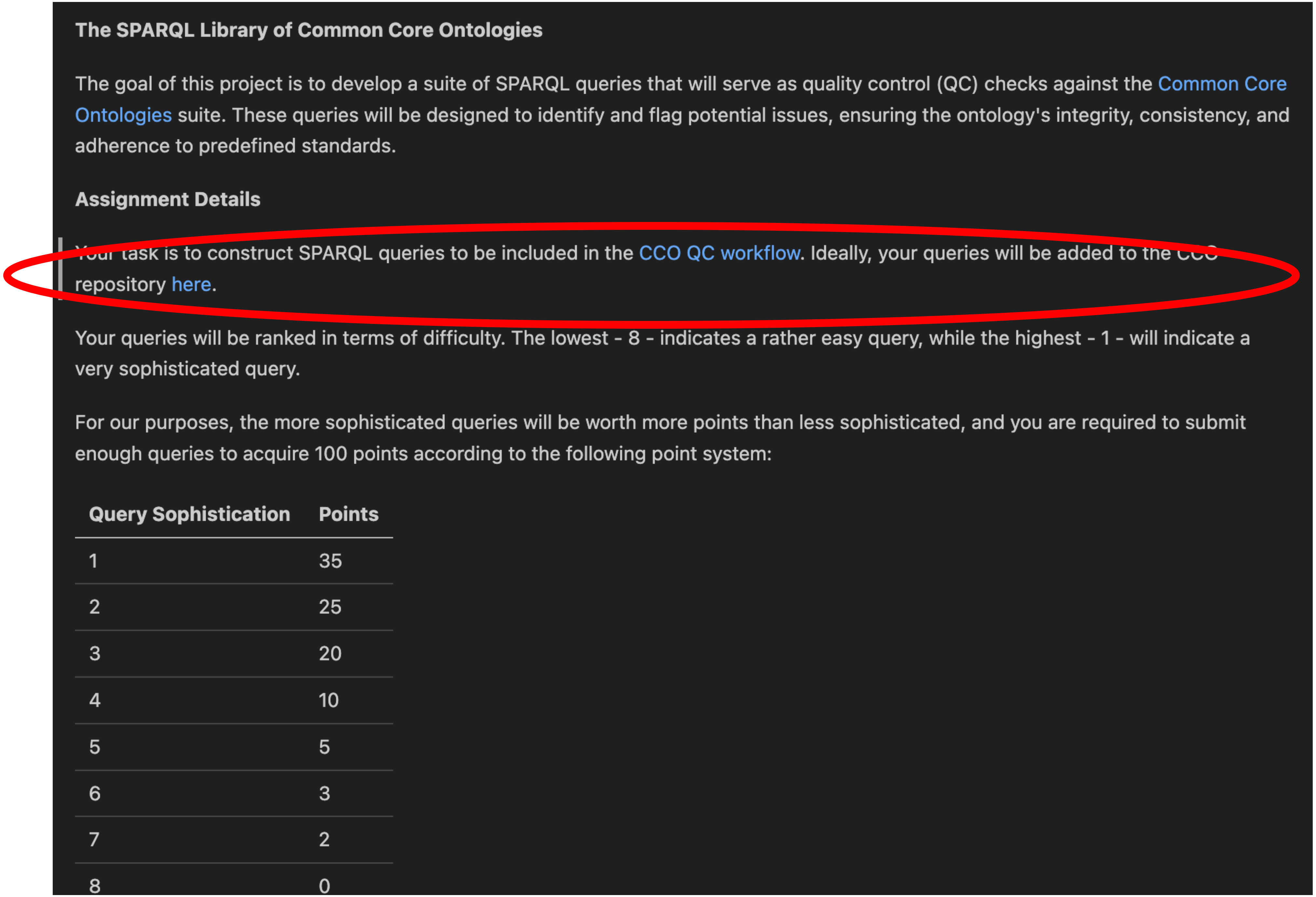
Assignment Details

> Your task is to construct SPARQL queries to be included in the CCO QC workflow. Ideally, your queries will be added to the CCO repository here.

Your queries will be ranked in terms of difficulty. The lowest - 8 - indicates a rather easy query, while the highest - 1 - will indicate a very sophisticated query.

For our purposes, the more sophisticated queries will be worth more points than less sophisticated, and you are required to submit enough queries to acquire 100 points according to the following point system:

Query Sophistication	Points
1	35
2	25
3	20
4	10
5	5
6	3
7	2
8	0



The SPARQL Library of Common Core Ontologies

The goal of this project is to develop a suite of SPARQL queries that will serve as quality control (QC) checks against the Common Core Ontologies suite. These queries will be designed to identify and flag potential issues, ensuring the ontology's integrity, consistency, and adherence to predefined standards.

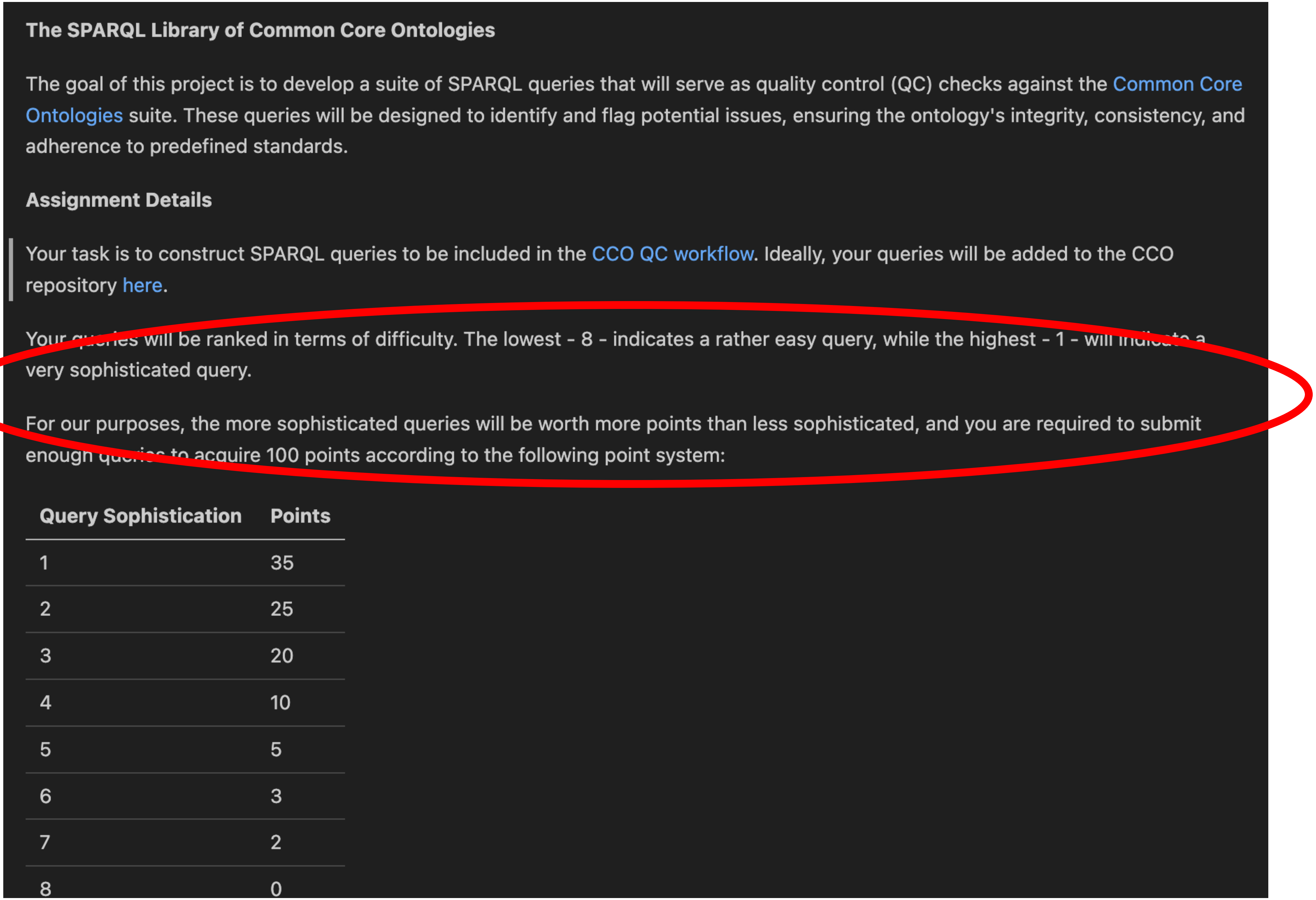
Assignment Details

> Your task is to construct SPARQL queries to be included in the CCO QC workflow. Ideally, your queries will be added to the CCO repository here.

Your queries will be ranked in terms of difficulty. The lowest - 8 - indicates a rather easy query, while the highest - 1 - will indicate a very sophisticated query.

For our purposes, the more sophisticated queries will be worth more points than less sophisticated, and you are required to submit enough queries to acquire 100 points according to the following point system:

Query Sophistication	Points
1	35
2	25
3	20
4	10
5	5
6	3
7	2
8	0



The SPARQL Library of Common Core Ontologies

The goal of this project is to develop a suite of SPARQL queries that will serve as quality control (QC) checks against the Common Core Ontologies suite. These queries will be designed to identify and flag potential issues, ensuring the ontology's integrity, consistency, and adherence to predefined standards.

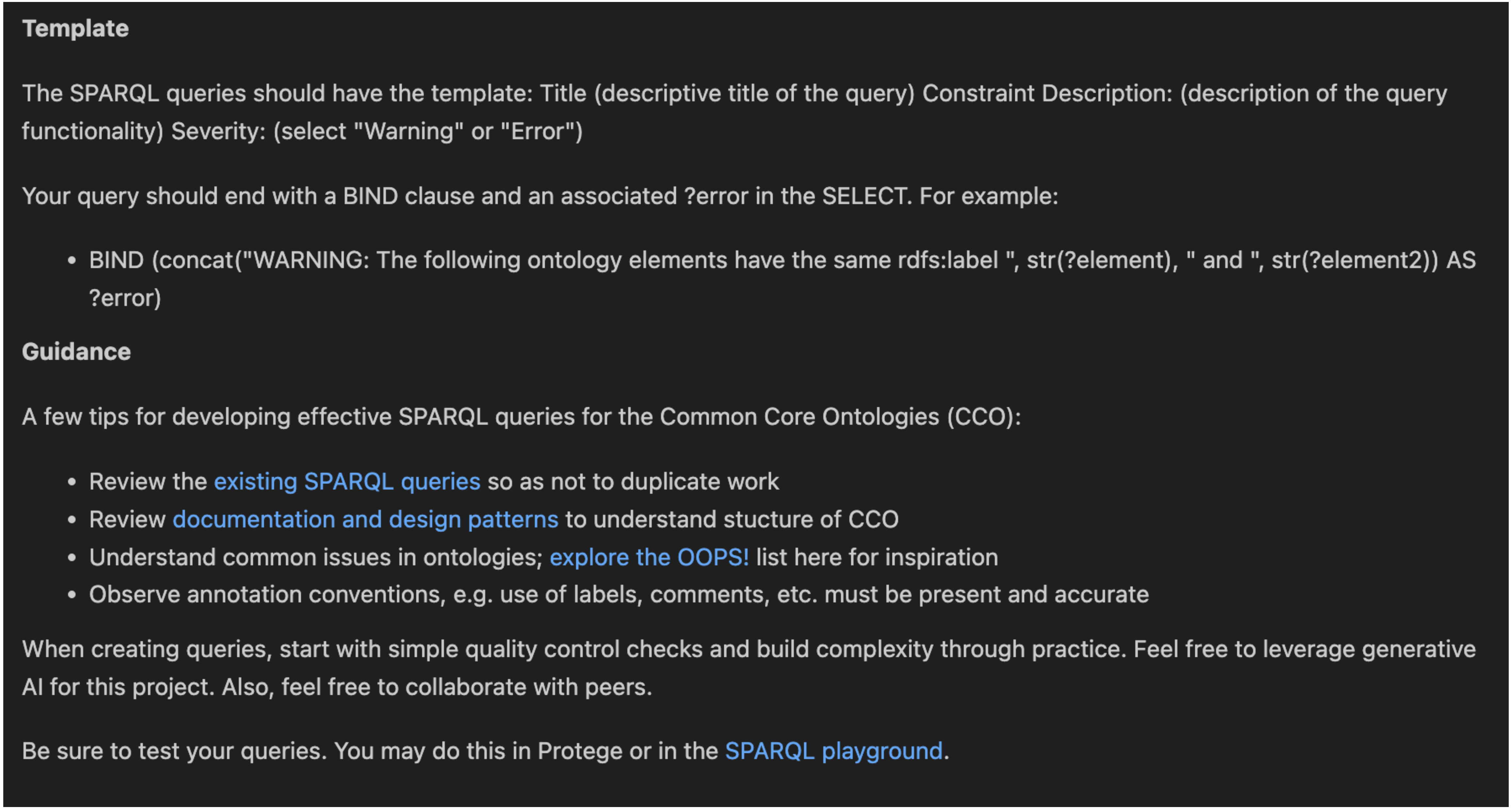
Assignment Details

> Your task is to construct SPARQL queries to be included in the CCO QC workflow. Ideally, your queries will be added to the CCO repository here.

Your queries will be ranked in terms of difficulty. The lowest - 8 - indicates a rather easy query, while the highest - 1 - will indicate a very sophisticated query.

For our purposes, the more sophisticated queries will be worth more points than less sophisticated, and you are required to submit enough queries to acquire 100 points according to the following point system:

Query Sophistication	Points
1	35
2	25
3	20
4	10
5	5
6	3
7	2
8	0



Template

The SPARQL queries should have the template: Title (descriptive title of the query) Constraint Description: (description of the query functionality) Severity: (select "Warning" or "Error")

Your query should end with a BIND clause and an associated ?error in the SELECT. For example:

- BIND (concat("WARNING: The following ontology elements have the same rdfs:label ", str(?element), " and ", str(?element2)) AS ?error)

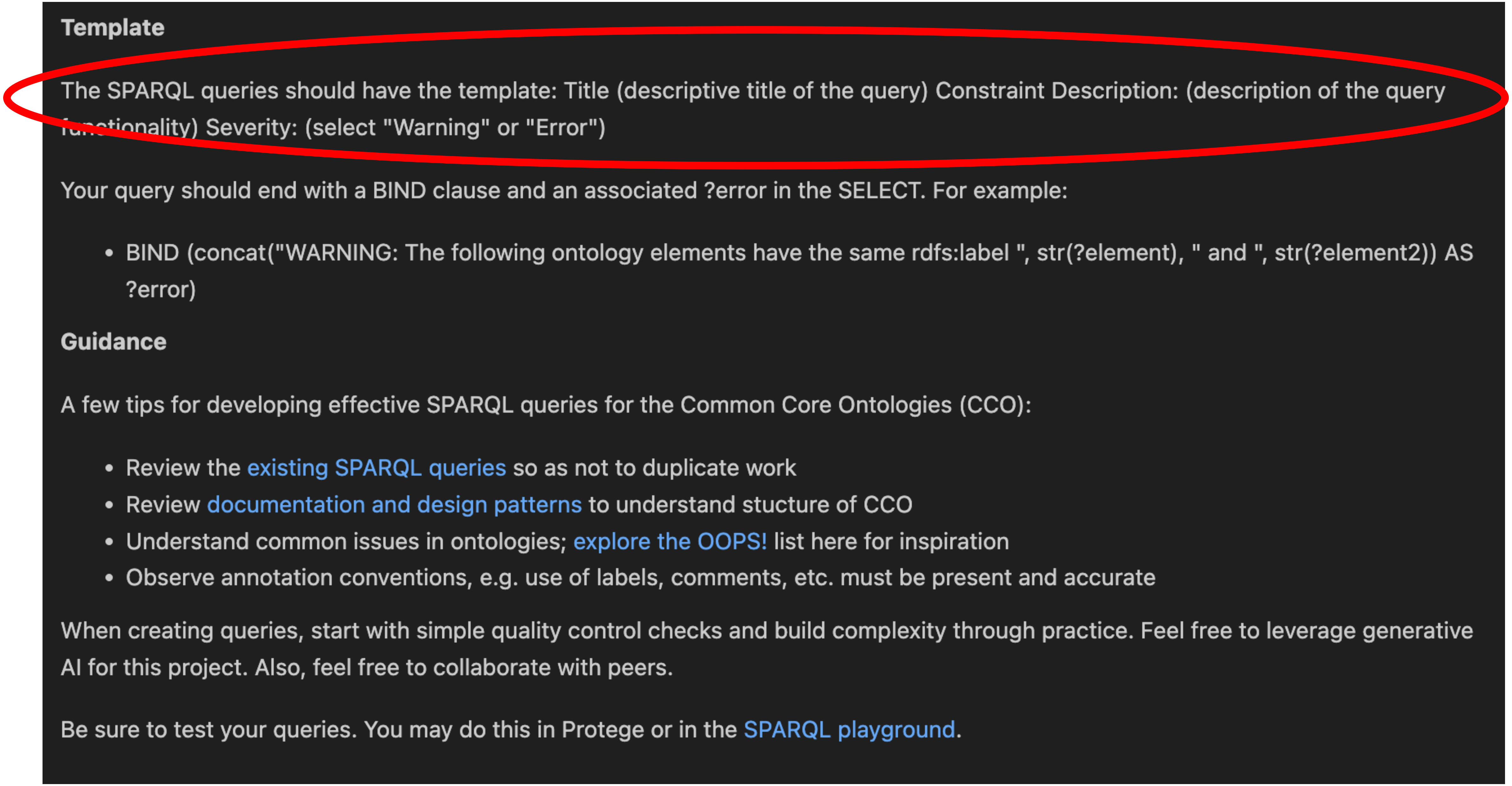
Guidance

A few tips for developing effective SPARQL queries for the Common Core Ontologies (CCO):

- Review the existing SPARQL queries so as not to duplicate work
- Review documentation and design patterns to understand stucture of CCO
- Understand common issues in ontologies; explore the OOPS! list here for inspiration
- Observe annotation conventions, e.g. use of labels, comments, etc. must be present and accurate

When creating queries, start with simple quality control checks and build complexity through practice. Feel free to leverage generative AI for this project. Also, feel free to collaborate with peers.

Be sure to test your queries. You may do this in Protege or in the SPARQL playground.



Template

The SPARQL queries should have the template: Title (descriptive title of the query) Constraint Description: (description of the query functionality) Severity: (select "Warning" or "Error")

Your query should end with a BIND clause and an associated ?error in the SELECT. For example:

- BIND (concat("WARNING: The following ontology elements have the same rdfs:label ", str(?element), " and ", str(?element2)) AS ?error)

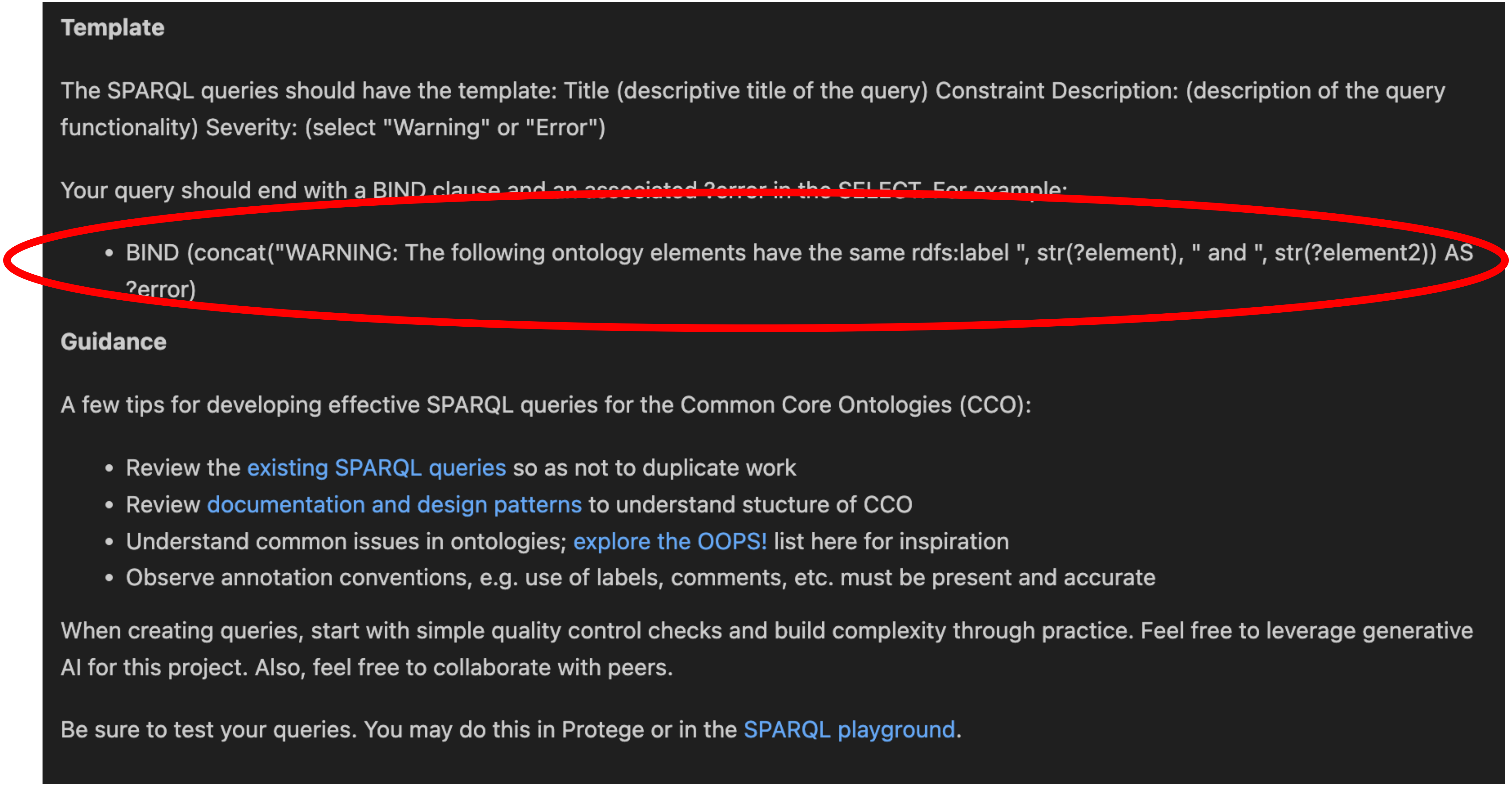
Guidance

A few tips for developing effective SPARQL queries for the Common Core Ontologies (CCO):

- Review the existing SPARQL queries so as not to duplicate work
- Review documentation and design patterns to understand stucture of CCO
- Understand common issues in ontologies; explore the OOPS! list here for inspiration
- Observe annotation conventions, e.g. use of labels, comments, etc. must be present and accurate

When creating queries, start with simple quality control checks and build complexity through practice. Feel free to leverage generative AI for this project. Also, feel free to collaborate with peers.

Be sure to test your queries. You may do this in Protege or in the SPARQL playground.



Template

The SPARQL queries should have the template: Title (descriptive title of the query) Constraint Description: (description of the query functionality) Severity: (select "Warning" or "Error")

Your query should end with a BIND clause and an associated ?error in the SELECT. For example:

- BIND (concat("WARNING: The following ontology elements have the same rdfs:label ", str(?element), " and ", str(?element2)) AS ?error)

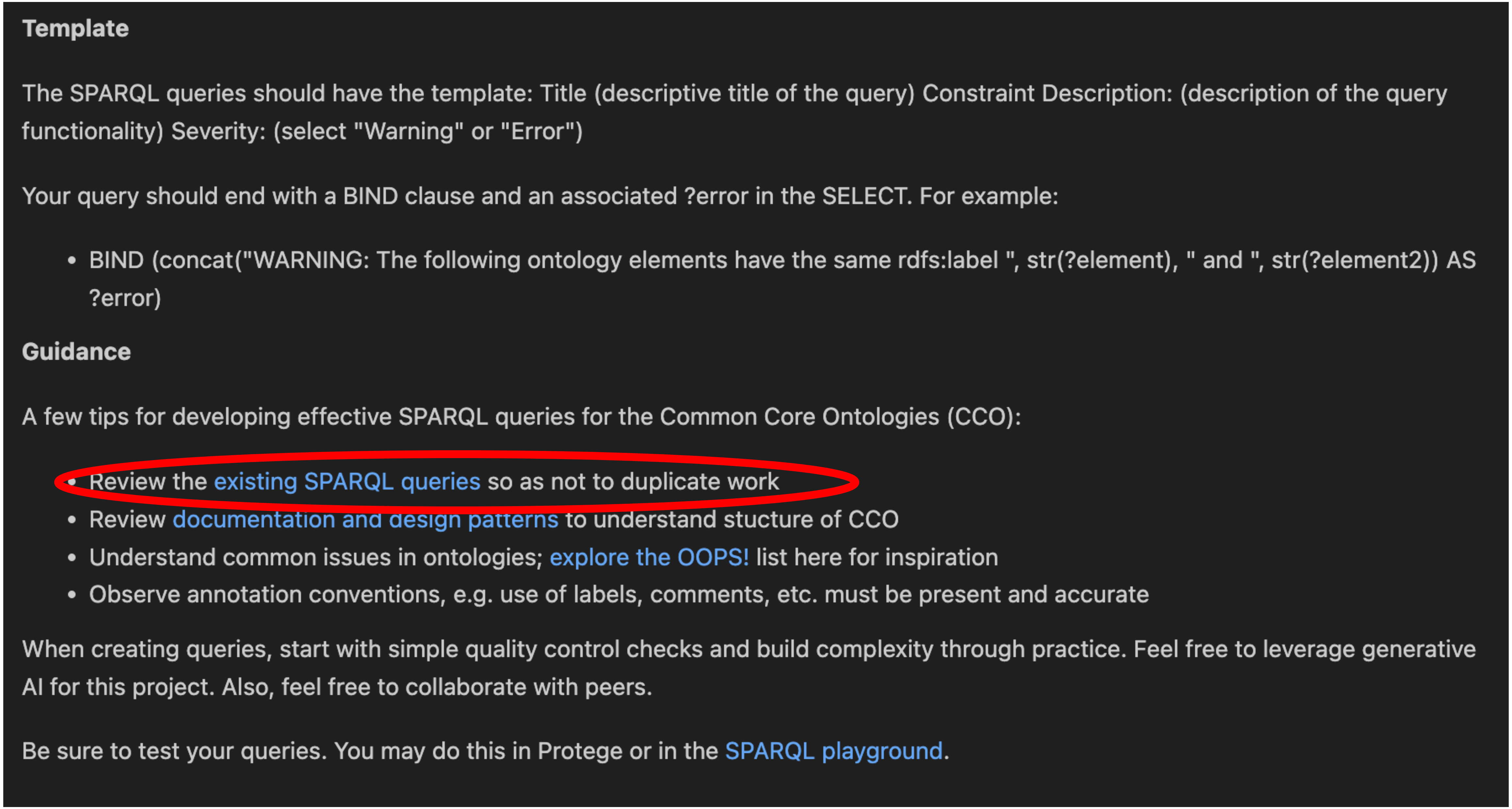
Guidance

A few tips for developing effective SPARQL queries for the Common Core Ontologies (CCO):

- Review the existing SPARQL queries so as not to duplicate work
- Review documentation and design patterns to understand stucture of CCO
- Understand common issues in ontologies; explore the OOPS! list here for inspiration
- Observe annotation conventions, e.g. use of labels, comments, etc. must be present and accurate

When creating queries, start with simple quality control checks and build complexity through practice. Feel free to leverage generative AI for this project. Also, feel free to collaborate with peers.

Be sure to test your queries. You may do this in Protege or in the SPARQL playground.



Template

The SPARQL queries should have the template: Title (descriptive title of the query) Constraint Description: (description of the query functionality) Severity: (select "Warning" or "Error")

Your query should end with a BIND clause and an associated ?error in the SELECT. For example:

- BIND (concat("WARNING: The following ontology elements have the same rdfs:label ", str(?element), " and ", str(?element2)) AS ?error)

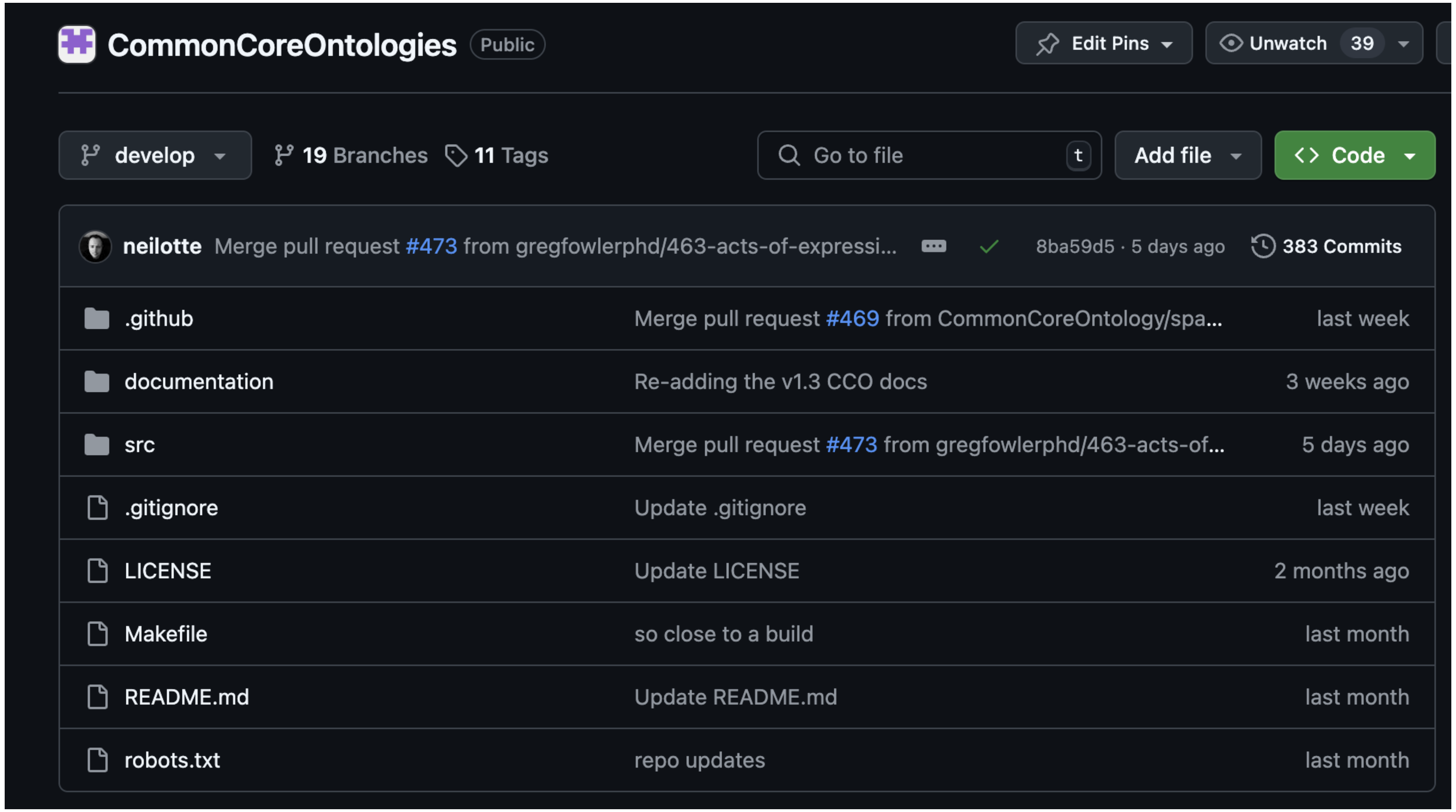
Guidance

A few tips for developing effective SPARQL queries for the Common Core Ontologies (CCO):

- Review the existing SPARQL queries so as not to duplicate work
- Review documentation and design patterns to understand stucture of CCO
- Understand common issues in ontologies; explore the OOPS! list here for inspiration
- Observe annotation conventions, e.g. use of labels, comments, etc. must be present and accurate

When creating queries, start with simple quality control checks and build complexity through practice. Feel free to leverage generative AI for this project. Also, feel free to collaborate with peers.

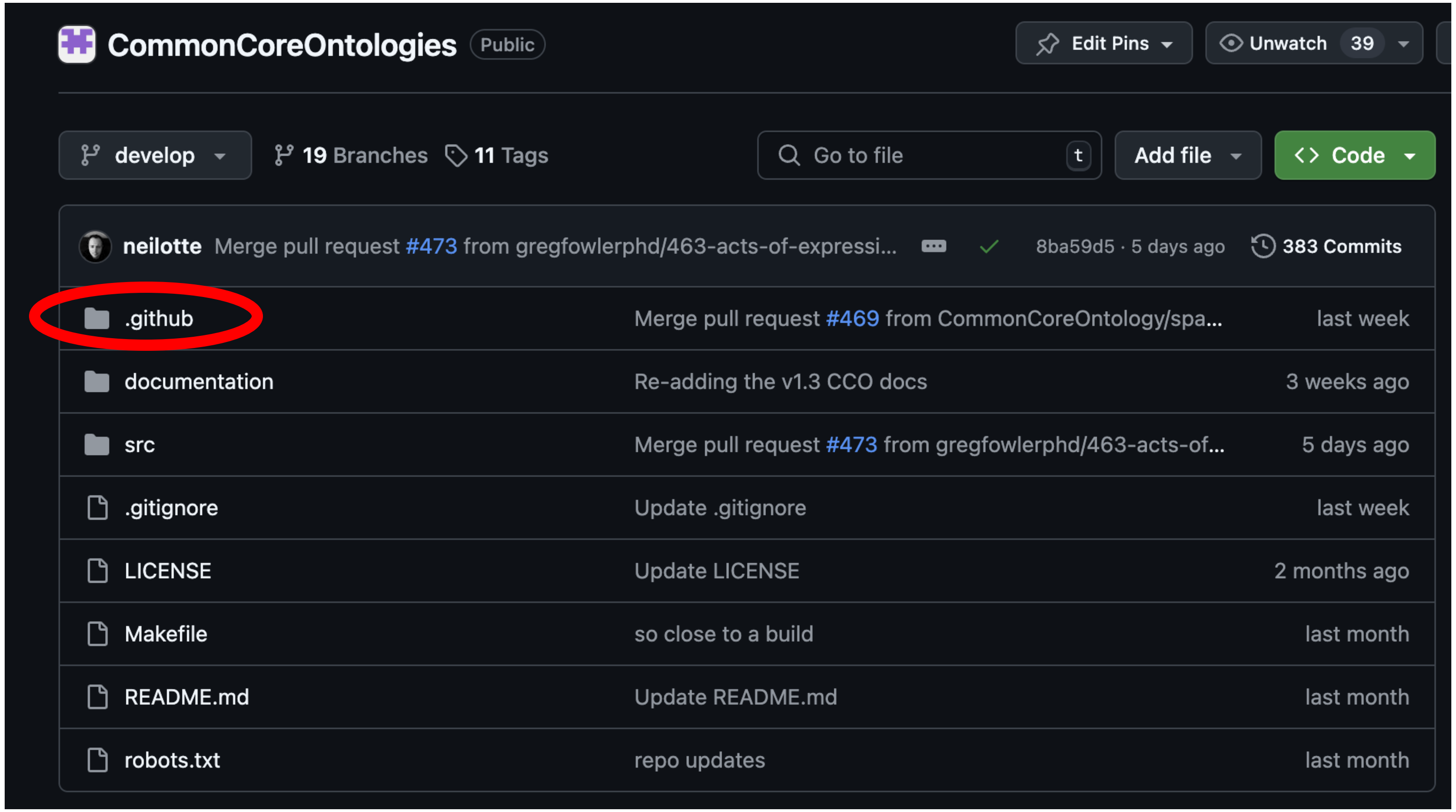
Be sure to test your queries. You may do this in Protege or in the SPARQL playground.



CommonCoreOntologies Public

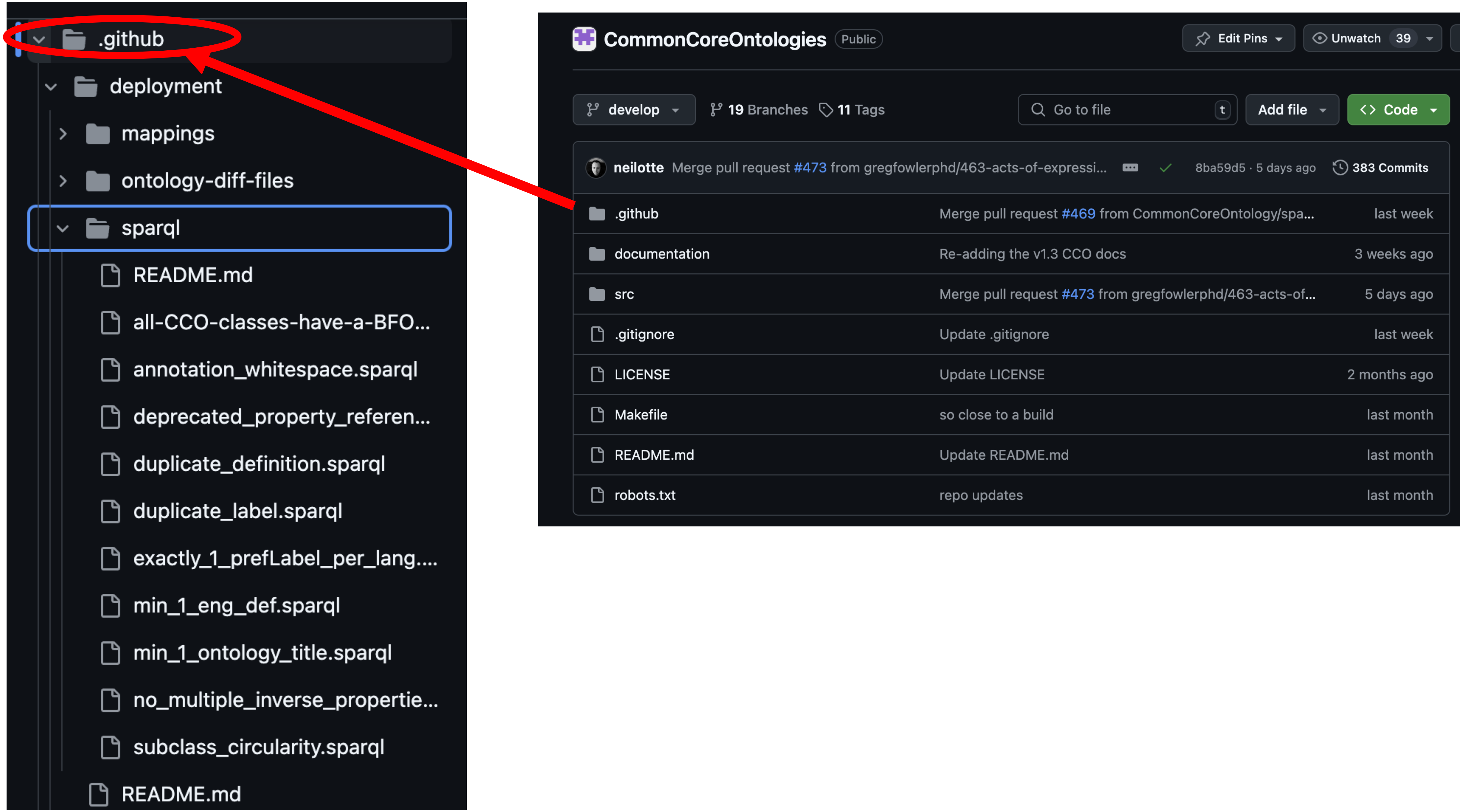
neilotte Merge pull request #473 from gregfowlerphd/463-acts-of-expressi... 8ba59d5 · 5 days ago 383 Commits

.github	Merge pull request #469 from CommonCoreOntology/spa...	last week
documentation	Re-adding the v1.3 CCO docs	3 weeks ago
src	Merge pull request #473 from gregfowlerphd/463-acts-of...	5 days ago
.gitignore	Update .gitignore	last week
LICENSE	Update LICENSE	2 months ago
Makefile	so close to a build	last month
README.md	Update README.md	last month
robots.txt	repo updates	last month



neilotte Merge pull request #473 from gregfowlerphd/463-acts-of-expressi... 8ba59d5 · 5 days ago 383 Commits

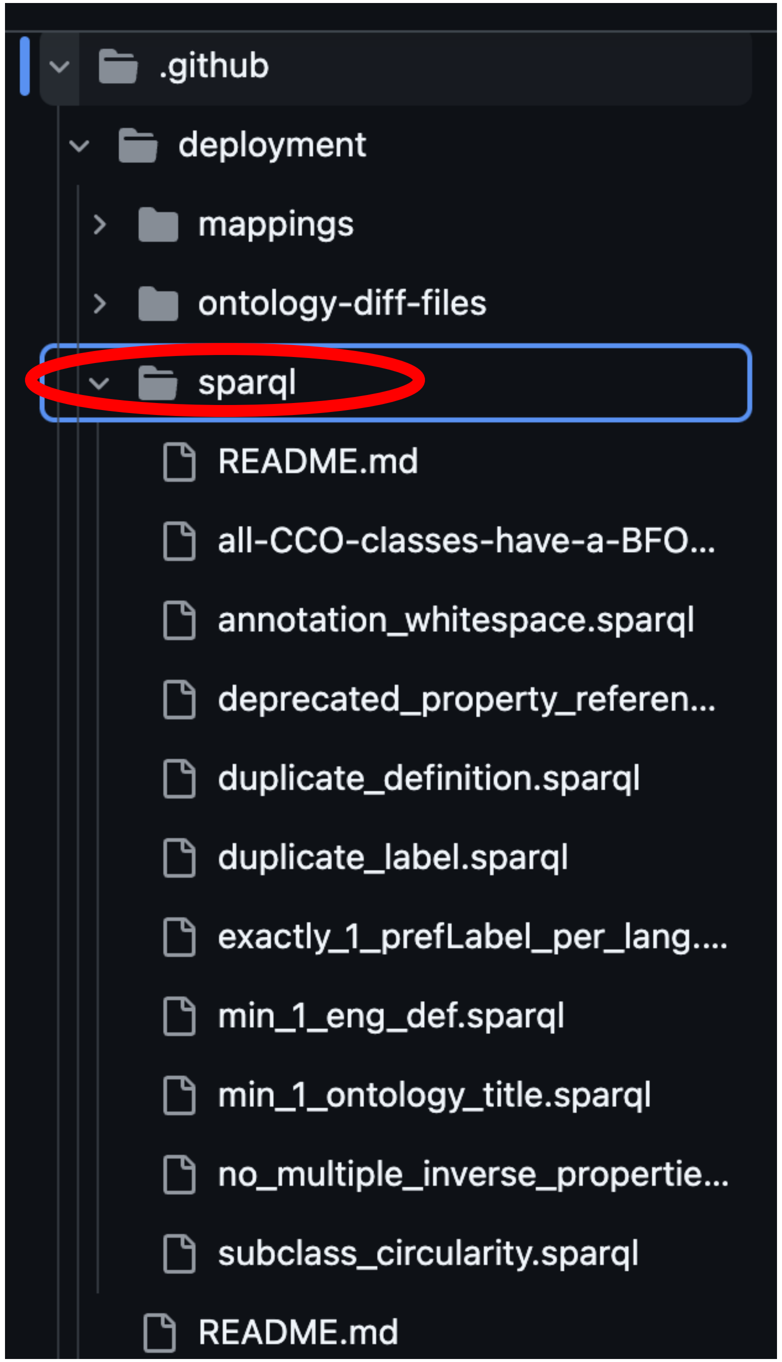
.github	Merge pull request #469 from CommonCoreOntology/spa...	last week
documentation	Re-adding the v1.3 CCO docs	3 weeks ago
src	Merge pull request #473 from gregfowlerphd/463-acts-of...	5 days ago
.gitignore	Update .gitignore	last week
LICENSE	Update LICENSE	2 months ago
Makefile	so close to a build	last month
README.md	Update README.md	last month
robots.txt	repo updates	last month



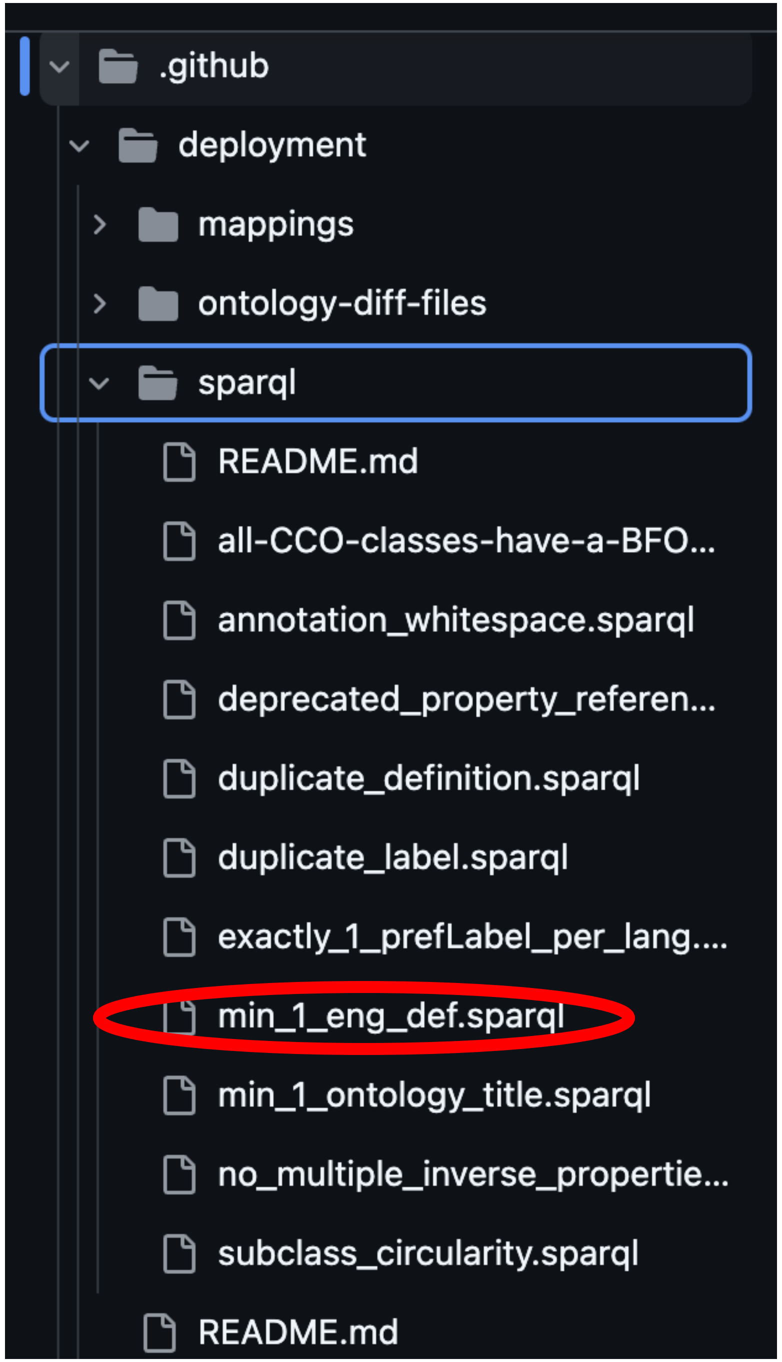
```
.github
  deployment
    mappings
    ontology-diff-files
    sparql
      README.md
      all-CCO-classes-have-a-BFO...
      annotation_whitespace.sparql
      deprecated_property_referen...
      duplicate_definition.sparql
      duplicate_label.sparql
      exactly_1_prefLabel_per_lang....
      min_1_eng_def.sparql
      min_1_ontology_title.sparql
      no_multiple_inverse_propertie...
      subclass_circularity.sparql
    README.md
```

neilotte Merge pull request #473 from gregfowlerphd/463-acts-of-expressi... 8ba59d5 · 5 days ago 383 Commits

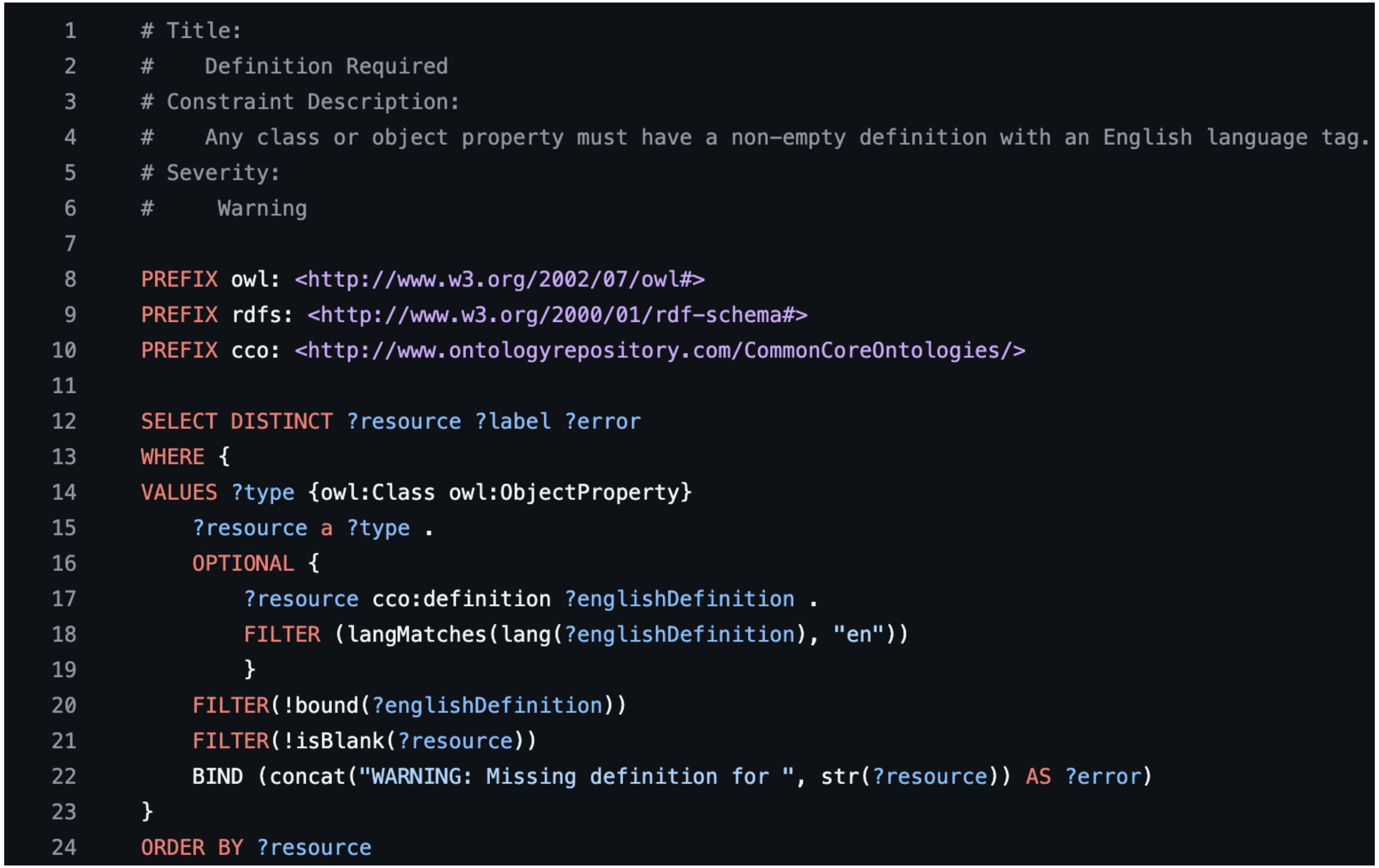
.github	Merge pull request #469 from CommonCoreOntology/spa...	last week
documentation	Re-adding the v1.3 CCO docs	3 weeks ago
src	Merge pull request #473 from gregfowlerphd/463-acts-of...	5 days ago
.gitignore	Update .gitignore	last week
LICENSE	Update LICENSE	2 months ago
Makefile	so close to a build	last month
README.md	Update README.md	last month
robots.txt	repo updates	last month



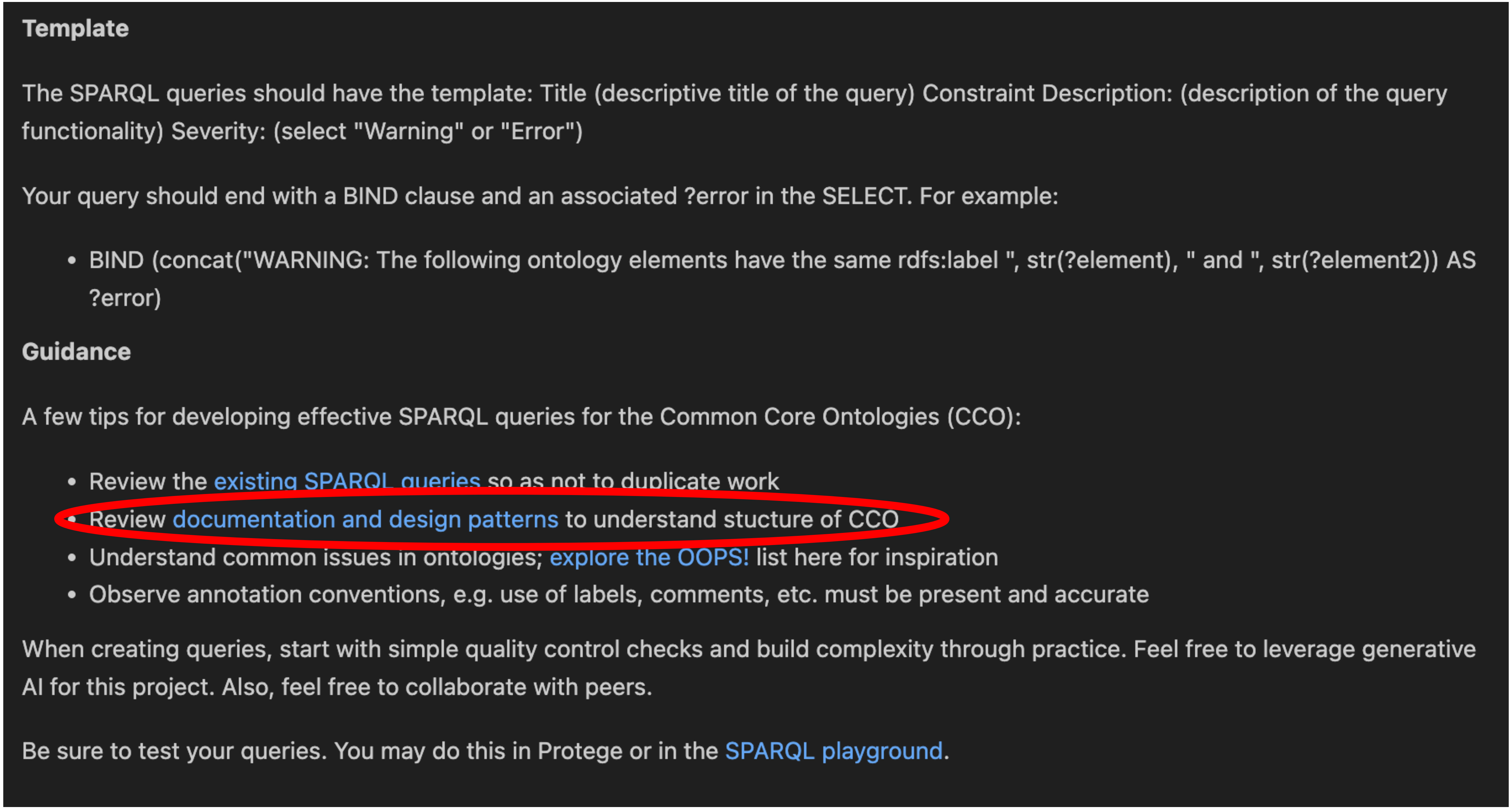
```
.github
  deployment
    mappings
    ontology-diff-files
    sparql
      README.md
      all-CCO-classes-have-a-BFO...
      annotation_whitespace.sparql
      deprecated_property_referen...
      duplicate_definition.sparql
      duplicate_label.sparql
      exactly_1_prefLabel_per_lang....
      min_1_eng_def.sparql
      min_1_ontology_title.sparql
      no_multiple_inverse_propertie...
      subclass_circularity.sparql
    README.md
```



```
.github
  deployment
    mappings
    ontology-diff-files
    sparql
      README.md
      all-CCO-classes-have-a-BFO...
      annotation_whitespace.sparql
      deprecated_property_referen...
      duplicate_definition.sparql
      duplicate_label.sparql
      exactly_1_prefLabel_per_lang....
      min_1_eng_def.sparql
      min_1_ontology_title.sparql
      no_multiple_inverse_propertie...
      subclass_circularity.sparql
    README.md
```



```
# Title:
#    Definition Required
# Constraint Description:
#    Any class or object property must have a non-empty definition with an English language tag.
# Severity:
#     Warning

PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX cco: <http://www.ontologyrepository.com/CommonCoreOntologies/>

SELECT DISTINCT ?resource ?label ?error
WHERE {
VALUES ?type {owl:Class owl:ObjectProperty}
    ?resource a ?type .
    OPTIONAL {
        ?resource cco:definition ?englishDefinition .
        FILTER (langMatches(lang(?englishDefinition), "en"))
        }
    FILTER(!bound(?englishDefinition))
    FILTER(!isBlank(?resource))
    BIND (concat("WARNING: Missing definition for ", str(?resource)) AS ?error)
}
ORDER BY ?resource
```



Template

The SPARQL queries should have the template: Title (descriptive title of the query) Constraint Description: (description of the query functionality) Severity: (select "Warning" or "Error")

Your query should end with a BIND clause and an associated ?error in the SELECT. For example:

- BIND (concat("WARNING: The following ontology elements have the same rdfs:label ", str(?element), " and ", str(?element2)) AS ?error)

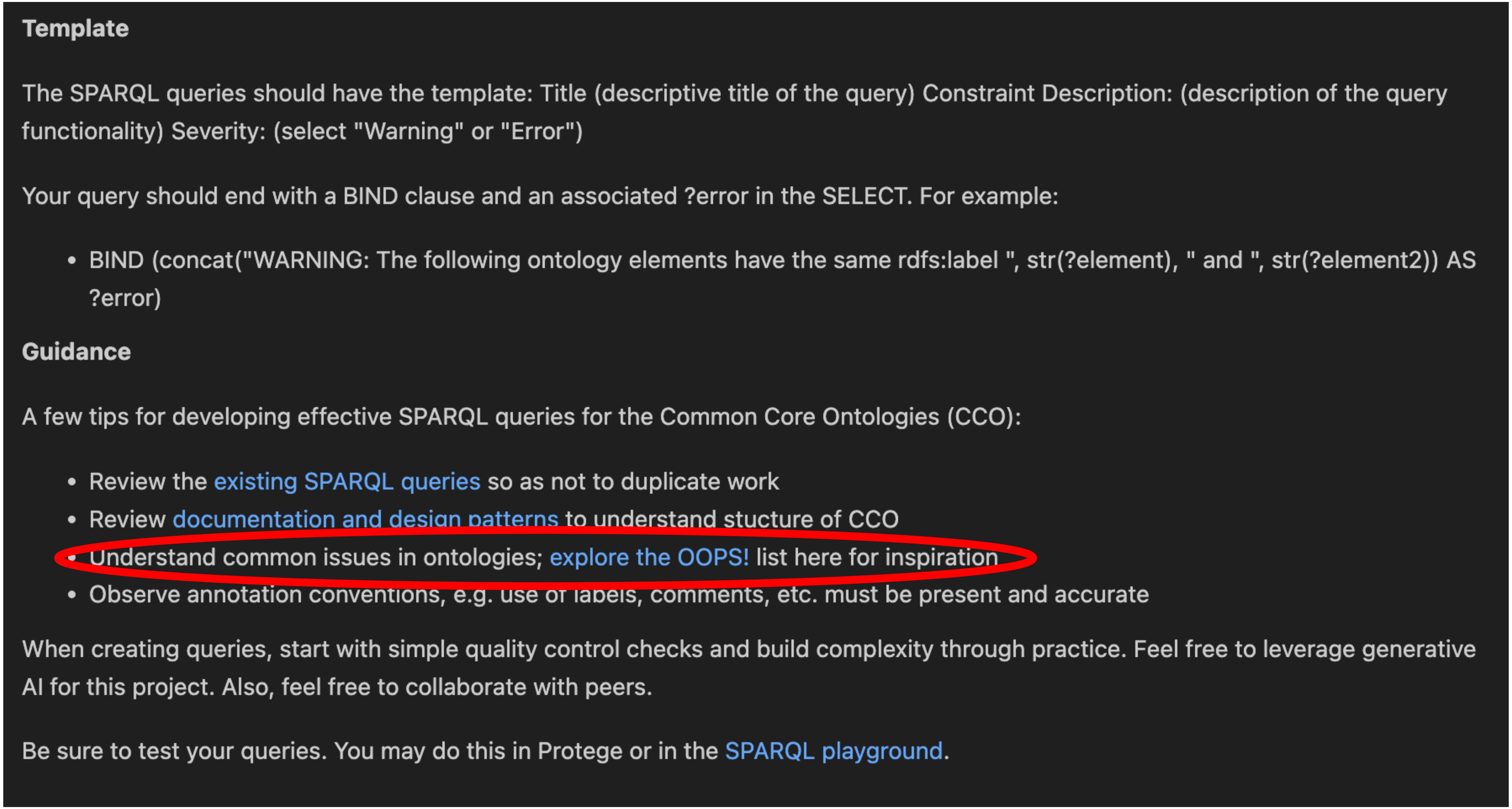
Guidance

A few tips for developing effective SPARQL queries for the Common Core Ontologies (CCO):

- Review the existing SPARQL queries so as not to duplicate work
- Review documentation and design patterns to understand stucture of CCO
- Understand common issues in ontologies; explore the OOPS! list here for inspiration
- Observe annotation conventions, e.g. use of labels, comments, etc. must be present and accurate

When creating queries, start with simple quality control checks and build complexity through practice. Feel free to leverage generative AI for this project. Also, feel free to collaborate with peers.

Be sure to test your queries. You may do this in Protege or in the SPARQL playground.



Template

The SPARQL queries should have the template: Title (descriptive title of the query) Constraint Description: (description of the query functionality) Severity: (select "Warning" or "Error")

Your query should end with a BIND clause and an associated ?error in the SELECT. For example:

- BIND (concat("WARNING: The following ontology elements have the same rdfs:label ", str(?element), " and ", str(?element2)) AS ?error)

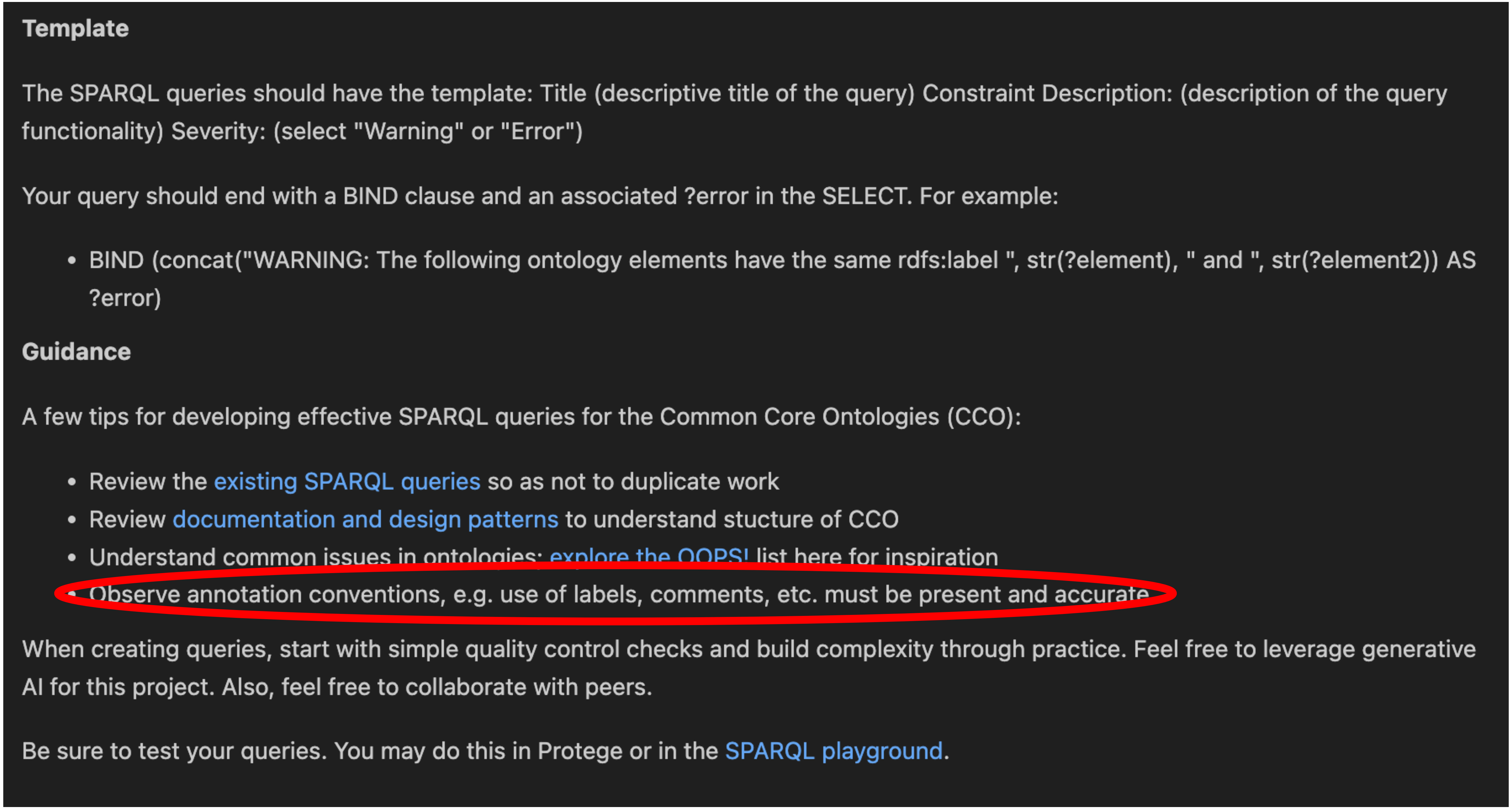
Guidance

A few tips for developing effective SPARQL queries for the Common Core Ontologies (CCO):

- Review the existing SPARQL queries so as not to duplicate work
- Review documentation and design patterns to understand stucture of CCO
- Understand common issues in ontologies; explore the OOPS! list here for inspiration
- Observe annotation conventions, e.g. use of labels, comments, etc. must be present and accurate

When creating queries, start with simple quality control checks and build complexity through practice. Feel free to leverage generative AI for this project. Also, feel free to collaborate with peers.

Be sure to test your queries. You may do this in Protege or in the SPARQL playground.



Template

The SPARQL queries should have the template: Title (descriptive title of the query) Constraint Description: (description of the query functionality) Severity: (select "Warning" or "Error")

Your query should end with a BIND clause and an associated ?error in the SELECT. For example:

- BIND (concat("WARNING: The following ontology elements have the same rdfs:label ", str(?element), " and ", str(?element2)) AS ?error)

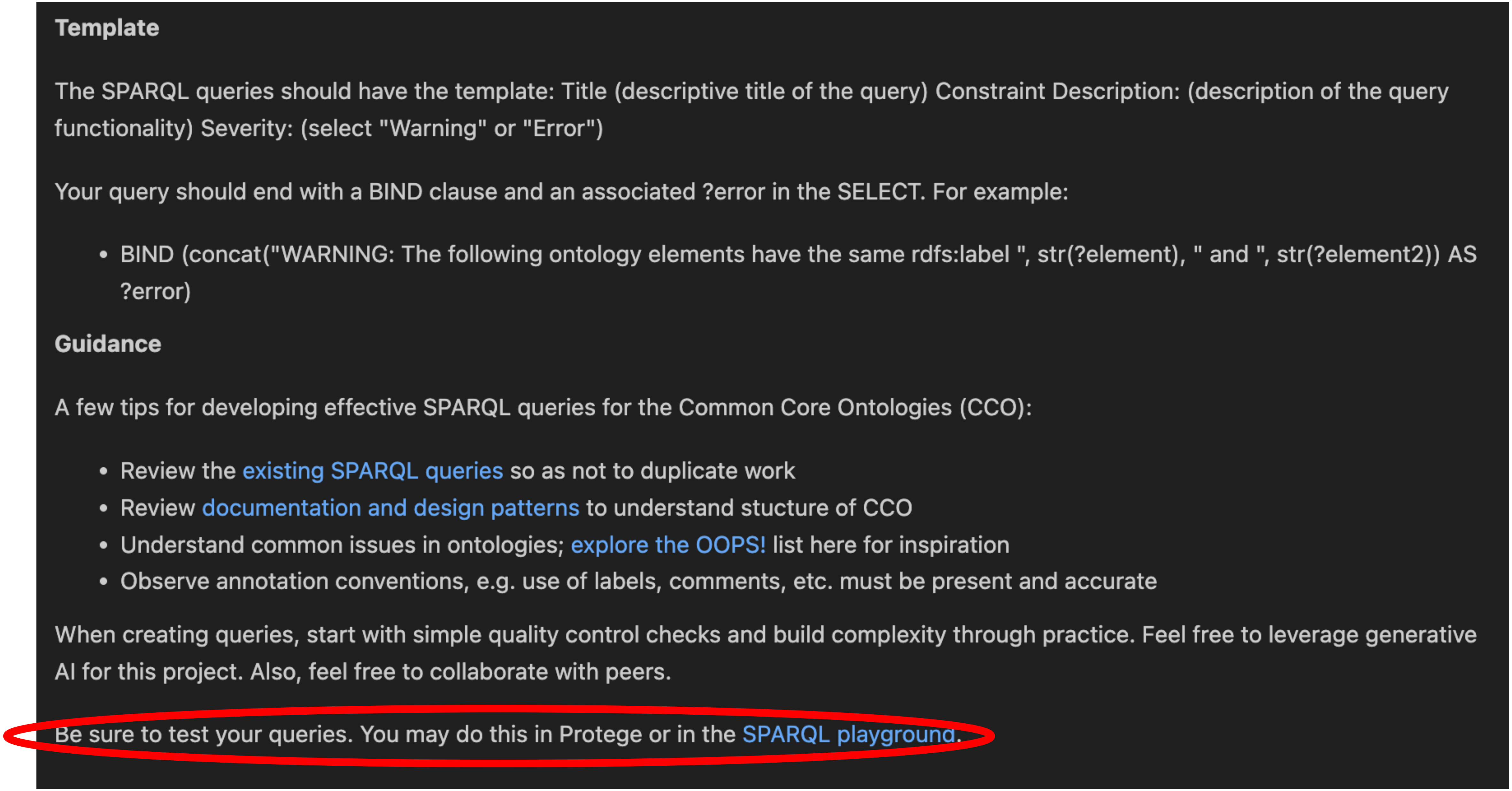
Guidance

A few tips for developing effective SPARQL queries for the Common Core Ontologies (CCO):

- Review the existing SPARQL queries so as not to duplicate work
- Review documentation and design patterns to understand stucture of CCO
- Understand common issues in ontologies: explore the OOPS! list here for inspiration
- Observe annotation conventions, e.g. use of labels, comments, etc. must be present and accurate

When creating queries, start with simple quality control checks and build complexity through practice. Feel free to leverage generative AI for this project. Also, feel free to collaborate with peers.

Be sure to test your queries. You may do this in Protege or in the SPARQL playground.



Template

The SPARQL queries should have the template: Title (descriptive title of the query) Constraint Description: (description of the query functionality) Severity: (select "Warning" or "Error")

Your query should end with a BIND clause and an associated ?error in the SELECT. For example:

- BIND (concat("WARNING: The following ontology elements have the same rdfs:label ", str(?element), " and ", str(?element2)) AS ?error)

Guidance

A few tips for developing effective SPARQL queries for the Common Core Ontologies (CCO):

- Review the existing SPARQL queries so as not to duplicate work
- Review documentation and design patterns to understand stucture of CCO
- Understand common issues in ontologies; explore the OOPS! list here for inspiration
- Observe annotation conventions, e.g. use of labels, comments, etc. must be present and accurate

When creating queries, start with simple quality control checks and build complexity through practice. Feel free to leverage generative AI for this project. Also, feel free to collaborate with peers.

Be sure to test your queries. You may do this in Protege or in the SPARQL playground.