BENAZIR BHUTTO SHAHEED UNIVERSITY

LYARI, KARACHI

"SEMANTIC WEB LAB MANUAL"

Submitted to:

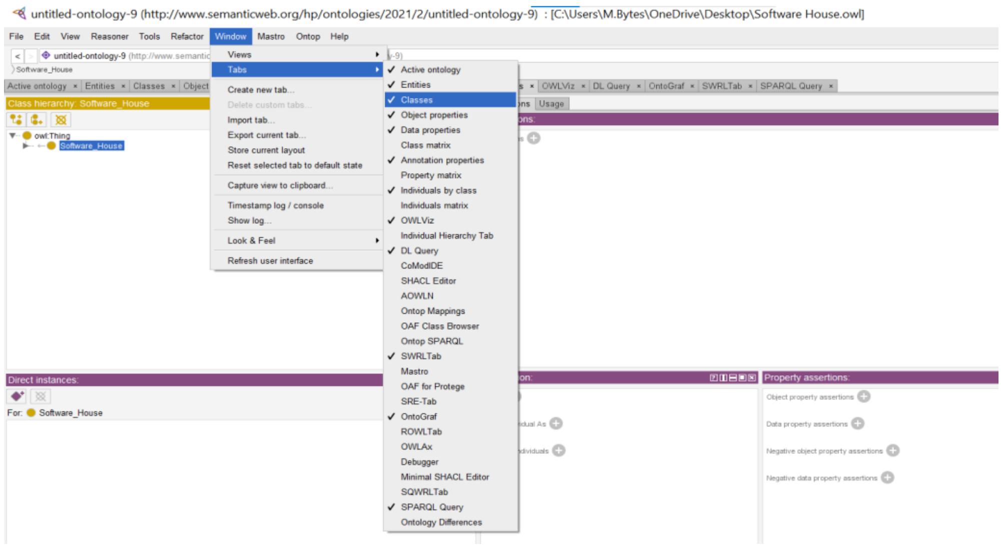
Ma'am Ambreen

Submitted by:

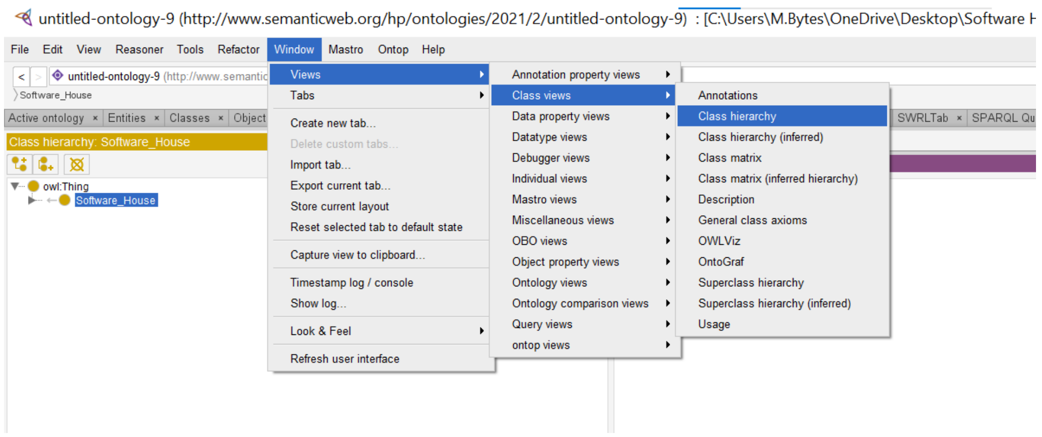
Name:	Salman Abdul Rahim
Department:	Computer Science
Semester:	7th – B
Roll #:	616

Lab 01: Classes Subclasses

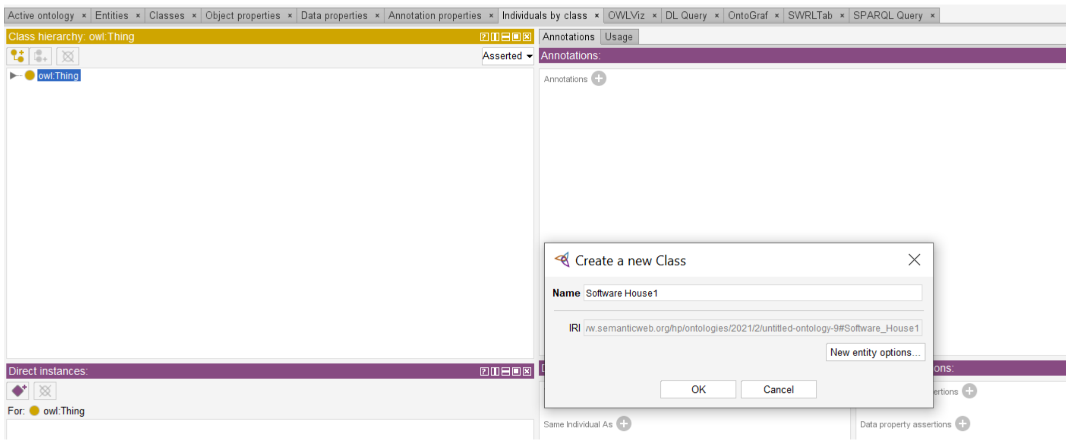
Adding Class Tab:



Adding Class Hierarchy view:



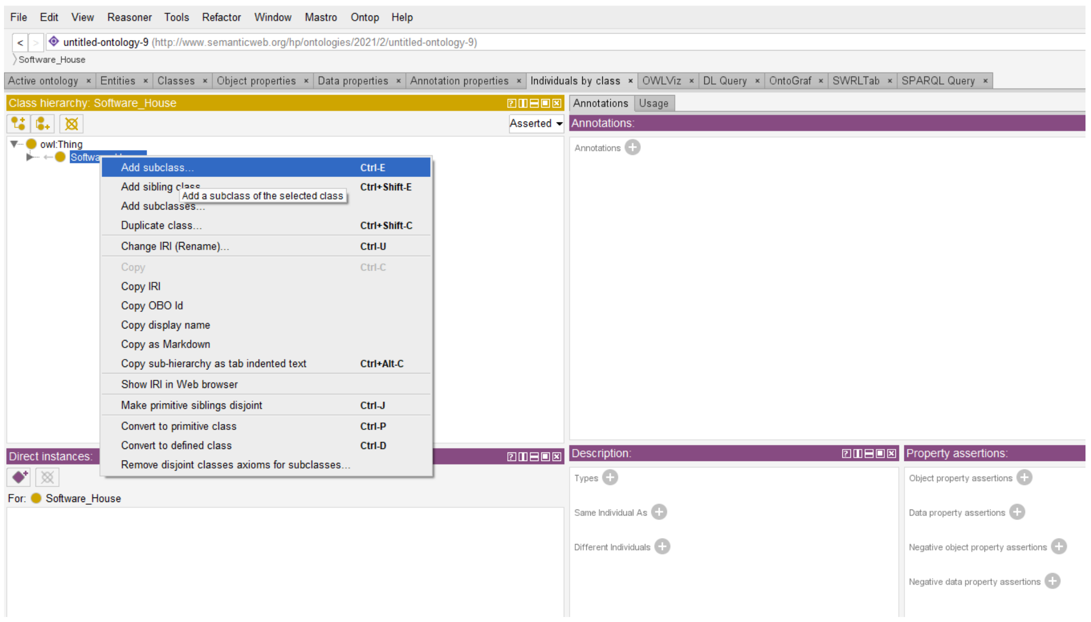
Creating a New Class:


Active ontology Entities Classes Object properties Data properties Annotation properties Individuals by class OWLViz DL Query OntoGraf SWRLTab SPARQL Query
Class hierarchy: owl:Thing
Asserted
owl:Thing
Annotations Usage
Annotations:
Direct instances:
For: owl:Thing
Same Individual As
Data property assertions
Create a new Class
New entity options...
OK
Cancel

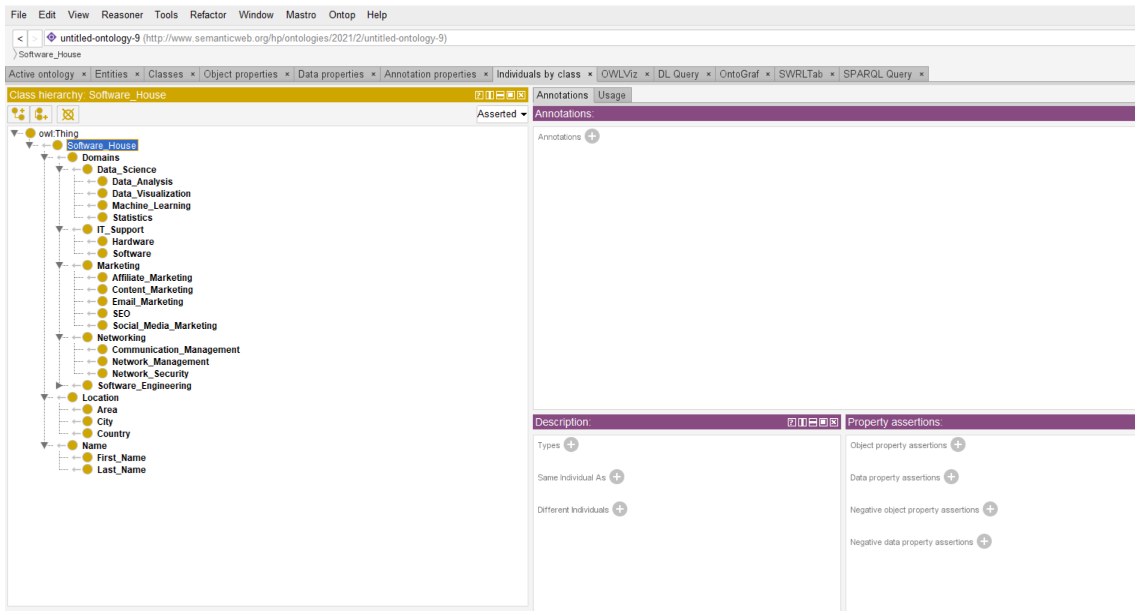
Name Software House1

IRI /w.semanticweb.org/hp/ontologies/2021/2/untitled-ontology-9#Software_House1

Creating a Subclass:


File Edit View Reasoner Tools Refactor Window Mastro Ontop Help
untitled-ontology-9 (http://www.semanticweb.org/hp/ontologies/2021/2/untitled-ontology-9)
Software_House
Active ontology Entities Classes Object properties Data properties Annotation properties Individuals by class OWLViz DL Query OntoGraf SWRLTab SPARQL Query
Class hierarchy: Software_House
Asserted
owl:Thing
Add subclass... Ctrl-E
Add sibling class Ctrl+Shift-E
Add a subclass of the selected class
Add subclasses...
Duplicate class... Ctrl+Shift-C
Change IRI (Rename)... Ctrl-U
Copy Ctrl-C
Copy IRI
Copy OBO Id
Copy display name
Copy as Markdown
Copy sub-hierarchy as tab indented text Ctrl+Alt-C
Show IRI in Web browser
Make primitive siblings disjoint Ctrl-J
Convert to primitive class Ctrl-P
Convert to defined class Ctrl-D
Remove disjoint classes axioms for subclasses...
Annotations Usage
Annotations:
Annotations
Direct instances:
For: Software_House
Description:
Types
Same Individual As
Different Individuals
Property assertions:
Object property assertions
Data property assertions
Negative object property assertions
Negative data property assertions

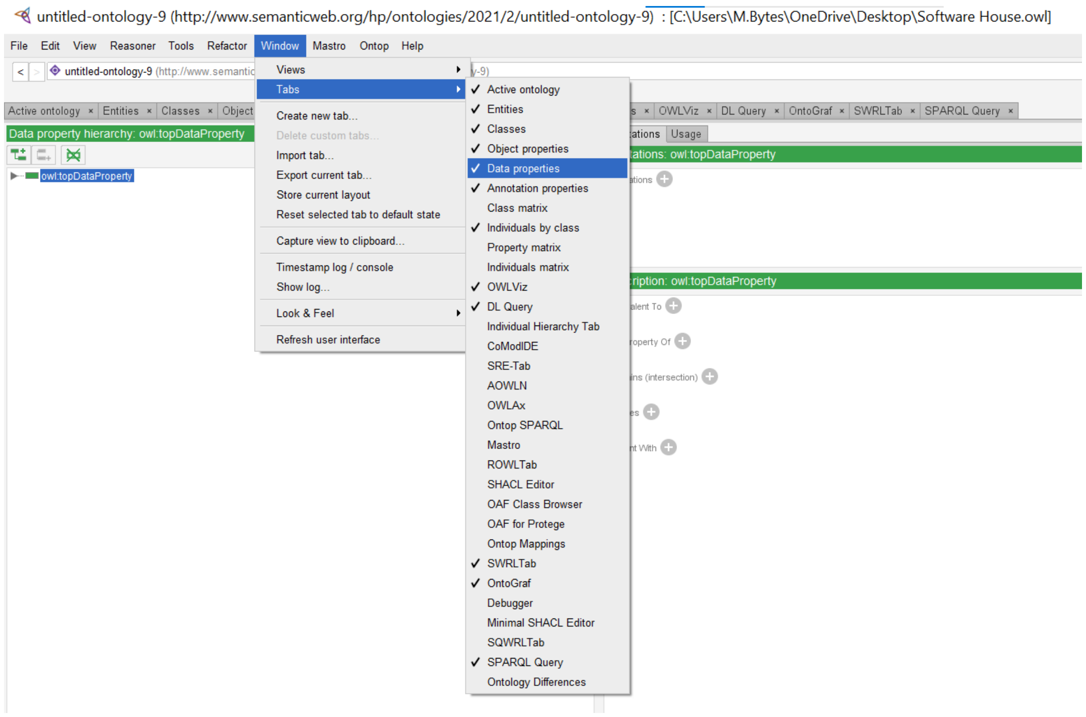
Classes and Subclasses:


File Edit View Reasoner Tools Refactor Window Mastro Ontop Help
untitled-ontology-9 (http://www.semanticweb.org/hp/ontologies/2021/2/untitled-ontology-9)
Software_House
Active ontology Entities Classes Object properties Data properties Annotation properties Individuals by class OWLViz DL Query OntoGraf SWRLTab SPARQL Query
Class hierarchy: Software_House
Asserted
Annotations Usage
Annotations:
Annotations
Description:
Types
Same Individual As
Different Individuals
Property assertions:
Object property assertions
Data property assertions
Negative object property assertions
Negative data property assertions

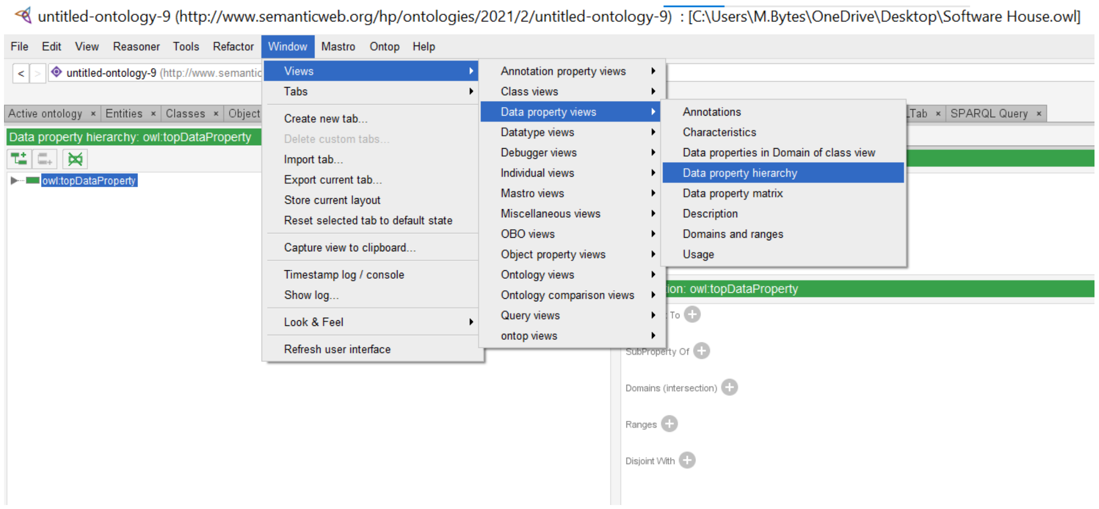
```
owl:Thing
  Software_House
    Domains
      Data_Science
        Data_Analysis
        Data_Visualization
        Machine_Learning
        Statistics
      IT_Support
        Hardware
        Software
      Marketing
        Affiliate_Marketing
        Content_Marketing
        Email_Marketing
        SEO
        Social_Media_Marketing
      Networking
        Communication_Management
        Network_Management
        Network_Security
      Software_Engineering
    Location
      Area
      City
      Country
    Name
      First_Name
      Last_Name
```

Lab 02: Data Property

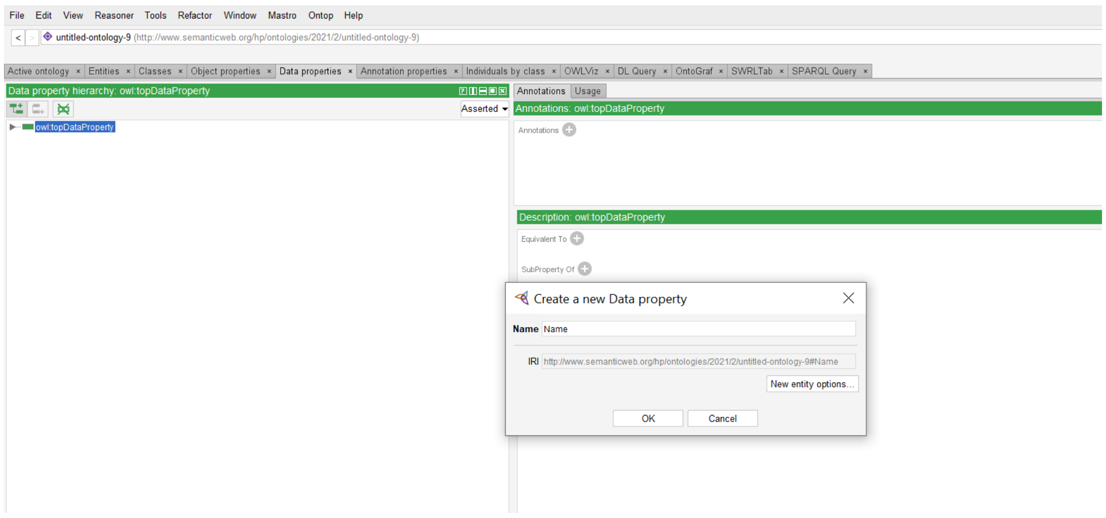
Adding Data Property Tab:



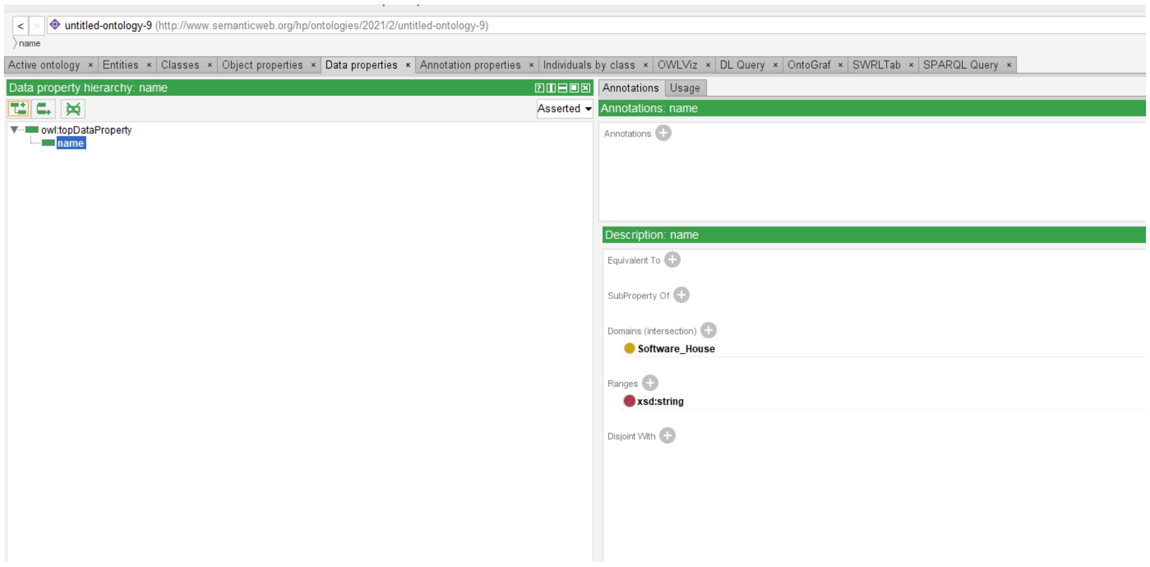
Adding Data Property Hierarchy View:


untitled-ontology-9 (http://www.semanticweb.org/hp/ontologies/2021/2/untitled-ontology-9) : [C:\Users\M.Bytes\OneDrive\Desktop\Software House.owl]
File Edit View Reasoner Tools Refactor Window Mastro Ontop Help
Views
Tabs
Annotation property views
Class views
Data property views
Datatype views
Debugger views
Individual views
Mastro views
Miscellaneous views
OBO views
Object property views
Ontology views
Ontology comparison views
Query views
ontop views
Annotations
Characteristics
Data properties in Domain of class view
Data property hierarchy
Data property matrix
Description
Domains and ranges
Usage
Data property hierarchy: owl:topDataProperty
owl:topDataProperty

Creating Data Property:

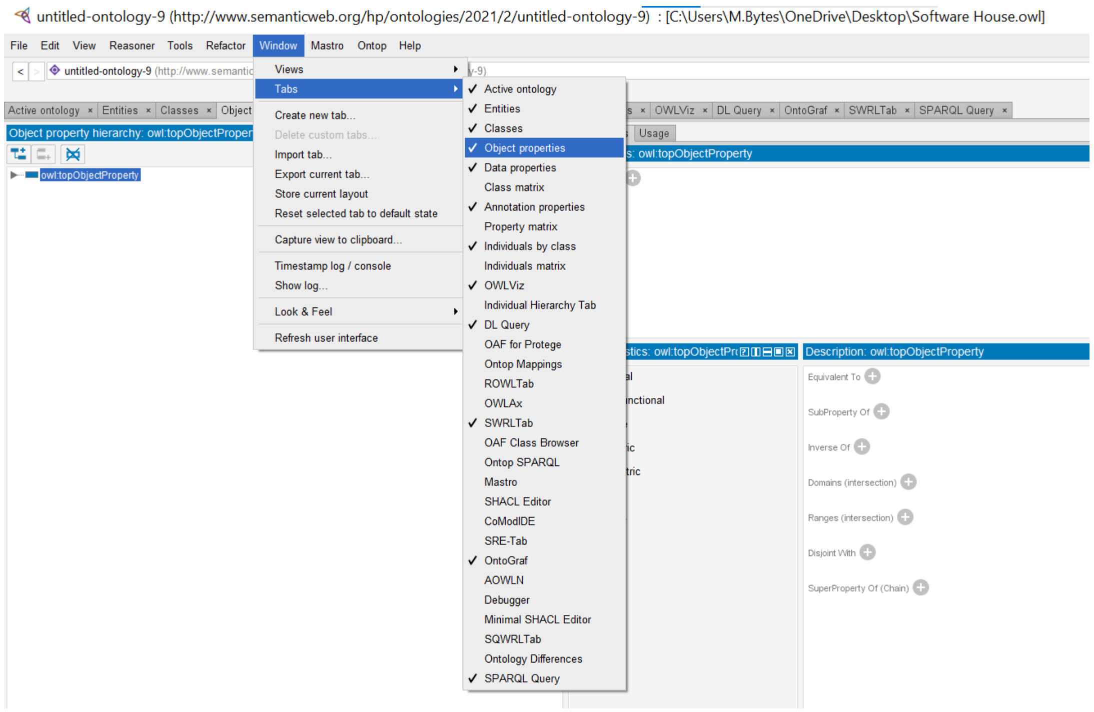

File Edit View Reasoner Tools Refactor Window Mastro Ontop Help
untitled-ontology-9 (http://www.semanticweb.org/hp/ontologies/2021/2/untitled-ontology-9)
Active ontology Entities Classes Object properties Data properties Annotation properties Individuals by class OWLViz DL Query OntoGraf SWRLTab SPARQL Query
Data property hierarchy: owl:topDataProperty
Asserted
owl:topDataProperty
Annotations Usage
Annotations: owl:topDataProperty
Annotations
Description: owl:topDataProperty
Equivalent To
SubProperty Of
Create a new Data property
Name Name
IRI http://www.semanticweb.org/hp/ontologies/2021/2/untitled-ontology-9#Name
New entity options...
OK
Cancel

Domain Range to Data Property:

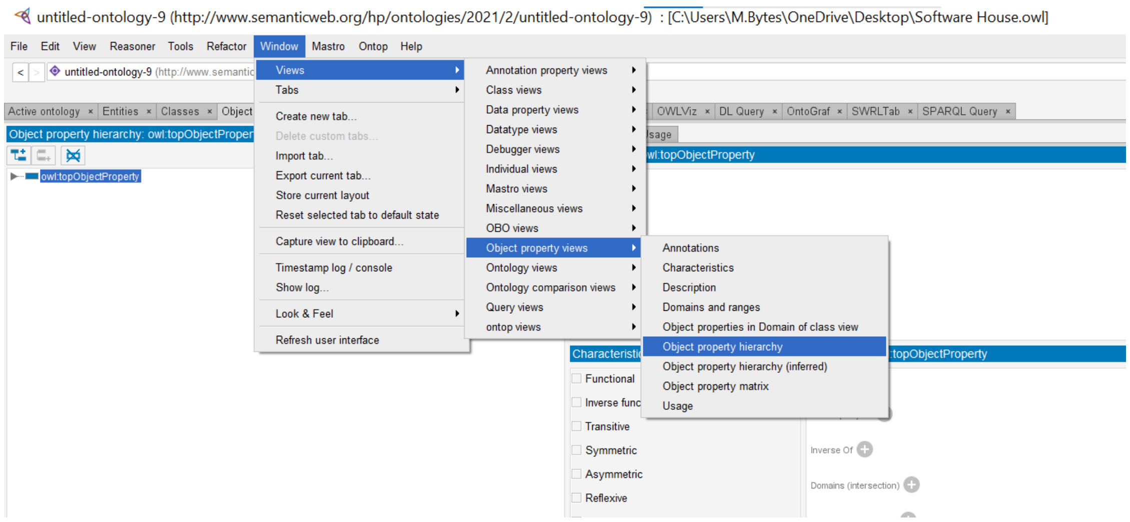

untitled-ontology-9 (http://www.semanticweb.org/hp/ontologies/2021/2/untitled-ontology-9)
name
Active ontology Entities Classes Object properties Data properties Annotation properties Individuals by class OWLViz DL Query OntoGraf SWRLTab SPARQL Query
Data property hierarchy: name
Asserted
owl:topDataProperty
name
Annotations Usage
Annotations: name
Annotations
Description: name
Equivalent To
SubProperty Of
Domains (intersection)
Software_House
Ranges
xsd:string
Disjoint With

Lab 03: Object Property

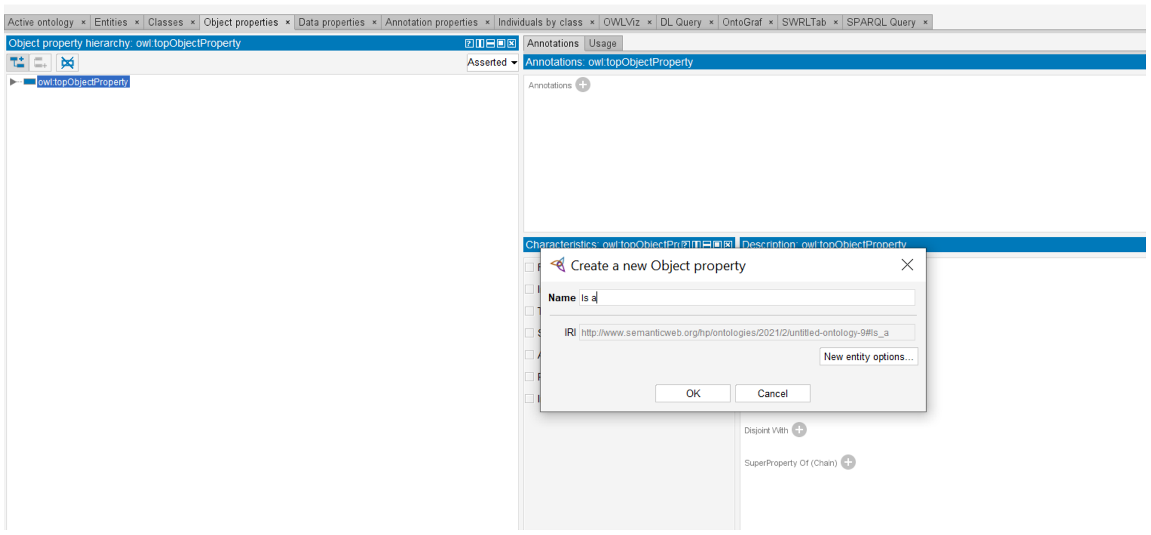
Adding Object Property Tab:



Adding Object Property View:



Creating Object Property:



Create a new Object property

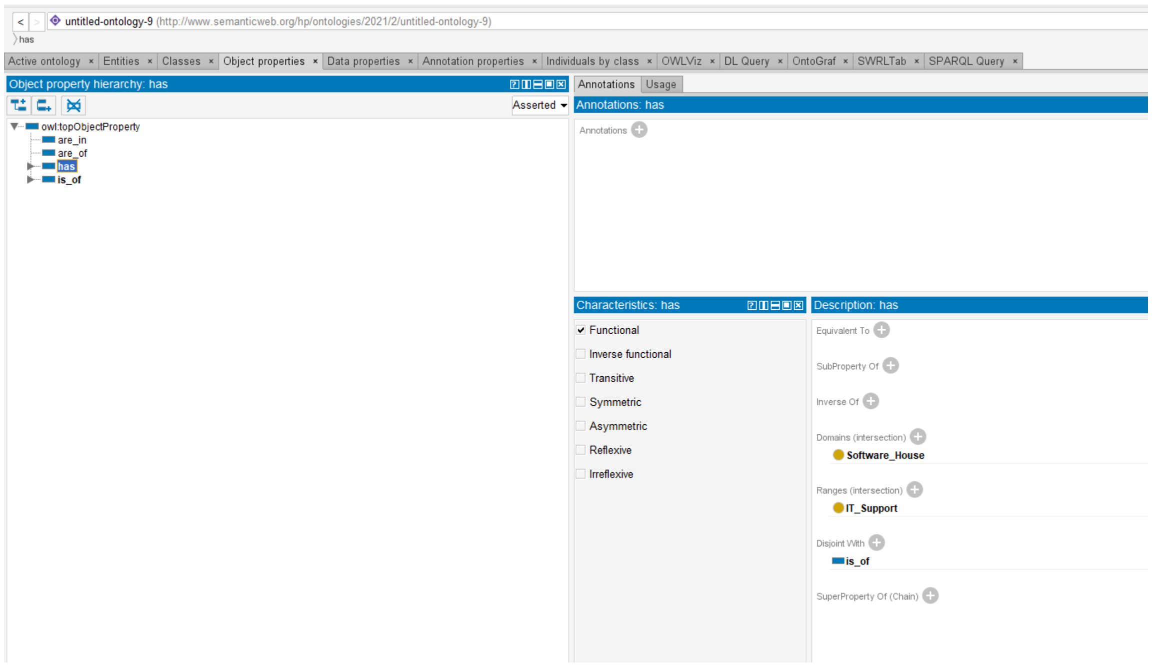
Name: Is a

IRI: http://www.semanticweb.org/hp/ontologies/2021/2/untitled-ontology-9#Is_a

New entity options...

OK Cancel

Domain Range of Object Property:



- ✓ Functional
- Inverse functional
- Transitive
- Symmetric
- Asymmetric
- Reflexive
- Irreflexive

Equivalent To

SubProperty Of

Inverse Of

Domains (intersection)
- Software_House

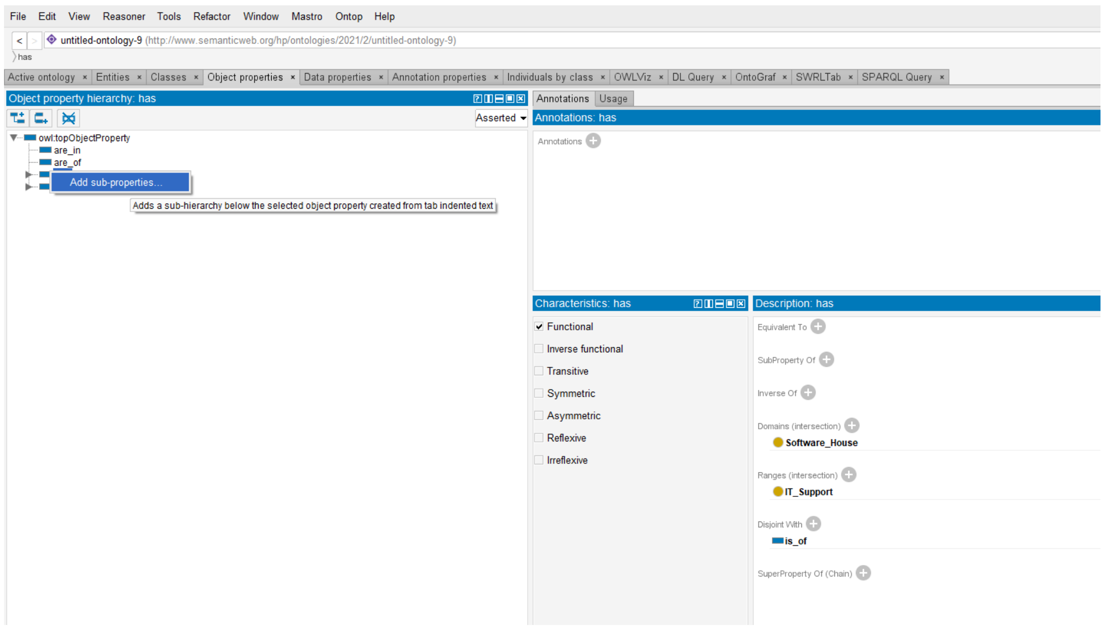
Ranges (intersection)
- IT_Support

Disjoint With
- is_of

SuperProperty Of (Chain)

Lab 04: Object Sub Property

Creating Object Sub Property:



owl:topObjectProperty
- are_in
- are_of

Add sub-properties...

Adds a sub-hierarchy below the selected object property created from tab indented text

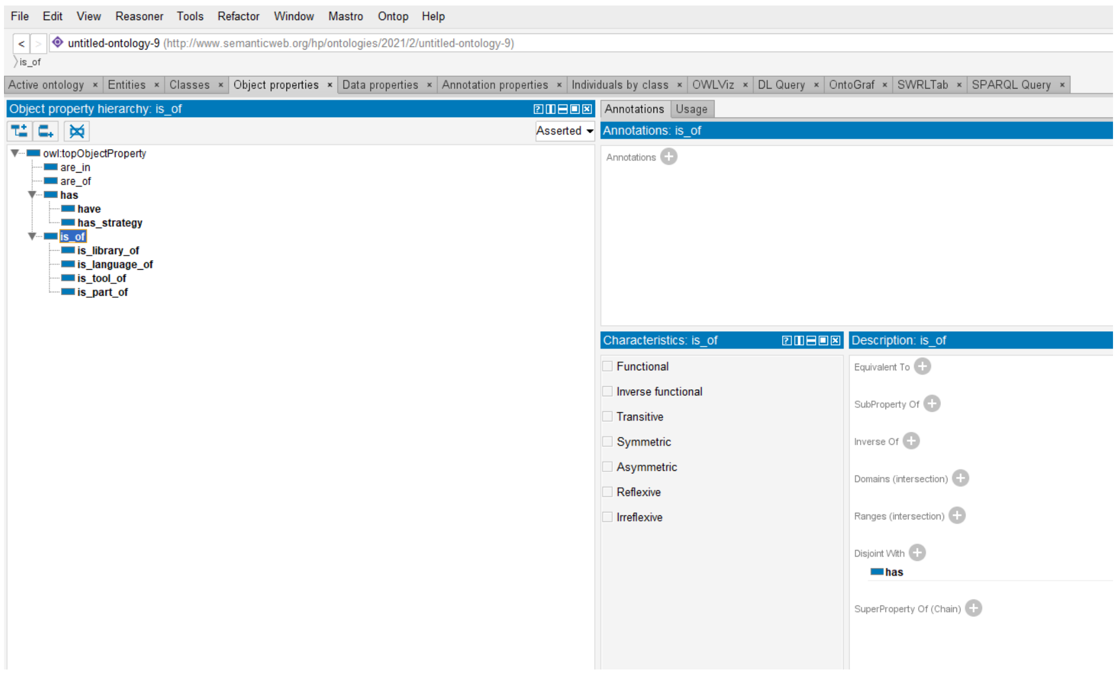
- [x] Functional
- [ ] Inverse functional
- [ ] Transitive
- [ ] Symmetric
- [ ] Asymmetric
- [ ] Reflexive
- [ ] Irreflexive

Domains (intersection): Software_House

Ranges (intersection): IT_Support

Disjoint With: is_of

Object Properties and Sub Properties:

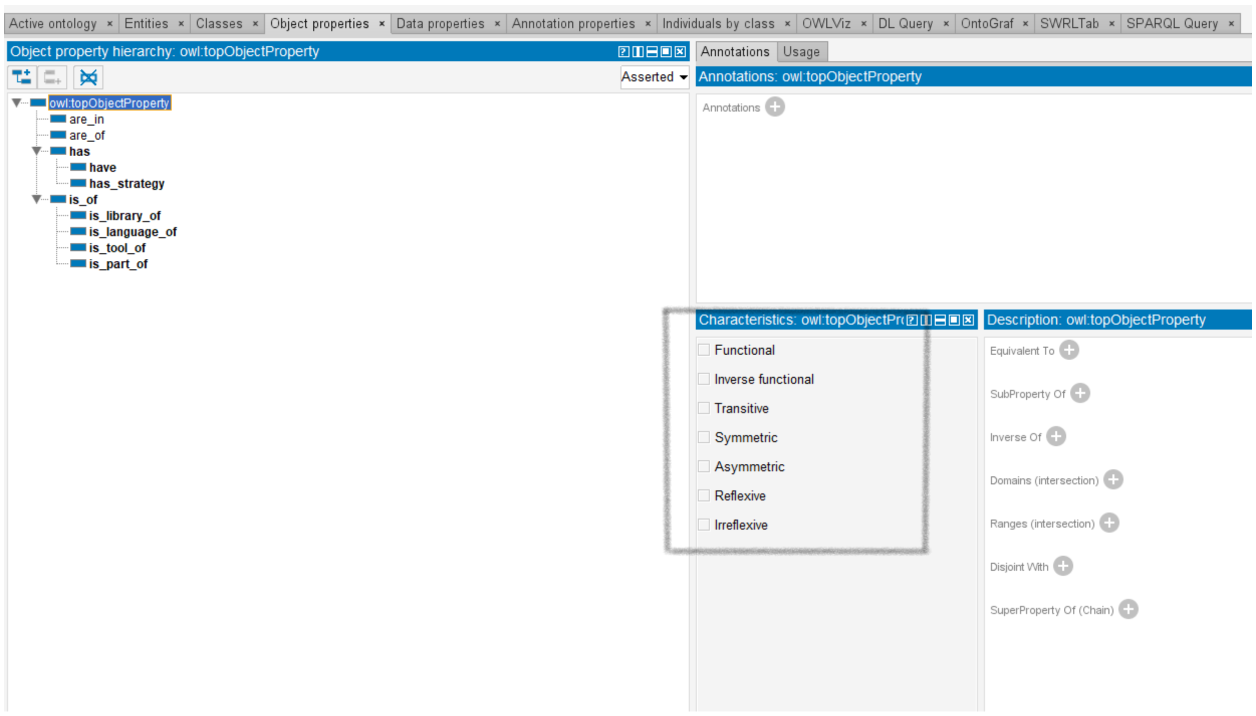


```
owl:topObjectProperty
    are_in
    are_of
    has
        have
        has_strategy
    is_of
        is_library_of
        is_language_of
        is_tool_of
        is_part_of
```

- [ ] Functional
- [ ] Inverse functional
- [ ] Transitive
- [ ] Symmetric
- [ ] Asymmetric
- [ ] Reflexive
- [ ] Irreflexive

Disjoint With: has

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



Object property hierarchy: owl:topObjectProperty

- owl:topObjectProperty
  - are_in
  - are_of
  - has
    - have
    - has_strategy
  - is_of
    - is_library_of
    - is_language_of
    - is_tool_of
    - is_part_of

Annotations: owl:topObjectProperty

Annotations

Characteristics: owl:topObjectPr

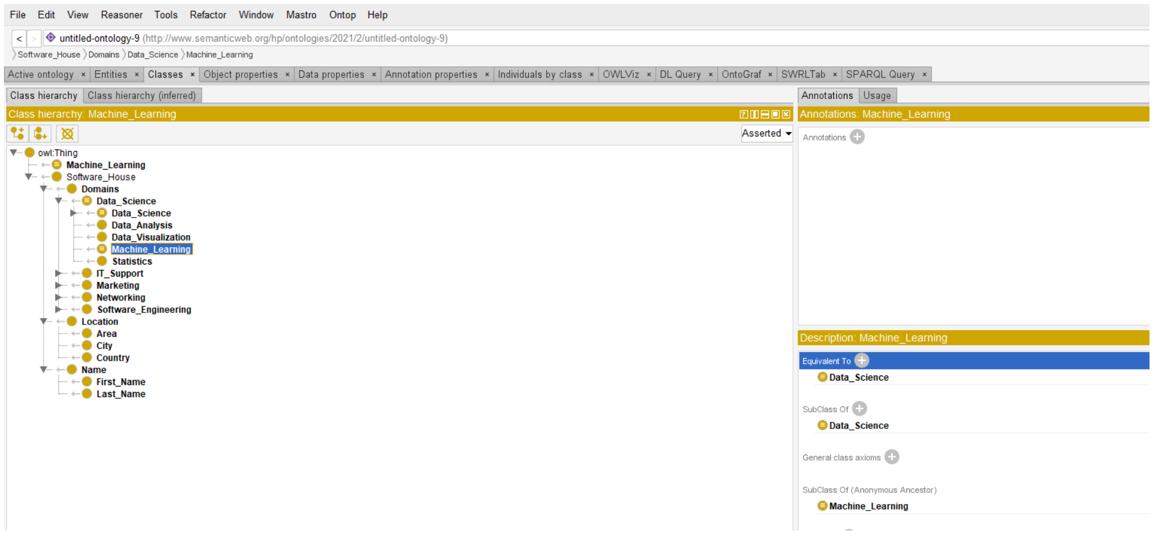
- Functional
- Inverse functional
- Transitive
- Symmetric
- Asymmetric
- Reflexive
- Irreflexive

Description: owl:topObjectProperty

- Equivalent To
- SubProperty Of
- Inverse Of
- Domains (intersection)
- Ranges (intersection)
- Disjoint With
- SuperProperty Of (Chain)

Lab 06: Equivalent to

Equivalent Classes:


File Edit View Reasoner Tools Refactor Window Mastro Ontop Help
untitled-ontology-9 (http://www.semanticweb.org/hp/ontologies/2021/2/untitled-ontology-9)
Software_House 〉Domains 〉Data_Science 〉Machine_Learning
Active ontology × Entities × Classes × Object properties × Data properties × Annotation properties × Individuals by class × OWLViz × DL Query × OntoGraf × SWRLTab × SPARQL Query ×
Class hierarchy | Class hierarchy (inferred)
Annotations | Usage
Asserted

Class hierarchy: Machine_Learning

```
owl:Thing
    Machine_Learning
    Software_House
        Domains
            Data_Science
                Data_Science
                Data_Analysis
                Data_Visualization
                Machine_Learning
                Statistics
            IT_Support
            Marketing
            Networking
            Software_Engineering
        Location
            Area
            City
            Country
        Name
            First_Name
            Last_Name
```

Annotations: Machine_Learning

Annotations

Description: Machine_Learning

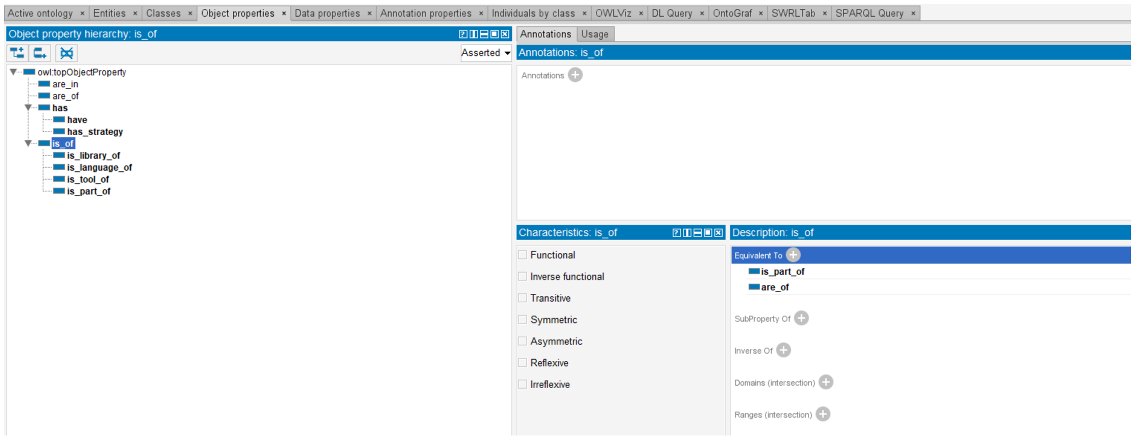
Equivalent To
- Data_Science

SubClass Of
- Data_Science

General class axioms

SubClass Of (Anonymous Ancestor)
- Machine_Learning

Equivalent Object Property:


Active ontology × Entities × Classes × Object properties × Data properties × Annotation properties × Individuals by class × OWLViz × DL Query × OntoGraf × SWRLTab × SPARQL Query ×
Annotations | Usage
Asserted

Object property hierarchy: is_of

```
owl:topObjectProperty
    are_in
    are_of
    has
        have
        has_strategy
    is_of
        is_library_of
        is_language_of
        is_tool_of
        is_part_of
```

Annotations: is_of

Annotations

Characteristics: is_of

- Functional
- Inverse functional
- Transitive
- Symmetric
- Asymmetric
- Reflexive
- Irreflexive

Description: is_of

Equivalent To
- is_part_of
- are_of

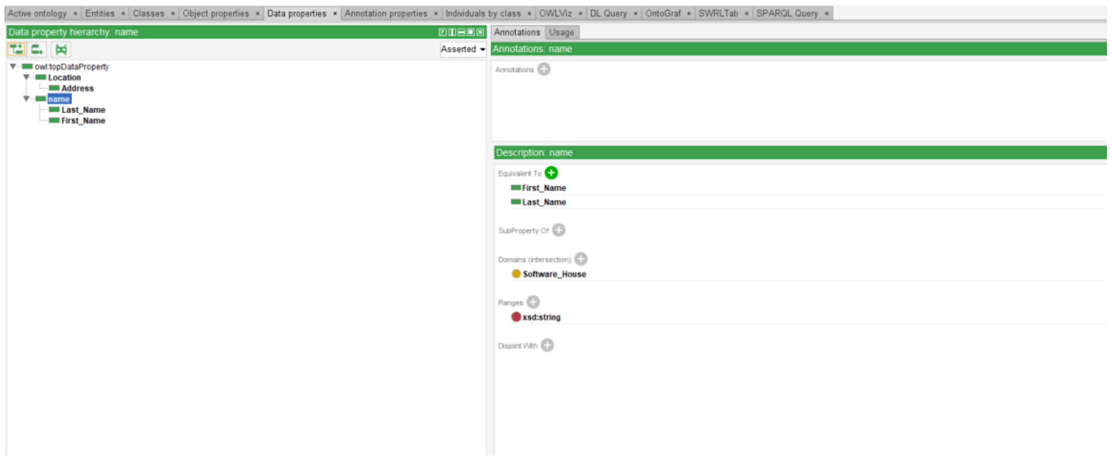
SubProperty Of

Inverse Of

Domains (intersection)

Ranges (intersection)

Equivalent Data Property:


Active ontology × Entities × Classes × Object properties × Data properties × Annotation properties × Individuals by class × OWLViz × DL Query × OntoGraf × SWRLTab × SPARQL Query ×
Annotations | Usage
Asserted

Data property hierarchy: name

```
owl:topDataProperty
    Location
        Address
    name
        Last_Name
        First_Name
```

Annotations: name

Annotations

Description: name

Equivalent To
- First_Name
- Last_Name

SubProperty Of

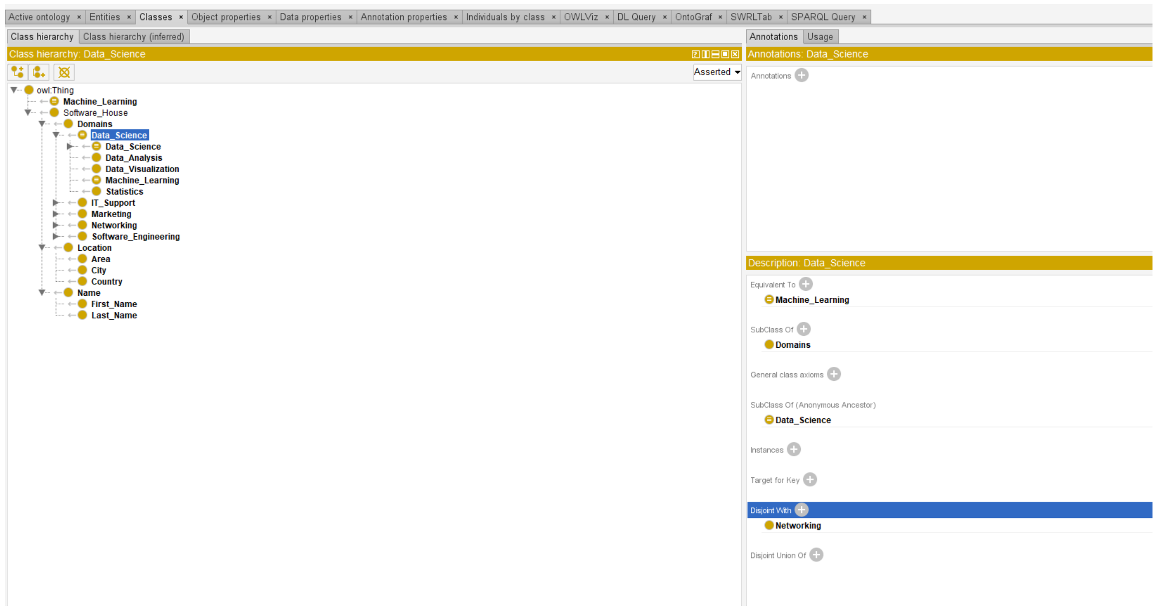
Domains (intersection)
- Software_House

Ranges
- xsd:string

Disjoint With

Lab 07: Disjoint

Disjoint Classes:



Class hierarchy: Data_Science

```
owl:Thing
    Machine_Learning
    Software_House
        Domains
            Data_Science
                Data_Science
                Data_Analysis
                Data_Visualization
                Machine_Learning
                Statistics
            IT_Support
            Marketing
            Networking
            Software_Engineering
        Location
            Area
            City
            Country
        Name
            First_Name
            Last_Name
```

Annotations: Data_Science

Annotations

Description: Data_Science

Equivalent To
Machine_Learning

SubClass Of
Domains

General class axioms

SubClass Of (Anonymous Ancestor)
Data_Science

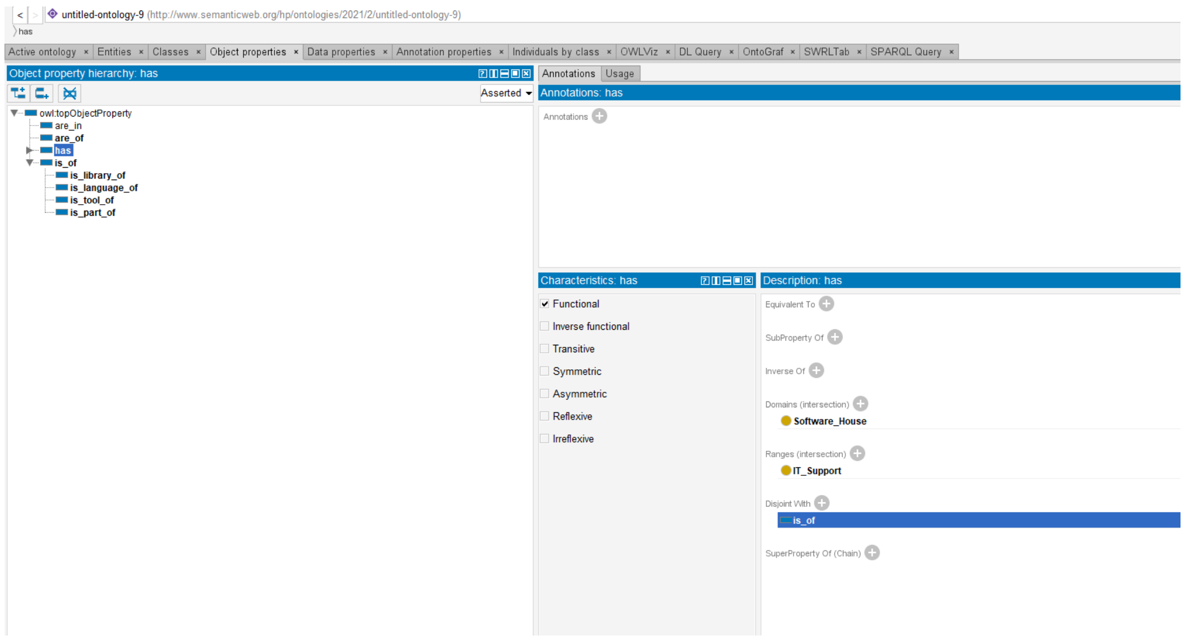
Instances

Target for Key

Disjoint With
Networking

Disjoint Union Of

Disjoint Object Properties:



Object property hierarchy: has

```
owl:topObjectProperty
    are_in
    are_of
    has
    is_of
        is_library_of
        is_language_of
        is_tool_of
        is_part_of
```

Annotations: has

Annotations

Characteristics: has

- [✓] Functional
- [ ] Inverse functional
- [ ] Transitive
- [ ] Symmetric
- [ ] Asymmetric
- [ ] Reflexive
- [ ] Irreflexive

Description: has

Equivalent To

SubProperty Of

Inverse Of

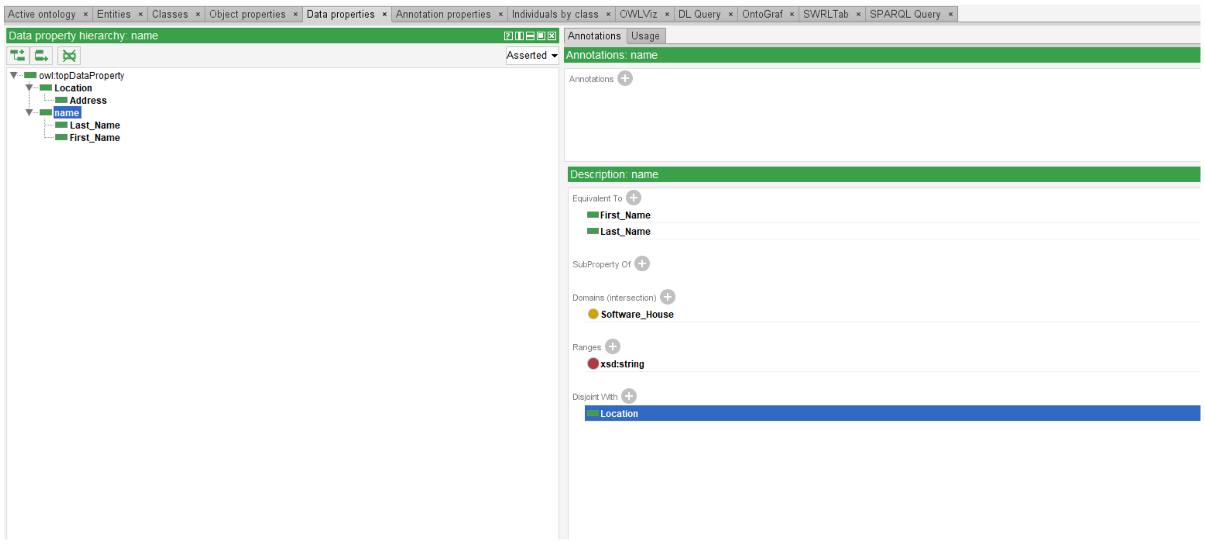
Domains (intersection)
Software_House

Ranges (intersection)
IT_Support

Disjoint With
is_of

SuperProperty Of (Chain)

Disjoint Data Properties:



Data property hierarchy: name

- owl:topDataProperty
  - Location
    - Address
  - name
    - Last_Name
    - First_Name

Annotations: name

Annotations

Description: name

Equivalent To
- First_Name
- Last_Name

SubProperty Of

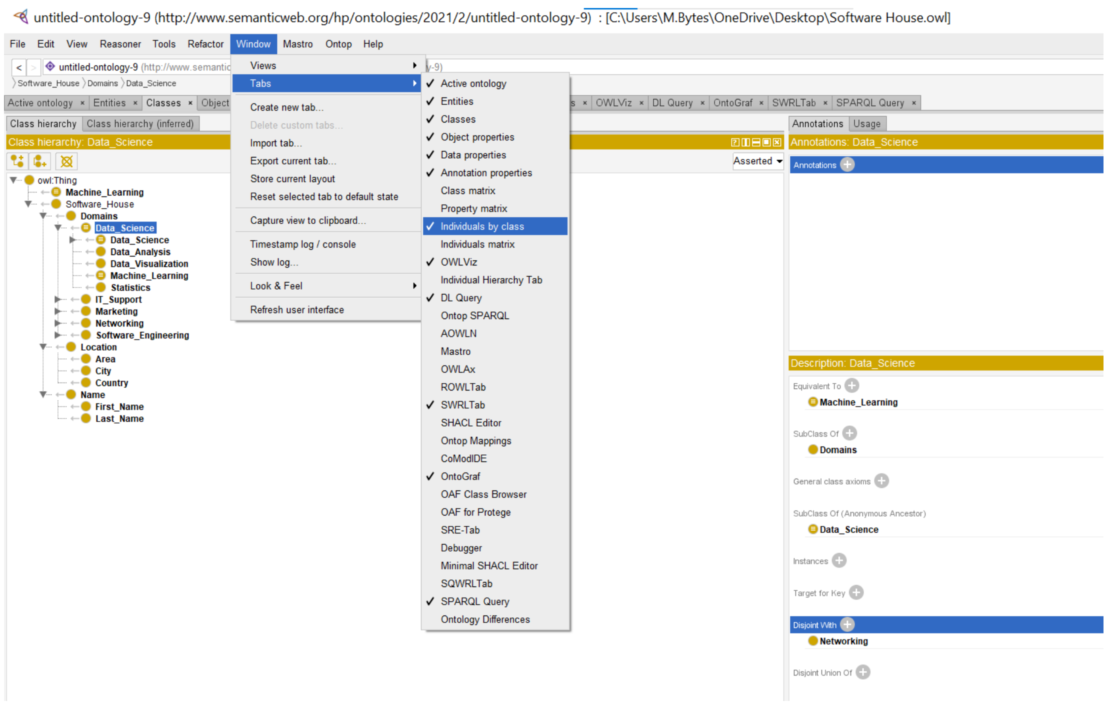
Domains (intersection)
- Software_House

Ranges
- xsd:string

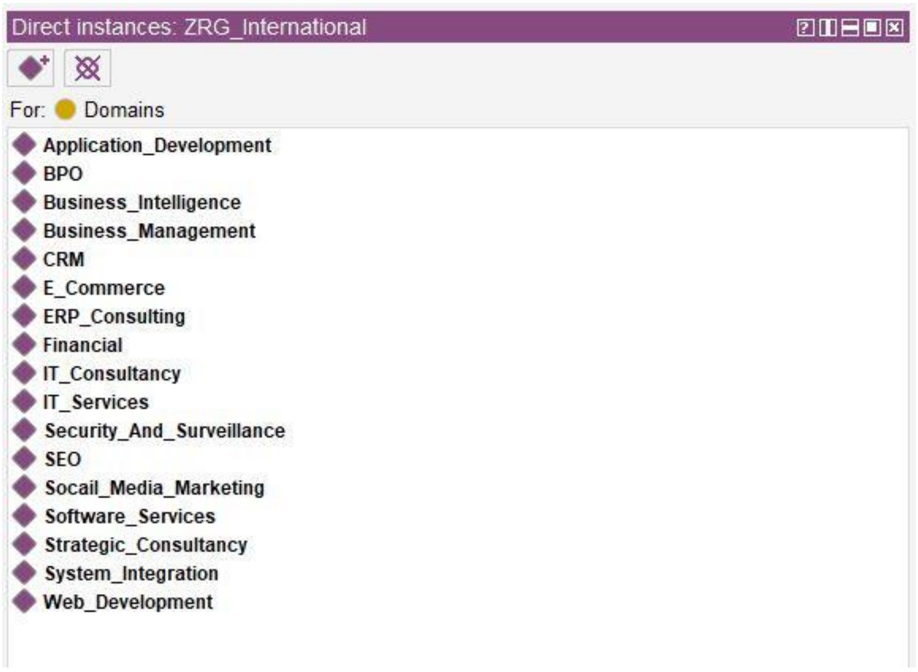
Disjoint With
- Location

Lab 08: Individuals

Adding Individuals Tab:

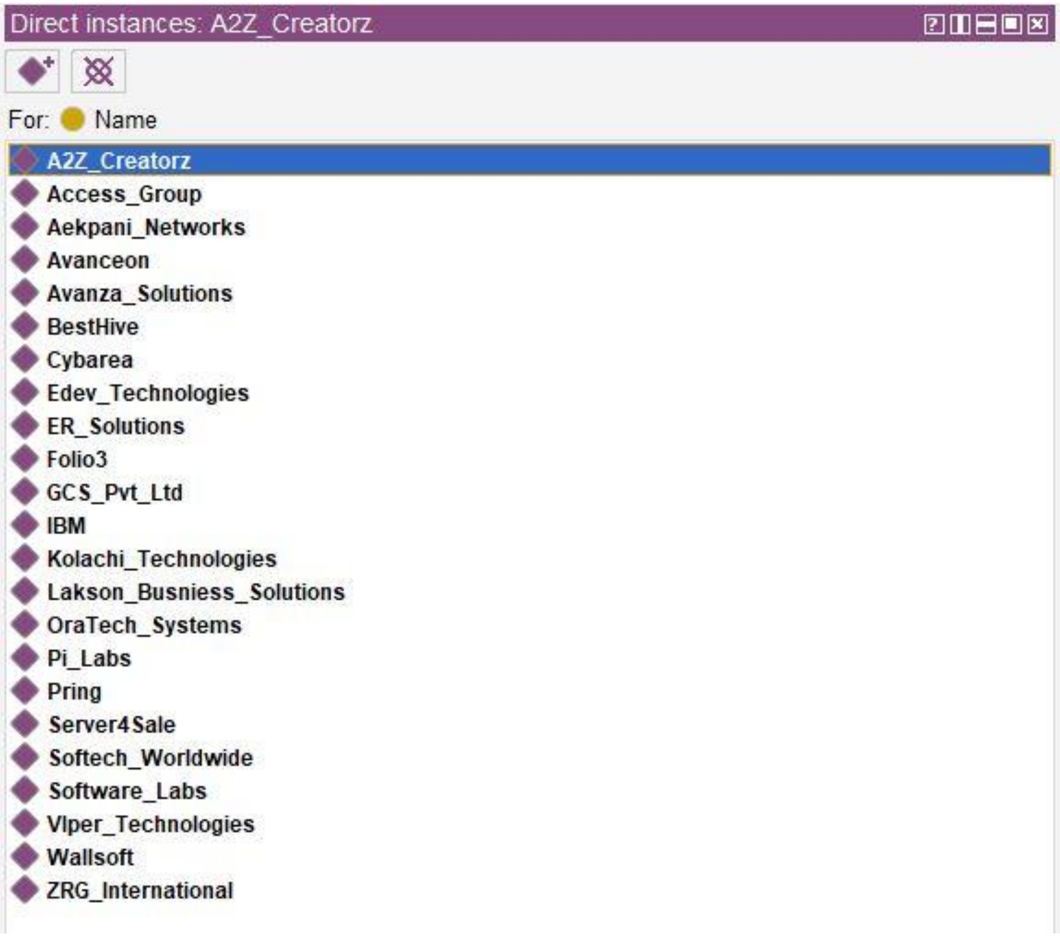


Individuals:


Direct instances: ZRG_International

For: Domains

- Application_Development
- BPO
- Business_Intelligence
- Business_Management
- CRM
- E_Commerce
- ERP_Consulting
- Financial
- IT_Consultancy
- IT_Services
- Security_And_Surveillance
- SEO
- Socail_Media_Marketing
- Software_Services
- Strategic_Consultancy
- System_Integration
- Web_Development



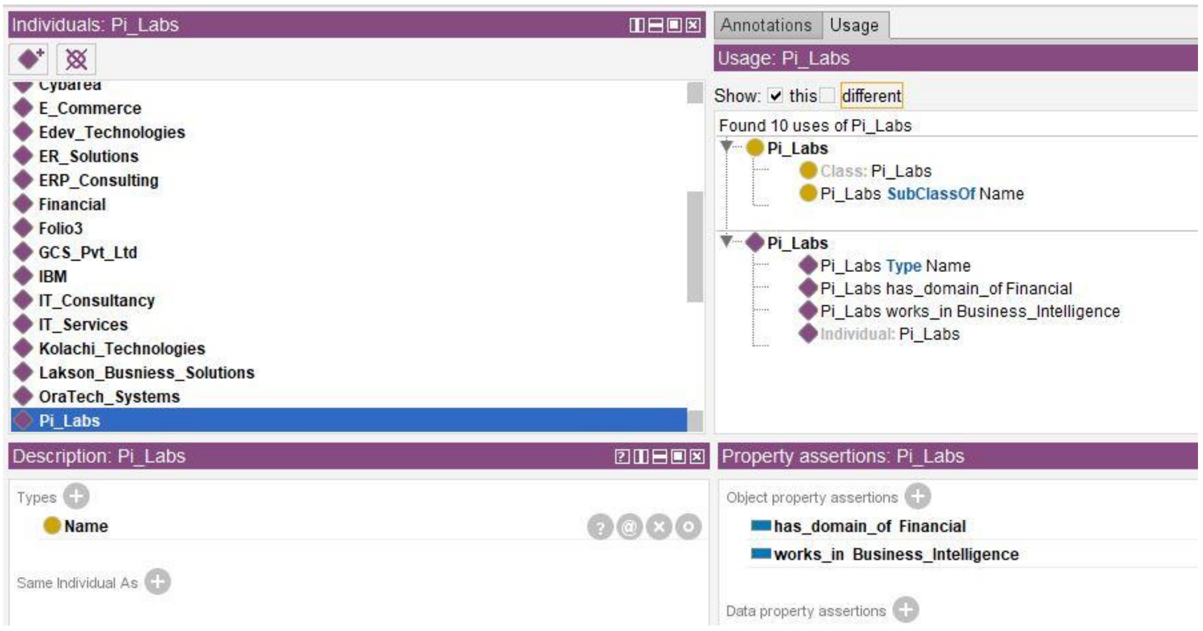
Direct instances: A2Z_Creatorz

For: Name

A2Z_Creatorz
Access_Group
Aekpani_Networks
Avanceon
Avanza_Solutions
BestHive
Cybarea
Edev_Technologies
ER_Solutions
Folio3
GCS_Pvt_Ltd
IBM
Kolachi_Technologies
Lakson_Busniess_Solutions
OraTech_Systems
Pi_Labs
Pring
Server4Sale
Softech_Worldwide
Software_Labs
VIper_Technologies
Wallsoft
ZRG_International

Lab 09: Object and data property assertion

Object Property Assertion:



Individuals: Pi_Labs

Cybarea
E_Commerce
Edev_Technologies
ER_Solutions
ERP_Consulting
Financial
Folio3
GCS_Pvt_Ltd
IBM
IT_Consultancy
IT_Services
Kolachi_Technologies
Lakson_Busniess_Solutions
OraTech_Systems
Pi_Labs

Annotations Usage

Usage: Pi_Labs

Show: ✓ this different

Found 10 uses of Pi_Labs

Pi_Labs
- Class: Pi_Labs
- Pi_Labs SubClassOf Name

Pi_Labs
- Pi_Labs Type Name
- Pi_Labs has_domain_of Financial
- Pi_Labs works_in Business_Intelligence
- Individual: Pi_Labs

Description: Pi_Labs

Types
Name

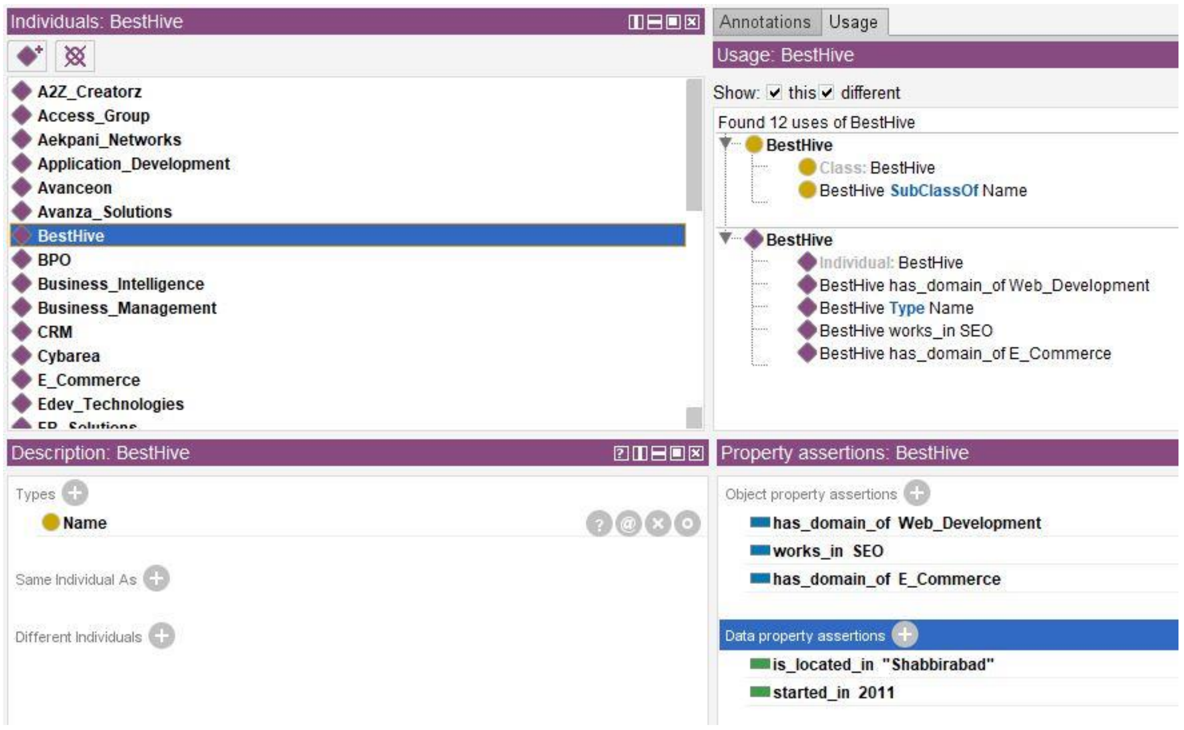
Same Individual As

Property assertions: Pi_Labs

Object property assertions
has_domain_of Financial
works_in Business_Intelligence

Data property assertions

Data Property Assertion:



Individuals: BestHive

A2Z_Creatorz
Access_Group
Aekpani_Networks
Application_Development
Avanceon
Avanza_Solutions
BestHive
BPO
Business_Intelligence
Business_Management
CRM
Cybarea
E_Commerce
Edev_Technologies
ER_Solutions

Annotations Usage

Usage: BestHive

Show: ✓ this ✓ different

Found 12 uses of BestHive

BestHive
- Class: BestHive
- BestHive SubClassOf Name

BestHive
- Individual: BestHive
- BestHive has_domain_of Web_Development
- BestHive Type Name
- BestHive works_in SEO
- BestHive has_domain_of E_Commerce

Description: BestHive

Types
Name

Same Individual As

Different Individuals

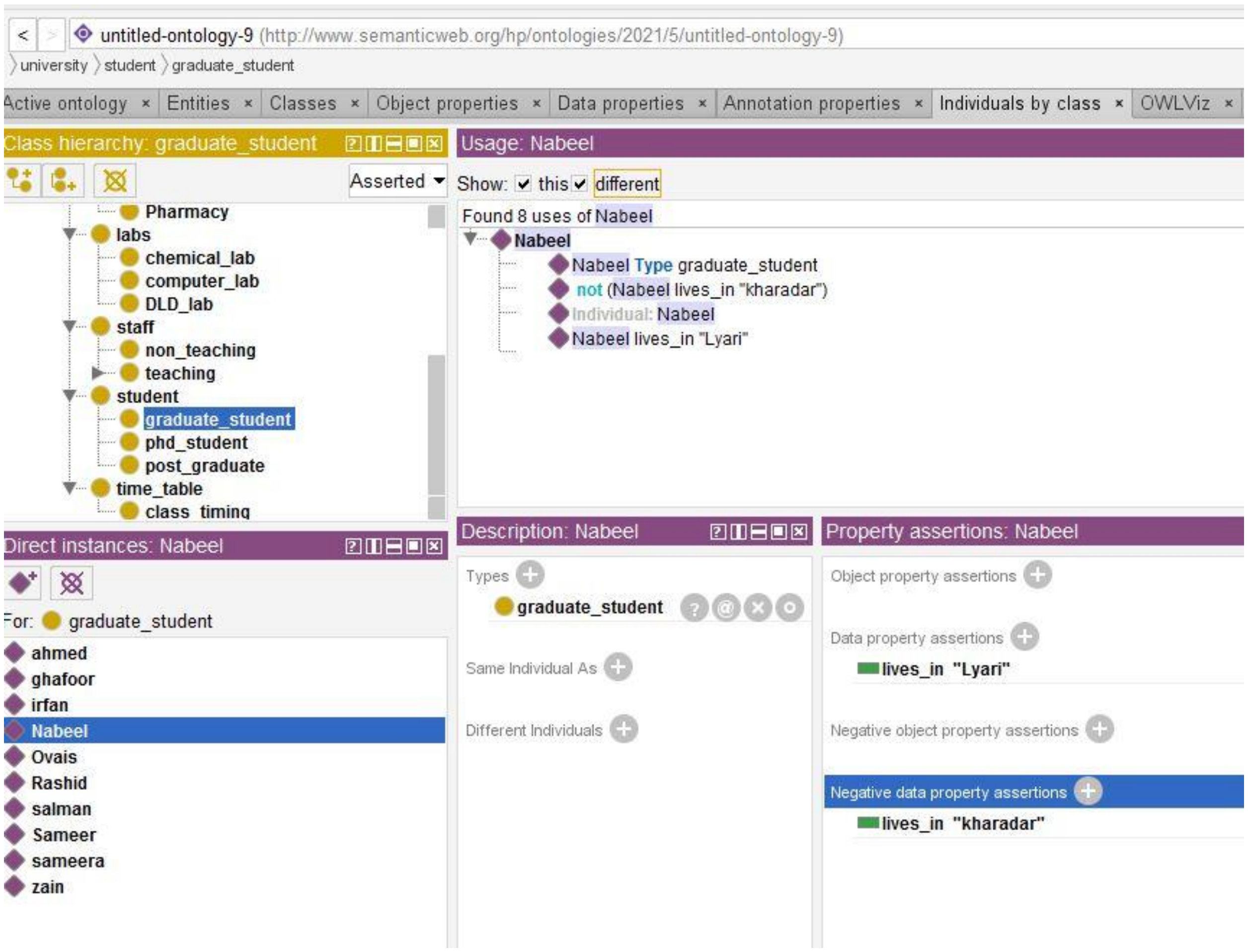
Property assertions: BestHive

Object property assertions
has_domain_of Web_Development
works_in SEO
has_domain_of E_Commerce

Data property assertions
is_located_in "Shabbirabad"
started_in 2011

Lab 10: Negative object and data property assertion

Negative Object Property Assertion:



untitled-ontology-9 (http://www.semanticweb.org/hp/ontologies/2021/5/untitled-ontology-9)
university > student > graduate_student
Active ontology | Entities | Classes | Object properties | Data properties | Annotation properties | Individuals by class | OWLViz
Class hierarchy: graduate_student
Asserted
Pharmacy
labs
chemical_lab
computer_lab
DLD_lab
staff
non_teaching
teaching
student
graduate_student
phd_student
post_graduate
time_table
class_timing
Direct instances: Nabeel
For: graduate_student
ahmed
ghafoor
irfan
Nabeel
Ovais
Rashid
salman
Sameer
sameera
zain
Usage: Nabeel
Show: this different
Description: Nabeel
Types
Same Individual As
Different Individuals
Property assertions: Nabeel
Object property assertions
Data property assertions
Negative object property assertions
Negative data property assertions

Found 8 uses of Nabeel

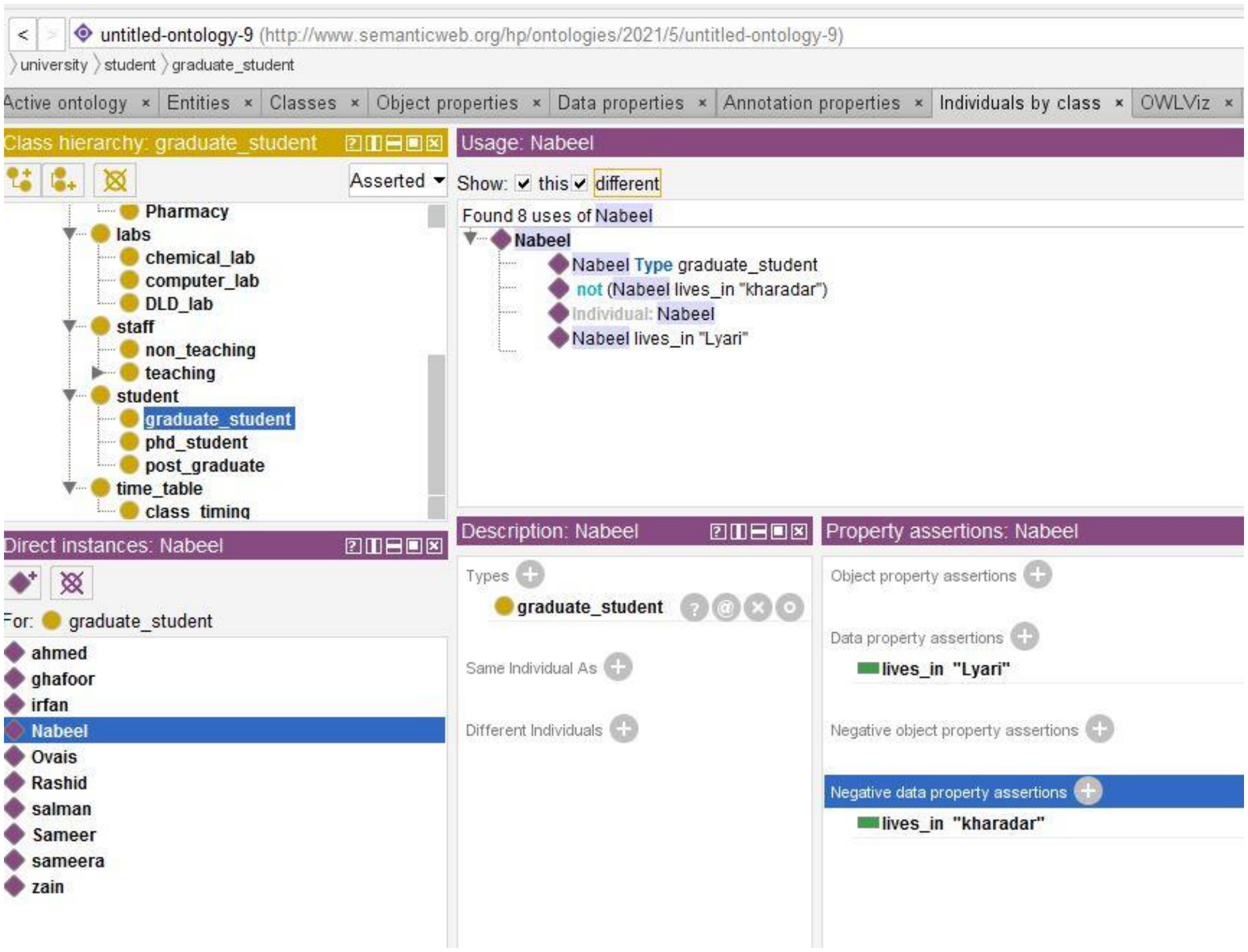
- Nabeel
  - Nabeel Type graduate_student
  - not (Nabeel lives_in "kharadar")
  - Individual: Nabeel
  - Nabeel lives_in "Lyari"

Types: graduate_student

Data property assertions: lives_in "Lyari"

Negative data property assertions: lives_in "kharadar"

Negative Data Property Assertion:



untitled-ontology-9 (http://www.semanticweb.org/hp/ontologies/2021/5/untitled-ontology-9)

university 〉 student 〉 graduate_student

Usage: Nabeel

Show: ✓ this ✓ different

Found 8 uses of Nabeel

Nabeel
- Nabeel Type graduate_student
- not (Nabeel lives_in "kharadar")
- Individual: Nabeel
- Nabeel lives_in "Lyari"

Direct instances: Nabeel

For: graduate_student

ahmed, ghafoor, irfan, Nabeel, Ovais, Rashid, salman, Sameer, sameera, zain

Description: Nabeel

Types: graduate_student

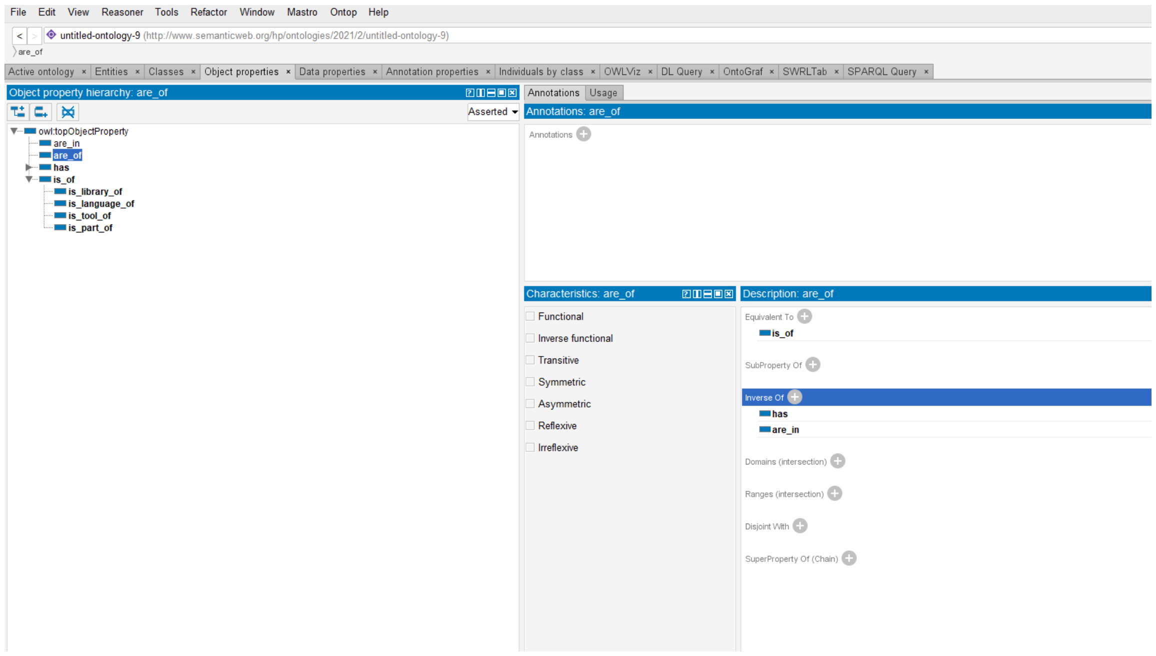
Property assertions: Nabeel

Data property assertions: lives_in "Lyari"

Negative data property assertions: lives_in "kharadar"

Lab 11: Inverse of

Inverse Of:


File Edit View Reasoner Tools Refactor Window Mastro Ontop Help
Active ontology × Entities × Classes × Object properties × Data properties × Annotation properties × Individuals by class × OWLViz × DL Query × OntoGraf × SWRLTab × SPARQL Query ×
Asserted
Annotations Usage
Annotations
Functional
Inverse functional
Transitive
Symmetric
Asymmetric
Reflexive
Irreflexive
Domains (intersection)
Ranges (intersection)
Disjoint With
SuperProperty Of (Chain)

untitled-ontology-9 (http://www.semanticweb.org/hp/ontologies/2021/2/untitled-ontology-9)

Object property hierarchy: are_of

```
owl:topObjectProperty
    are_in
    are_of
    has
    is_of
        is_library_of
        is_language_of
        is_tool_of
        is_part_of
```

Annotations: are_of

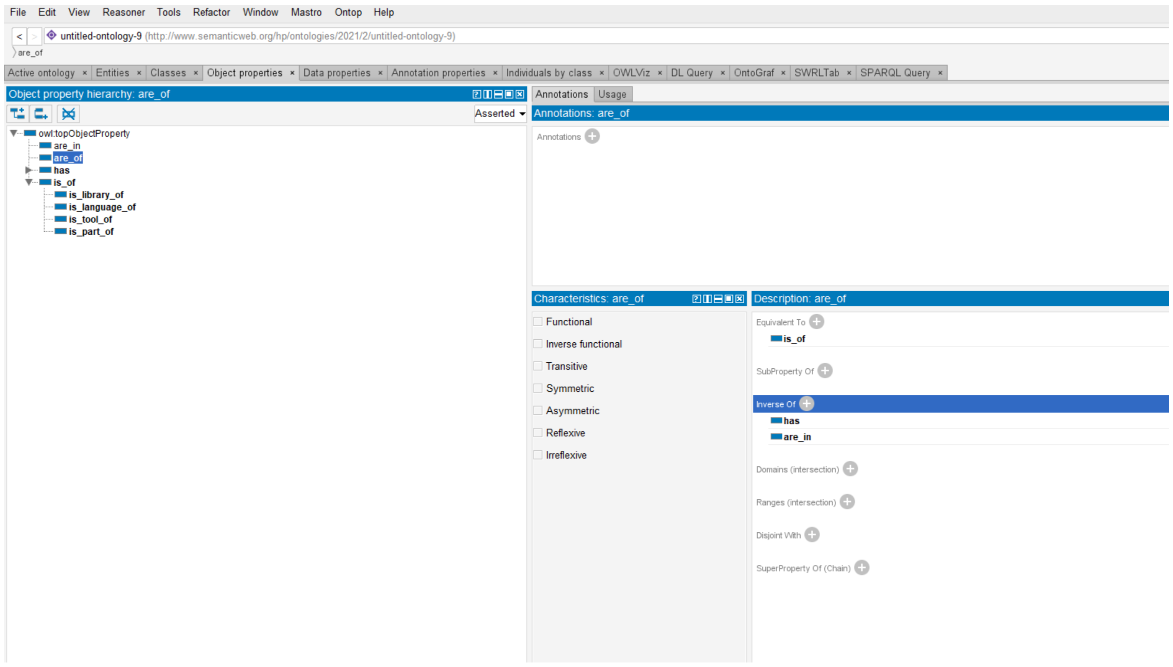
Characteristics: are_of

Description: are_of

Equivalent To
- is_of

SubProperty Of

Inverse Of
- has
- are_in


File Edit View Reasoner Tools Refactor Window Mastro Ontop Help
Active ontology × Entities × Classes × Object properties × Data properties × Annotation properties × Individuals by class × OWLViz × DL Query × OntoGraf × SWRLTab × SPARQL Query ×
Asserted
Annotations Usage
Annotations
Functional
Inverse functional
Transitive
Symmetric
Asymmetric
Reflexive
Irreflexive
Domains (intersection)
Ranges (intersection)
Disjoint With
SuperProperty Of (Chain)

untitled-ontology-9 (http://www.semanticweb.org/hp/ontologies/2021/2/untitled-ontology-9)

Object property hierarchy: are_of

```
owl:topObjectProperty
    are_in
    are_of
    has
    is_of
        is_library_of
        is_language_of
        is_tool_of
        is_part_of
```

Annotations: are_of

Characteristics: are_of

Description: are_of

Equivalent To
- is_of

SubProperty Of

Inverse Of
- has
- are_in

Lab 12: Same individual as Different individual as

Same Individual as:

Different Individual as:

Lab 13: Cardinality

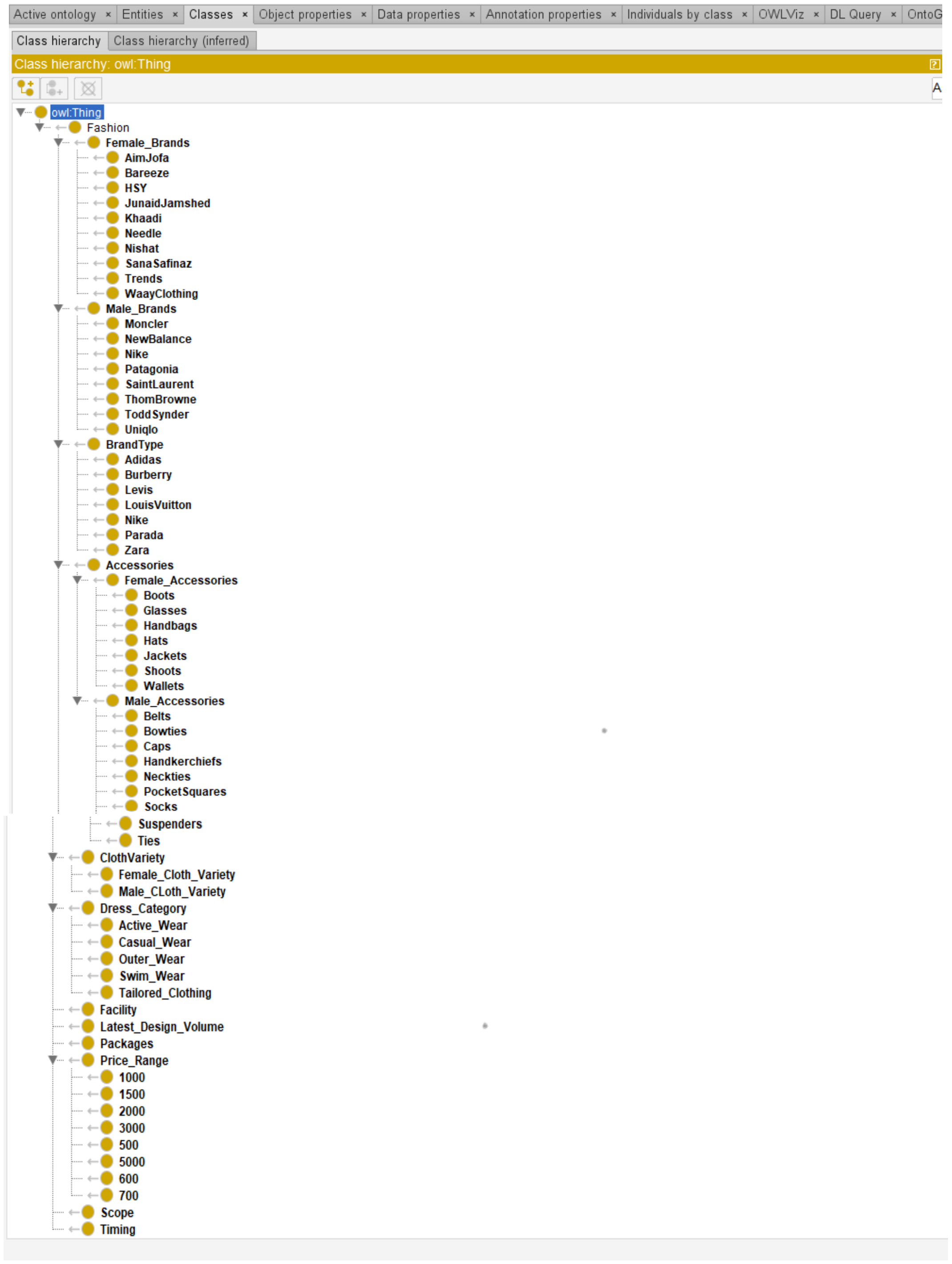
Setting Up Cardinality:

Cardinality:

Lab 14: Allvaluesfrom Somevaluesfrom Hasvalue

Lab 15: Fashion Ontology

Classes:



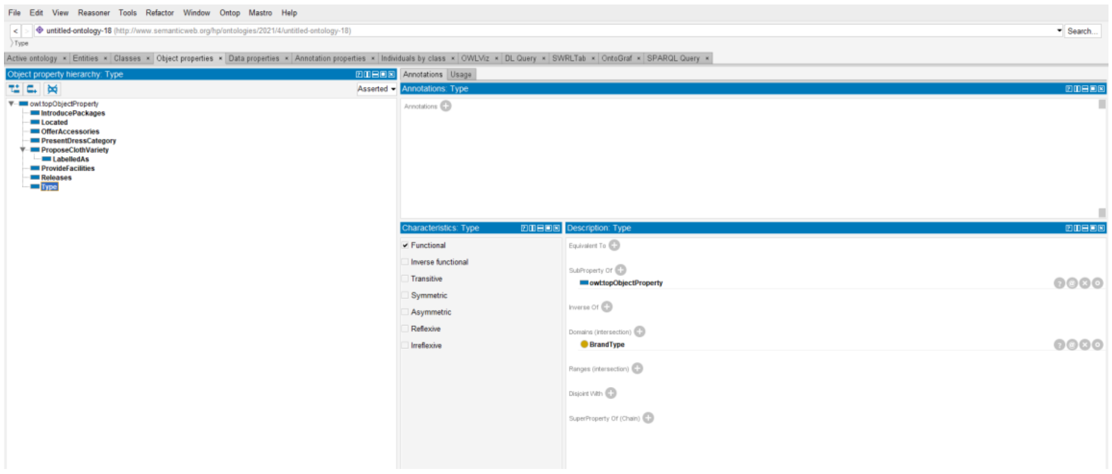
Active ontology × | Entities × | Classes × | Object properties × | Data properties × | Annotation properties × | Individuals by class × | OWLViz × | DL Query × | OntoG

Class hierarchy | Class hierarchy (inferred)

Class hierarchy: owl:Thing

```
owl:Thing
  Fashion
    Female_Brands
      AimJofa
      Bareeze
      HSY
      JunaidJamshed
      Khaadi
      Needle
      Nishat
      SanaSafinaz
      Trends
      WaayClothing
    Male_Brands
      Moncler
      NewBalance
      Nike
      Patagonia
      SaintLaurent
      ThomBrowne
      ToddSynder
      Uniqlo
    BrandType
      Adidas
      Burberry
      Levis
      LouisVuitton
      Nike
      Parada
      Zara
    Accessories
      Female_Accessories
        Boots
        Glasses
        Handbags
        Hats
        Jackets
        Shoots
        Wallets
      Male_Accessories
        Belts
        Bowties
        Caps
        Handkerchiefs
        Neckties
        PocketSquares
        Socks
        Suspenders
        Ties
    ClothVariety
      Female_Cloth_Variety
      Male_CLoth_Variety
    Dress_Category
      Active_Wear
      Casual_Wear
      Outer_Wear
      Swim_Wear
      Tailored_Clothing
    Facility
    Latest_Design_Volume
    Packages
    Price_Range
      1000
      1500
      2000
      3000
      500
      5000
      600
      700
    Scope
    Timing
```

Object Properties:


File Edit View Reasoner Tools Refactor Window Ontop Mastro Help
untitled-ontology-18 (http://www.semanticweb.org/hp/ontologies/2021/4/untitled-ontology-18)
Type
Active ontology Entities Classes Object properties Data properties Annotation properties Individuals by class OWLViz DL Query SWRLTab OntoGraf SPARQL Query
Object property hierarchy: Type
Asserted
Annotations Usage
Annotations: Type
Annotations
Characteristics: Type
Description: Type

owl:topObjectProperty
- IntroducePackages
- Located
- OfferAccessories
- PresentDressCategory
- ProposeClothVariety
  - LabelledAs
- ProvideFacilities
- Releases
- Type

[✓ Functional]
Inverse functional
Transitive
Symmetric
Asymmetric
Reflexive
Irreflexive

Equivalent To

SubProperty Of
owl:topObjectProperty

Inverse Of

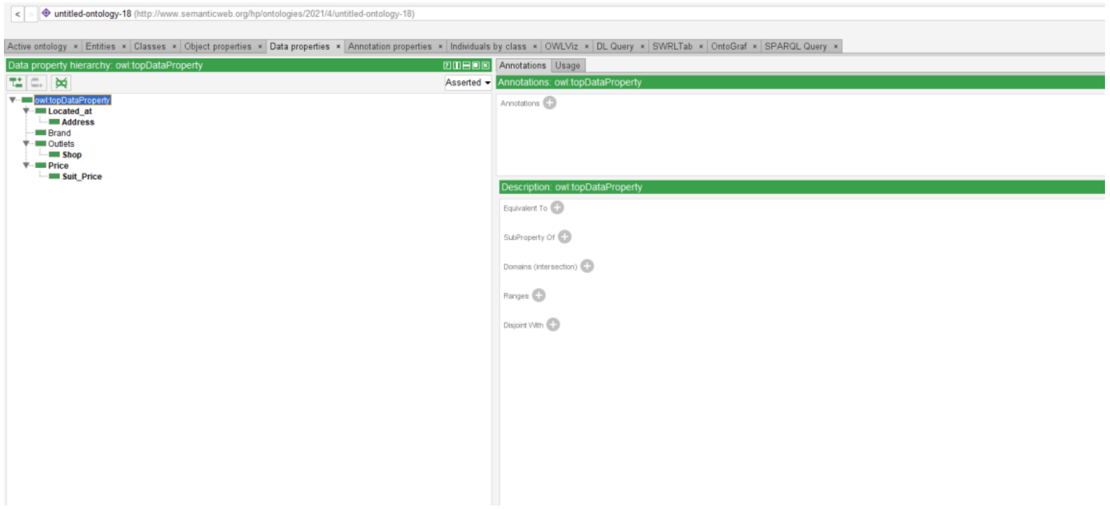
Domains (intersection)
BrandType

Ranges (intersection)

Disjoint With

SuperProperty Of (Chain)

Data Properties:


untitled-ontology-18 (http://www.semanticweb.org/hp/ontologies/2021/4/untitled-ontology-18)
Active ontology Entities Classes Object properties Data properties Annotation properties Individuals by class OWLViz DL Query SWRLTab OntoGraf SPARQL Query
Data property hierarchy: owl:topDataProperty
Asserted
Annotations Usage
Annotations: owl:topDataProperty
Annotations
Description: owl:topDataProperty

owl:topDataProperty
- Located_at
  - Address
- Brand
- Outlets
  - Shop
- Price
  - Suit_Price

Equivalent To

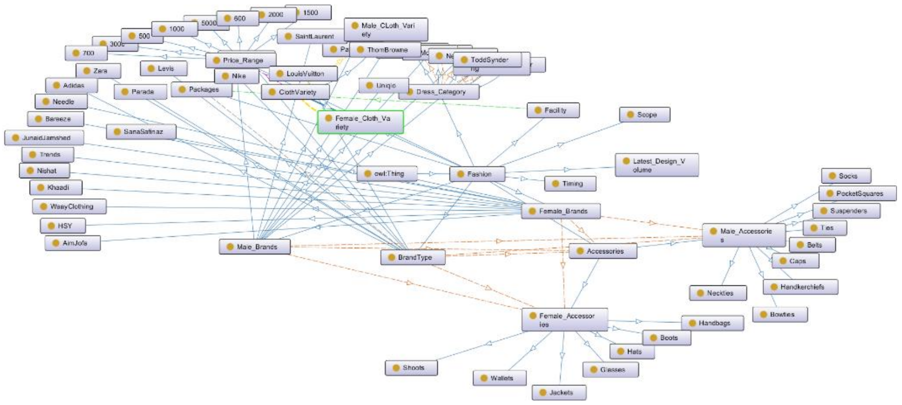
SubProperty Of

Domains (intersection)

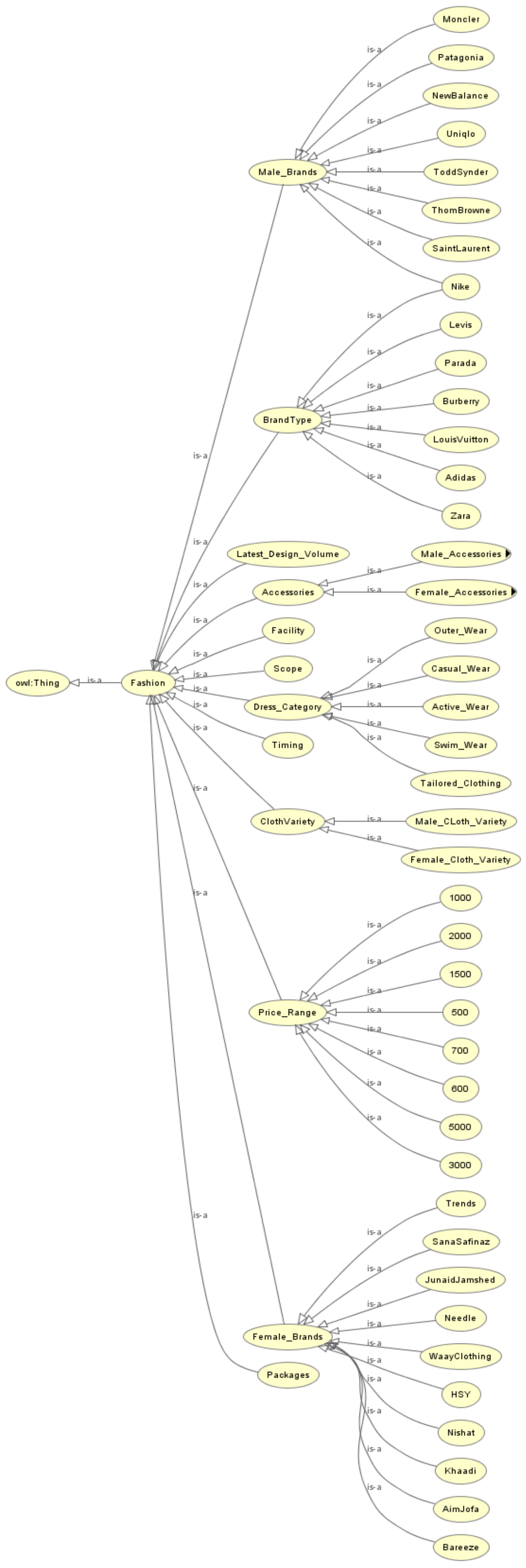
Ranges

Disjoint With

Ontograf View:

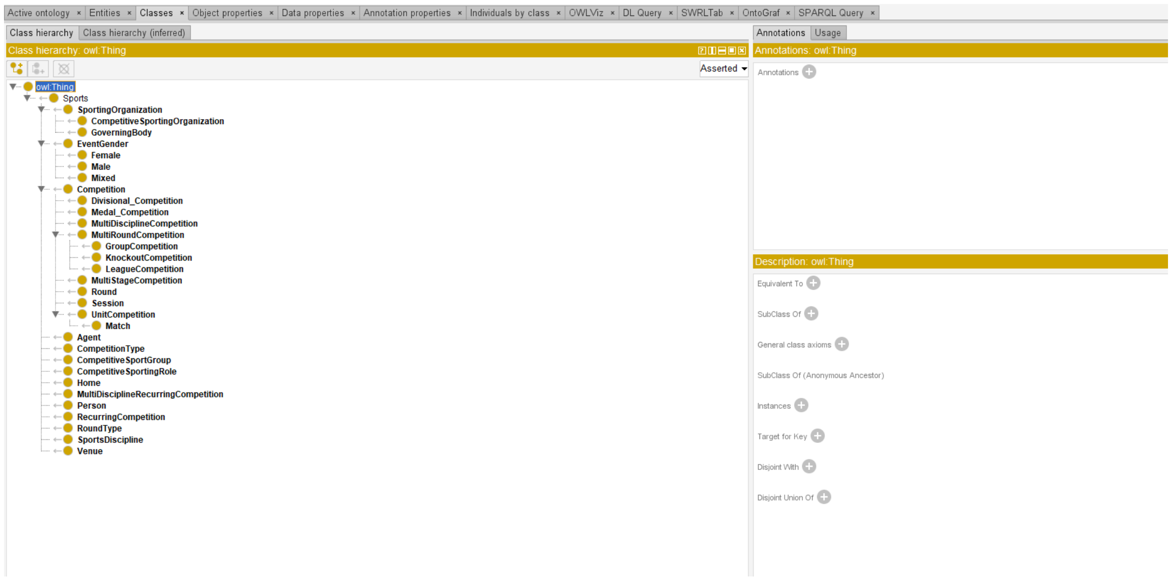


OwlVIZ:


owl:Thing
Fashion
Male_Brands
Moncler
Patagonia
NewBalance
Uniqlo
ToddSynder
ThomBrowne
SaintLaurent
Nike
BrandType
Levis
Parada
Burberry
LouisVuitton
Adidas
Zara
Latest_Design_Volume
Accessories
Male_Accessories
Female_Accessories
Facility
Scope
Dress_Category
Outer_Wear
Casual_Wear
Active_Wear
Swim_Wear
Tailored_Clothing
Timing
ClothVariety
Male_CLoth_Variety
Female_Cloth_Variety
Price_Range
1000
2000
1500
500
700
600
5000
3000
Female_Brands
Trends
SanaSafinaz
JunaidJamshed
Needle
WaayClothing
HSY
Nishat
Khaadi
AimJofa
Bareeze
Packages
is-a

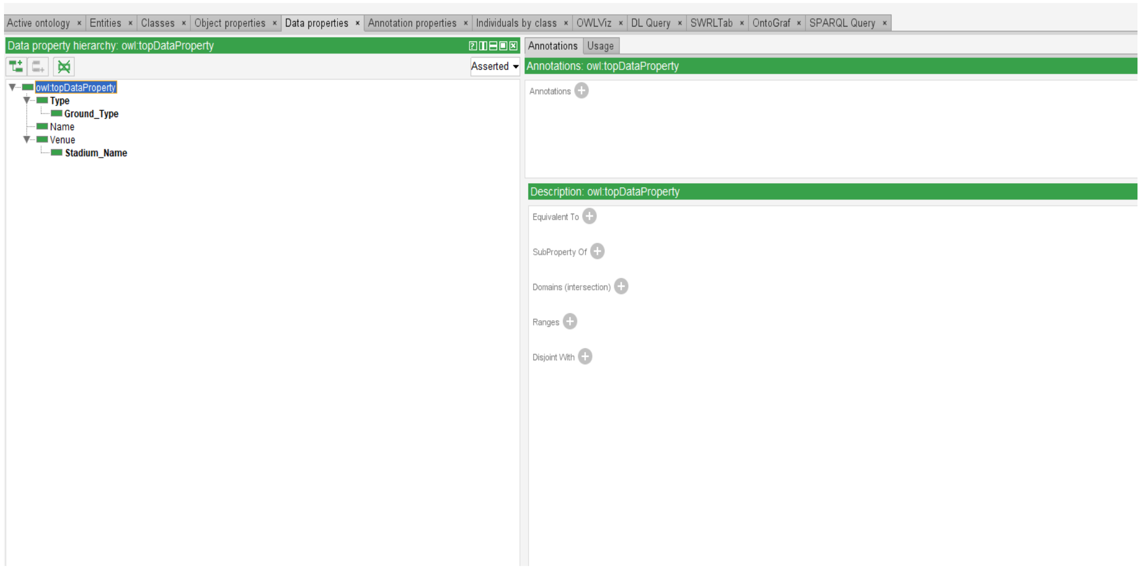
Lab 16: Sports Ontology

Classes:



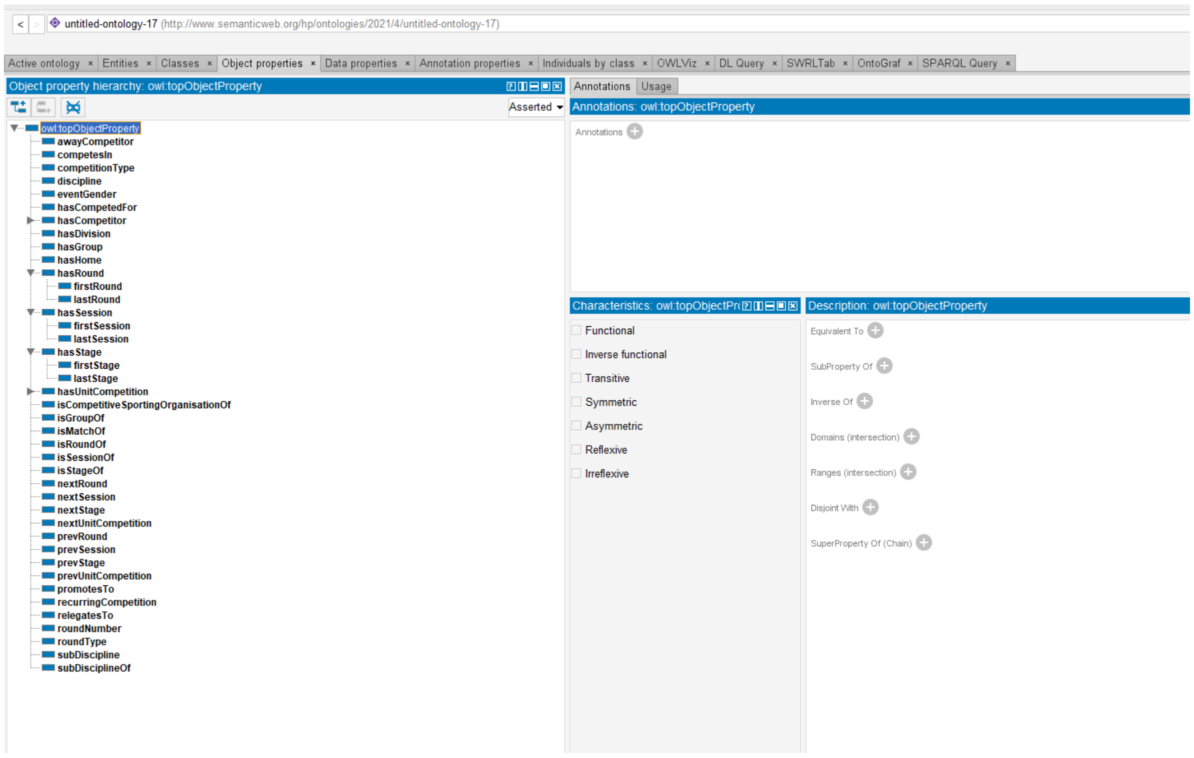
```
owl:Thing
    Sports
        SportingOrganization
            CompetitiveSportingOrganization
            GoverningBody
        EventGender
            Female
            Male
            Mixed
        Competition
            Divisional_Competition
            Medal_Competition
            MultiDisciplineCompetition
            MultiRoundCompetition
                GroupCompetition
                KnockoutCompetition
                LeagueCompetition
            MultiStageCompetition
            Round
            Session
            UnitCompetition
                Match
        Agent
        CompetitionType
        CompetitiveSportGroup
        CompetitiveSportingRole
        Home
        MultiDisciplineRecurringCompetition
        Person
        RecurringCompetition
        RoundType
        SportsDiscipline
        Venue
```

Data Properties:



```
owl:topDataProperty
    Type
        Ground_Type
    Name
    Venue
        Stadium_Name
```

Object Properties:



untitled-ontology-17 (http://www.semanticweb.org/hp/ontologies/2021/4/untitled-ontology-17)

Active ontology × Entities × Classes × Object properties × Data properties × Annotation properties × Individuals by class × OWLViz × DL Query × SWRLTab × OntoGraf × SPARQL Query ×

Object property hierarchy: owl:topObjectProperty

Asserted

```
owl:topObjectProperty
    awayCompetitor
    competesIn
    competitionType
    discipline
    eventGender
    hasCompetedFor
    hasCompetitor
    hasDivision
    hasGroup
    hasHome
    hasRound
        firstRound
        lastRound
    hasSession
        firstSession
        lastSession
    hasStage
        firstStage
        lastStage
    hasUnitCompetition
    isCompetitiveSportingOrganisationOf
    isGroupOf
    isMatchOf
    isRoundOf
    isSessionOf
    isStageOf
    nextRound
    nextSession
    nextStage
    nextUnitCompetition
    prevRound
    prevSession
    prevStage
    prevUnitCompetition
    promotesTo
    recurringCompetition
    relegatesTo
    roundNumber
    roundType
    subDiscipline
    subDisciplineOf
```

Annotations | Usage

Annotations: owl:topObjectProperty

Annotations

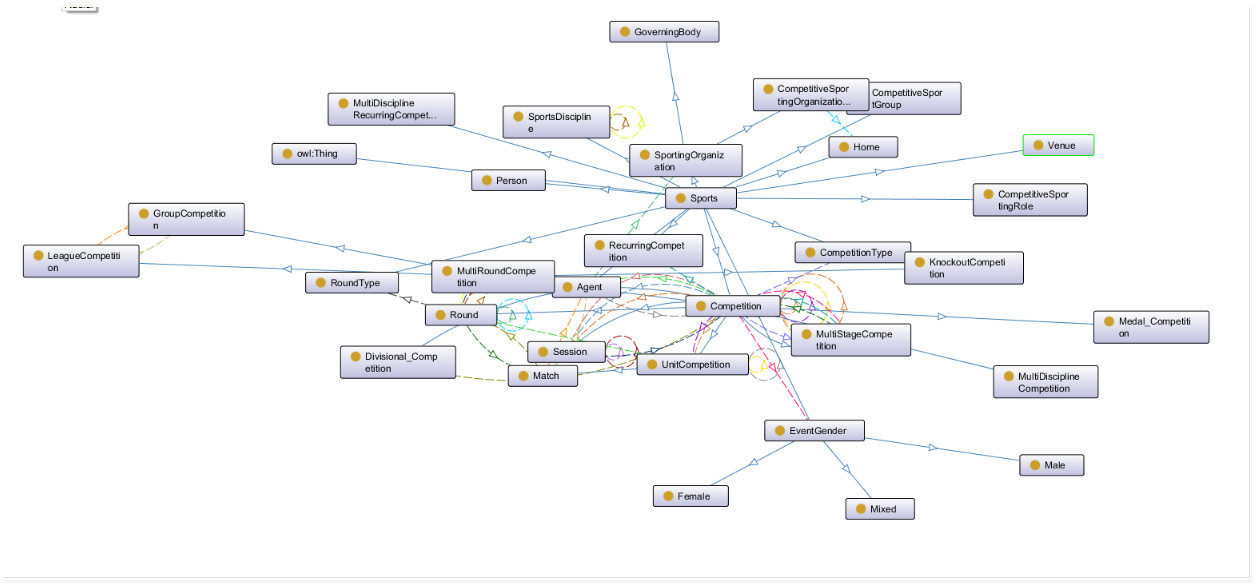
Characteristics: owl:topObjectPr

- Functional
- Inverse functional
- Transitive
- Symmetric
- Asymmetric
- Reflexive
- Irreflexive

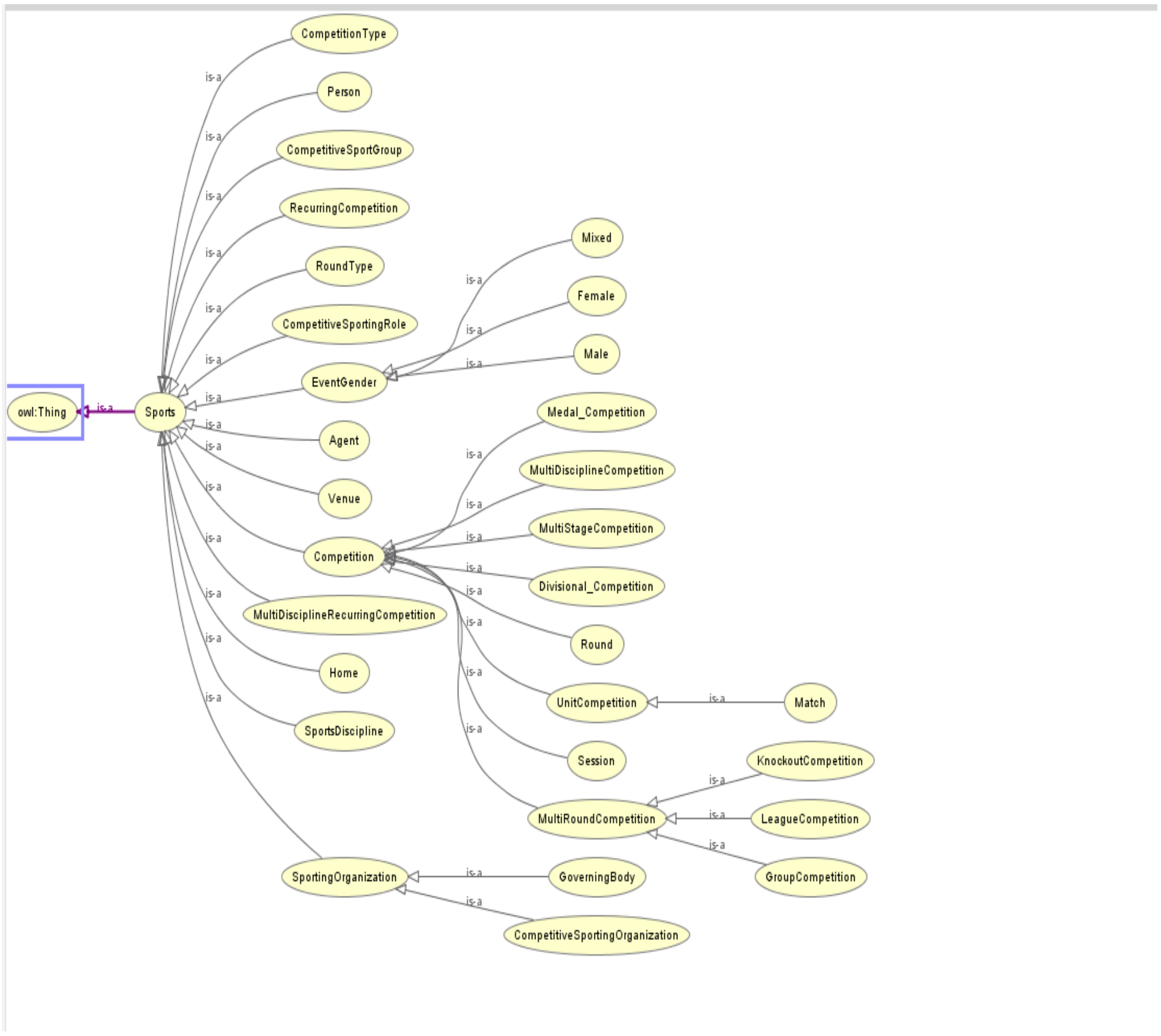
Description: owl:topObjectProperty

- Equivalent To
- SubProperty Of
- Inverse Of
- Domains (intersection)
- Ranges (intersection)
- Disjoint With
- SuperProperty Of (Chain)

Ontograf View:

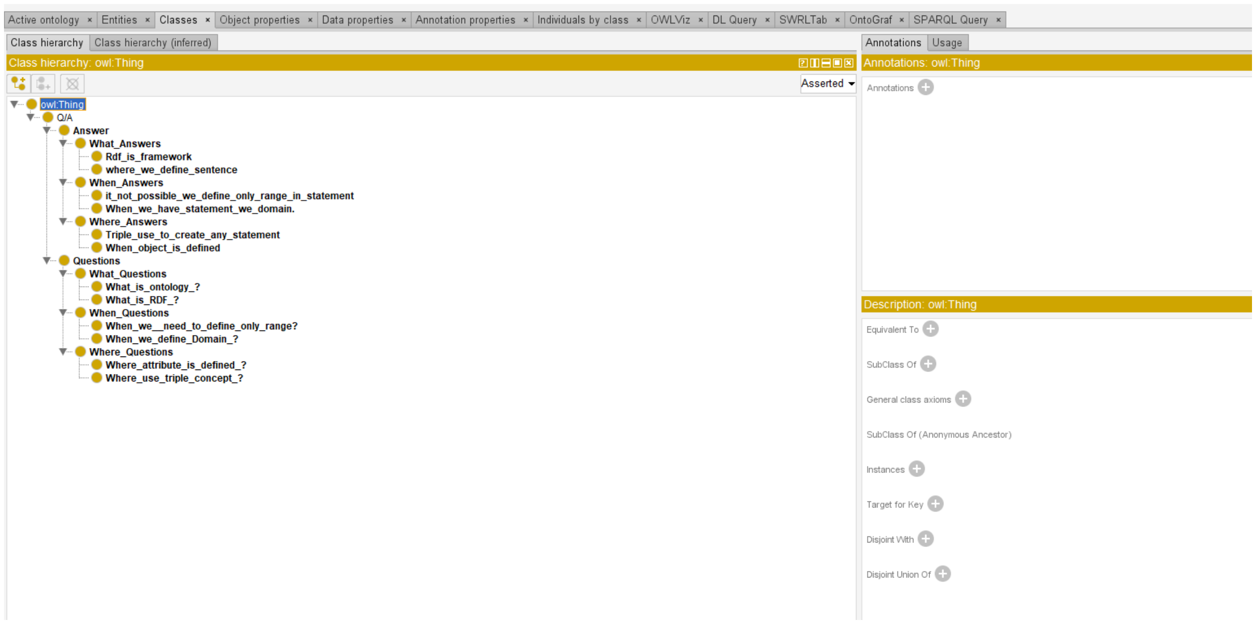


OwlVIZ:


owl:Thing
Sports
CompetitionType
Person
CompetitiveSportGroup
RecurringCompetition
RoundType
CompetitiveSportingRole
EventGender
Agent
Venue
Competition
MultiDisciplineRecurringCompetition
Home
SportsDiscipline
SportingOrganization
Mixed
Female
Male
Medal_Competition
MultiDisciplineCompetition
MultiStageCompetition
Divisional_Competition
Round
UnitCompetition
Match
Session
MultiRoundCompetition
KnockoutCompetition
LeagueCompetition
GroupCompetition
GoverningBody
CompetitiveSportingOrganization
is-a

Lab 17: Q/A system

Classes:



Active ontology × | Entities × | Classes × | Object properties × | Data properties × | Annotation properties × | Individuals by class × | OWLViz × | DL Query × | SWRLTab × | OntoGraf × | SPARQL Query ×

Class hierarchy | Class hierarchy (inferred)

Class hierarchy: owl:Thing

Asserted

```
owl:Thing
  Q/A
    Answer
      What_Answers
        Rdf_is_framework
        where_we_define_sentence
      When_Answers
        it_not_possible_we_define_only_range_in_statement
        When_we_have_statement_we_domain.
      Where_Answers
        Triple_use_to_create_any_statement
        When_object_is_defined
    Questions
      What_Questions
        What_is_ontology_?
        What_is_RDF_?
      When_Questions
        When_we__need_to_define_only_range?
        When_we_define_Domain_?
      Where_Questions
        Where_attribute_is_defined_?
        Where_use_triple_concept_?
```

Annotations | Usage

Annotations: owl:Thing

Annotations

Description: owl:Thing

Equivalent To

SubClass Of

General class axioms

SubClass Of (Anonymous Ancestor)

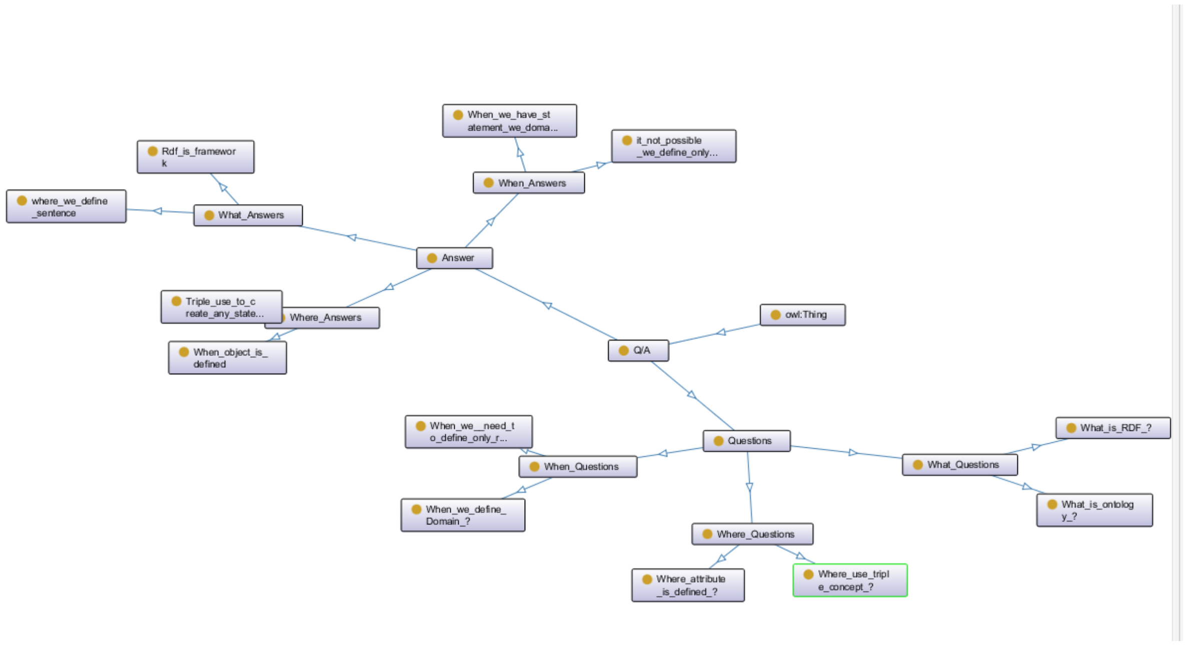
Instances

Target for Key

Disjoint With

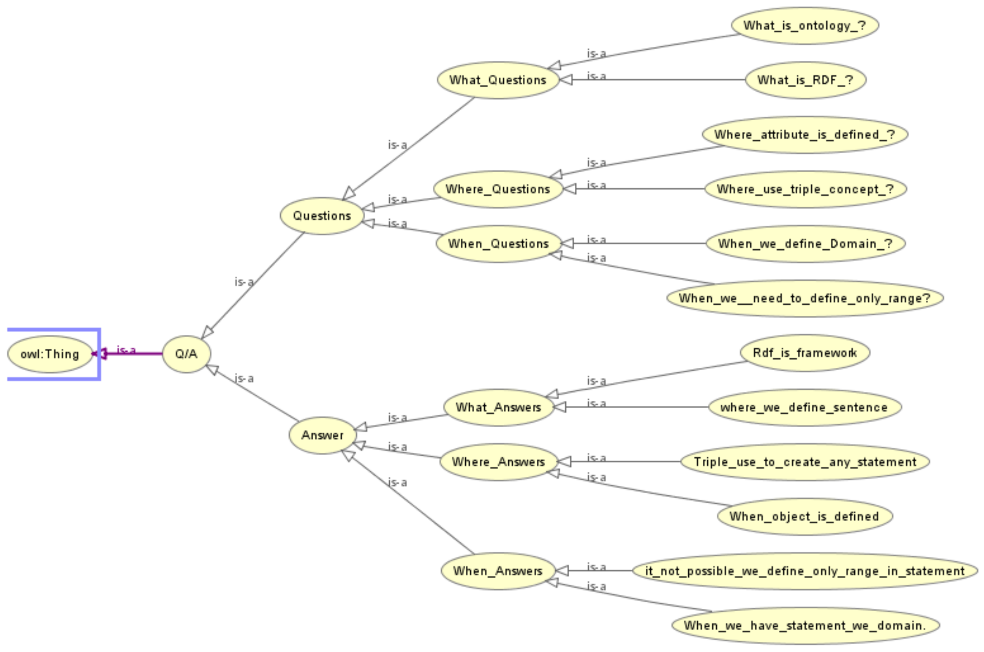
Disjoint Union Of

Ontograf View:



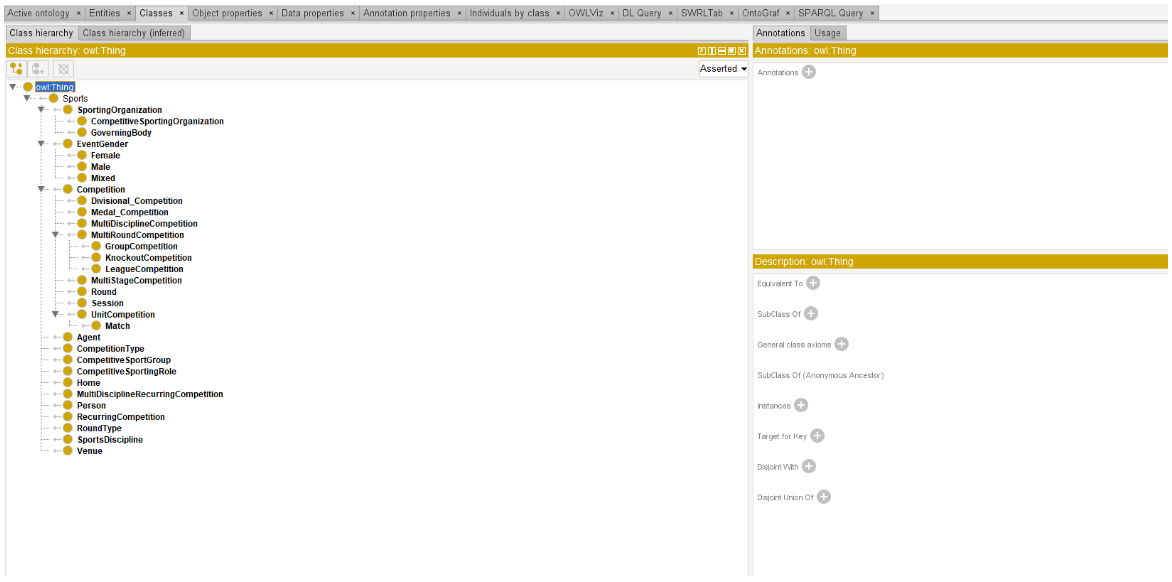
When_we_have_st atement_we_doma...
it_not_possible _we_define_only...
Rdf_is_framewor k
When_Answers
where_we_define _sentence
What_Answers
Answer
Triple_use_to_c reate_any_state...
Where_Answers
owl:Thing
When_object_is_ defined
Q/A
When_we__need_t o_define_only_r...
What_is_RDF_?
Questions
When_Questions
What_Questions
When_we_define_ Domain_?
What_is_ontolog y_?
Where_Questions
Where_attribute _is_defined_?
Where_use_tripl e_concept_?

OwlVIZ:


owl:Thing
is-a
Q/A
Questions
Answer
What_Questions
Where_Questions
When_Questions
What_is_ontology_?
What_is_RDF_?
Where_attribute_is_defined_?
Where_use_triple_concept_?
When_we_define_Domain_?
When_we__need_to_define_only_range?
What_Answers
Where_Answers
When_Answers
Rdf_is_framework
where_we_define_sentence
Triple_use_to_create_any_statement
When_object_is_defined
it_not_possible_we_define_only_range_in_statement
When_we_have_statement_we_domain.

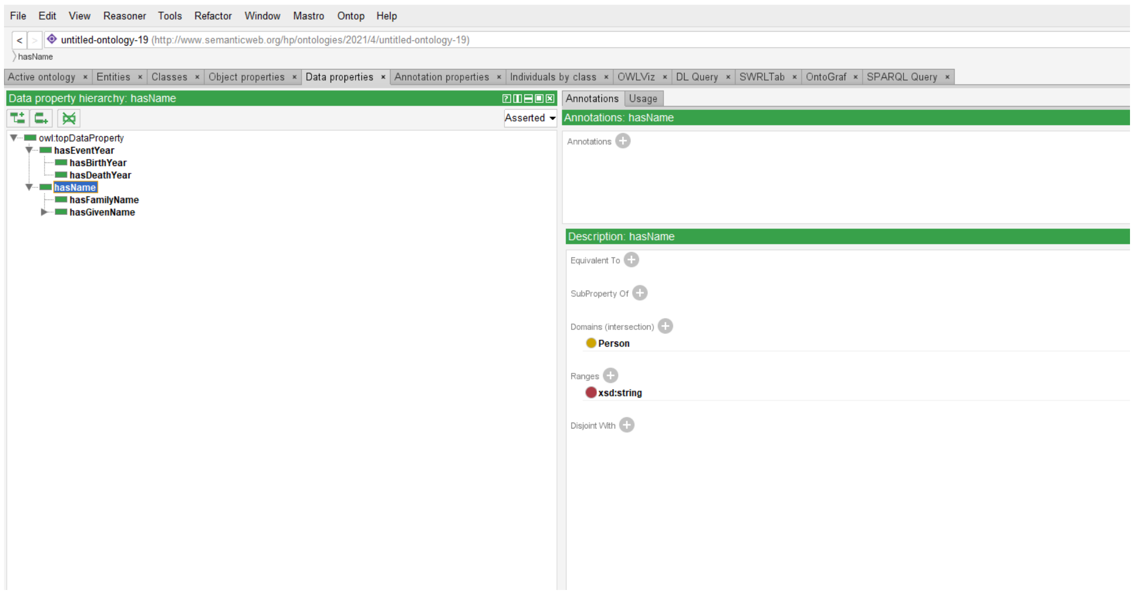
Lab 18: family Ontology

Classes:



```
owl:Thing
    Sports
        SportingOrganization
            CompetitiveSportingOrganization
            GoverningBody
        EventGender
            Female
            Male
            Mixed
        Competition
            Divisional_Competition
            Medal_Competition
            MultiDisciplineCompetition
            MultiRoundCompetition
                GroupCompetition
                KnockoutCompetition
                LeagueCompetition
            MultiStageCompetition
            Round
            Session
            UnitCompetition
                Match
        Agent
        CompetitionType
        CompetitiveSportGroup
        CompetitiveSportingRole
        Home
        MultiDisciplineRecurringCompetition
        Person
        RecurringCompetition
        RoundType
        SportsDiscipline
        Venue
```

Data Properties:

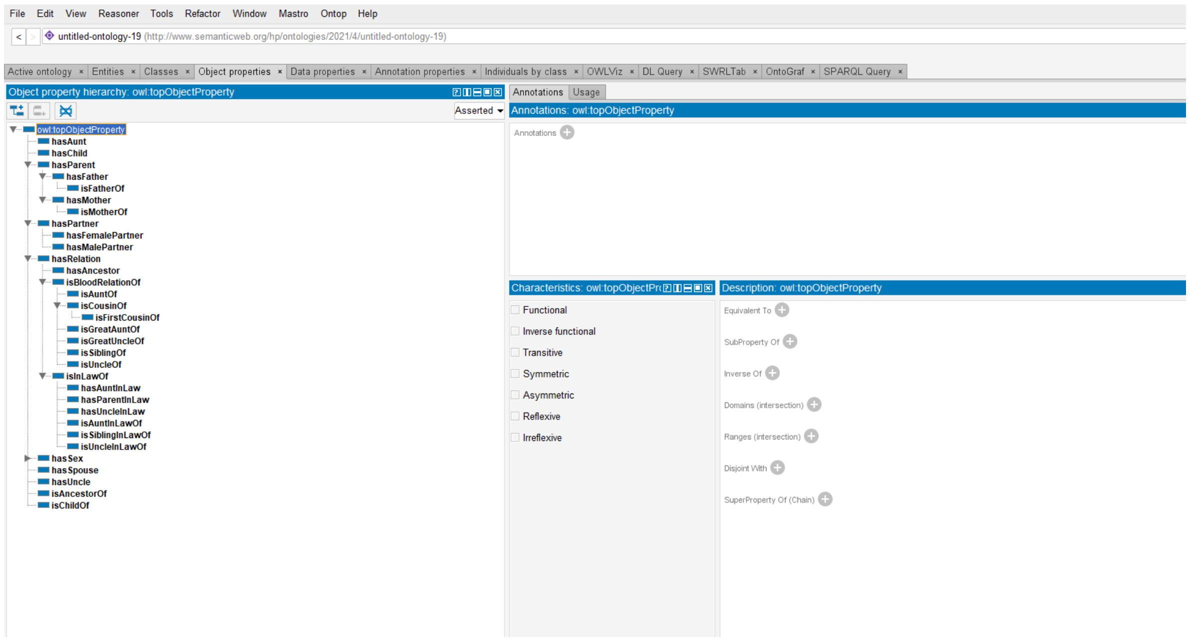


```
owl:topDataProperty
    hasEventYear
        hasBirthYear
        hasDeathYear
    hasName
        hasFamilyName
        hasGivenName
```

Domains (intersection)
Person

Ranges
xsd:string

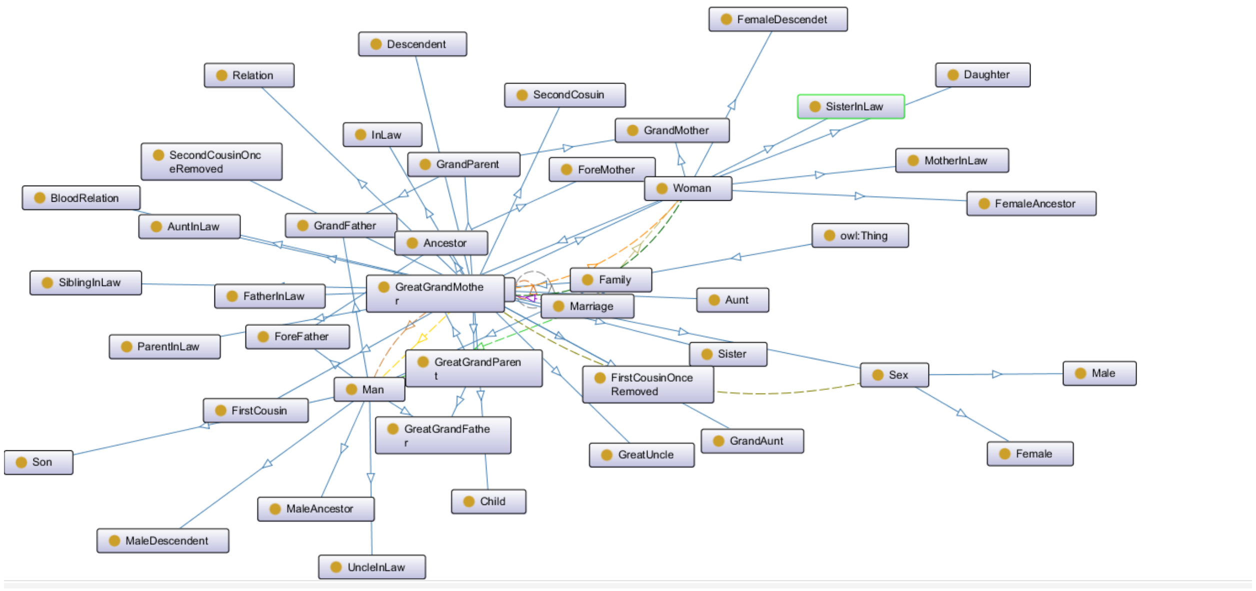
Object Properties:


File Edit View Reasoner Tools Refactor Window Mastro Ontop Help
untitled-ontology-19 (http://www.semanticweb.org/hp/ontologies/2021/4/untitled-ontology-19)
Active ontology × Entities × Classes × Object properties × Data properties × Annotation properties × Individuals by class × OWLViz × DL Query × SWRLTab × OntoGraf × SPARQL Query ×
Object property hierarchy: owl:topObjectProperty
Asserted
Annotations Usage
Annotations: owl:topObjectProperty
Annotations
Characteristics: owl:topObjectPr
Description: owl:topObjectProperty
Equivalent To
SubProperty Of
Inverse Of
Domains (intersection)
Ranges (intersection)
Disjoint With
SuperProperty Of (Chain)

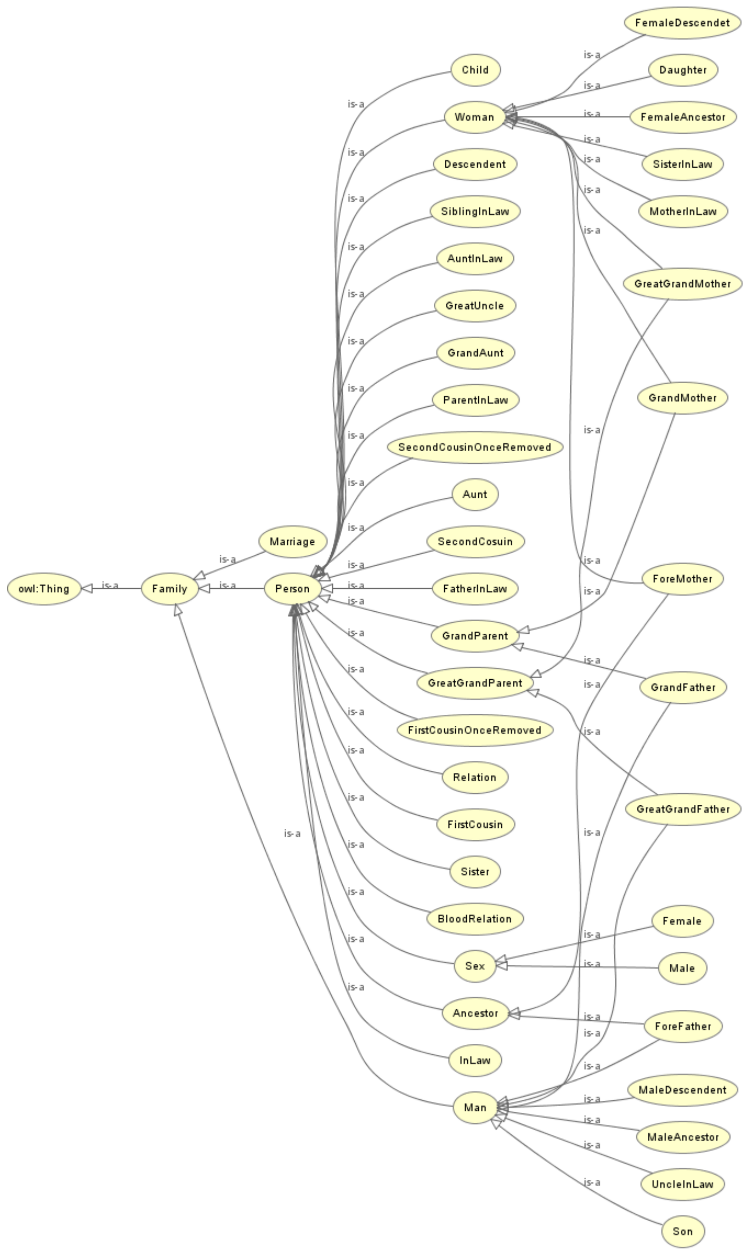
```
owl:topObjectProperty
    hasAunt
    hasChild
    hasParent
        hasFather
            isFatherOf
        hasMother
            isMotherOf
    hasPartner
        hasFemalePartner
        hasMalePartner
    hasRelation
        hasAncestor
        isBloodRelationOf
            isAuntOf
            isCousinOf
                isFirstCousinOf
            isGreatAuntOf
            isGreatUncleOf
            isSiblingOf
            isUncleOf
        isInLawOf
            hasAuntInLaw
            hasParentInLaw
            hasUncleInLaw
            isAuntInLawOf
            isSiblingInLawOf
            isUncleInLawOf
    hasSex
    hasSpouse
    hasUncle
    isAncestorOf
    isChildOf
```

- [ ] Functional
- [ ] Inverse functional
- [ ] Transitive
- [ ] Symmetric
- [ ] Asymmetric
- [ ] Reflexive
- [ ] Irreflexive

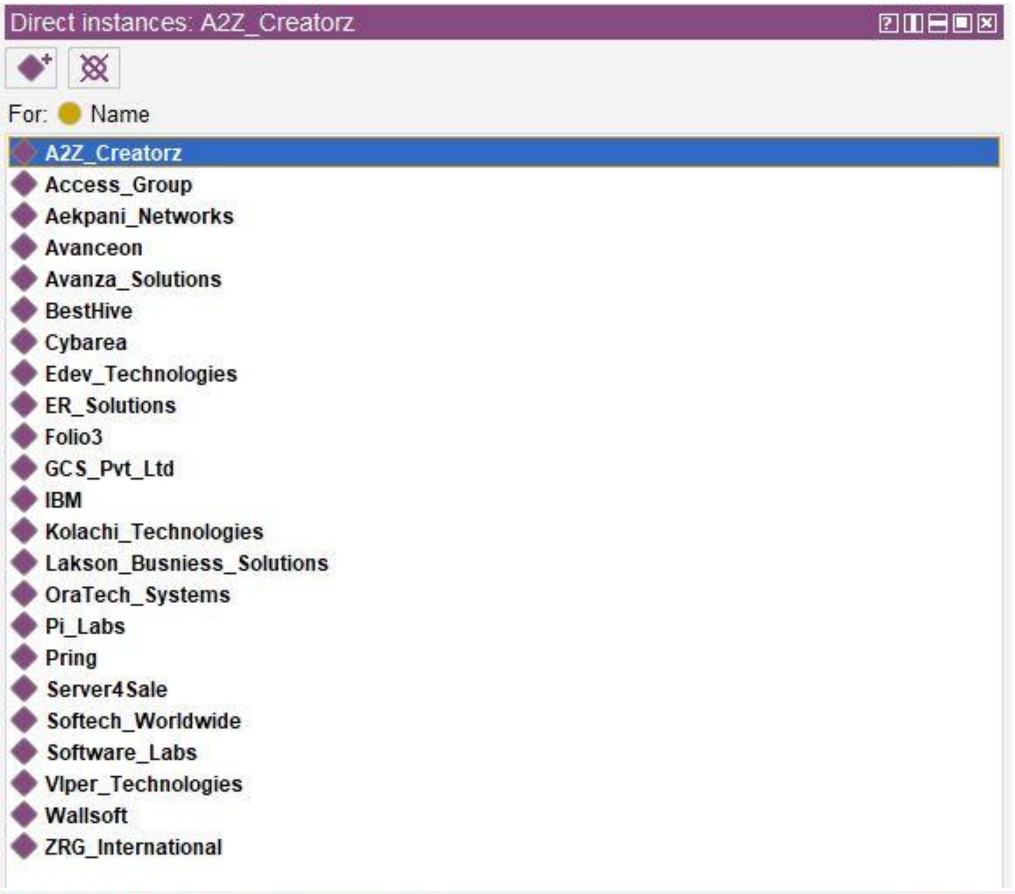
Ontograf View:



OwlVIZ:


owl:Thing
Family
Marriage
Person
Child
Woman
Descendent
SiblingInLaw
AuntInLaw
GreatUncle
GrandAunt
ParentInLaw
SecondCousinOnceRemoved
Aunt
SecondCosuin
FatherInLaw
GrandParent
GreatGrandParent
FirstCousinOnceRemoved
Relation
FirstCousin
Sister
BloodRelation
Sex
Ancestor
InLaw
Man
FemaleDescendet
Daughter
FemaleAncestor
SisterInLaw
MotherInLaw
GreatGrandMother
GrandMother
ForeMother
GrandFather
GreatGrandFather
Female
Male
ForeFather
MaleDescendent
MaleAncestor
UncleInLaw
Son
is-a

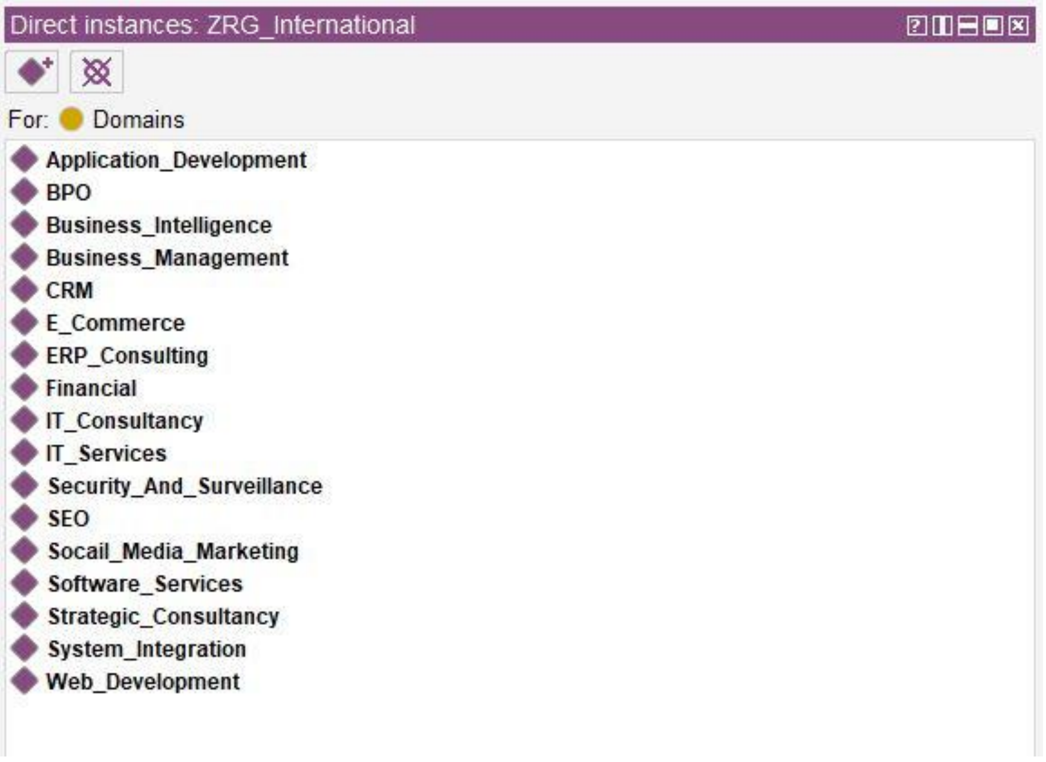
Lab 19: Lab task date (5/5/21)



Direct instances: A2Z_Creatorz

For: Name

A2Z_Creatorz
Access_Group
Aekpani_Networks
Avanceon
Avanza_Solutions
BestHive
Cybarea
Edev_Technologies
ER_Solutions
Folio3
GCS_Pvt_Ltd
IBM
Kolachi_Technologies
Lakson_Busniess_Solutions
OraTech_Systems
Pi_Labs
Pring
Server4Sale
Softech_Worldwide
Software_Labs
VIper_Technologies
Wallsoft
ZRG_International

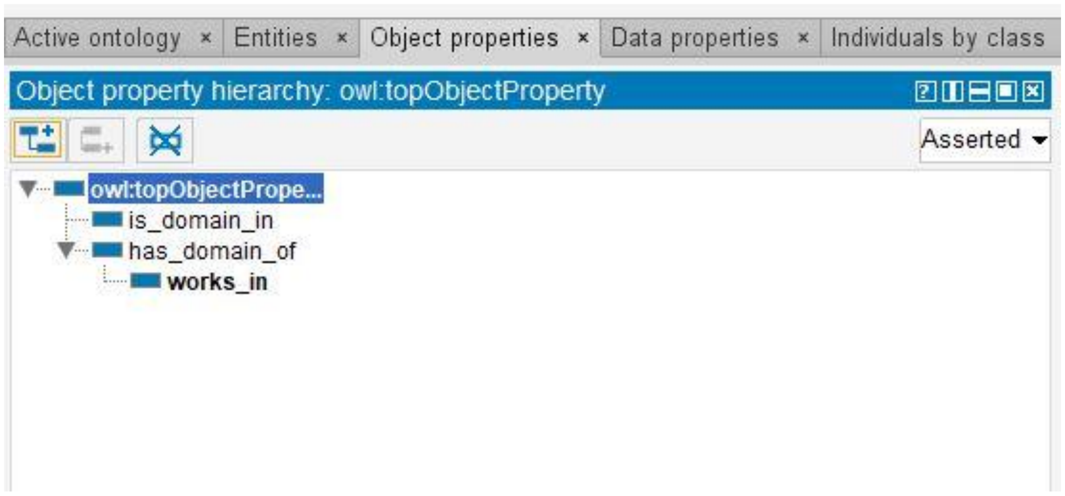


Direct instances: ZRG_International

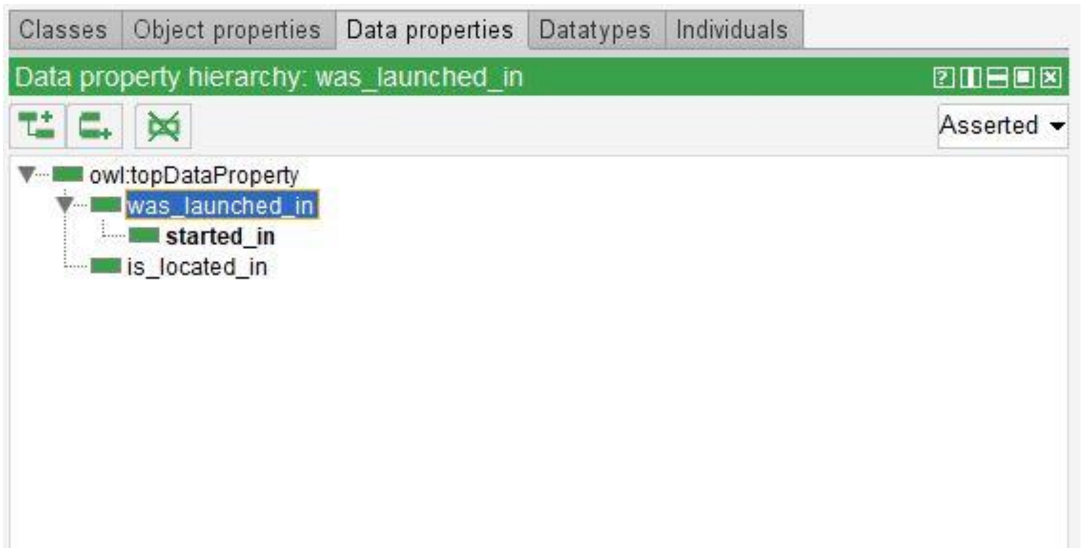
For: Domains

Application_Development
BPO
Business_Intelligence
Business_Management
CRM
E_Commerce
ERP_Consulting
Financial
IT_Consultancy
IT_Services
Security_And_Surveillance
SEO
Socail_Media_Marketing
Software_Services
Strategic_Consultancy
System_Integration
Web_Development

Lab 20: Lab task date (21/4/21)



```
owl:topObjectPrope...
    is_domain_in
    has_domain_of
        works_in
```



```
owl:topDataProperty
    was_launched_in
        started_in
    is_located_in
```

Lab 21: assignment-01 (19/05/21)

Q1: Discuss RDF vocabulary?

Answer: The language defined in this specification consists of a collection of RDF resources that can be used to describe properties of other RDF resources including properties which define application specific RDF vocabularies. The vocabulary is defined in a namespace informally called 'RDFS' here, and identified by the URI.

Q2: How can we maintain consistency and inconsistency between classes and properties? Support your answer through different example.

Answer: Consistency & Inconsistency : consistency is a flow information suppose we design a class so every next class is depended upon previous class or every next class is reflecting the information about every previous class. Every next class is derived almost previous classes. If any inconsistency comes in different classes or different properties, ontologies or data sources may be contradictory so we maintain and set.

$$\begin{tikzcd}
\boxed{\text{Teacher}} \arrow[r, "\text{Age}"] & \boxed{35} \arrow[r, "\text{Course}"] & \boxed{\text{Subject}}
\end{tikzcd}$$

the first query is to collect all the teachers aged 37. The second query asks for all the teachers that have declared an age. The third query, instead, requires collecting all the teachers that teach some courses but not Database, where dashed nodes and lines are introduced to allow a form of negation.

We can see a W-graph model as a Kripke structure and a semantically equivalency query as a formula of the temporal logic CTL. In this way, the inconsistency checking problem is reduced to the problem of finding out the states of the model.

Q3: How do all discussed layering issues will be solved?

The relationship between the various layers in this Semantic Web tower well this does depend somewhat on which layers are being considered, but there are some principles and some basic kinds of relationships that can be considered.

1- **Same-syntax semantic extension:** In this proposed layering, this is the same layering relationship as that between RDF and RDF Schema.
2- **Syntax and semantic extension:** In this proposed layering the semantics of OWL is defined as an extension of the semantics of RDF Schema.

3- **Same-syntax but diverging semantics:** In this proposal layering, OWL syntax would again be RDF syntax, or a subset of RDF syntax, but the meaning of some constructs would be different from their meaning in RDF or RDF Schema
4- **Differing syntax and semantics:** In this proposal layering, OWL differs from RDF and RDF Schema in both syntax and semantics. Again, although the formalisms would diverge, considerable overlap is possible and even desirable.

Solution For layering: One approach would be to change RDF or RDF Schema in some way, perhaps by removing some of the parts of RDF Schema that participate in the paradoxes. However, if we want to keep all of RDF and RDF Schema, it is necessary to pick one of the other solutions

The natural way of defining OWL in this layering proposal is to make the restrictions of OWL be new syntax. These restrictions would have a separate meaning defined by the OWL model theory. For example, a possible syntax for a maximum cardinality restriction like the one above could be:

<owl:cardinality maximum=’’0’’ property=’’friend’’> Person </owl:cardinality>

Q4: What are major requirements for OWL and how do they support our work?

Answer:

The Working Group currently feels that the requirements described below are essential to the language.

Explicit ontology extension: Ontologies must be able to explicitly extend other ontologies in order to reuse concepts while adding new classes and properties.

Ontologies as distinct resources: Ontologies must be resources that have their own unique identifiers, such as a URI reference.

Ontology metadata: It must be possible to provide meta-data for each ontology, such as author, publish-date, etc.

Class and property equivalence: The language must include features for stating that two classes or properties are equivalent

Classes as instances: The language must support the ability to treat classes as instances. This is because the same concept can often be seen as a class or an individual

Cardinality constraints: The language must support the specification of cardinality restrictions on properties.
XML syntax: The language should have XML serialization syntax. XML has become widely accepted by industry and numerous tools for processing XML have been developed.

Q5: Write expressions which can be driven from following:

Answer:

Statement of RDF:

1- Data Analysis with Python Expert.

Domain: Data Analysis

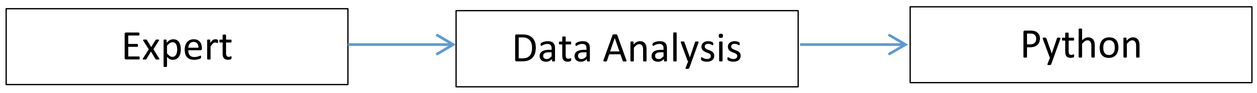
Relation: with

Range: Python

Super Class: Expert

Class: Data Analysis

Subclass: Python



Lab 22: assignment-01 (28/04/21)

Q1: Translate the following sentences into RDF:

1. **Mary is a woman.**

```
<rdf:RDF>
 <rdf:Description about="woman ">
 <s:is a>Mary</s:is a >
 </rdf:Description>
<rdf:RDF>
```

2. **Every mother is a woman.**

```
<rdf:RDF>
 <rdf:Description about="woman ">
 <s:is a>Mother</s:is a >
 </rdf:Description>
</rdf:RDF>
```

3. **Mary is John’s wife.**

```
<rdf:RDF>
 <rdf:Description about="John’s Wife ">
 <s:is>Mary</s:is>
 </rdf:Description>
</rdf:RDF>
```

4. Mothers are women who are also parents.

```
<rdf:RDF>
     <rdf:Description about="women ">
```

```
 <s:are>Mothers</s:are >

 </rdf:Description>

 <rdf:Description about="parents">
 <s:are>Mothers</s:are >

 </rdf:Description>

</rdf:RDF>
```

5. At least one child of a grandparent has also a child.

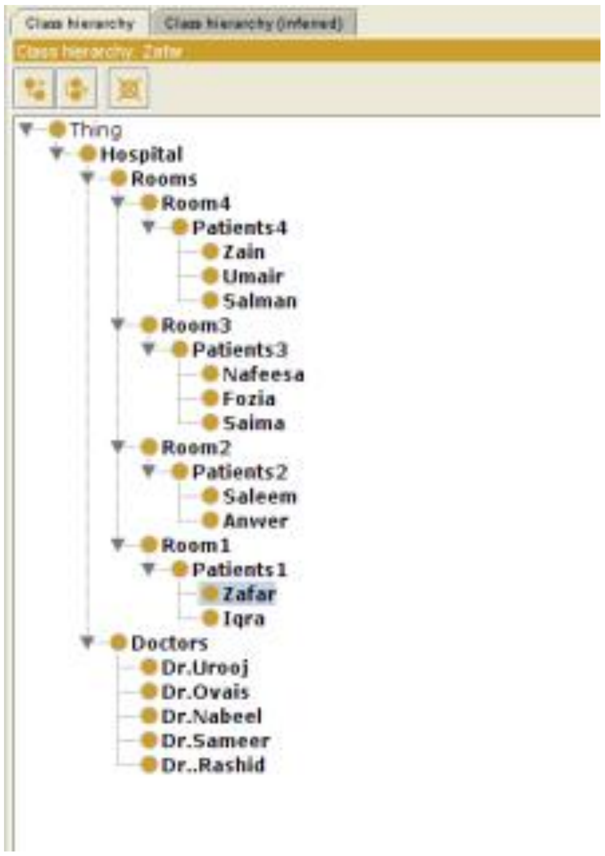
```
<rdf:RDF>

 <rdf:Description about="grandparent">

 <s:is of>child</s:is of>

 </rdf:Description>

 <rdf:Description about="child">

 <s:has a>child</s:has a>

 </rdf:Description>

</rdf:RDF>
```

Q2: Define a RDFS file for a hospital that includes:

(a) four rooms, five doctors, ten patients, and three employees.

(b) Also each room has at least one patient and utmost three patients. Then transform your file with protégé tool to graphical model.

Classes:



Graphical Representation:

Q4: Write SPARQL query for given questions.

1- Which course(s) does a student take?

```
SELECT ?Courses ?Student
        WHERE { ?Course rdfs:CanTake ?Student }
```

2- Which lecturer(s) teach a specific student?

```
SELECT ?Lectures ?Student
        WHERE { ?Lecturer rdfs:Teach ?Student }
```

3- What are the courses offered in first semester?

```
SELECT ?Courses ?Semester
        WHERE { ?Course rdfs:OfferedIn ?Semester }
        FILTER(Semester == First)
```

4- What are the degree awarding institutes?

```
SELECT ?Institute ?Degree
        WHERE { ?Institute rdfs:Awarding ?Degree }
```

5- Which teacher teaches more than 3 courses?

```
SELECT ?Teacher ?CoursesCount
        WHERE { ?Teacher rdfs:Teaches ?CourseCount }
        FILTER(CourseCount == 3)
```

Q3: Open file in Protégé then write and execute the following queries with SPARQL language. Attach queries and the output of them in your report.

a- (label) of the type (e.g., article, book chapter, etc.) and year of the paper.

```
SELECT ?Label ?Year
        WHERE { ?Label rdfs:Type ?Year }
```

b- Find all papers that appear in a venue with"ABC" in the name (using partial string matching), and return a list of title, venue, year and topic label (i.e., not the URI of the topic), sorted by year in descending order, followed by topic in ascending order (for those papers in the same year).

```
SELECT ?Name ?Title ?Venue ?Year
        WHERE { ?Name rdfs:Venue ?"ABC"
                        ?Title rdfs:Venue ?"ABC"
```

```
 ?Venue rdfs:Venue ?"ABC"

 ?Year rdfs:Venue ?"ABC"

                                  }

ORDER BY (?Year)

ORDER BY (?Name)
```

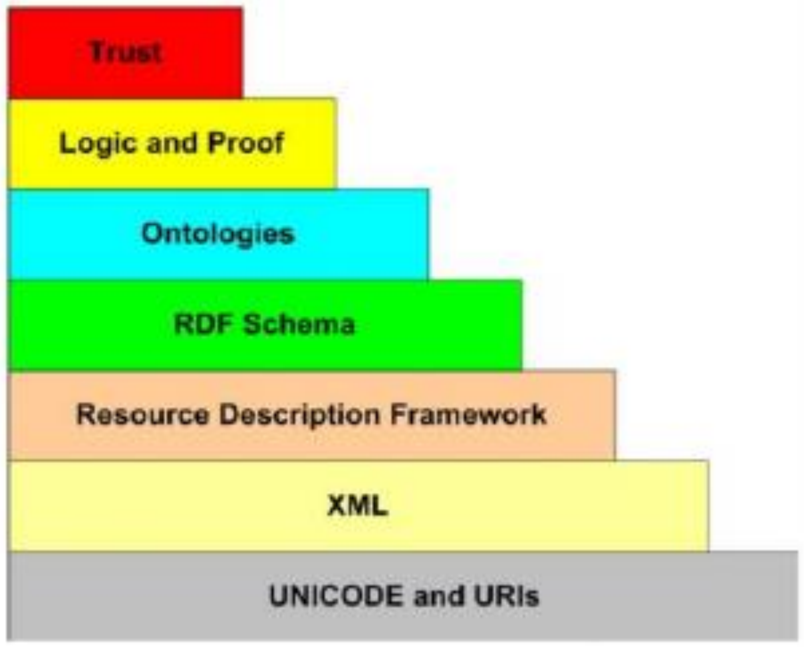
c- For each author, retrieve their name and a count of the number of papers they have authored.

```
SELECT (COUNT (?Papers) as ?Name

          WHERE { ?Paper rdfs:Author ?Name }
```

d- Sort the result by author name. You should assume that two authors are the same if and only if their full names match exactly. For sorting purposes, just use the name as it appears, do not worry about first name or last name distinctions.

```
SELECT ?Name ?Author

          WHERE { ?Name rdfs:is_a ?Author }

          ORDER BY (?Name)
```

Q5: Define major building blocks of semantic web



BUILDING BLOCKS OF SEMANTIC WEB

· **Unicode and URI:**

Unicode, the Standard for computer character representation, and URIs, the standard for identifying and locating resources (such as pages on the Web), provide a baseline for representing characters used in most of the languages in the world, and for identifying resources.

· **XML:**

XML and its related Standards, such as Namespaces, and Schemas, form a common means for structuring data on the Web but without communicating the meaning of the data. These are well established within the Web already.

· **Resource Description Framework:**

RDF is the first layer of the Semantic Web proper. RDF is a simple metadata representation framework, using URIs to identify Web- based resources and a graph model for describing relationships between resources. Several syntactic representations are available, including a standard XML format. RDF Schema: a simple type modeling language for describing classes of resources and properties between them in the basic RDF model. It provides a simple reasoning framework for inferring types of resources.

· **Ontologies:**

A richer language for providing more complex constraints on the types of resources and their properties.

· **Logic and Proof:**

An (automatic) reasoning system provided on top of the ontology structure to make new inferences. Thus, using such a system, a software agent can make deductions as to whether a particular resource satisfies its requirements (and vice versa).

· **Trust:**

The final layer of the stack addresses issues of trust that the Semantic Web can support. This component has not progressed far beyond a vision of allowing people to ask questions of the trustworthiness of the information on the Web, in order to provide an assurance of its quality.

QUIZ :

Q1) Define Queries And Statements from following properties.

Statements (S) And Queries (Q) from properties given:

1. takesCourse:
   S = Ali takes course of physics
   Q = Does Ali take course of physics?

2. isTakenBy:
   S = Physics is taken by Ali.
   Q = Who has taken physics course?

3. isStudiesBy:
   S = Physics is studied by Ali
   Q = Who studies physics?

4. hasLevel:
   S = Temperature has level low
   Q =What is the level of temperature?

5. teacherOf:
   S = Rana is the teacher of Ali
   Q = Who is the teacher of Ali?

6. isTaughtBy:
   S = Akram is taught by Salman.
   Q = Who teaches Akram?

7. tutorOf:
   S = Haseeb is tutor of Uzaif
   Q = Who is Uzaif's tutor?

8. hasGrade:
   S = Talha has grade A.
   Q = What grade does Talha has?

9. IsMemberOf:
   S = Zain is member of Sports Society
   Q = Zain is the member of which Society?

10. hasMember:
    S = Music Society has member as Rashid.
    Q = Who is the member of Music Society?

11. inTutorialGroup:
    S = He is in Tutorial group of PhDs.
    Q = Is he in the PhD’s Tutorial Group?

12. hasTutorial:
    S = Zain has tutorial of python
    Q = Who has tutorial of python?

13. TaughtIn:
    S = List and Tuples were taught in third lecture.
    Q = When were List and Tuples taught?

14. hasAge:
    S = Kaleem has age 23
    Q = What is Kaleem’s age?

15. hasMaxClassSize:
    S = University has max class size for 20 people
    Q = What’s the maximum class size in University?

16. Disjoint:
    S = A.I class disjoints Web Development class.
    Q = What disjoints the Web Development class?

17. teachesCourse:
    S = Nasir Hussain teaches course of A.I.
    Q = Who teaches the A.I course?

18. studiesCourse:
    S = Fayyaz studies course of Cloud Computing.
    Q = Which course do Fayyaz study?

19. hasAssignmentGrade:
    S = Raheel has assignment grade as Zero.
    Q = Who has the Zero assignment grade?

20. hasCourseSemester:
    S = Masters has course semesters containing 4 courses.
    Q = Which degree has four courses in semesters?

21. hasExamGrade:
    S = Yaseen has exam grades above 90 percent.
    Q = Who has exam grades above 90 percent?

Q3) Write down SPARQL Queries for following statements.

1- Access students who are teaching BS program:

```
SELECT ?Student ?BS Program
        WHERE { ?Student rdfs:Teaching ?BS program }
```

2- Name of students starts with letter "A":

```
SELECT ?Name ?A
        WHERE { ?Name rdfs:Startswith ?A }
```

3- Name of teachers whose salary ranges between 50-80k:

```
SELECT ?Teacher ?  Salary
        WHERE { ?subject rdfs:Type ?Salary }
        Filter (Salary > 50 & Salary < 80)
```

4- Name of all drivers who drives on Karachi road:

```
SELECT ?Name ?Karachi road
        WHERE { ?Name rdfs:DrivesOn ?Karachi road }
```

5- Students who mostly gain 65 marks:

```
SELECT ?Student ?mark
        WHERE { ?Student rdfs:gain ?marks }
        FILTER (mark <= 65)
```

6- Age of people lies between 18-25 years:

```
SELECT ?People ?  Age
```

```
WHERE { ?People rdfs:Type ?Age }
Filter (Age > 18 & Age < 25)
```

7- **Birds class who can't survive in hot weather:**

```
SELECT ?birds ?  Hot weather
        WHERE { ?birds rdfs:Survives ?Hot weather }
        Filter ()
```

8- **Different author for same entitled book/chapter:**

```
SELECT ?Auther ?  Book
        WHERE { ?Auther rdfs:Type ?Book }
        Filter (Bookname)
```

9- **Restaurants which open up to 8 hours:**

```
SELECT ?Resturants ?  8 hours
        WHERE { ?Resturants rdfs:Type ?8 hours}
```

10- **Devices which supports only 3G:**

```
SELECT ?Devices ?  3G
        WHERE { ?Device rdfs:Supports ?3G }
```

Q4) Write down summaries of each of the presentation topics.

Summary of Topics:

Semantic Web Background:

It defines about the syntax and semantics. Version of web include

1) Web 1.0 which was used for reading purpose.
2) Web 2.0, used for reading and writing purpose.

3) Web 3.0 is the most advanced than these versions. It is based on ontologies and has ability to read and write at the same time.
4) Some tools which used in Semantic Web are Protege, Jeno, Vowl, etc.

Background of Protege:

Protege is an open source tool to create terminologies and ontologies. In Protégé, we describe the relationship between objects through creating an ontology. The Latest version of Protege is 5.0 and through Protege we can create the ontologies in an efficient way and there is no restriction of creating and expanding an ontology.

Rules Used in Semantic Web: Semantic Web Rule Language (SWRL) such a language for Semantic Web from which one can easily express rules and logics. It basically combines the two of the OWL's species i.e. OWL DL or OWL Lite with a successor of the RML (Rule Markup Language).

Concept in Semantic Web:

Concept refers to an abstract idea or thought that comes from in mind or speech. Everything in semantic web has a different concept. It can be a class, a relation, an attribute and even a statement. Combination of different concepts is called statement.

Terminologies Used in Semantic Web:

- Semantic: meaning
- Ontology: It is a representation of knowledge, properties and relation between the concepts.
- OWL: web ontology language
- RDF: It defines the building blocks of the semantic web such as classes and properties and how they interact to create meaning.
- Triple: A triple is a set of three elements: a subject, a property, and an object.
- SPARQL: Simply a query language used in Semantic Web.
- Class: A concept in the domain
- Properties: Relation between the classes.
- URI: It is a string of characters used to identify a name or a web resource.
- Individuals: individuals are the objects in the domain.
- Disjoint: Two classes in an ontology are disjoint if they cannot share an instance.
- Resource: We can think of a resource as an object.

XML:

The Extensible Markup Language (XML) is a general purpose language which is used to design and describe structured document. XML is based on tags (Own tags) like HTML, However, XML does not have a fixed set of tags but allows users to create their own tags. An XML document consists plain text and markup, in the form of tags and it is represented as a tree. An XML document may contain following nodes, elements, attributes, text, comments, processing instructions and namespaces.

Species of OWL:

There are three species of OWL namely, OWL Full, OWL DL, and OWL Lite. These species define primitives and restrictions of rules within the OWL language use.

1) The OWL Full uses all the OWL language primitives and is fully upward compatible with RDF.

2) OWL DL is the sublanguage of "OWL Full" in which application of constructors to each other is restricted. It is less compatible with RDF.

3) OWL Lite is further restricted as it excludes enumerated classes, disjoints and cardinalities. This makes it easier to understand and implement.

Individuals:

Individual are the objects of class and in Protégé we can describes the individuals using property assertions there are 4 property assertions in individuals. Object property assertion, Negative Object Property Assertion, Data Property Assertion, and Negative Data Property Assertion.

Properties:

Properties are binary relation between domain and range. Property are divided into two categories data and object property. While object have inverse, transitive, inverse functional property, etc. Each have different set of rule to stand uniquely.

Constraints:

Constraints are something that has limitations and restrictions. There are two types of constraints restrictions: values constraints include all values from, some values from and has value and another one is cardinality constraints include min, max and exactly.

Value Constraints:

Constrain is a restriction on someone's behavior. Value constraint is a property restrictions and property restrictions is a special kind of class description. By using constraints we can save our time instead of searching for a specific needed class in whole ontology.it has three types.

1. All value: it helps us to define all values form the class which is asked.

2. Some value: it helps us to define some values from the class/property which is asked.

3. Has value: it describes the individual from the class which is asked.

Reasoner:

Reasoner is service provided by Protege for performing some reasoning task like: ontology consistency, instance checking, classification and class equivalence. There are 2 types of Reasoner used in Protege Hermit and Fact++.

Case Study on Pizza Ontology:

It is a well-known ontology of semantic web community that was designed in University of Manchester for educational purposes only. Basically it is a demo ontology for the students who want to learn semantic web. This ontology includes classes, subclass, object properties, sub properties, individuals, types, same individual as, different individual as, equivalence and visualization.

Case Study on Wine Ontology:

Wine had deployed major impact from the ancient times. By the time and the rise of technology, it has become easier to study and differentiate the variety of wine types and ontologies have taken a major part in this subject. This has put a positive effect for the e-commerce and smart recommendation systems. In the work discussed, there was a clear idea about the effects and influence of web semantics and ontologies on the Machine Learning and recommendation systems. Moreover, how the algorithms make use of these ontologies to work more efficiently and accurately was also discussed.

Q2) Consider Automatic ticket Reservation system to describe basic concepts of RDF. Do you really this everything around us, which we use, is semantic? If yes then how and why? If No then how and why. Support your answer by giving 10 real life examples.

Yes, it's all about semantics as there is advanced web nowadays which is none other than the Web 3.0. It all depends on the words, their relations with each other, statements, competency questions and their dependencies, etc. The data we enter in the Automatic Ticket Reservation System is processed in the backend and not just used to show on the screen or tab. This all possible due to the Web 3.0, and the ontologies and query language support.

Examples:

- Student fills form for admission in university utilizes Web Semantics
- You search for best food across town, it uses Web Semantics and give answers according to ontologies.
- A person Amazon search engine to find product he likes, what he searches, the keywords are processed in the form of ontologies and the answer comes on the screen.