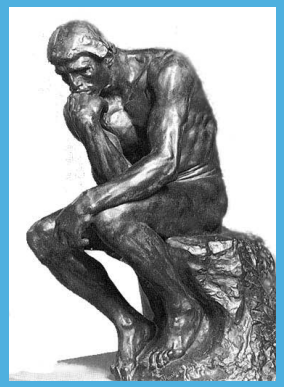
Think Ontologically

A brief account about ontology modelling for archaeologists



Yi Hong and Emilio Tuosto
Department of Computer Sciences
University of Leicester

What is archaeology?

- A few definitions:
  - "[Archaeology] is the method of finding out about the past of the human race in its material aspects, and the study of the products of this past." Kathleen Kenyon, 1956. *Beginning in Archaeology*. Phoenix House, London.
  - "Archaeology is what archaeologists do." David Clarke, 1973 Archaeology: the loss of innocence. *Antiquity* 47:6-18.
  - "Field Archaeology is the application of scientific method to the excavation of ancient objects, and it is based on the theory that the historical value of an object depends not so much on the nature of the object itself as on its associations" C. Leonard Woolley, 1961. *Digging up the Past. Penguin, Harmondsworth.*

Source: K. Kris Hirst, A Compilation of Quotes by Geoff Carver

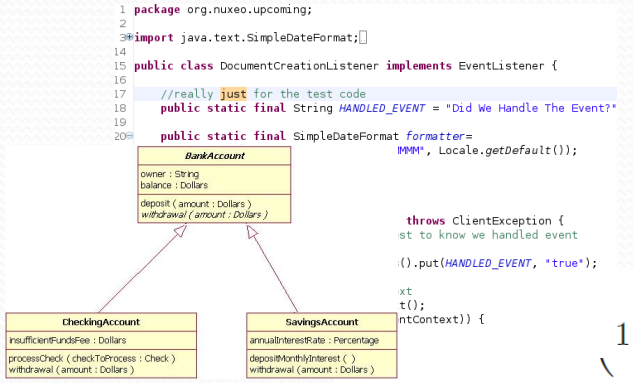
How archaeologists describe knowledge?

[Apologies for the oversimplification]

- "Unstructured" text
  - natural language
  - **human readable**
  - highly expressive
  - non-standard terminology
- To allow individuals to have maximal freedom in the analysis archaeological evidence

How computer scientists describe knowledge?

- **Formal Specification**
  - programming language, metadata
  - machine readable and **processable**
  - controlled vocabulary
  - limited expressiveness
  - mathematical definition and notations

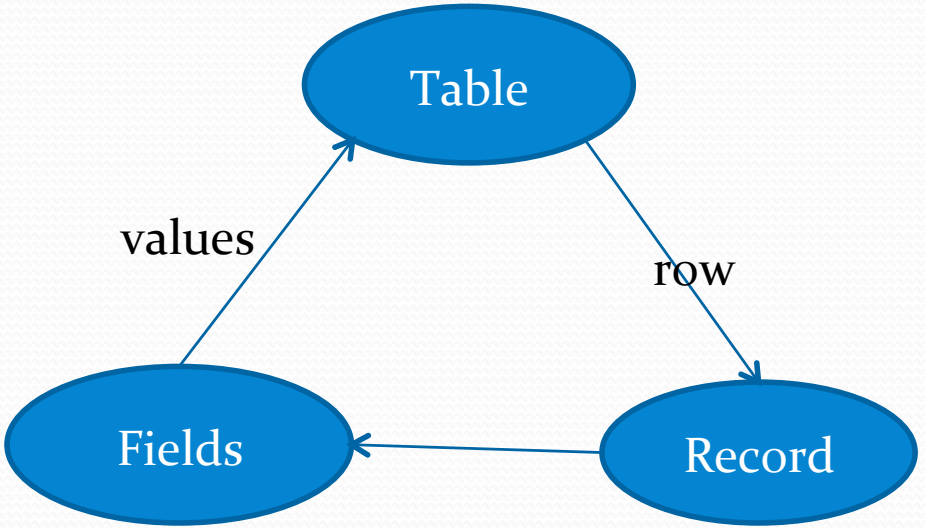


"Computer language design is just like a stroll in the park. Jurassic Park, that is."
-Larry Wall inventor of "Perl"

$$\left(1 - \frac{\Sigma_{s \in D(s)} \Sigma_{a \in U(s)} |CF(u,s) - CF(a,s)|}{|D(s)|}\right)$$

Data bases… traditionally

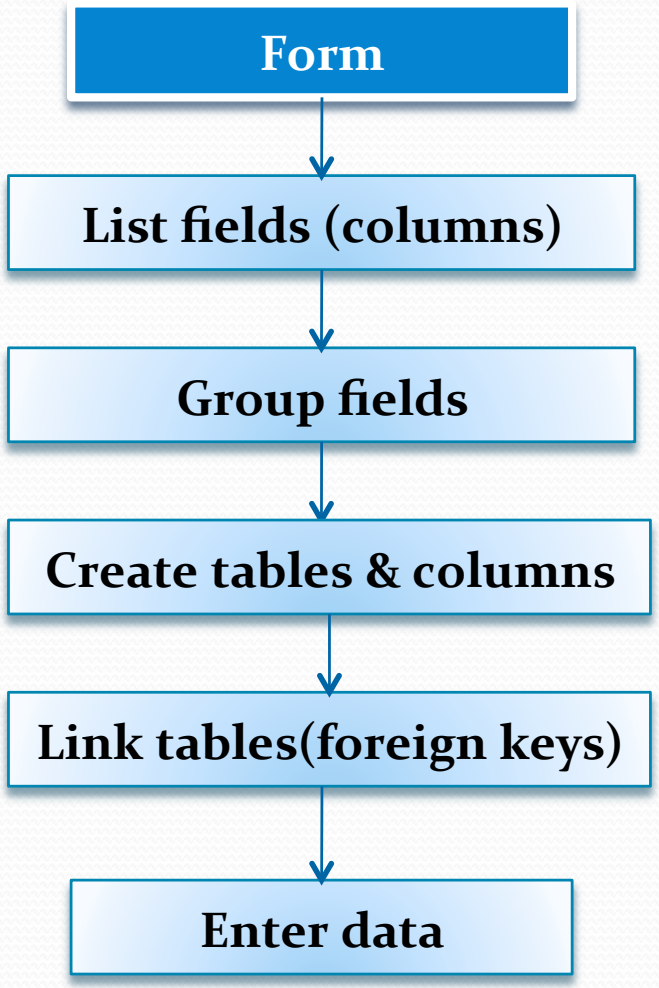
Relational DBs



Identified/linked by

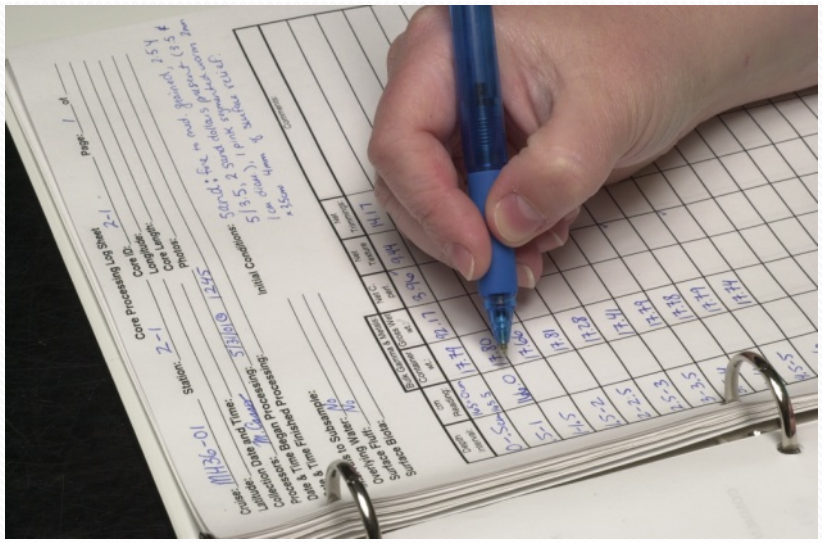
Primary/Foreign Keys

Process to create a database

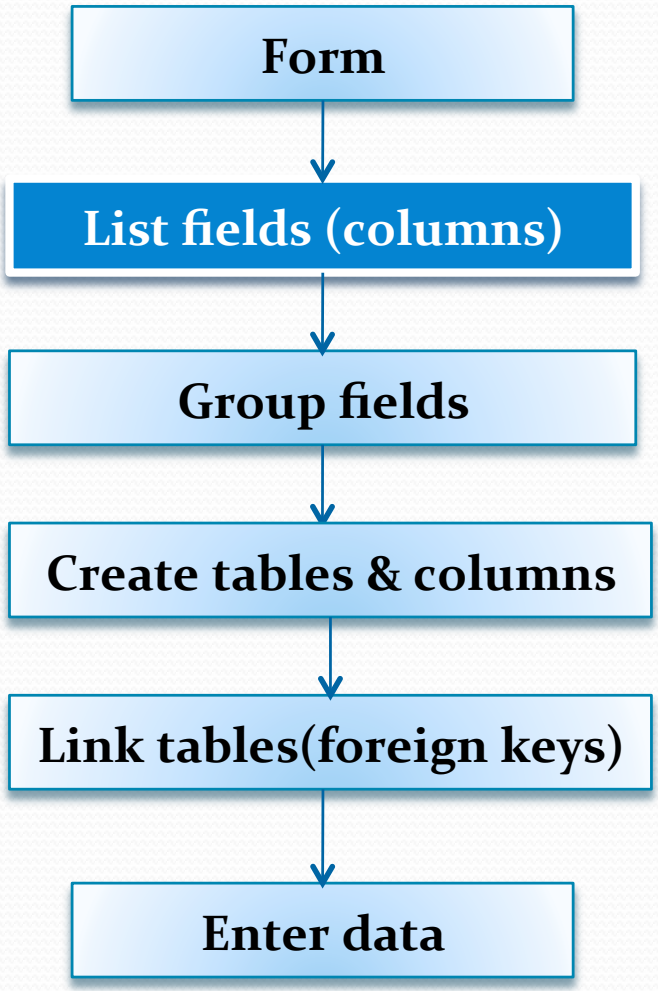


Relational Database approach

Step 1.
paper-based or electronic form



Process to create a database

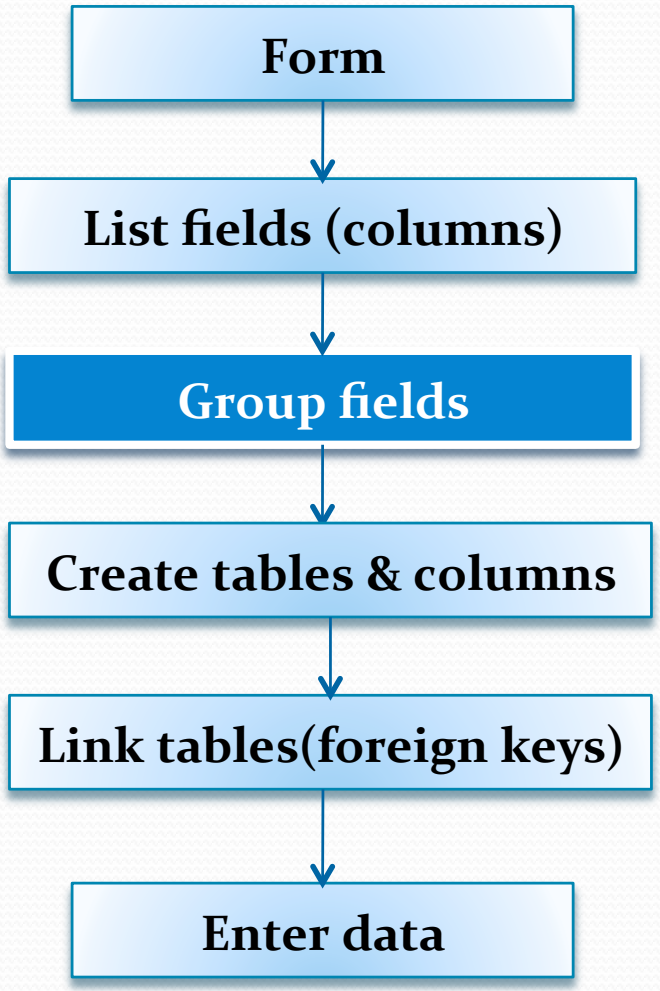


Relational Database approach

Step 2. List all fields

- Inventory Number,
- Excavation Date,
- Site name,
- Site description,
- Material,
- Weight,
- Catalog,
- Images,
- Object Description

Process to create a database

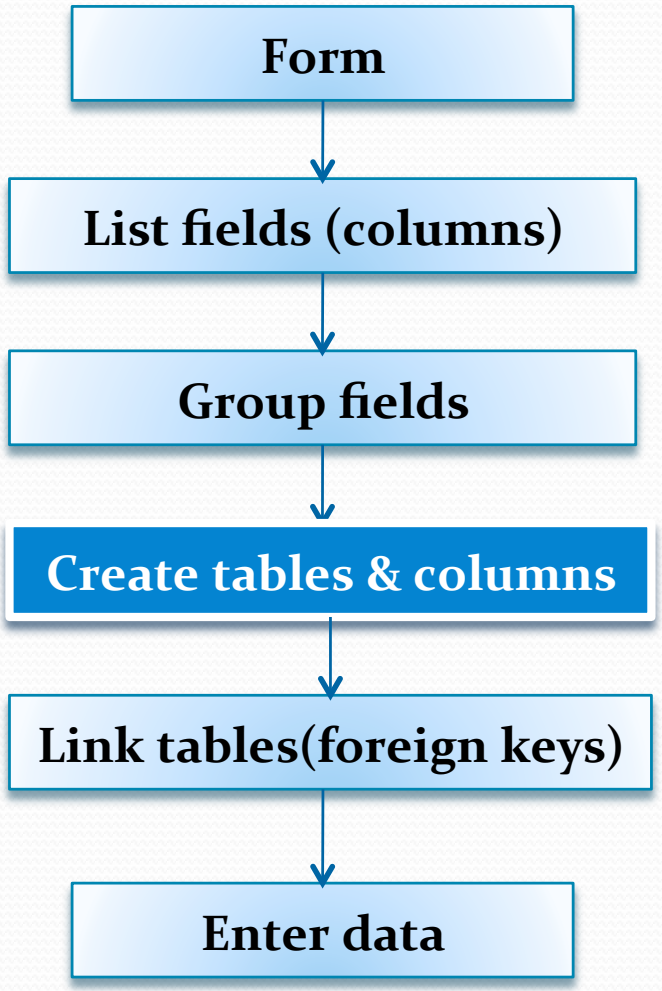


Relational Database approach

Step 3. Group fields, give a group name

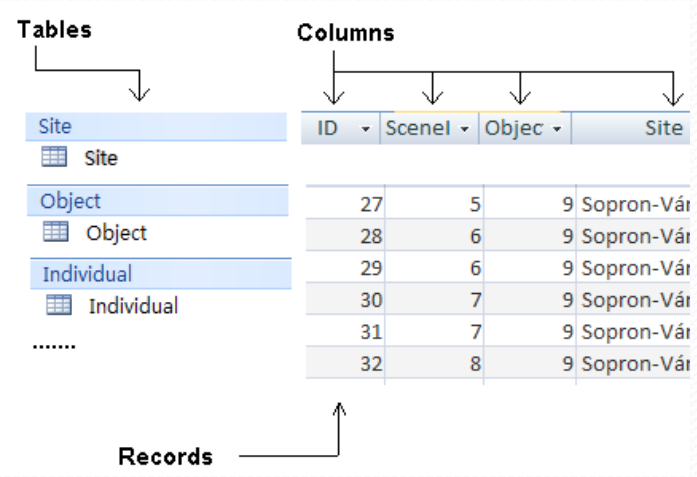
- **Object**
  - Inventory Number
  - Excavation Date.
  - Material
  - Weight
  - Catalog
  - Images
  - Object description
- **Site**
  - Site name
  - Site description

Process to create a database

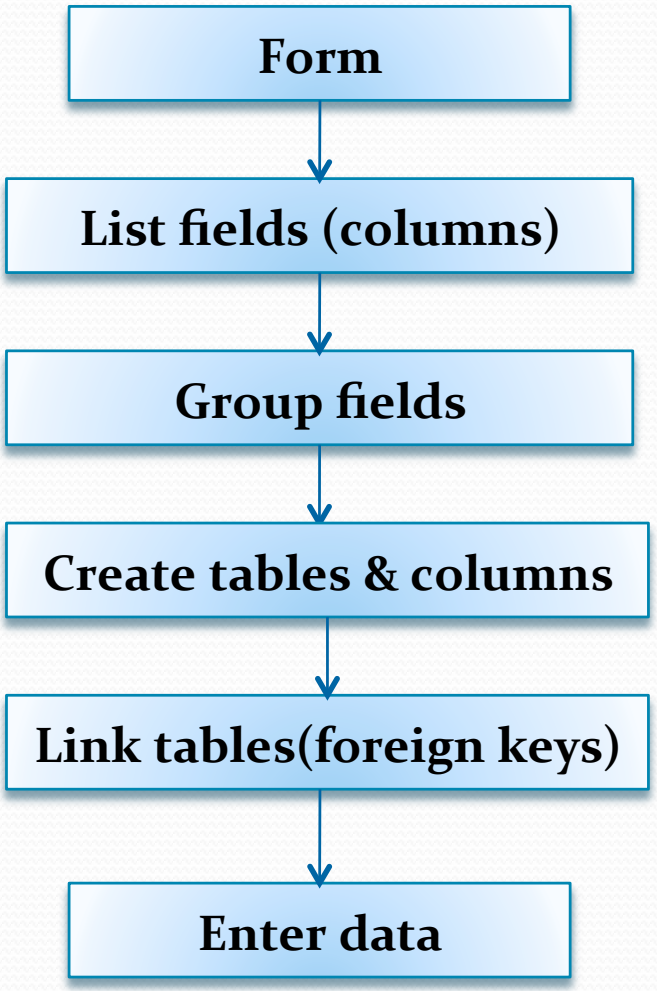


Relational Database approach

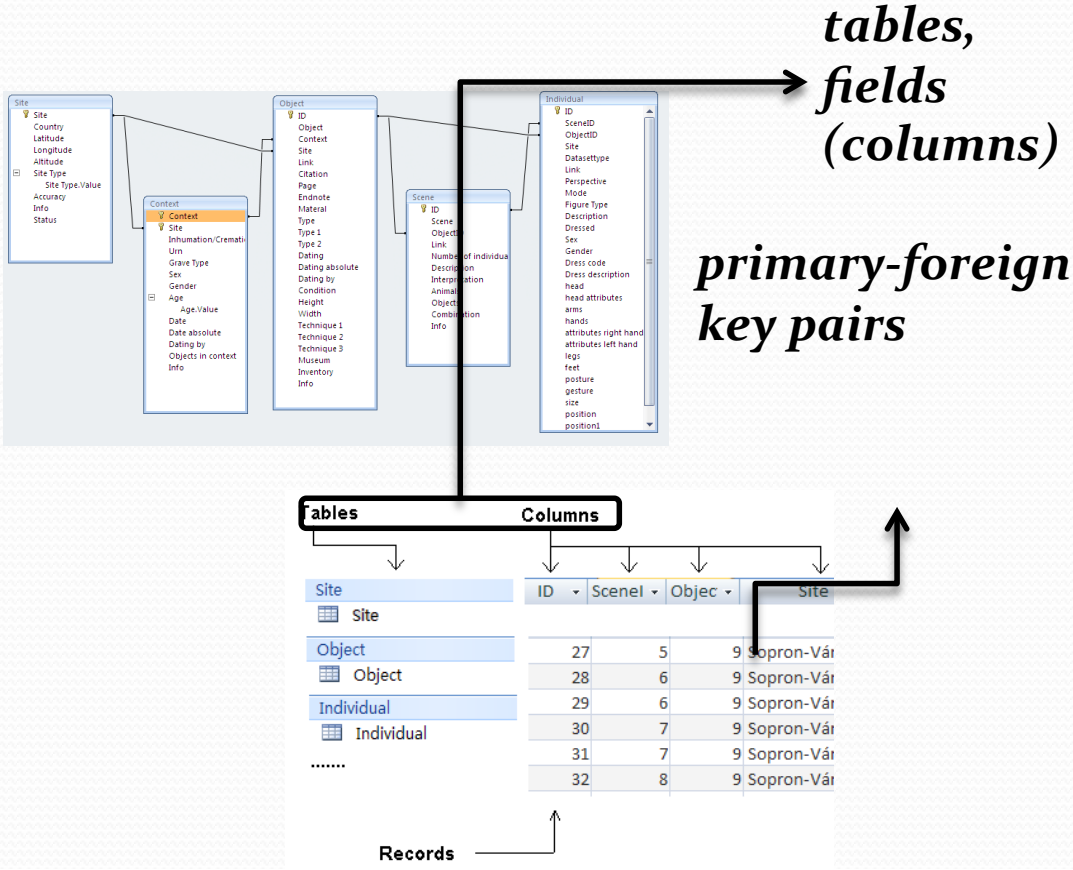
Step 4. create tables, columns



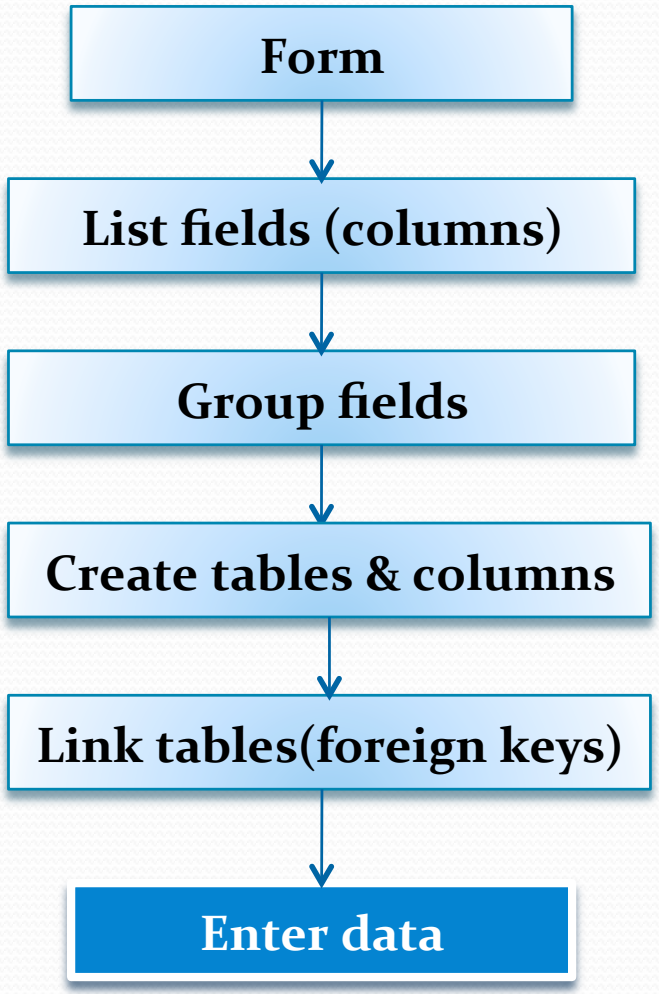
Process to create a database



Relational Database approach

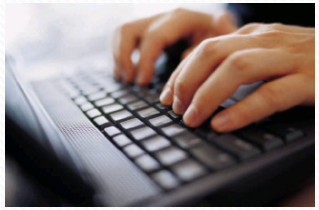


Process to create a database



Relational Database approach

Step 4. Data entry



Example

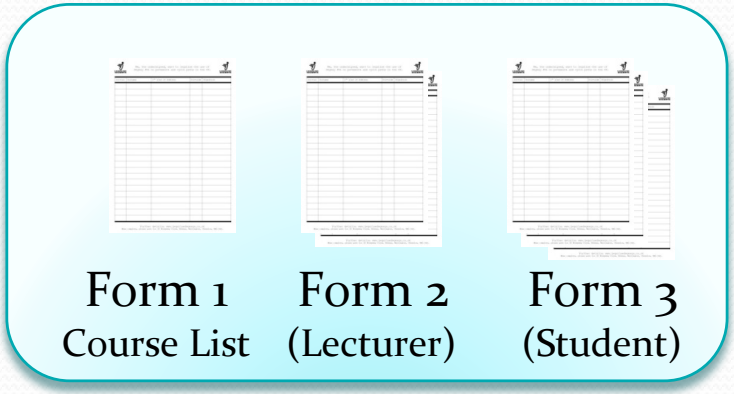
Scenario :

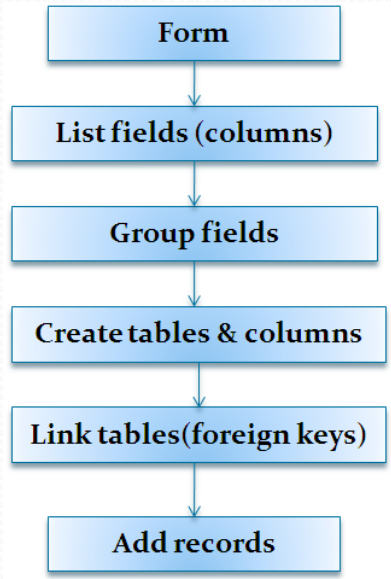
Design a data structure for **Course Modules Selection**

Requirements:

(1) Every course module can only be taught by one Lecturer
(2) A Lecturer can teach many courses.
(3) A student must enrol in at least **3** courses.
(4) Max enrolment number for a course is **100.**

Example continued….





* Every student has a unique student ID
* Every staff has a unique staff ID
* Many to many relation

Tables

Student

Student ID	Name	Email	….

Lecturer

Staff ID	Name	Email	….

Course

Course ID	Course Name	….

ModuleSelection

Course ID	Staff ID	Student ID

Problems with Relational database

What if…

- someone is both a student and staff (e.g. staff doing part-time PhD)
- we want to add a new table for distant learning module?
- we want to store multiple email addresses of a person?

How to …

- Specify the constraints that a student must enrol in at least **3** courses and the max enrolment number for a course is **100?**

Student

Student ID	Name	Email	….

Lecturer

Staff ID	Name	Email	….

Course

Course ID	Course Name	….

ModuleSelection

Course ID	Staff ID	Student ID

Problems with Relational database

- Fixed schema
  - Not flexible enough to describe complex scenarios
  - Changes to schema might lead to major unintended impacts
- Relationships
  - The relationships among tables and various logical constraints are not always clear to the user because they are defined implicitly by primary/foreign key pairs.
- Interoperability
  - Importing/reusing table structures is hard.
- Usability of the search
  - Form-based query filters are still very limited in their expressivity
- Insufficient reasoning support
  - Limited automatic reasoning
  - Unable to query over relationship

Problems with Relational database

Querying

- Queries relational database support:
  - List all students taking a specific module
  - List course modules taught by a specific Lecturer
  - List students attending any lecture that is taught by a specific Lecturer

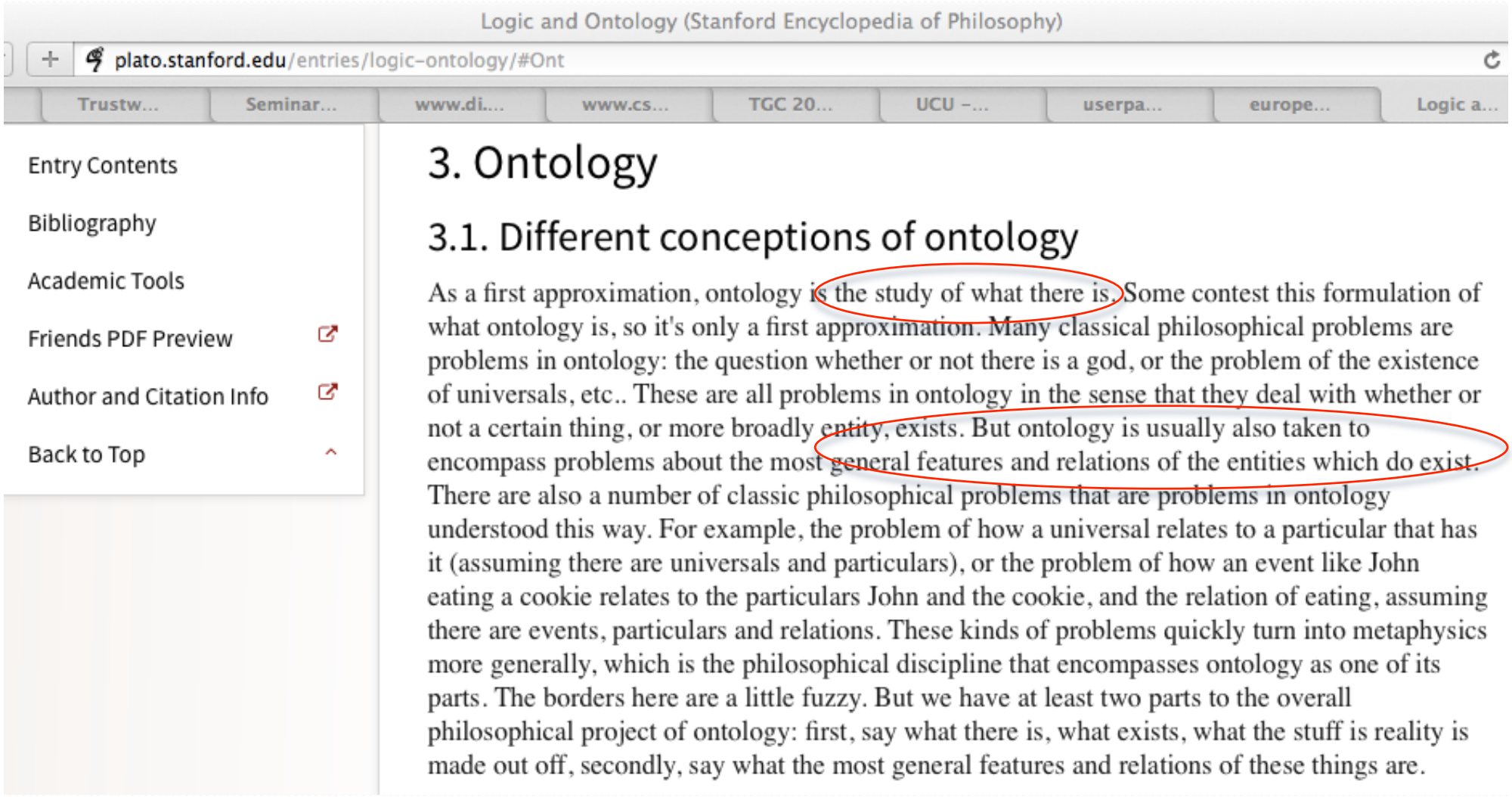
  ....

- Queries relational database are not able to deal with:
  - show all possible relationship between two person, Tom and Kate. (e.g. They could be classmate, teacher-student relationship colleague etc. But whether Tom is a lecturer or a student is not known to us)
  - show all person that are related to a given module.

  .....

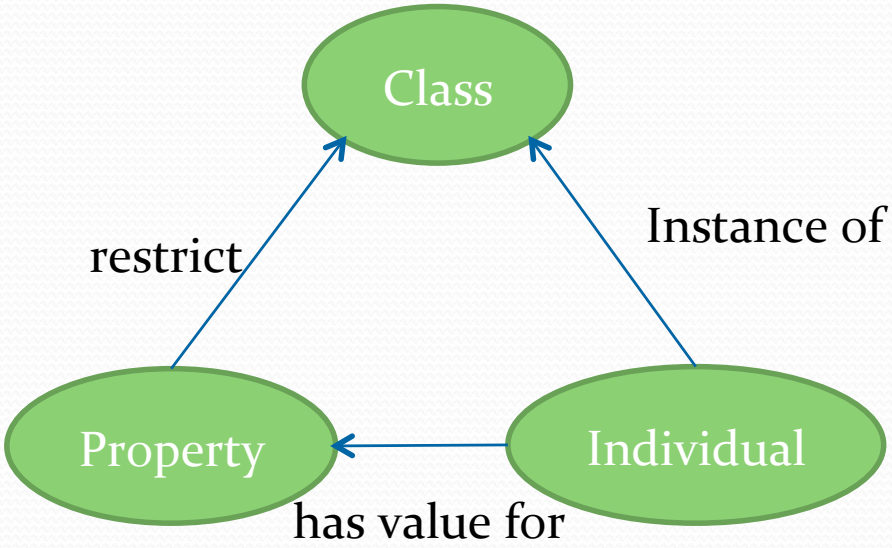
Think Ontologically

What does ontology mean?

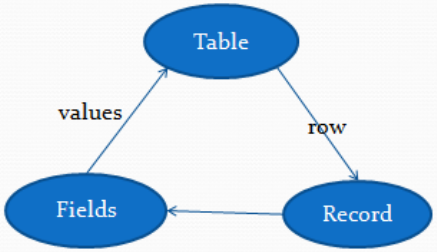


A conceptual view of ontologies

Ontology

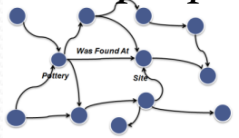
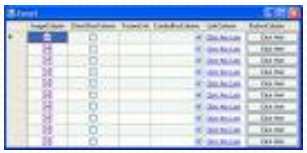
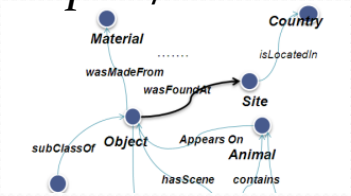


Identified/linked by
(URI) Uniform Resource Identifier



Identified/linked by
Primary/Foreign Keys

Relational Database vs Ontological Database

Define Structure	***Schema (SQL)*** *(table, field)* 	***Ontology (RDF/OWL)*** *(class, property)* 
Basic elements	*Records* 	*Triples/ axioms* 
Products/ Editor	Relational DB ***Microsoft Access, FileMaker, Oracle, MySQL etc***	OntoDB (Triple store) ***Protége, SWOOP***
		Virtuoso , Jena SDB, TDB etc.
Query	SQL (Structured Query Language) Form-based filter and search	SPARQL Graph pattern

Open vs Closed World Assumption

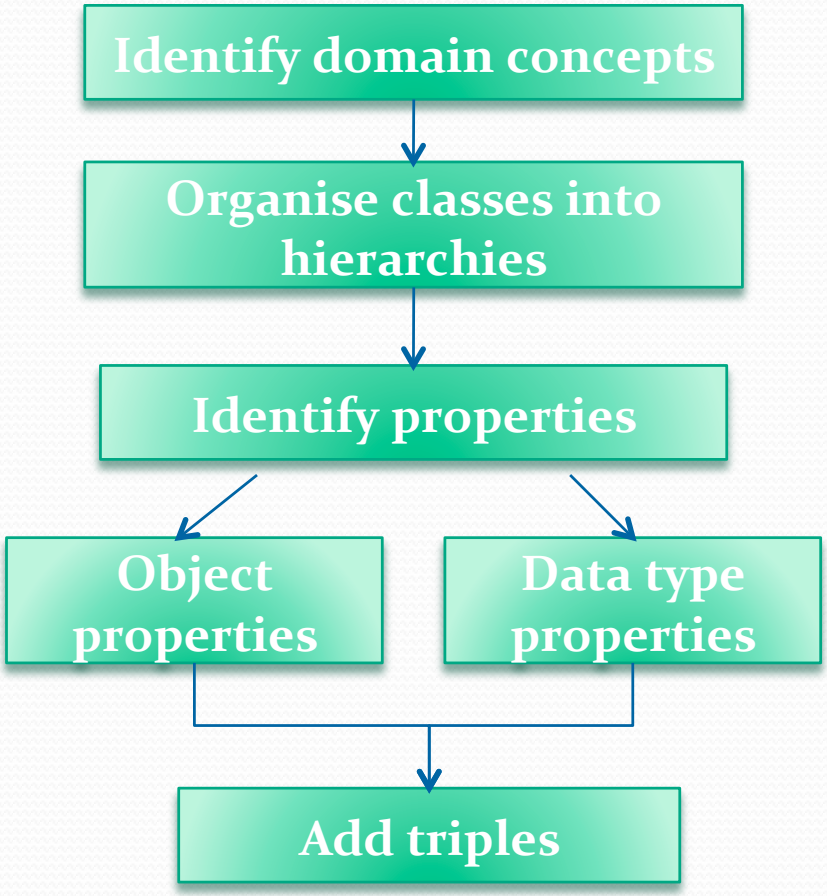
Yi eats
chicken tikka
boiled rice
potato salads
tofu

Alessandro eats
pasta
boiled rice
potato salads
hash browns

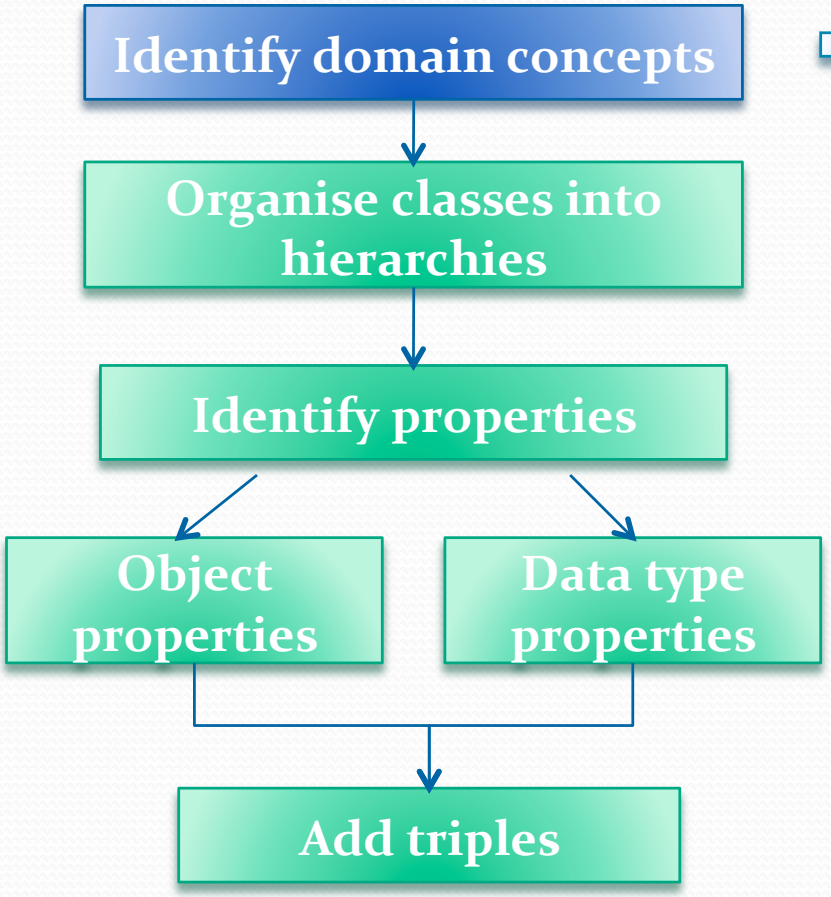
Open World Assumption
Knowledge is **incomplete**, hence we have to admit **undefined** answers

Vegetarian?	Yi	Alessandro
Closed world assumption	No	Yes
Open world assumption	No	???

Abstractions on classes and properties



Classes abstract away Concepts



Class (Concept)

A template of definition of a particular kind of object

e.g. Person, Ceramic Pot , Excavation Site, Color

Describe Classes with Description Logic

$\top$	$\top$
$\bot$	$\bot$
$\sqcap$	$C \sqcap D$
$\sqcup$	$C \sqcup D$
$\neg$	$\neg C$
$\forall$	$\forall R.C$
$\exists$	$\exists R.C$
$\sqsubseteq$	$C \sqsubseteq D$
$\equiv$	$C \equiv D$
$\doteq$	$C \doteq D$

Description Logic

Examples

"Circular terracotta loom weight"

→ $WeavingTool \sqcap Disc \sqcap \forall madeFrom.Terracotta$

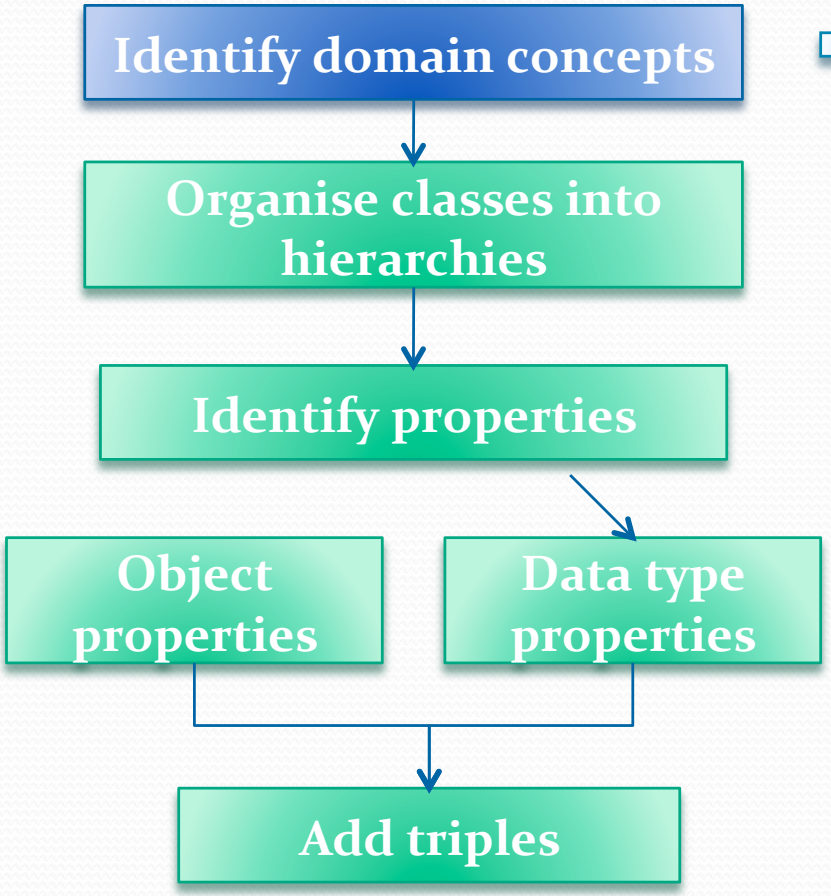
"Excavation site"

→ $Site \sqcap \exists foundObject.Artefact$

"Settlement"

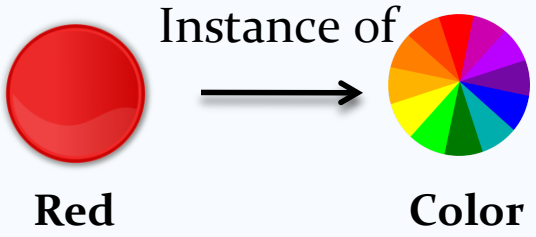
→ $Site \sqcap \exists hasResident.Human$

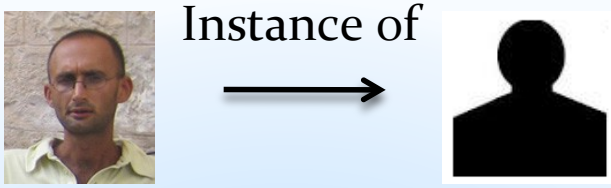
Classes Have Individuals as Members



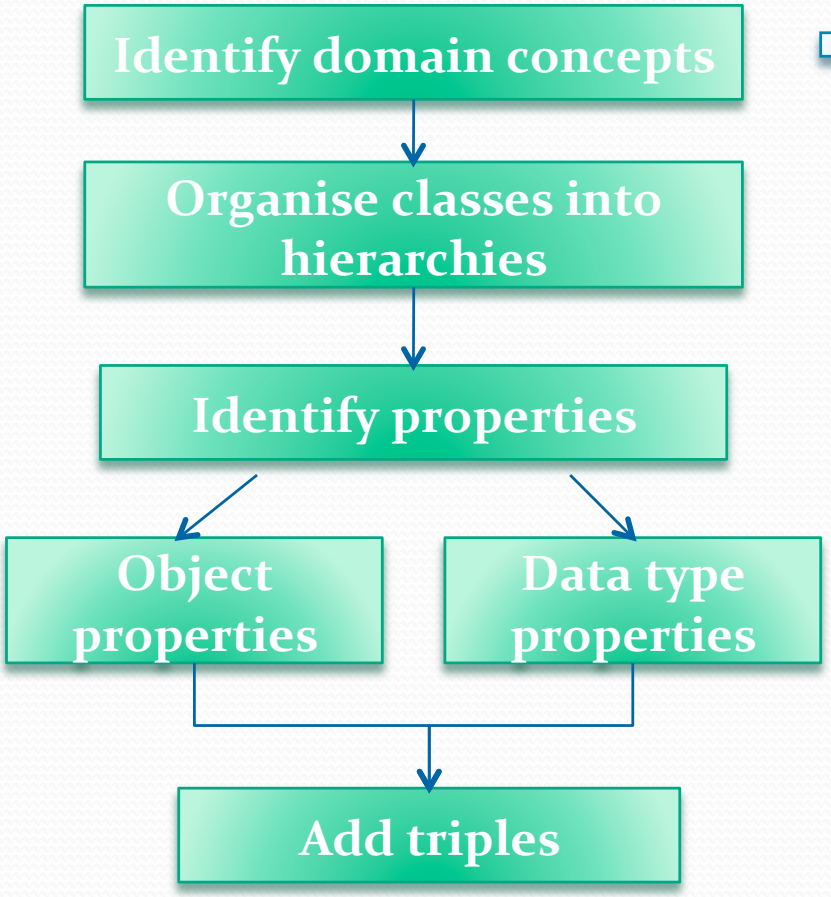
Individual (aka object/ instance)

Examples:





Individual



Individual (object)

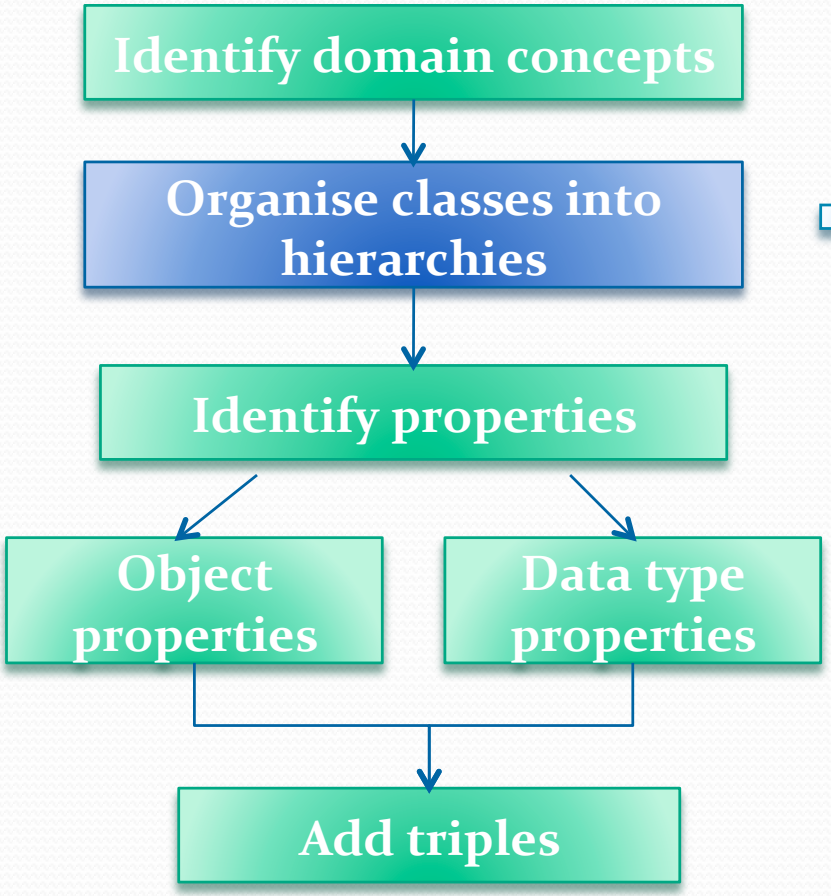
Individuals are instance of class

Example:

"**A two holes disc loom weight found at Satriano**" is an instance of **loom weight**

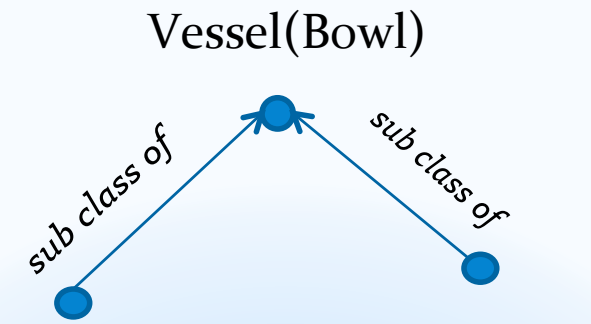
Instance of

Class Hierarchy

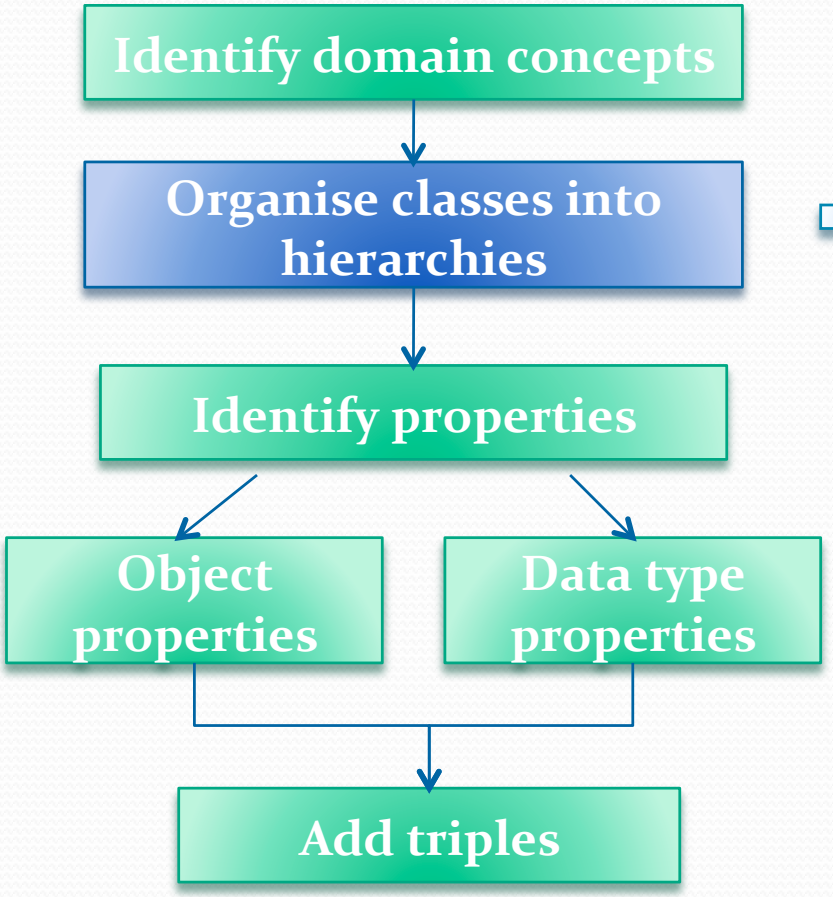


Class hierarchy: a class X **IS_A** subclass of Y when every member of X is also a member of Y

Example

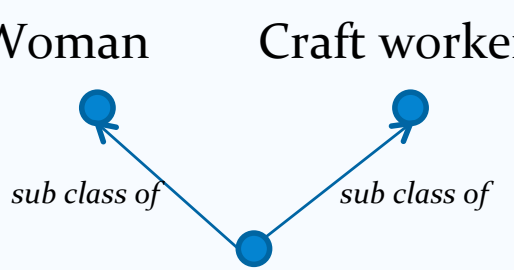


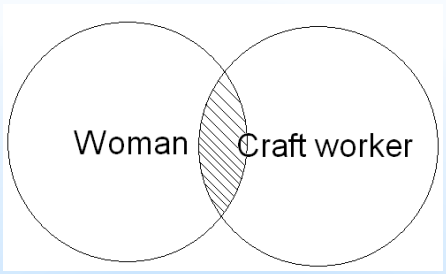
Class Hierarchy



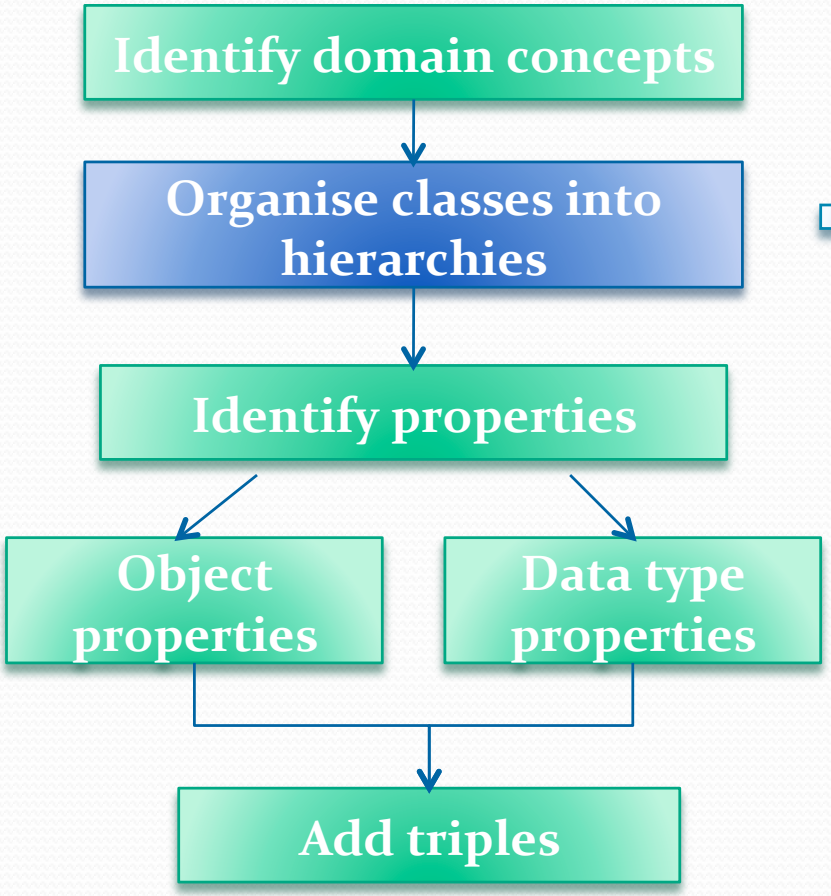
Class hierarchy

multiple inheritance



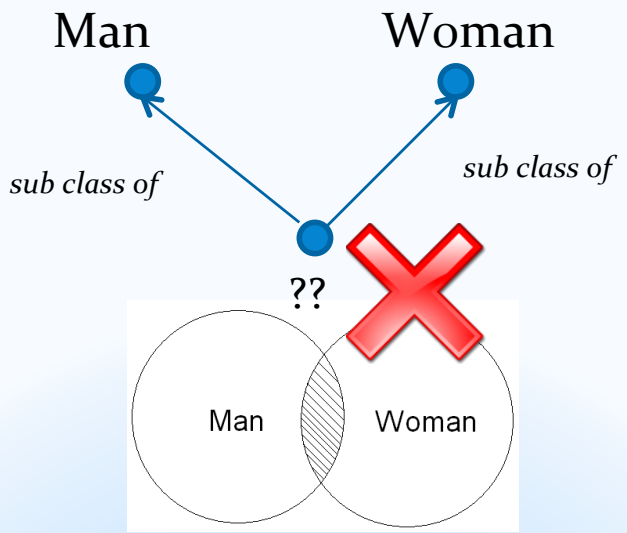


Class Hierarchy



Class hierarchy

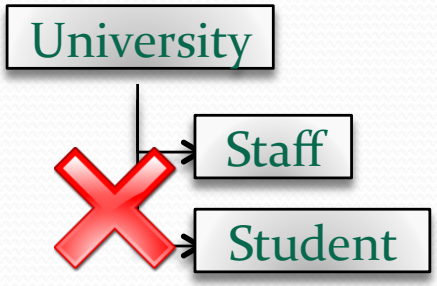
multiple inheritance



Disjoined class

Exercise: ontology of HE

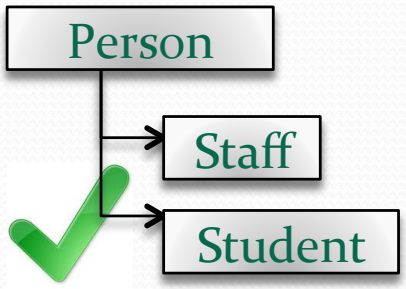
Question: are the following correct hierarchies? [HINT: translate them in English]



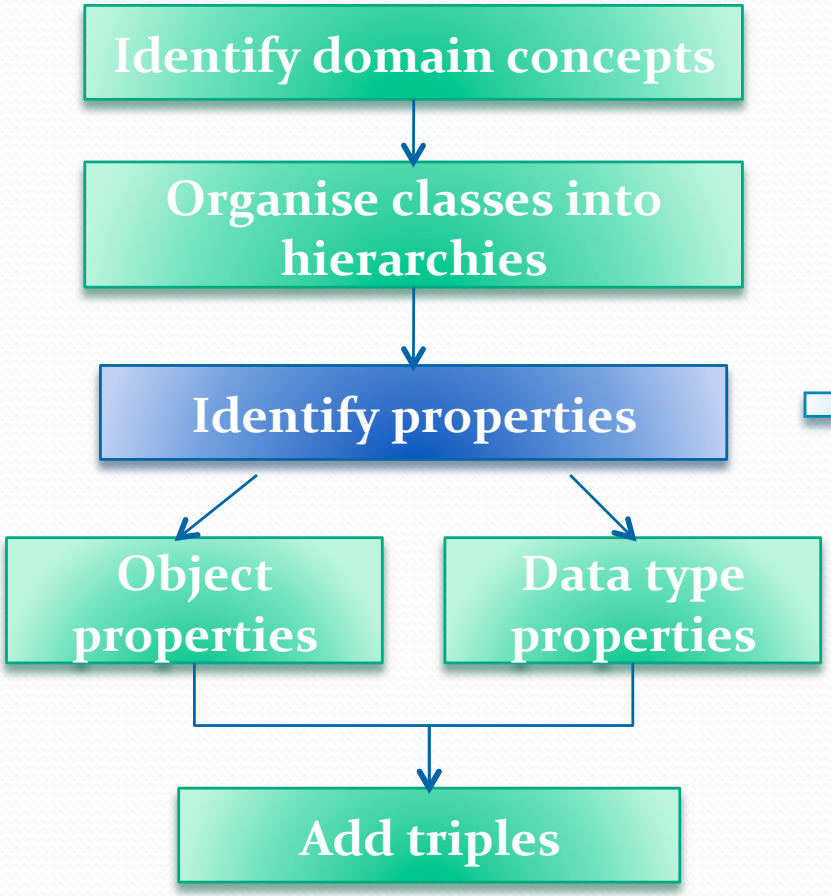
The subclass relationship must be an "**is a**" relationship

UoL and UoG are ***indiviuals*** of class ***University*** , not classes

Question: are the following correct hierarchies?



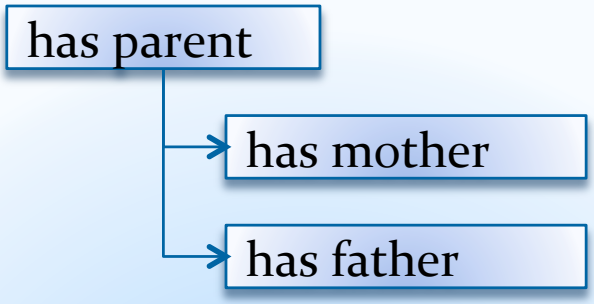
Property



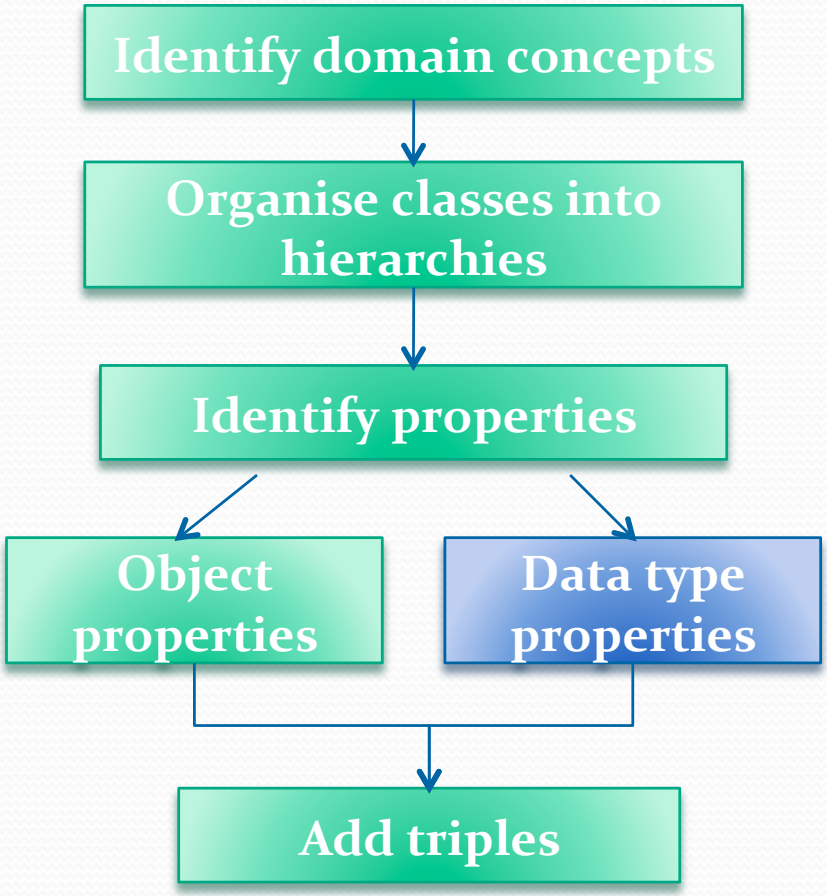
Properties specify either **attributes** of a concept or **relations** between concepts.

- ***Data type property***
- ***Object property***

There might be a hierarchy of properties:



Data Type Property



Data type property *is an attribute of the objects of a class; the attribute* takes values (a data literal) specific for each object.

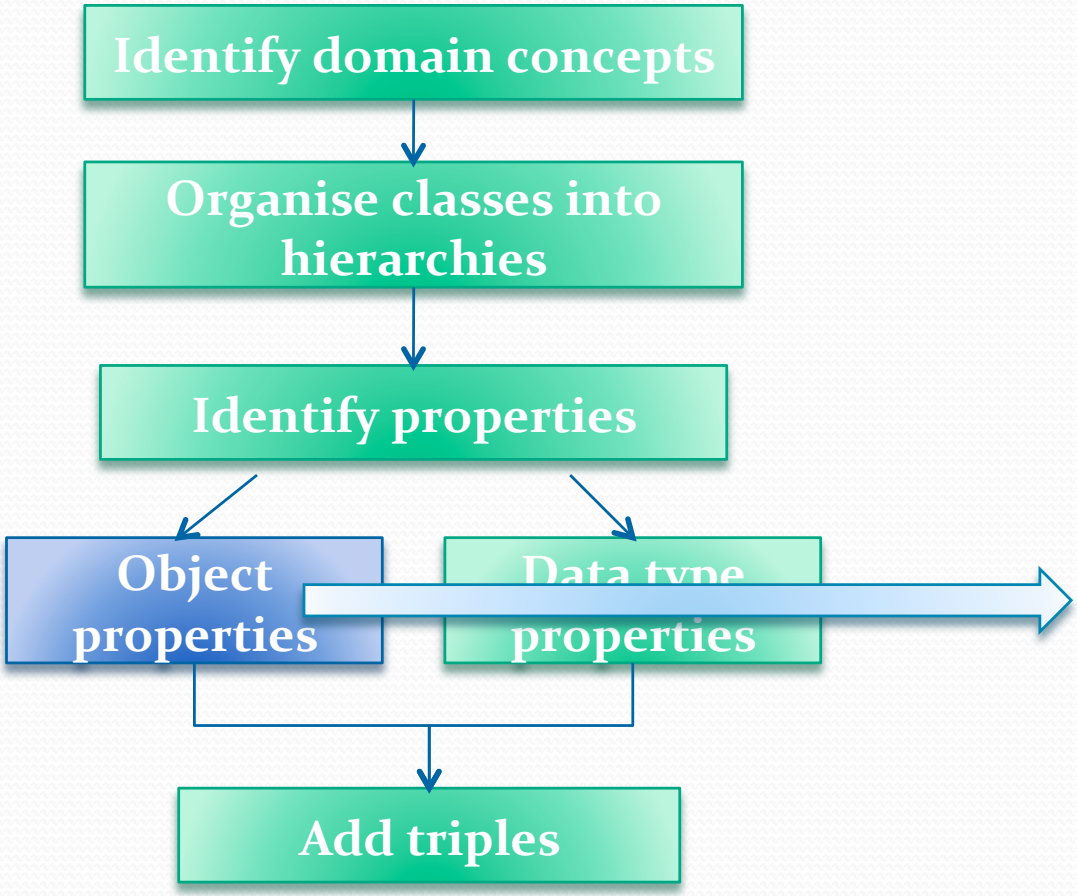
Typical types of such attributes are

- Strings
- Numbers
- Dates
- …

Example

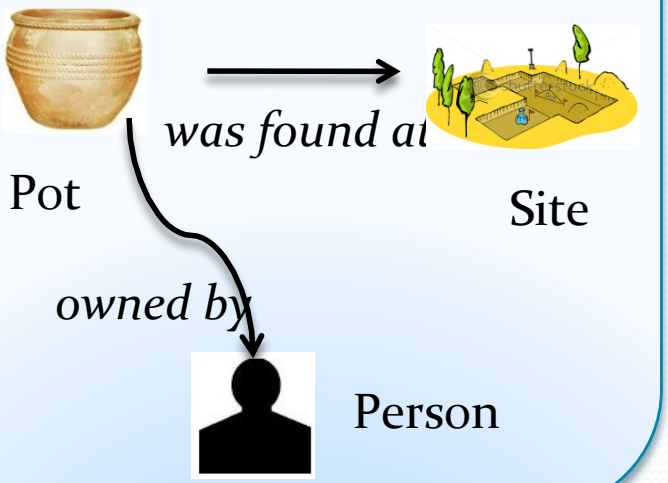
Weight: **1.4kg**
Description: "**Beautiful**"
Year of Excavation: **1980**

Object Property

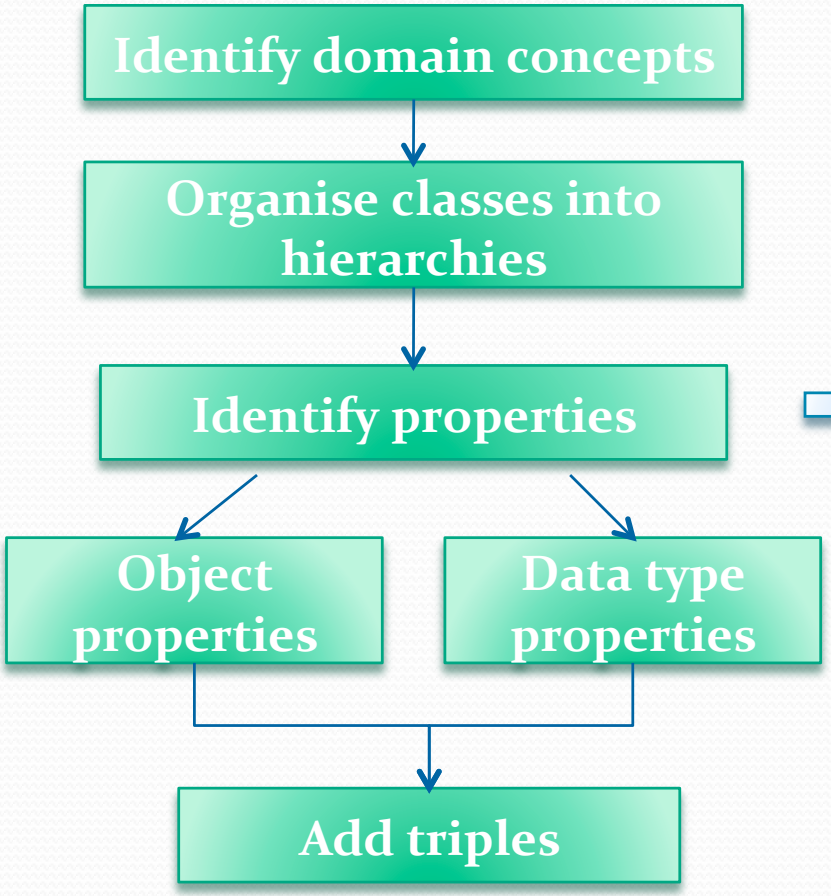


Object Property is a direct binary **relation** between two classes

Example



Domain & Range

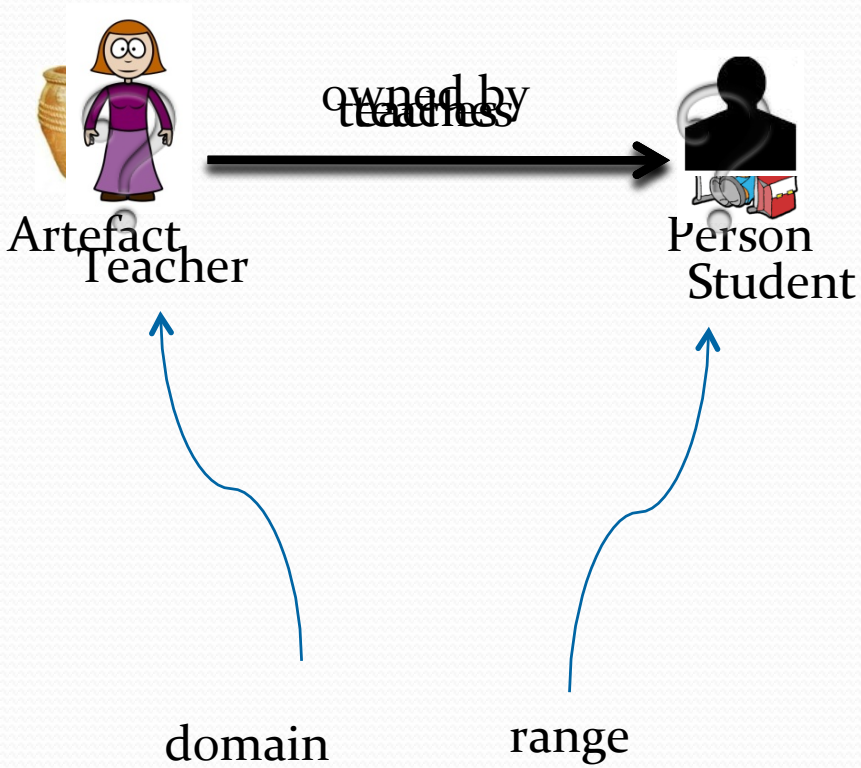


Property

Domain:
A domain of a property limits the individuals to which the property can be applied.

Range:
The range of a property limits the individuals that the property may have as its value

Domain & Range



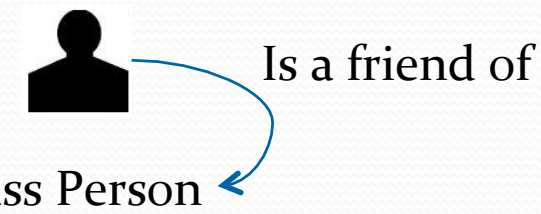
Property

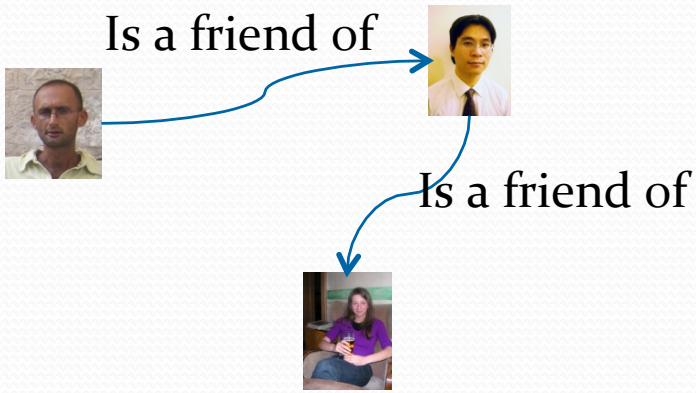
Domain:
A domain of a property limits the individuals to which the property can be applied.

Range:
The range of a property limits the individuals that the property may have as its value

Domain & Range

Domain & range can be the same class





Property

Domain:
A domain of a property limits the individuals to which the property can be applied.

Range:
The range of a property limits the individuals that the property may have as its value

Cardinality

found_at

exactly 1

Artifact

Site

has_decoration

Decorated Pot

min **1**

Motif

has_handles

max **2**

Amphora

Cardinality

cardinality specifies the occurrences of a relationship

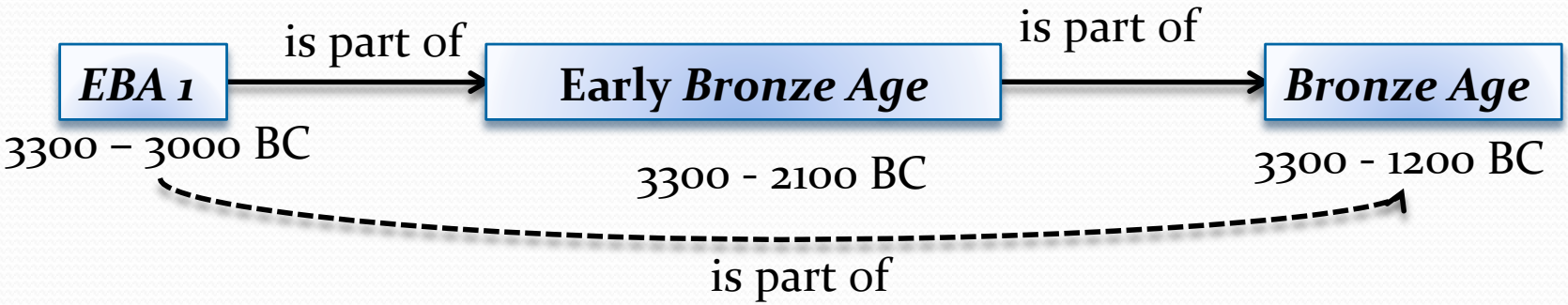
A property can have:

Exact cardinality
Min cardinality
Max cardinality

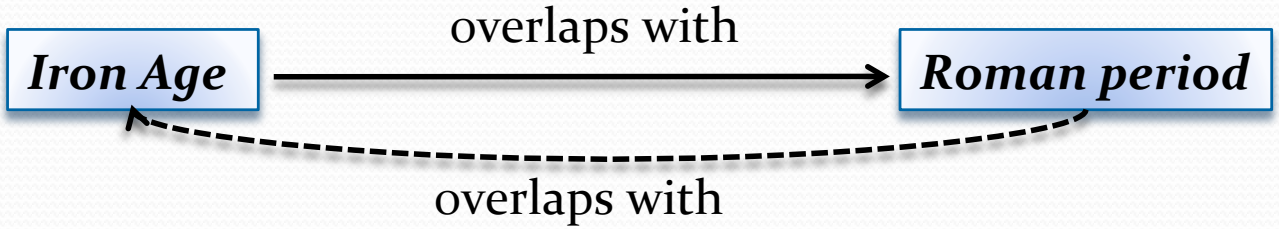
Characteristics of a property

Object properties have other characteristics than cardinality

Transitivity



Symmetry



Properties for Free…

Question: what is the inverse property of the property below?

Amphora

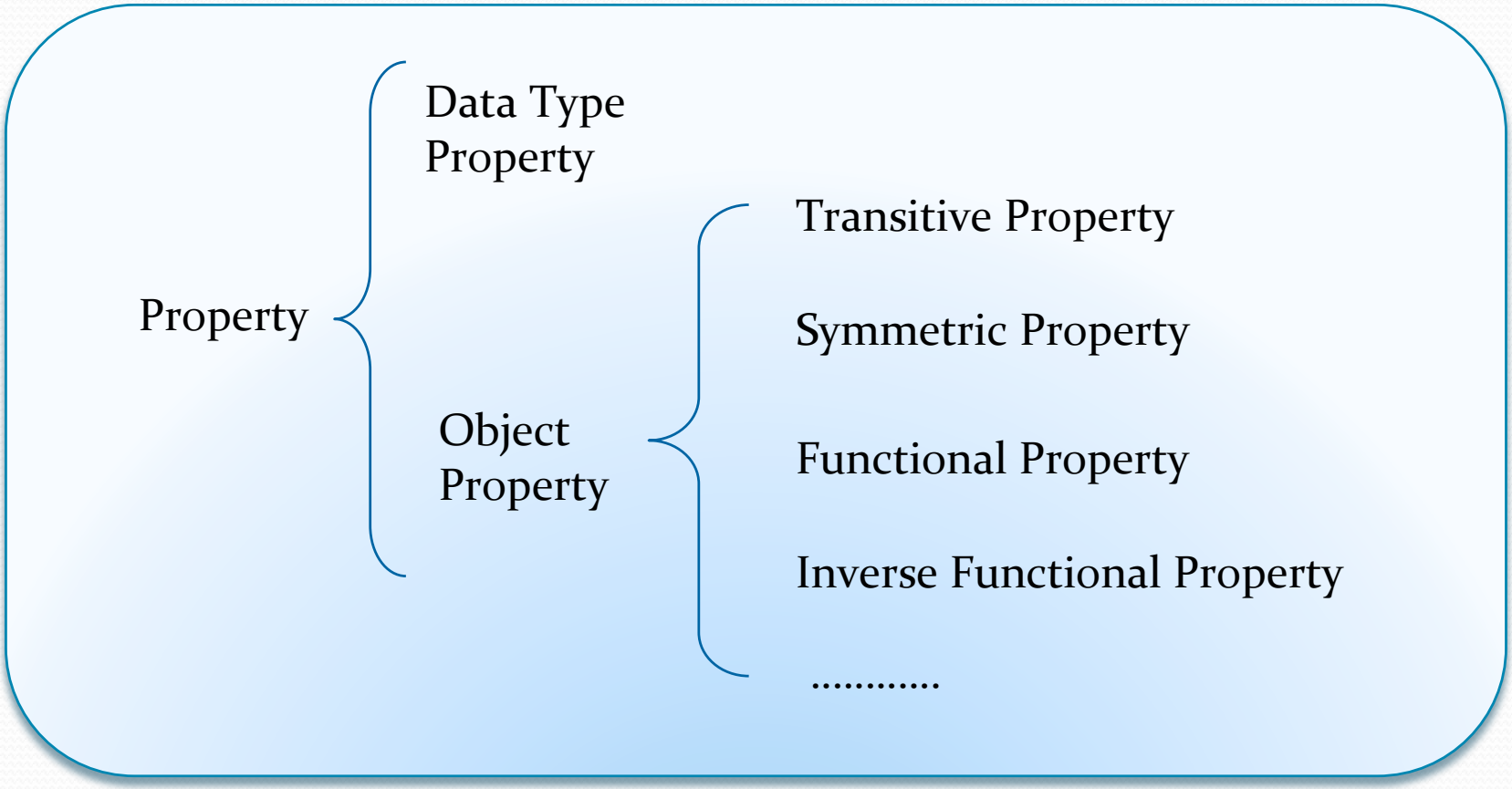
has_handles

max **2**

Once an object property is defined, one immediately gets the **inverse property**, "for free".

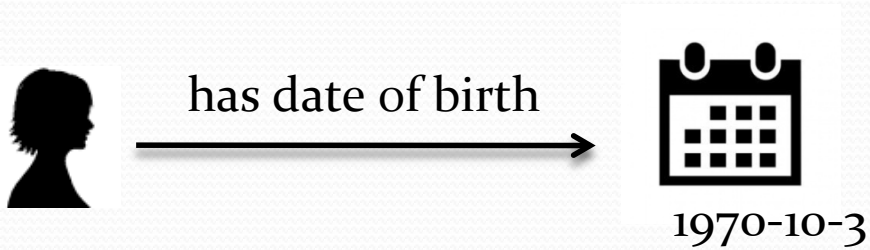
P1 is the inverse of property P2 when object X has the property P2 with object Y if, and only if, object Y has the property P1 with object X.

More Properties for Free



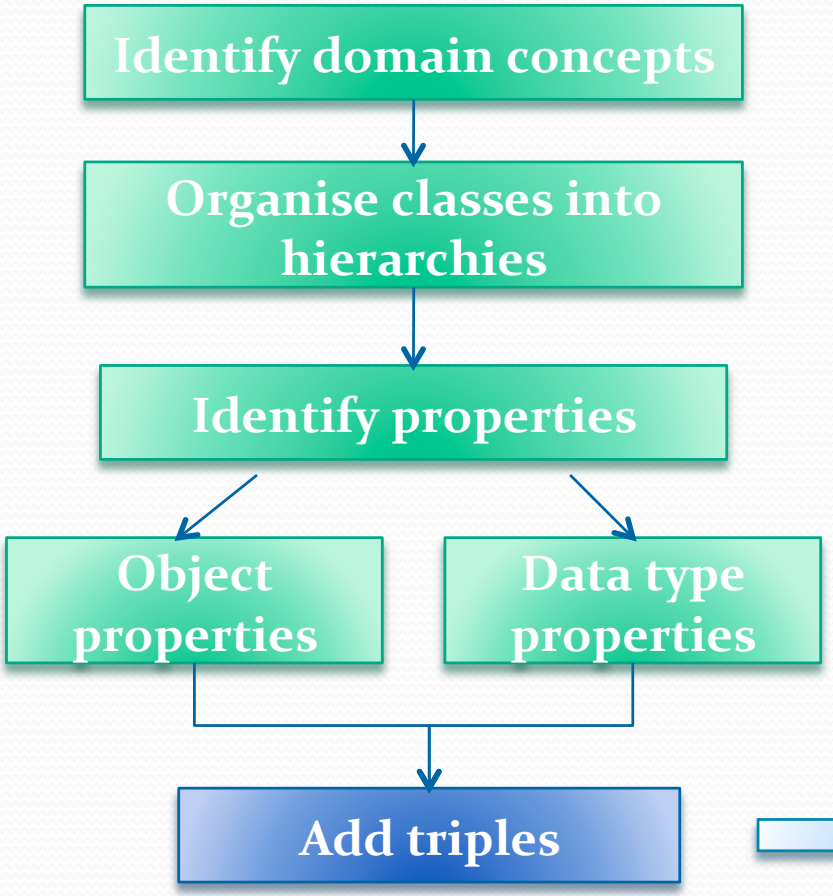
Characteristics of a property

Functional Property



Inverse Functional Property

Triple



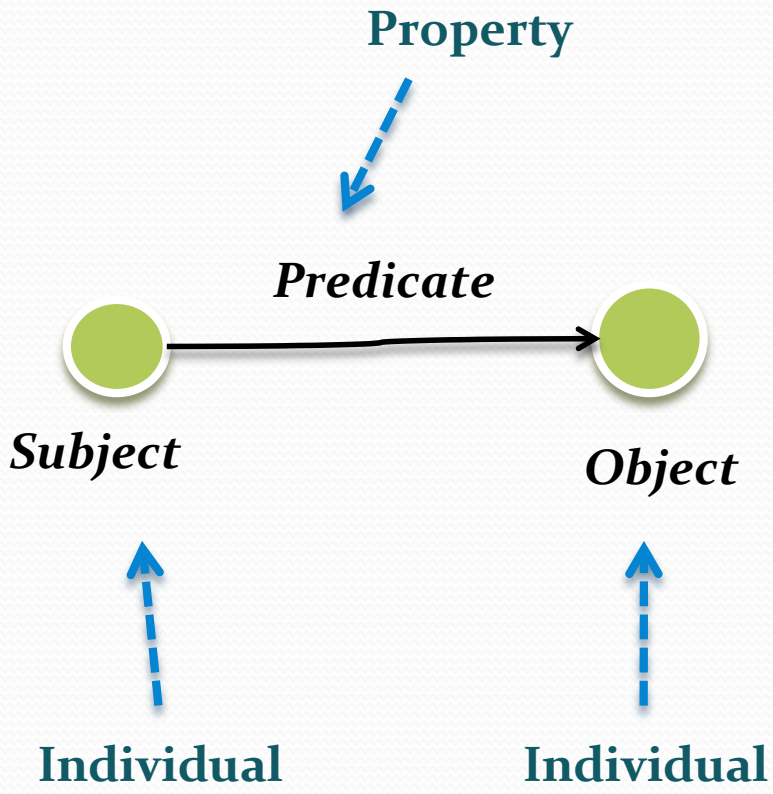
Triples are the basic elements of an ontological DB to "populate" properties.

A ***Triple*** *is* conventionally written in the order

1. **subject,**
2. **predicate,**
3. **object**.

Triple

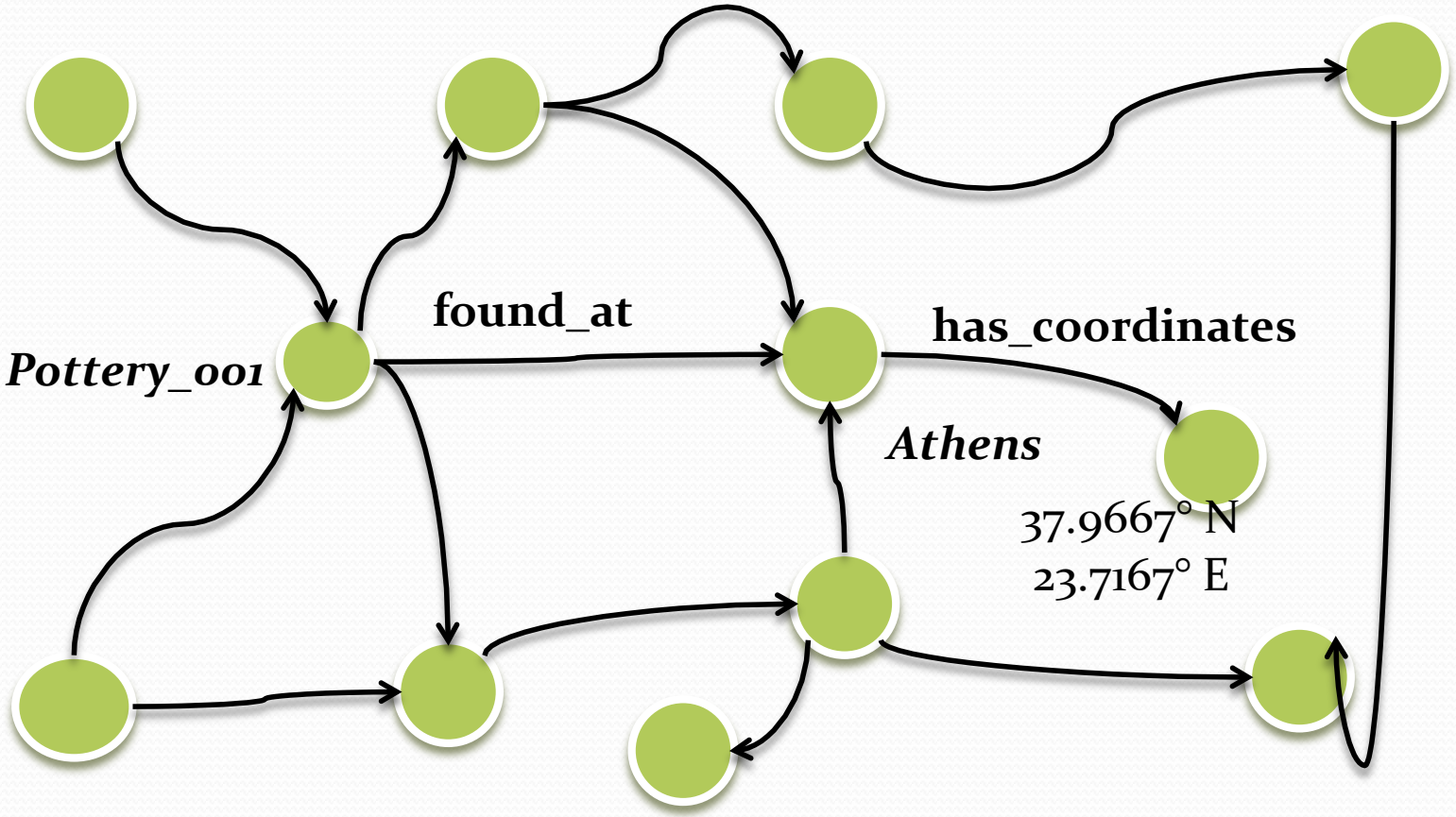
Basic element
contains three parts: ***subject, predicate*** *and* ***object****.*



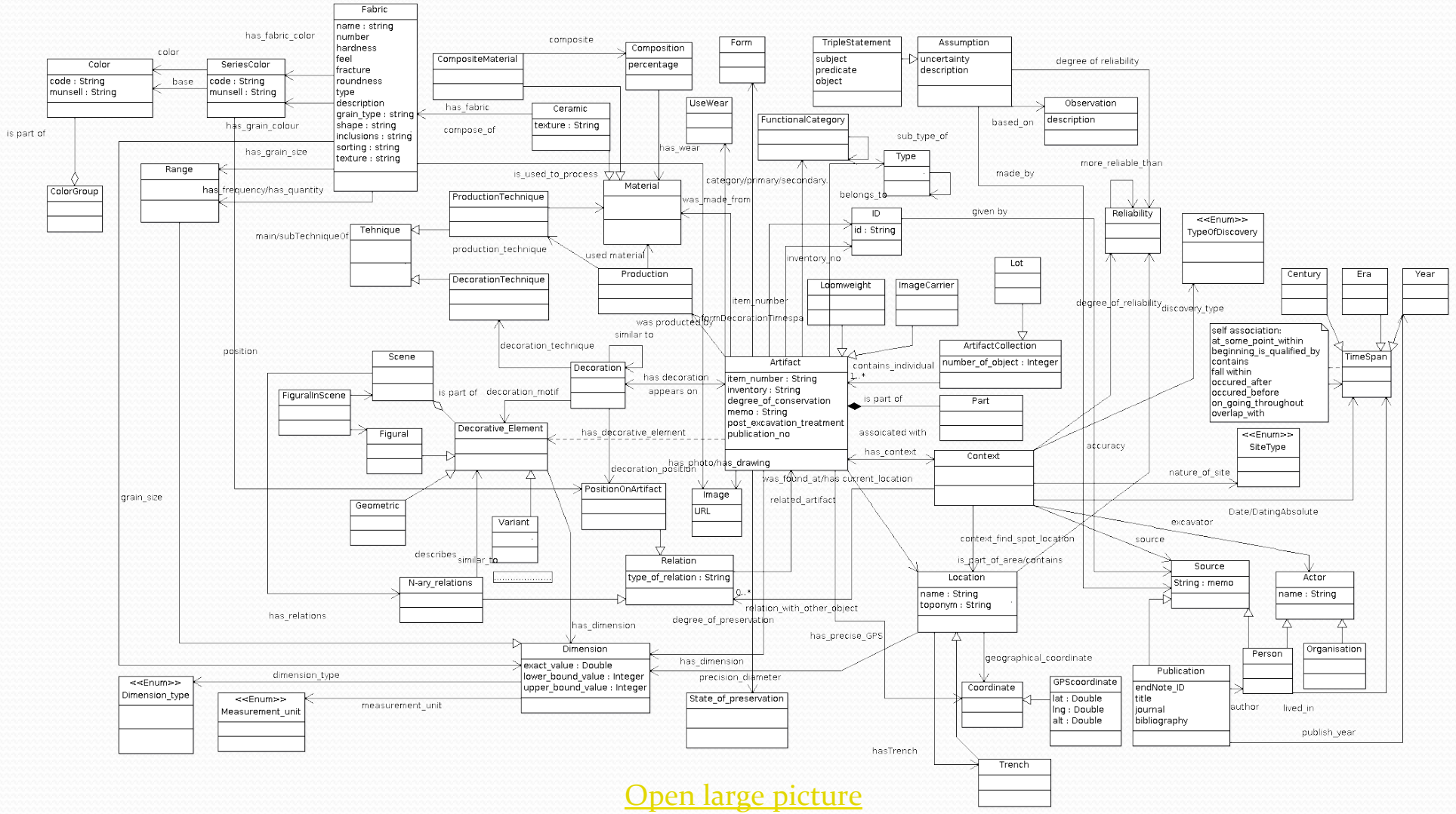
Triple

Triple Graph

A set of triples becomes a graph and actually, an ontology-based DB is a graph



A glimpse of our TN Ontology



Open large picture

Example

Scenario :

Design a data structure for **Course Modules Selection**

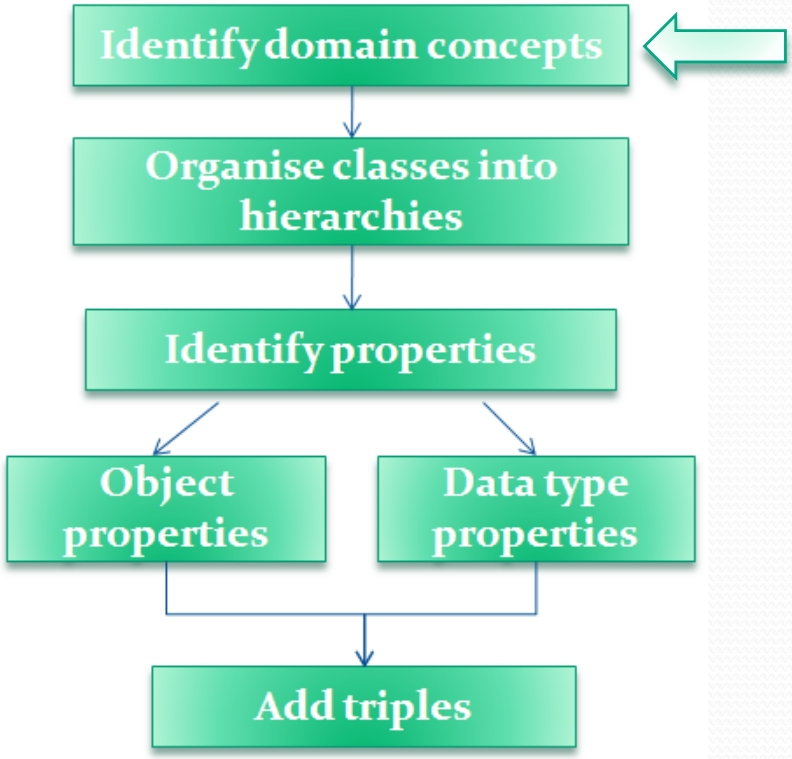
Requirement:

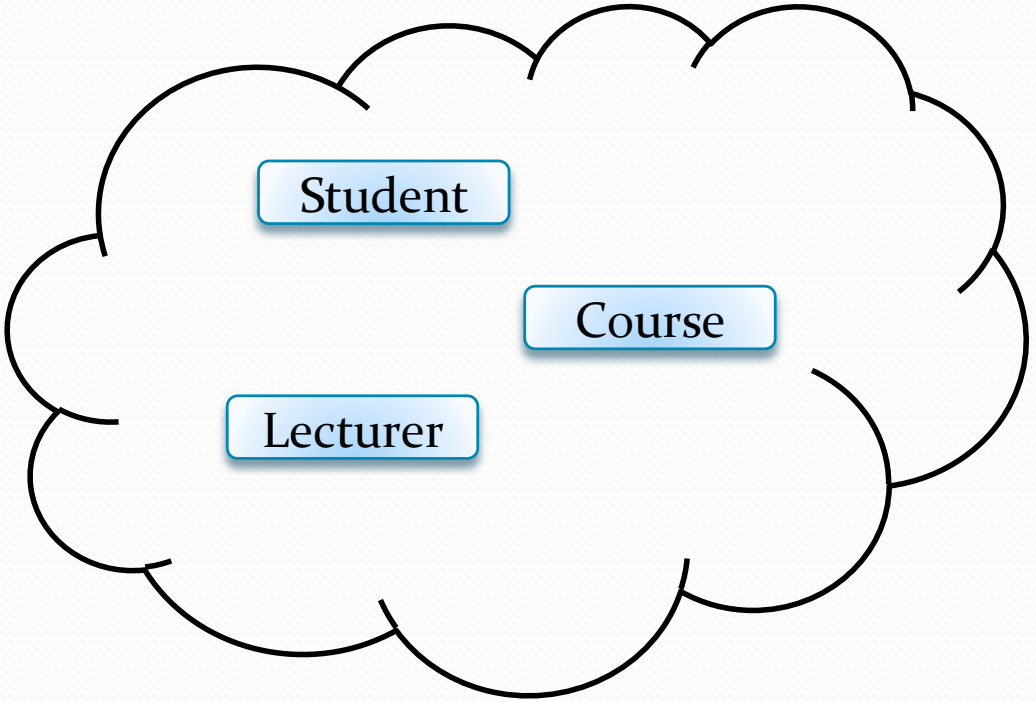
(1) Every course module can only be taught by one Lecturer
(2) A Lecturer can teach many courses..
(3) An student must enrol in at least **3** courses.
(4) Max enrolment number for a course is **100.**

* Every student has a unique student ID
* Every staff has a unique staff ID

Ontological approach

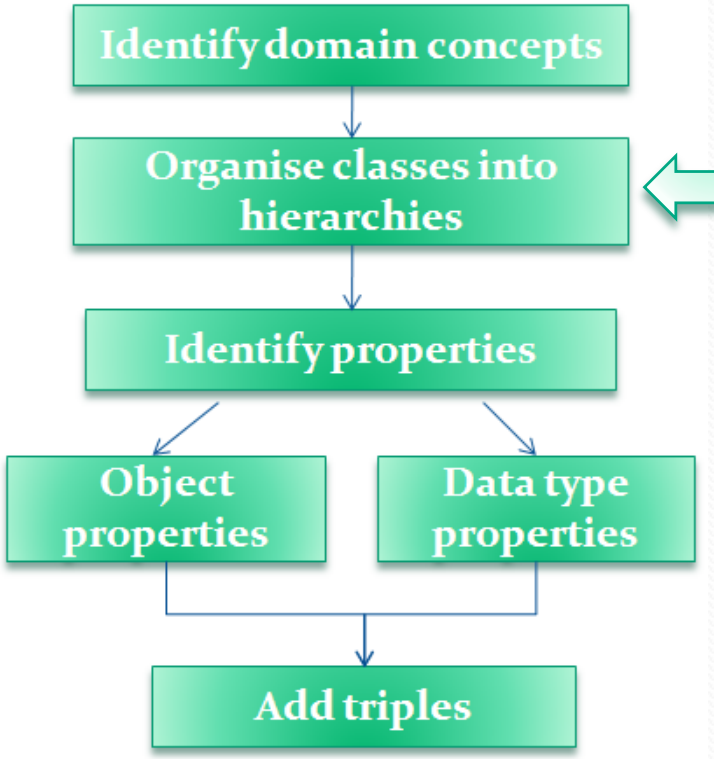
Identify concepts

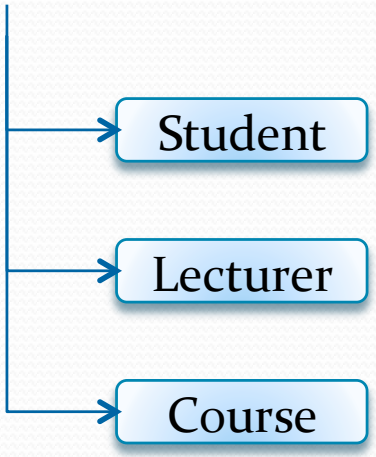




Ontological approach

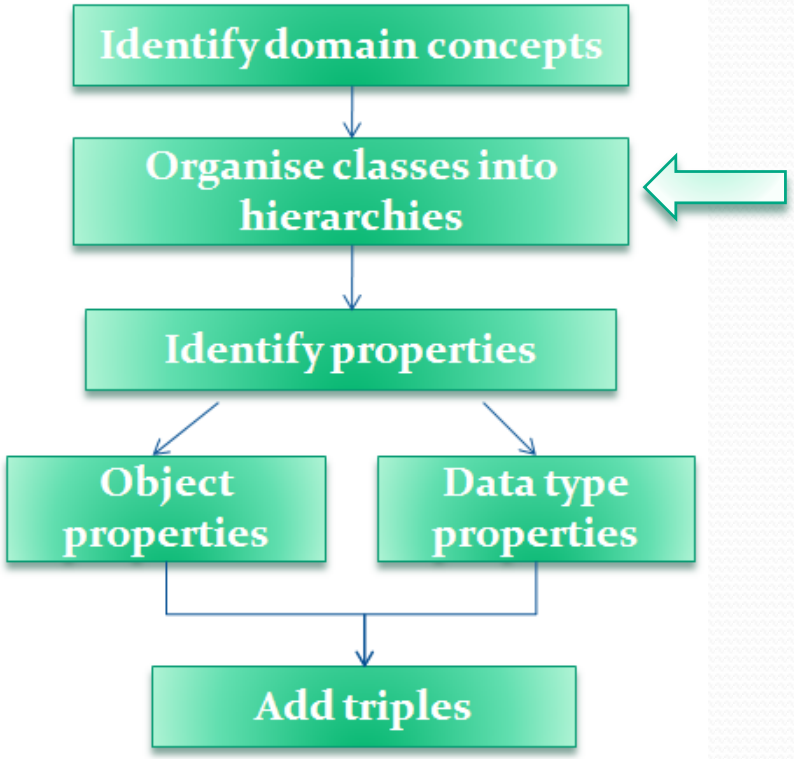
Organise classes into hierarchies

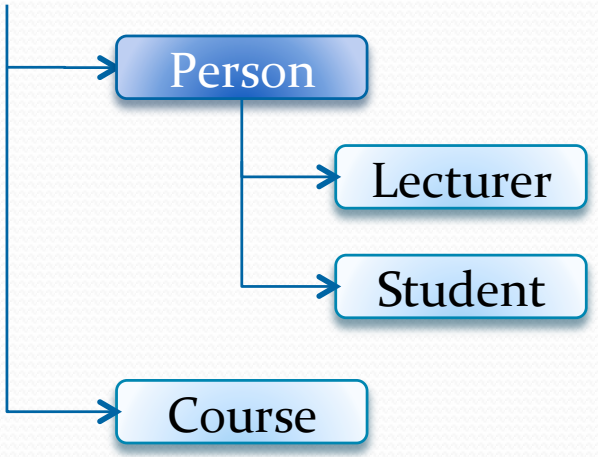




Making relationships explicit and clear

Organise classes into hierarchies

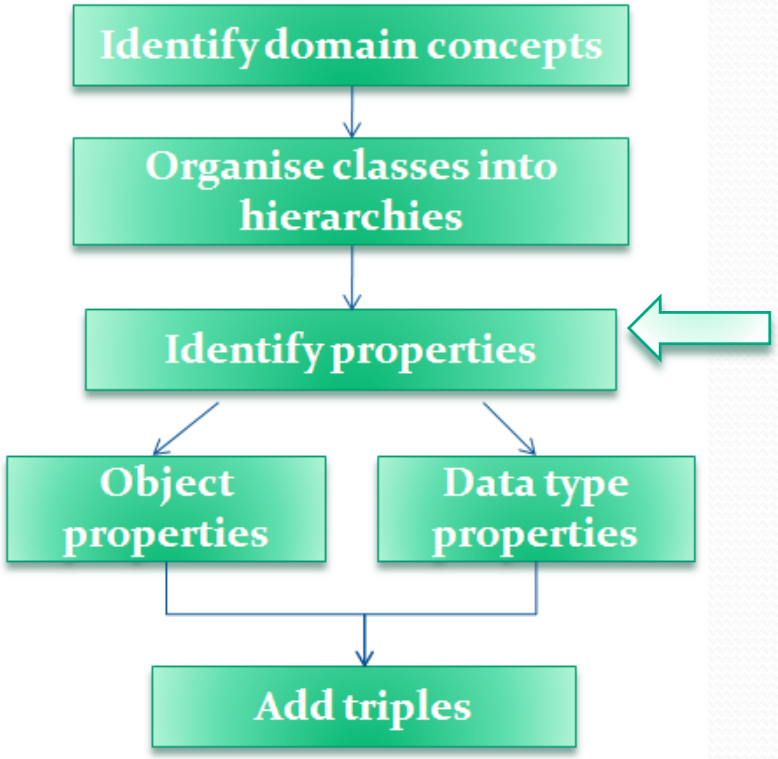


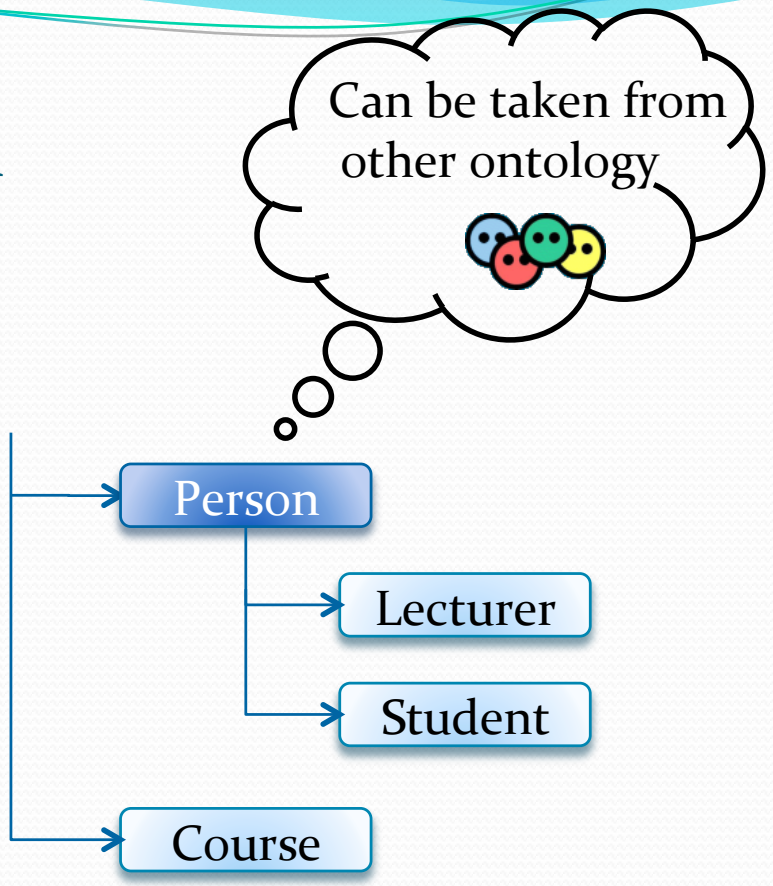


Enable reuse of domain knowledge

Ontological approach

Ontological Approach

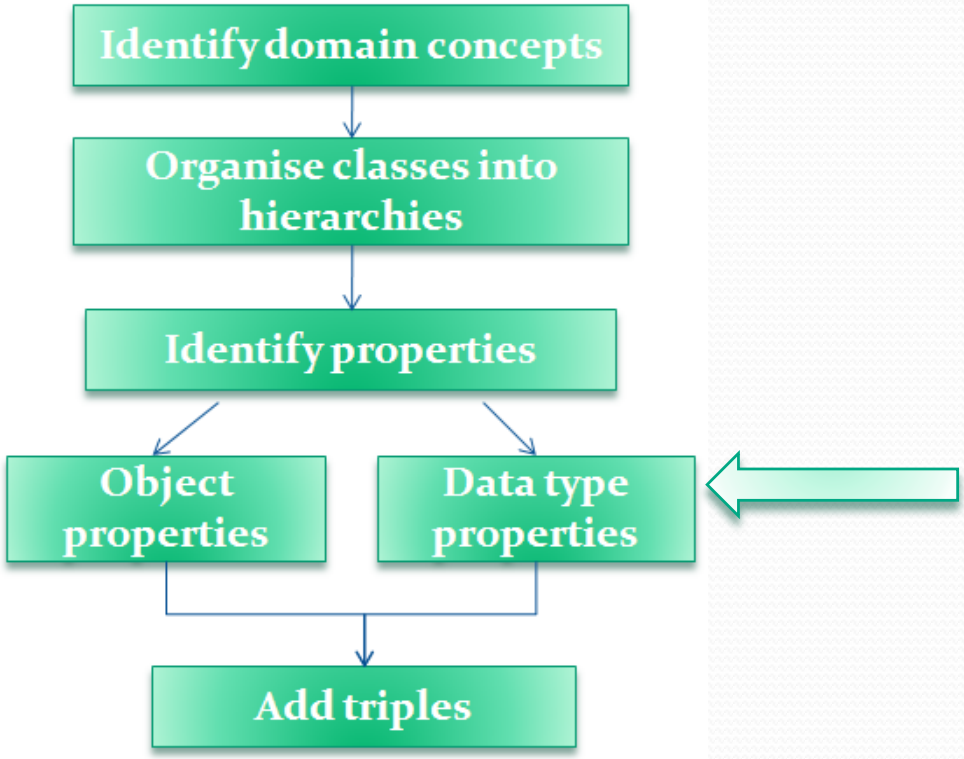


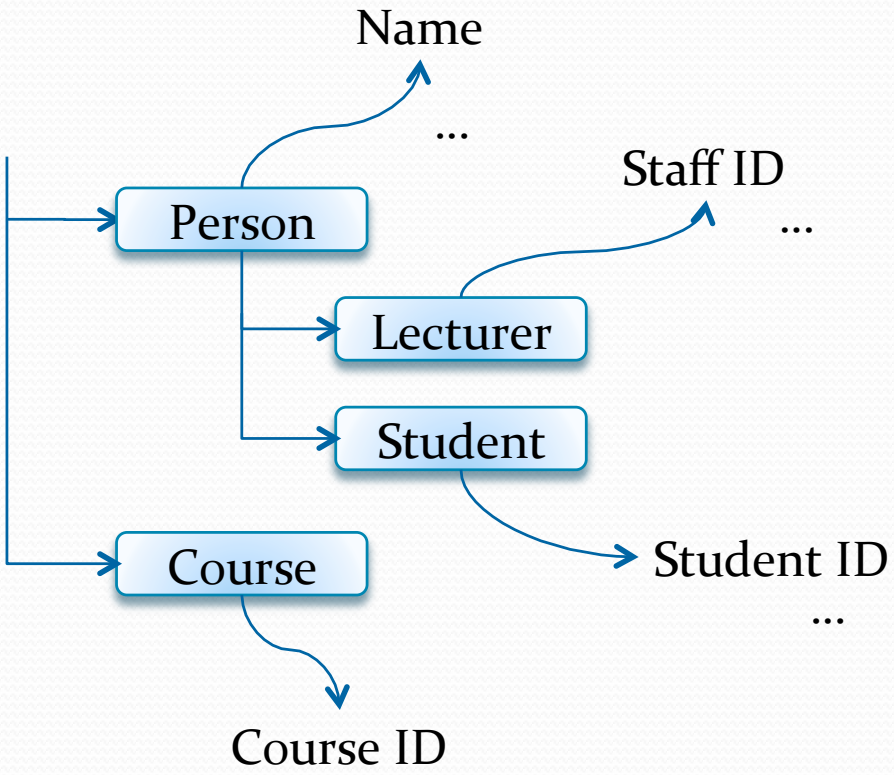


Enable reuse of domain knowledge

Ontological approach

Ontological Approach

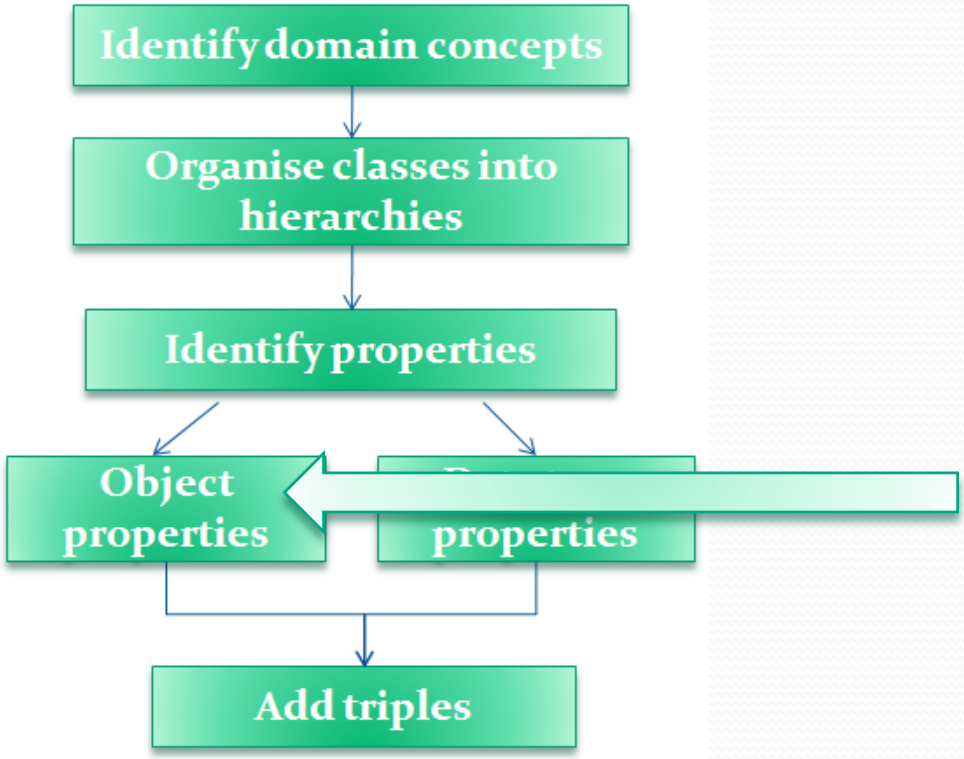


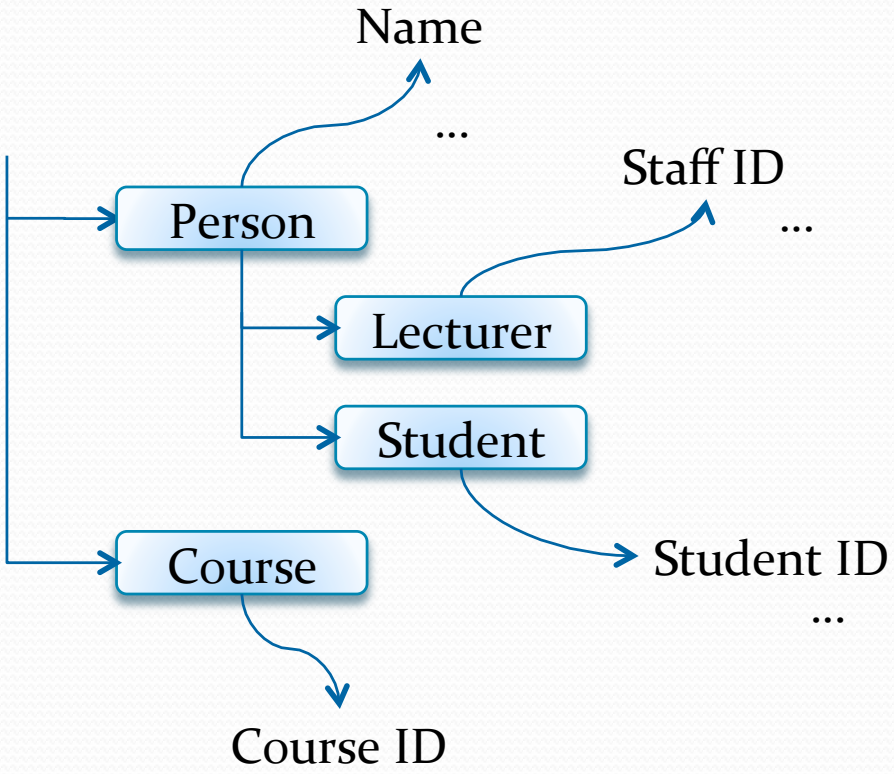


Add Data Type Property

Ontological approach

Ontological Approach

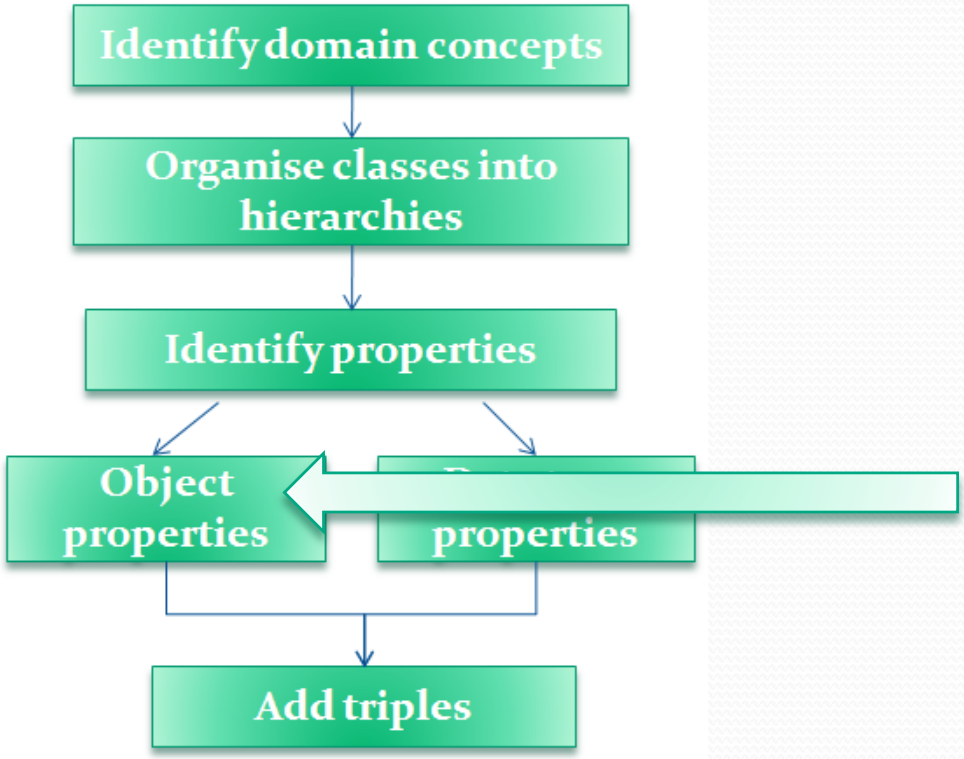


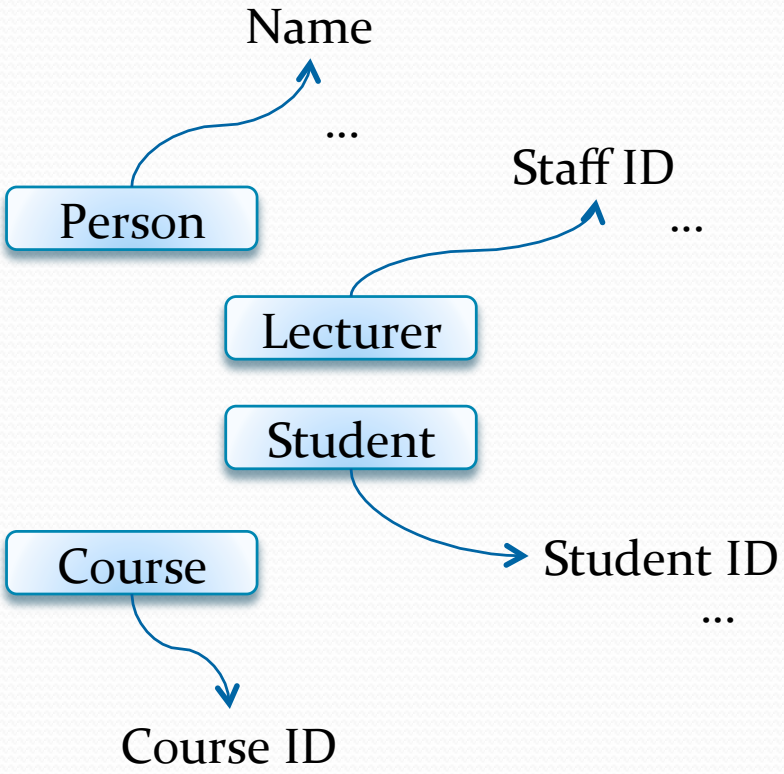


Add Object Property

Ontological approach

Ontological Approach

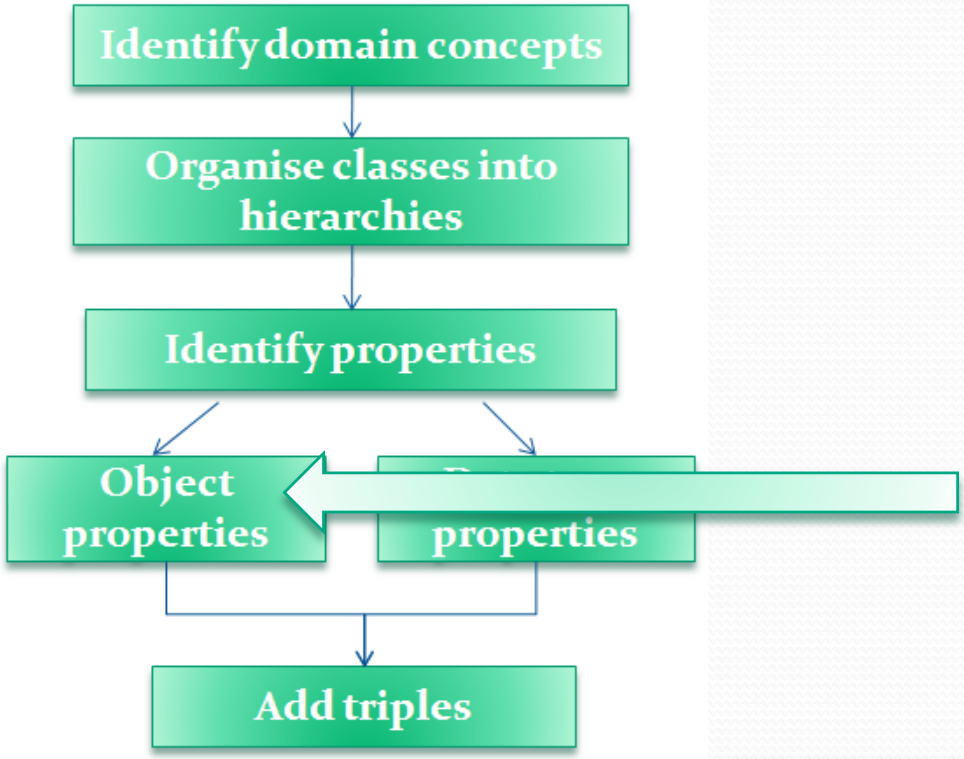




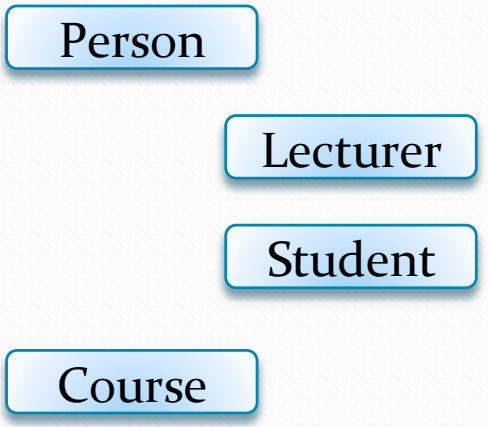
Object Property

Ontological approach

Ontological Approach



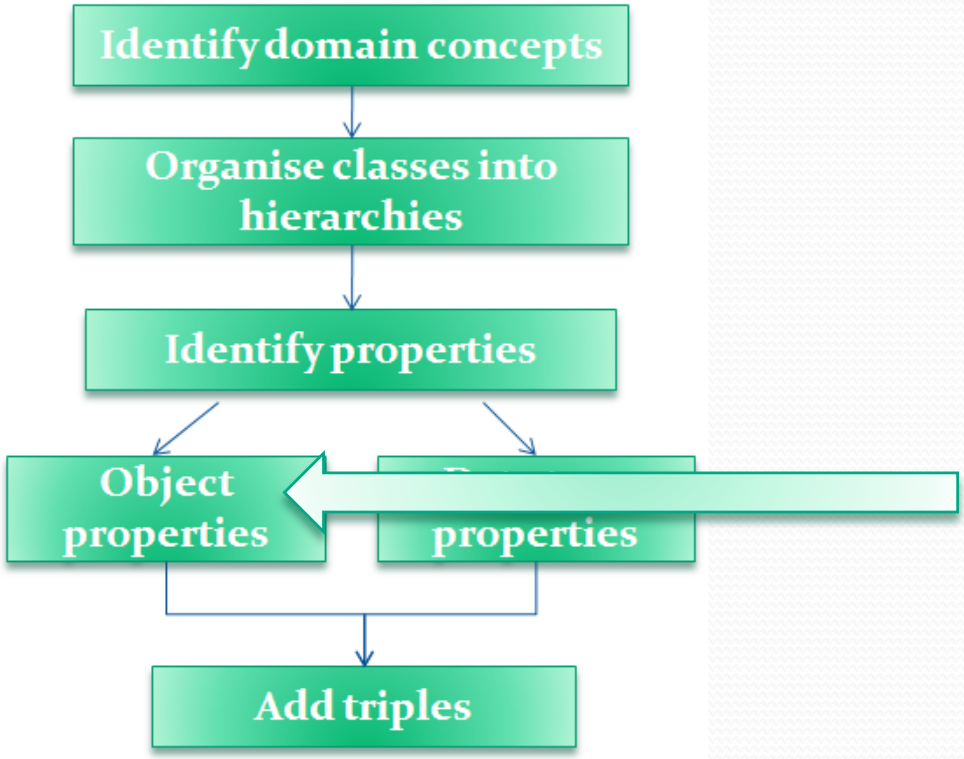
Add ObjectProperty



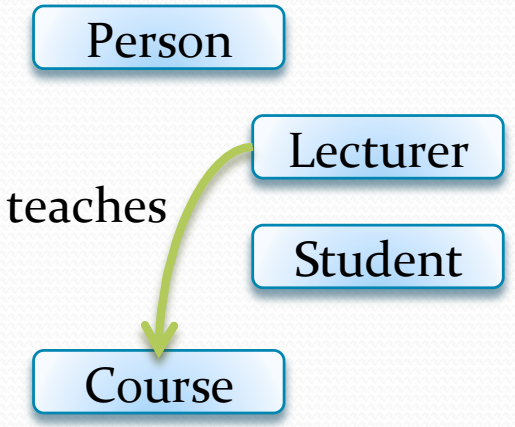
Object Property

Ontological approach

Ontological Approach



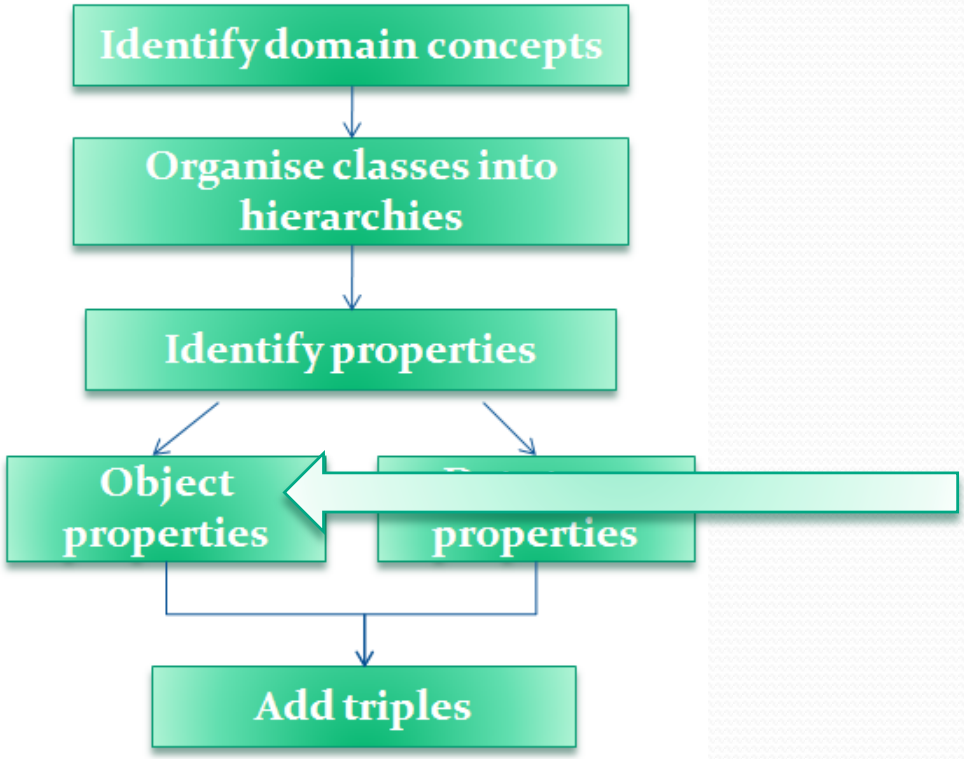
Add ObjectProperty



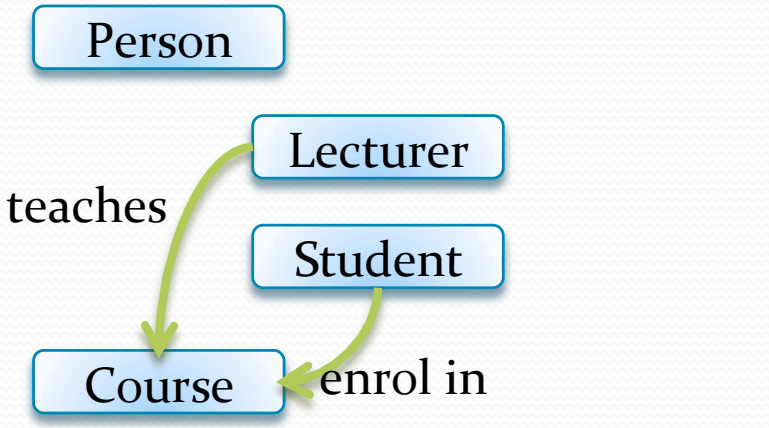
Object Property

Ontological approach

Ontological Approach



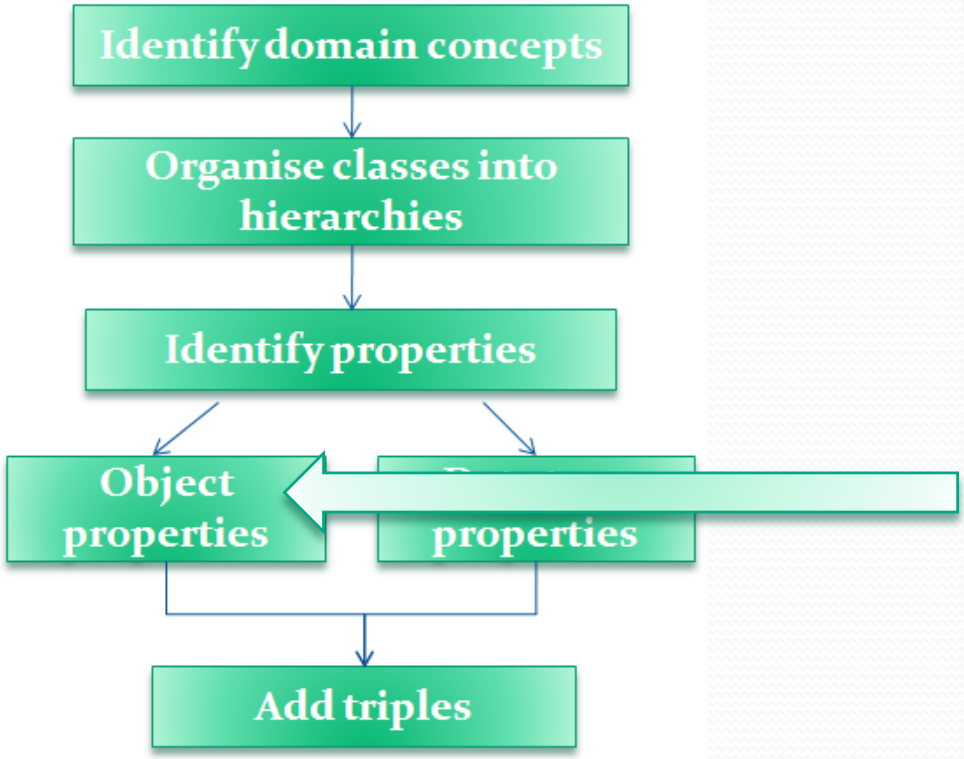
Add ObjectProperty



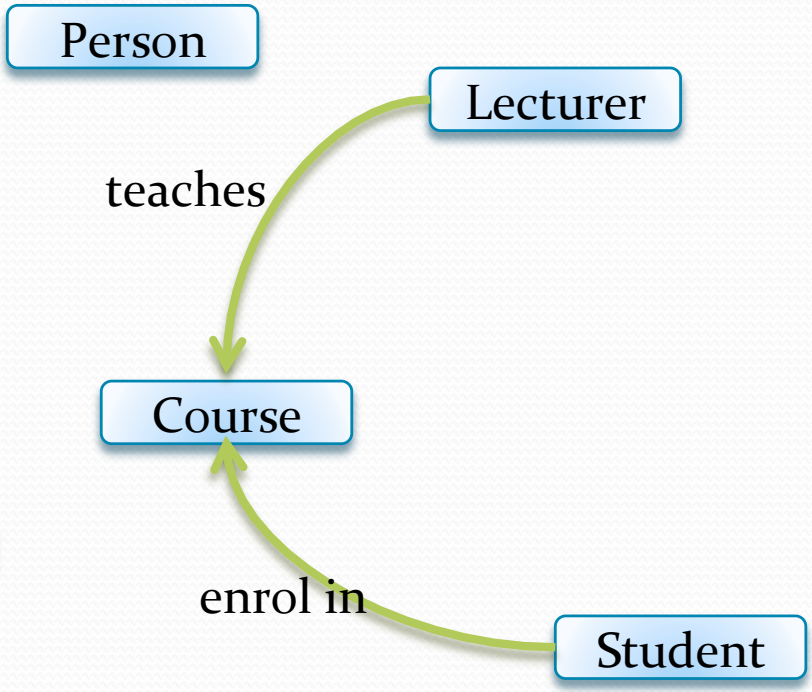
Object Property

Ontological approach

Ontological Approach



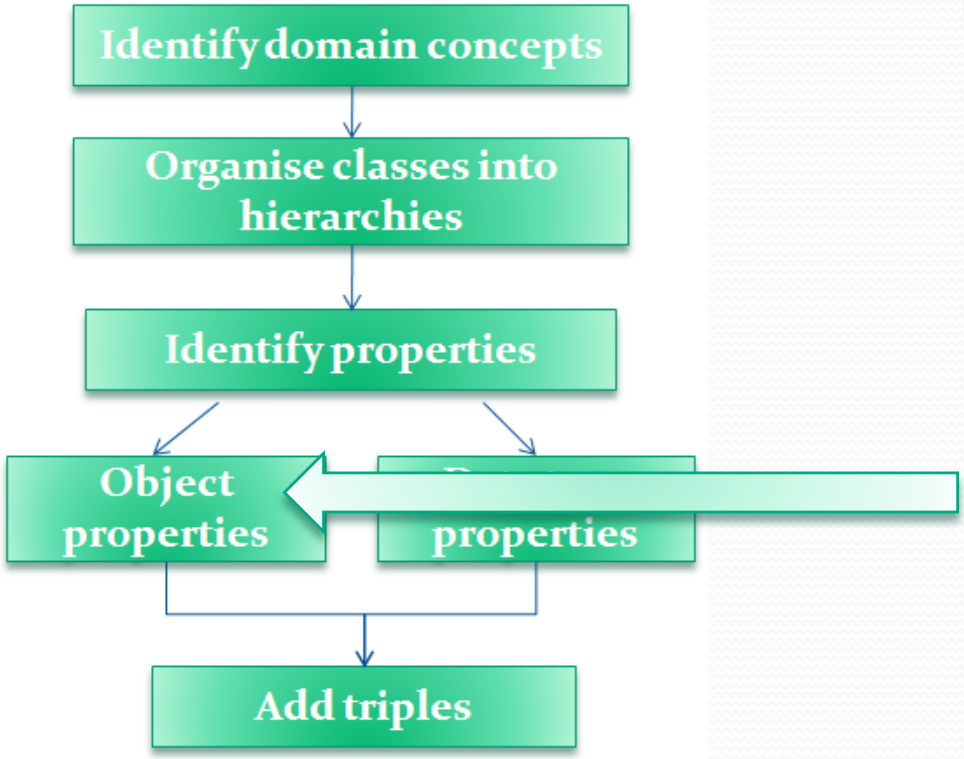
Add ObjectProperty



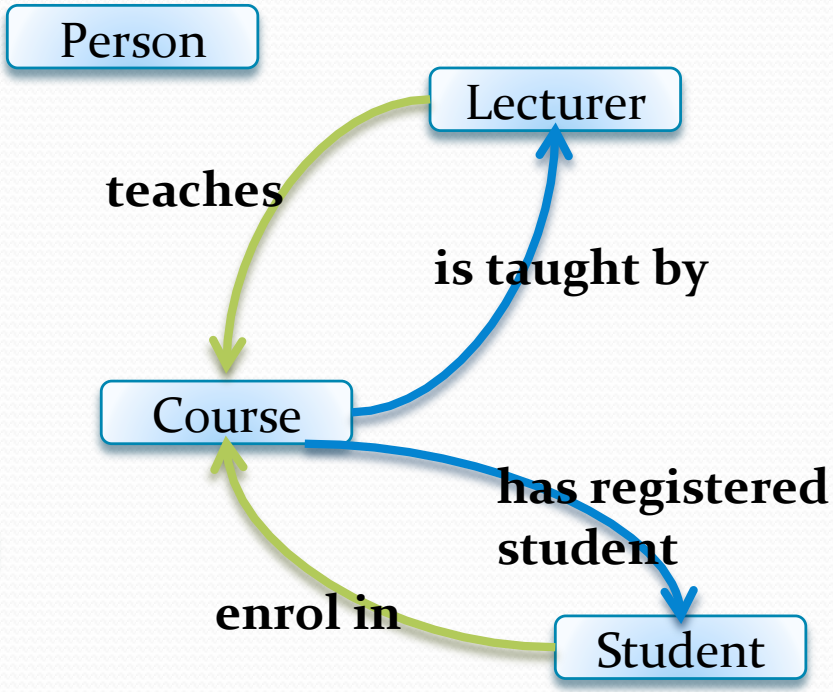
Object Property

Ontological approach

Ontological Approach



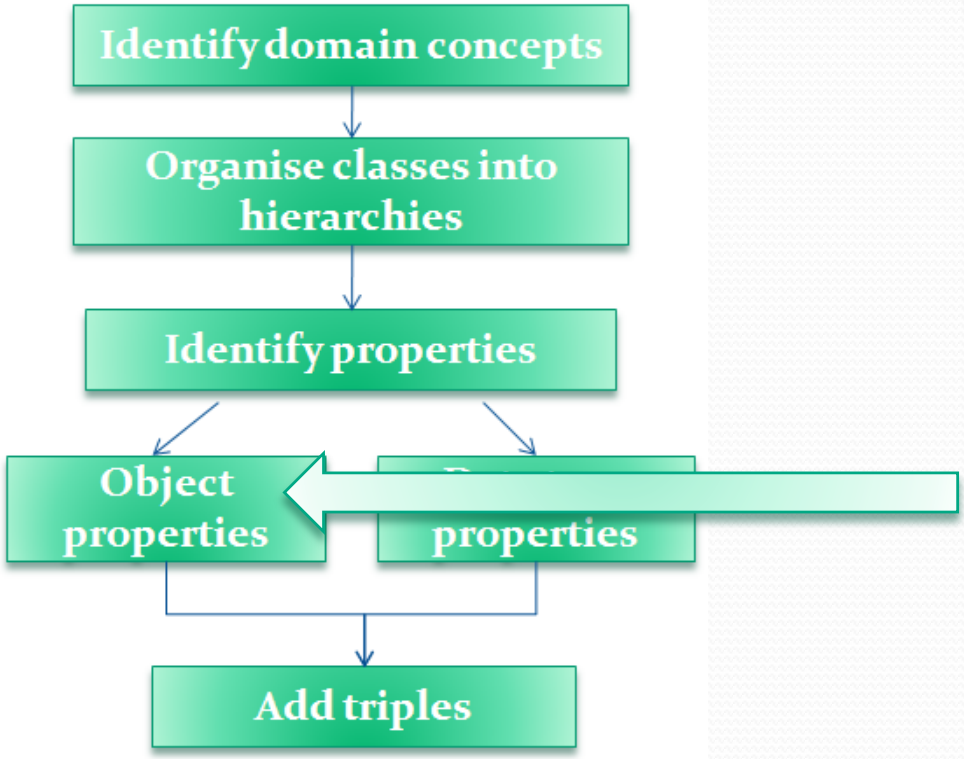
Add Inverse Property



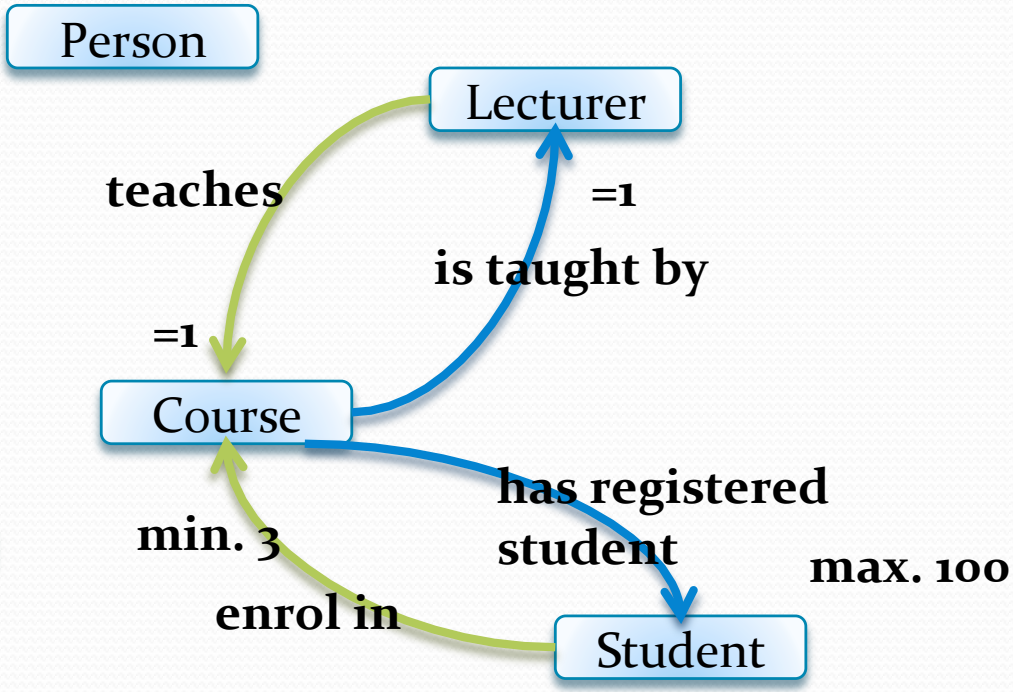
Object Property

Ontological approach

Ontological Approach



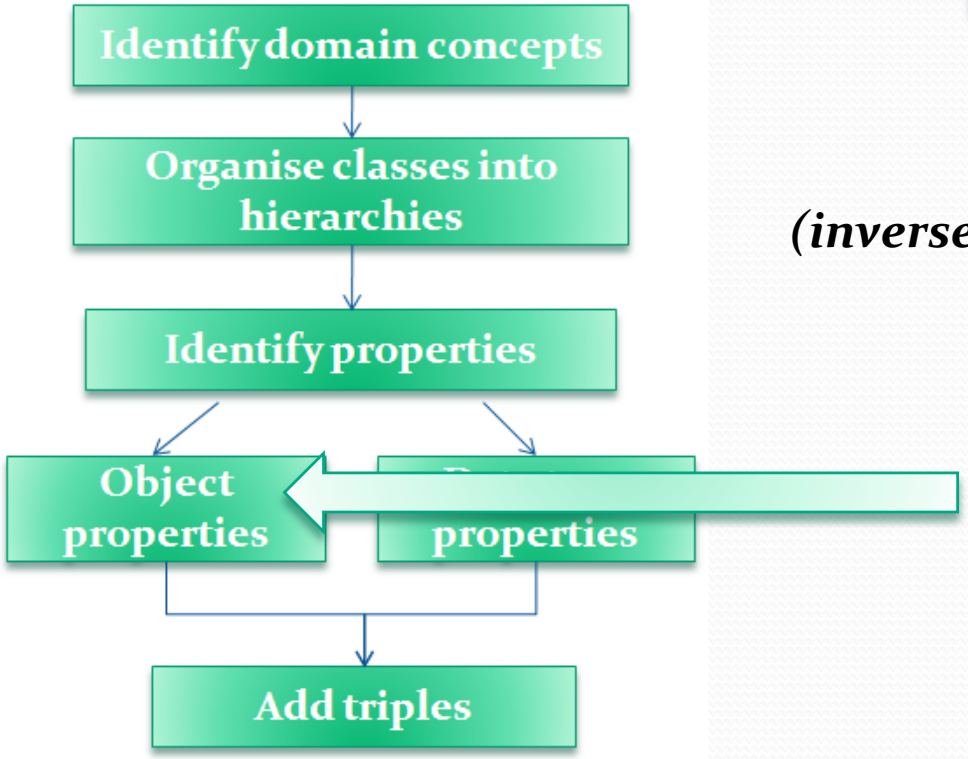
Specify cardinality



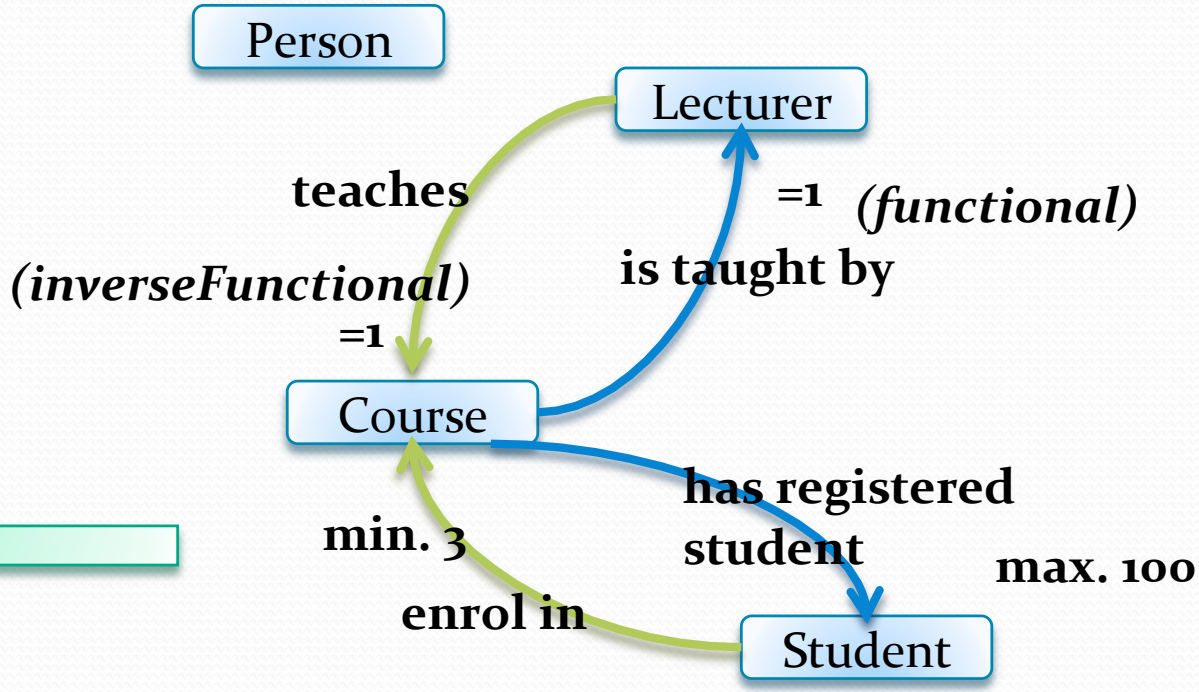
Object Property

Ontological approach

Ontological Approach



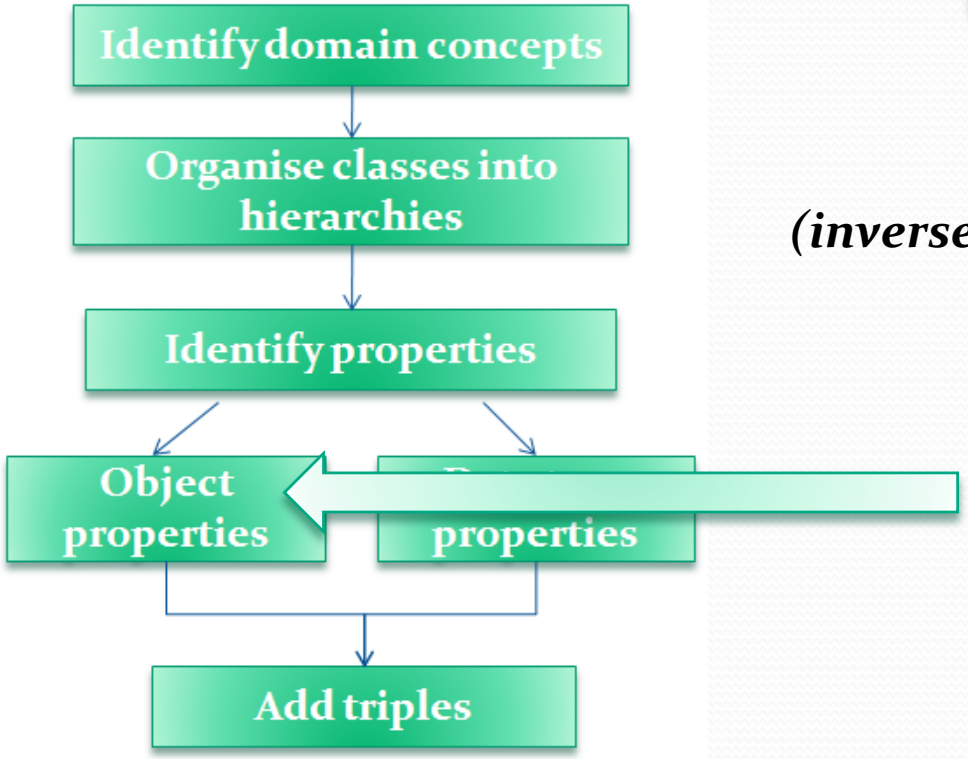
Identify type of property



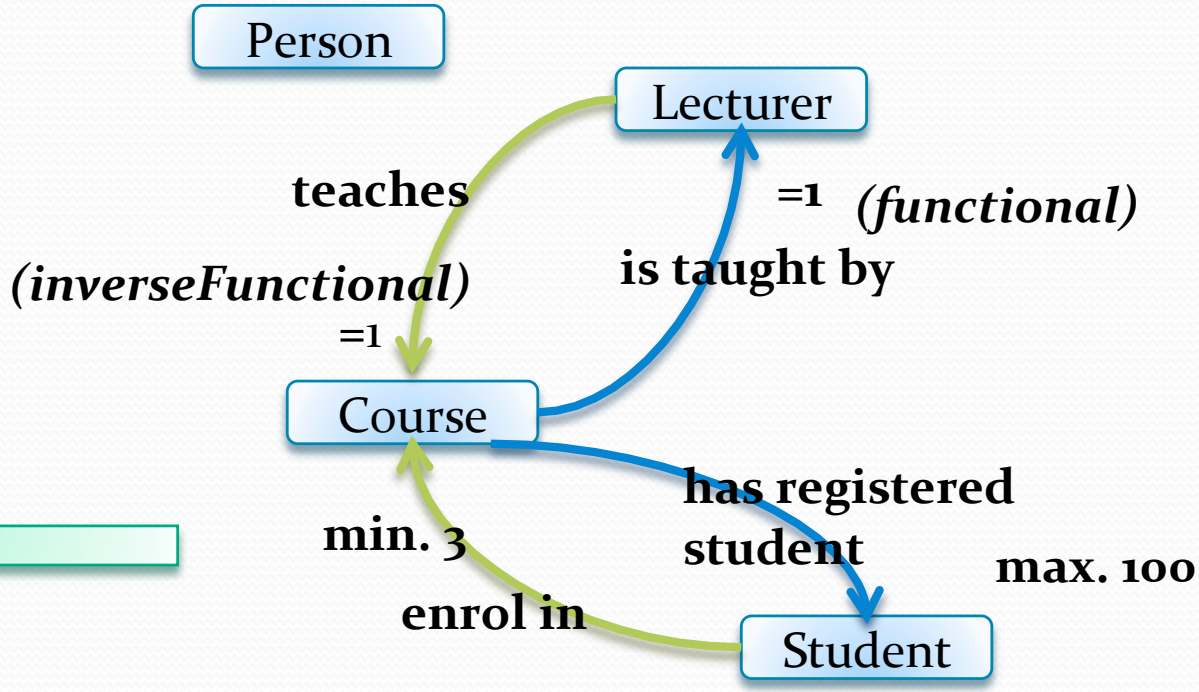
Object Property

Ontological approach

Ontological Approach



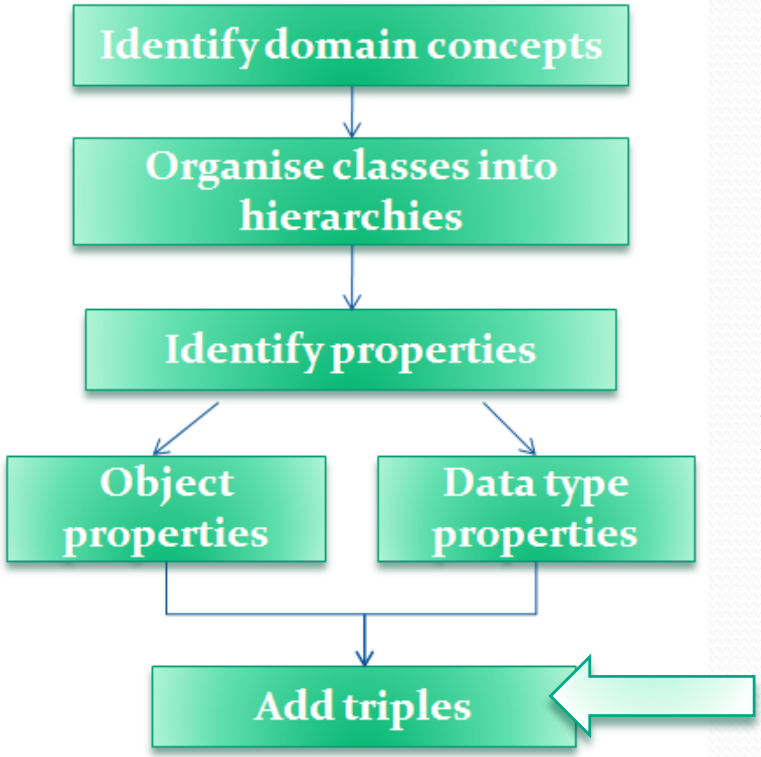
Done!



Object Property

Ontological approach

Ontological Approach



C1

L1

course

lecturer

student

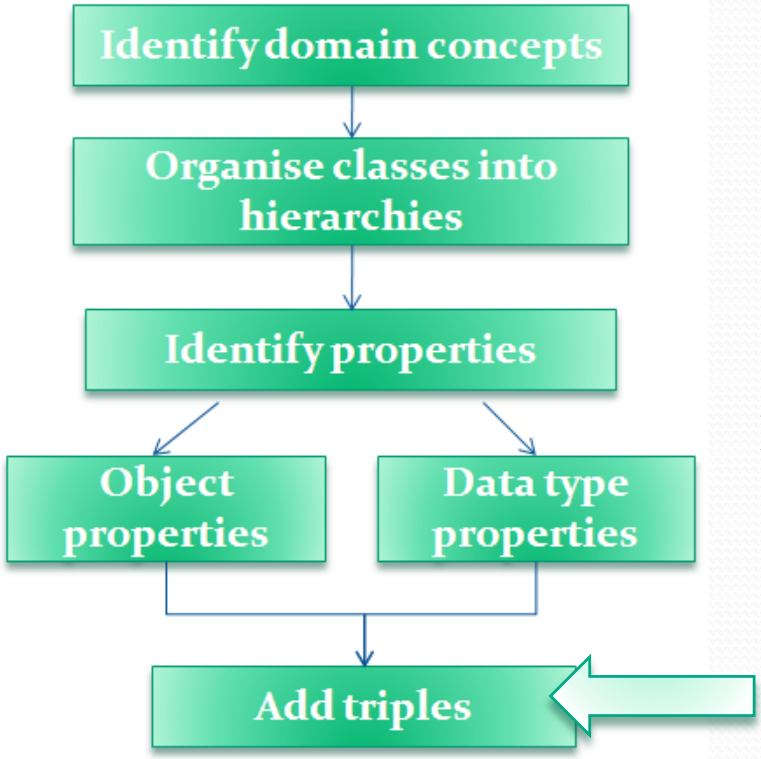
enrol in

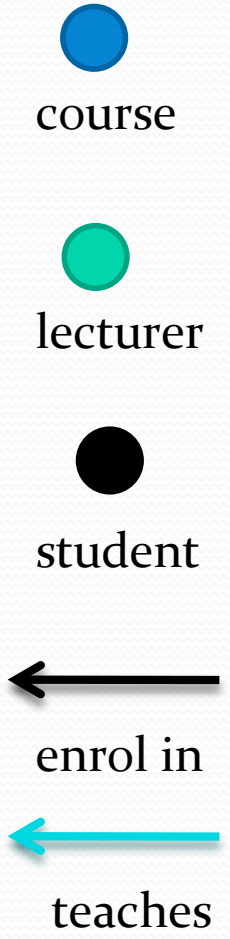
teaches

Add triple

Ontological approach

Ontological Approach

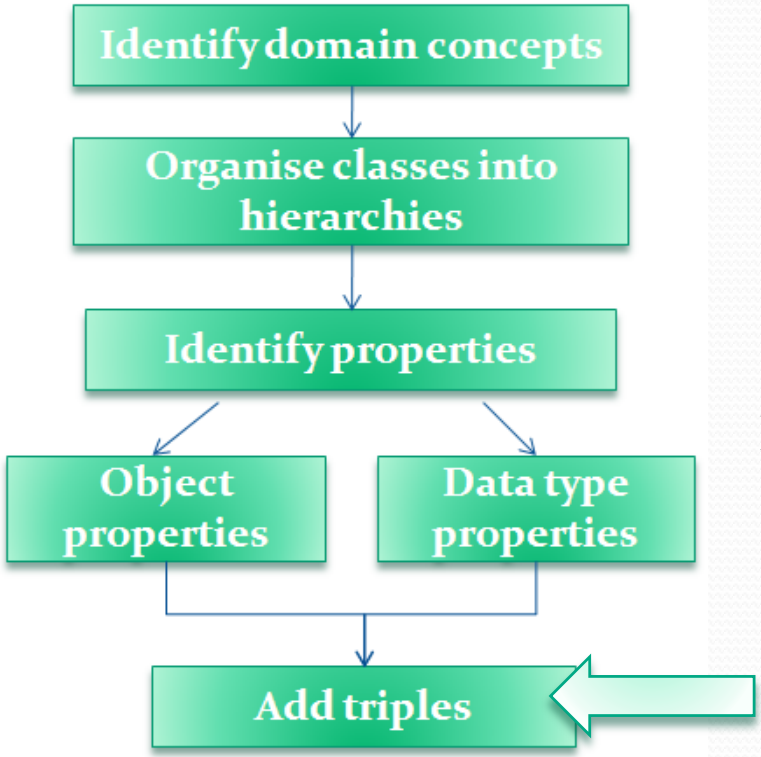


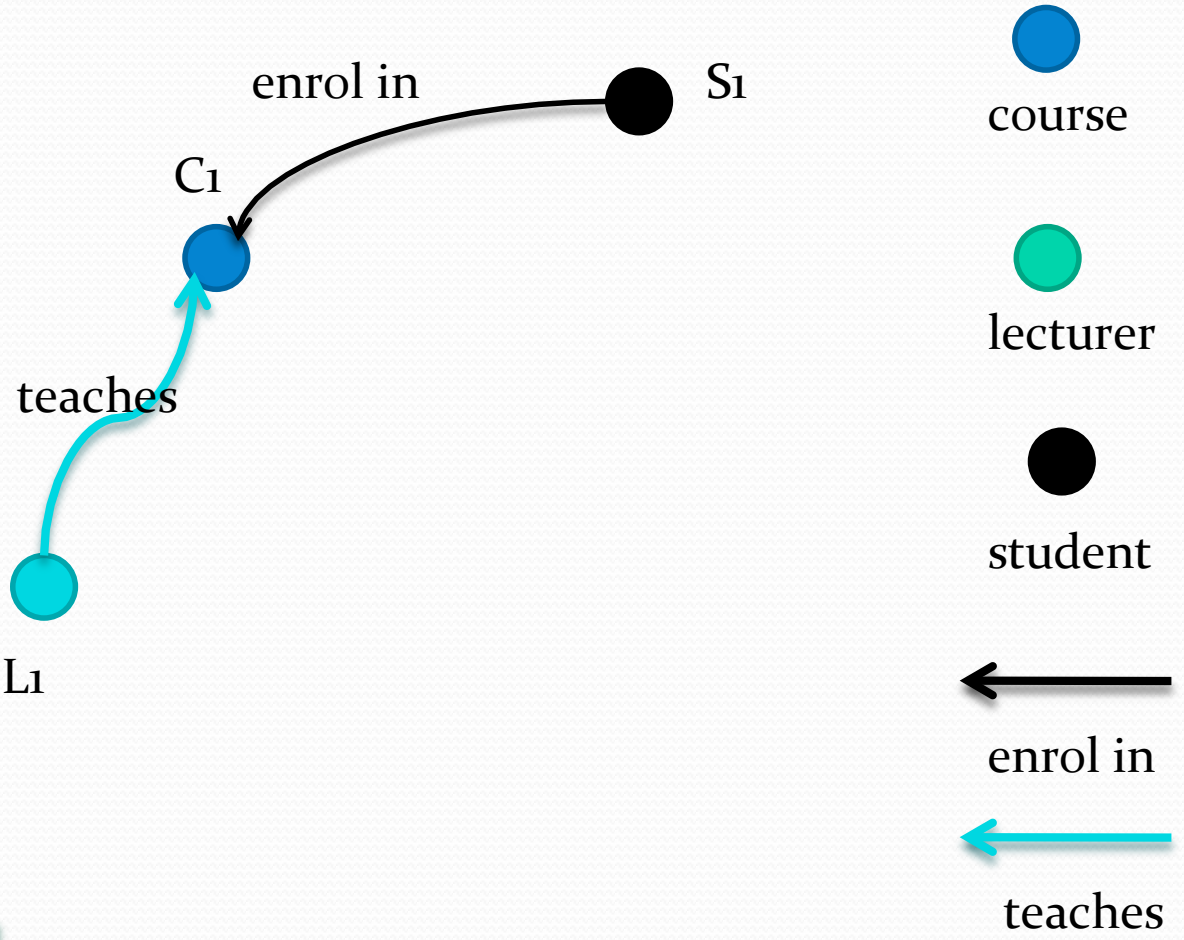


Add triple

Ontological approach

Ontological Approach

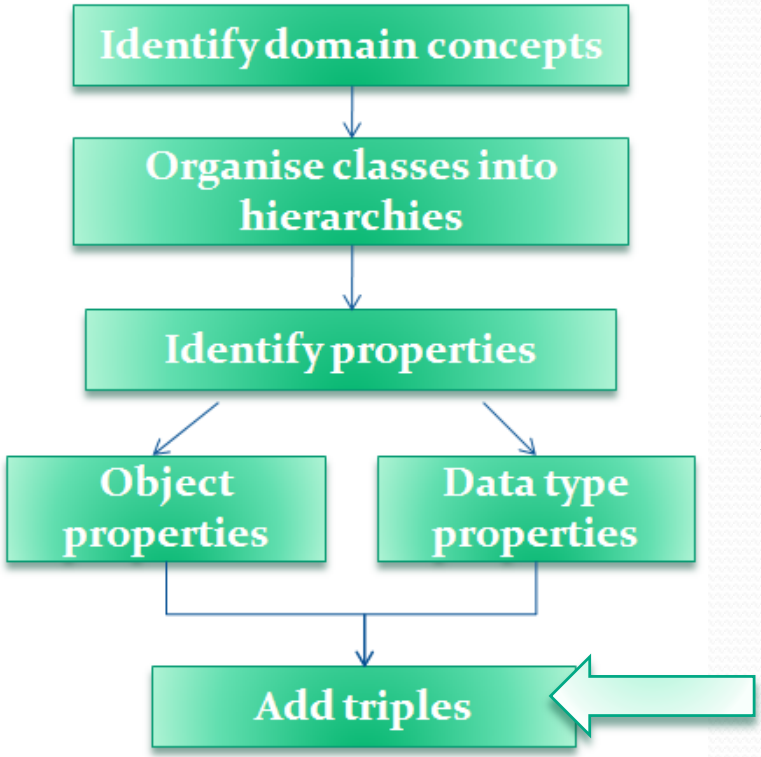


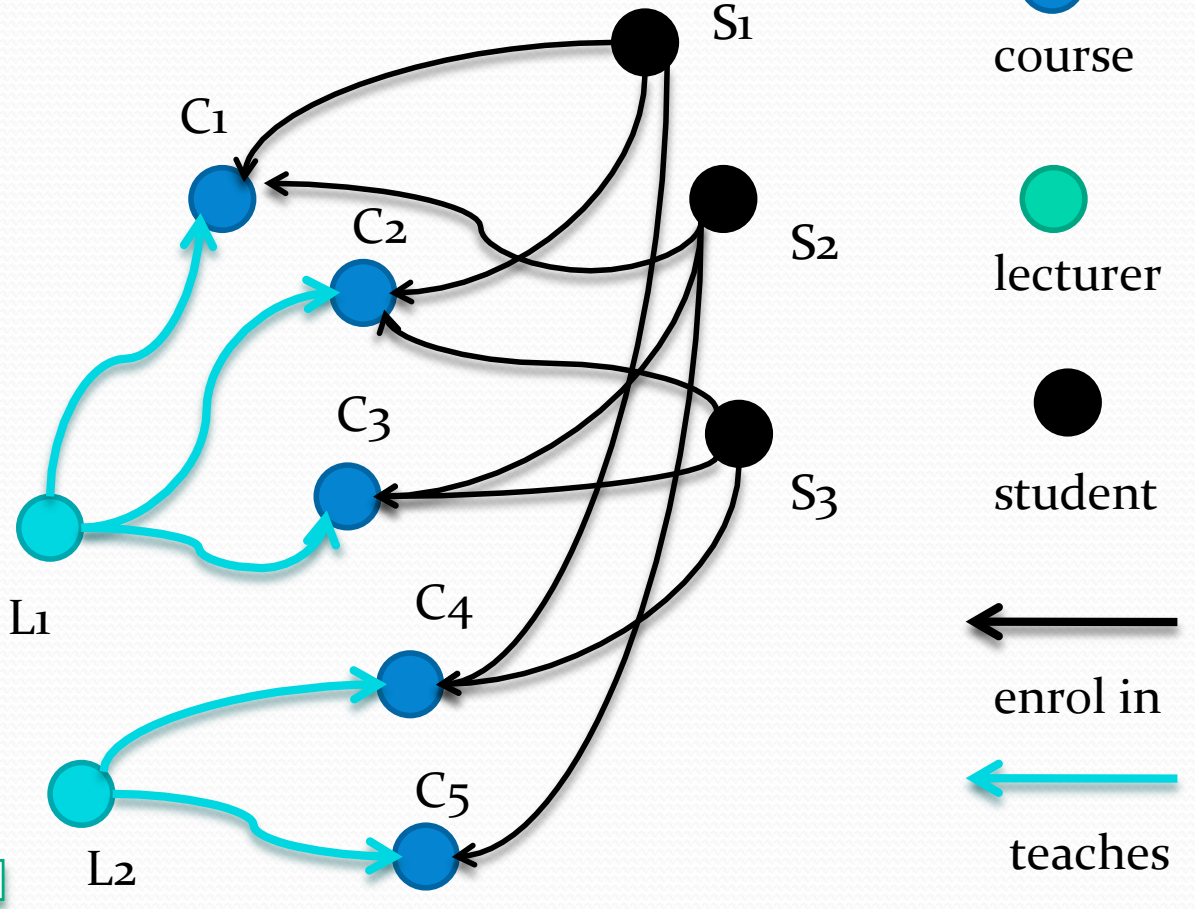


Add triple

Ontological approach

Ontological Approach

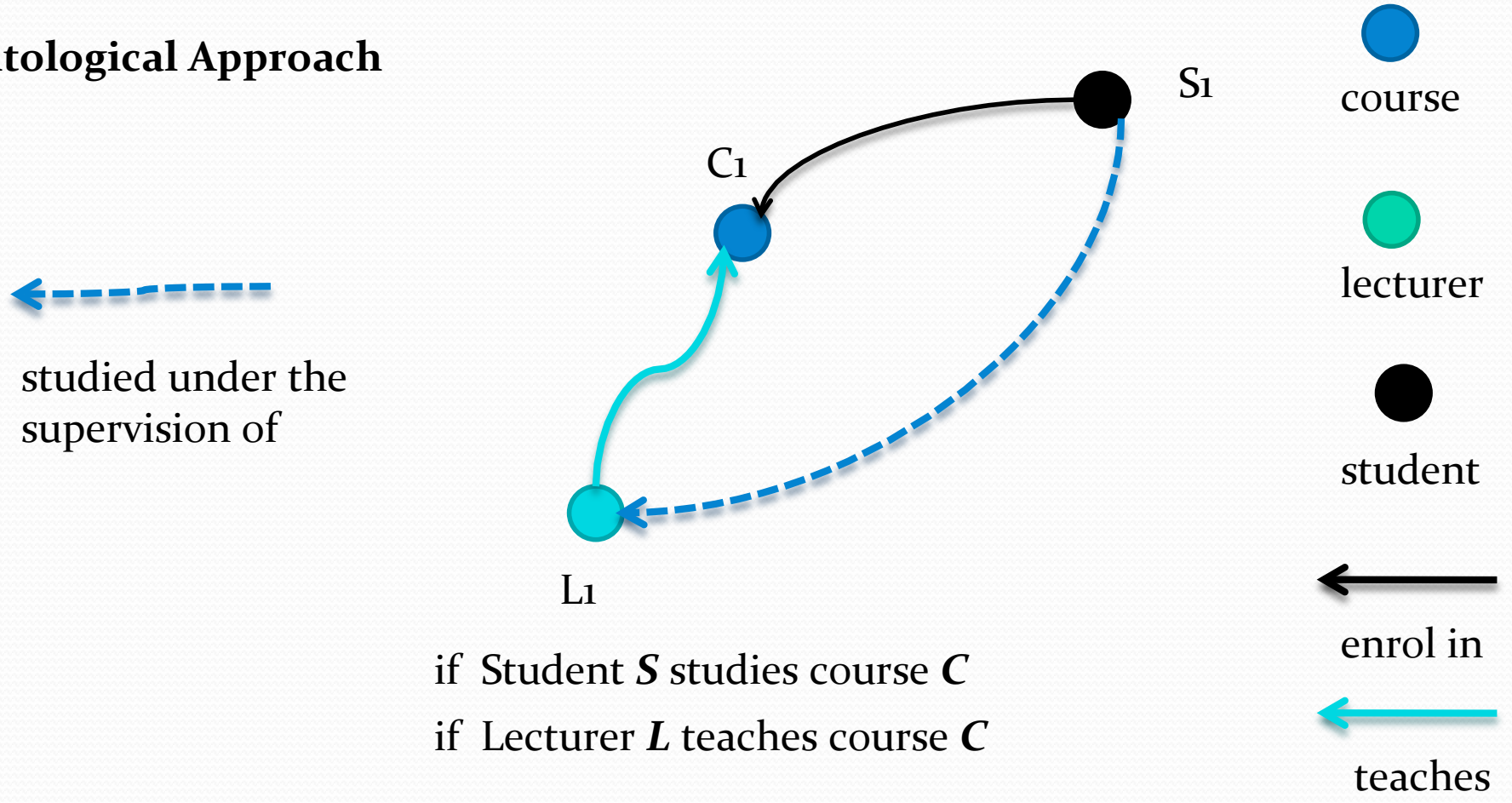




Add triple

Explore implicit relationship

Ontological Approach



if Student ***S*** studies course ***C***

if Lecturer ***L*** teaches course ***C***

Reasoning

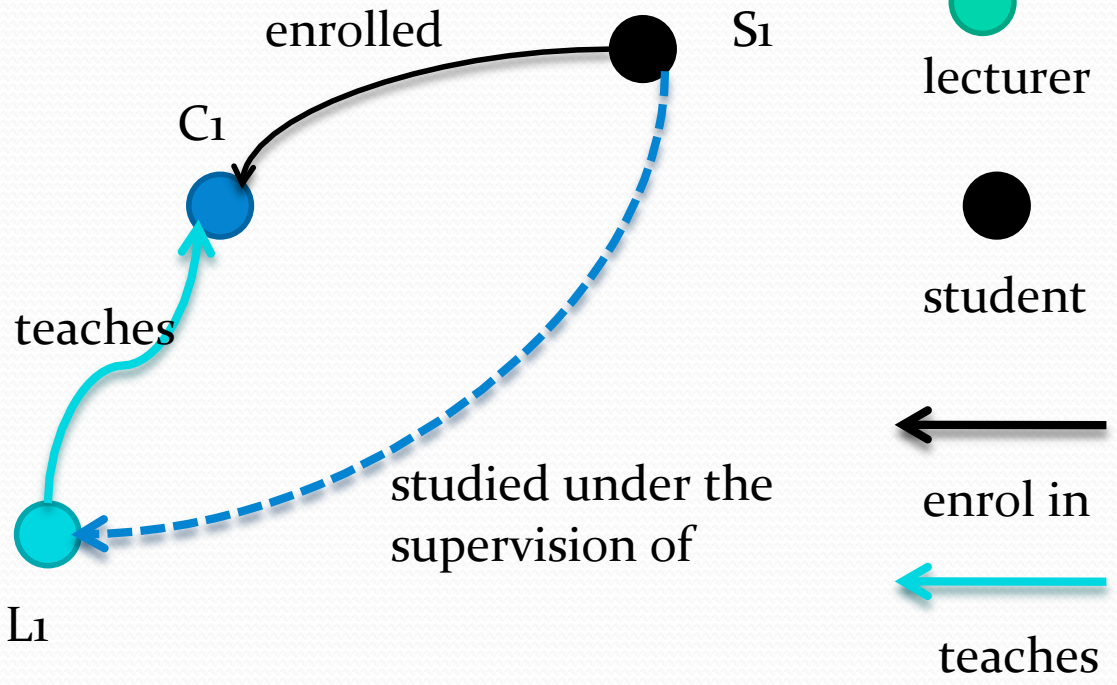
Explore implicit relationship

Deductive reasoning

if Student ***S*** enrolled in course ***C***
AND
Lecturer ***L*** teaches course ***C***

THEN

Student ***S*** studied under the supervision of ***L***



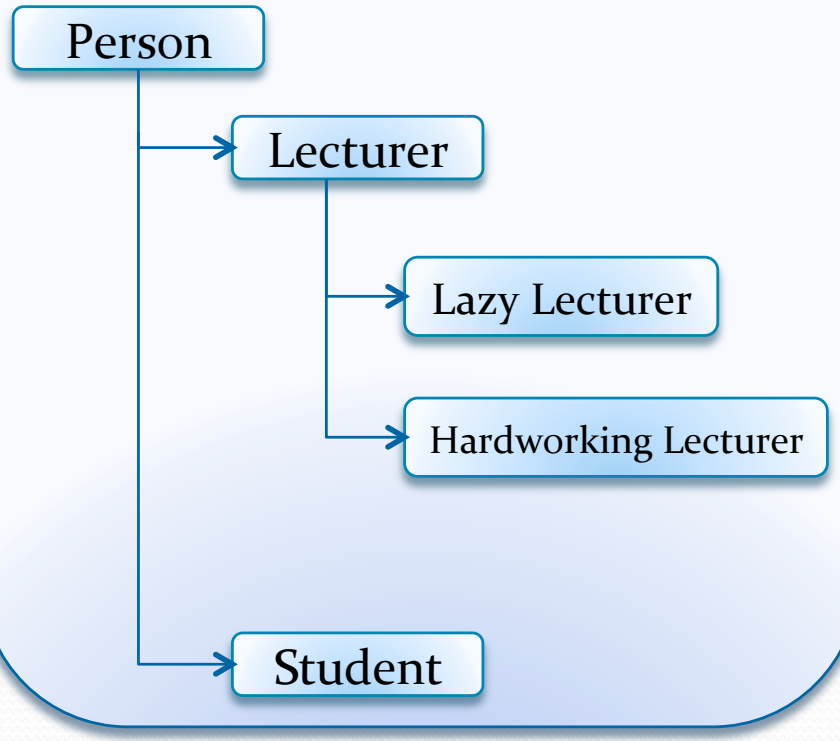
(?s rdf:type Student) (?s rdf:type Student)(?s ***enrolled ?c***) (?l ***teaches*** ?c)
→(?s studies_under the supervision of ?l)

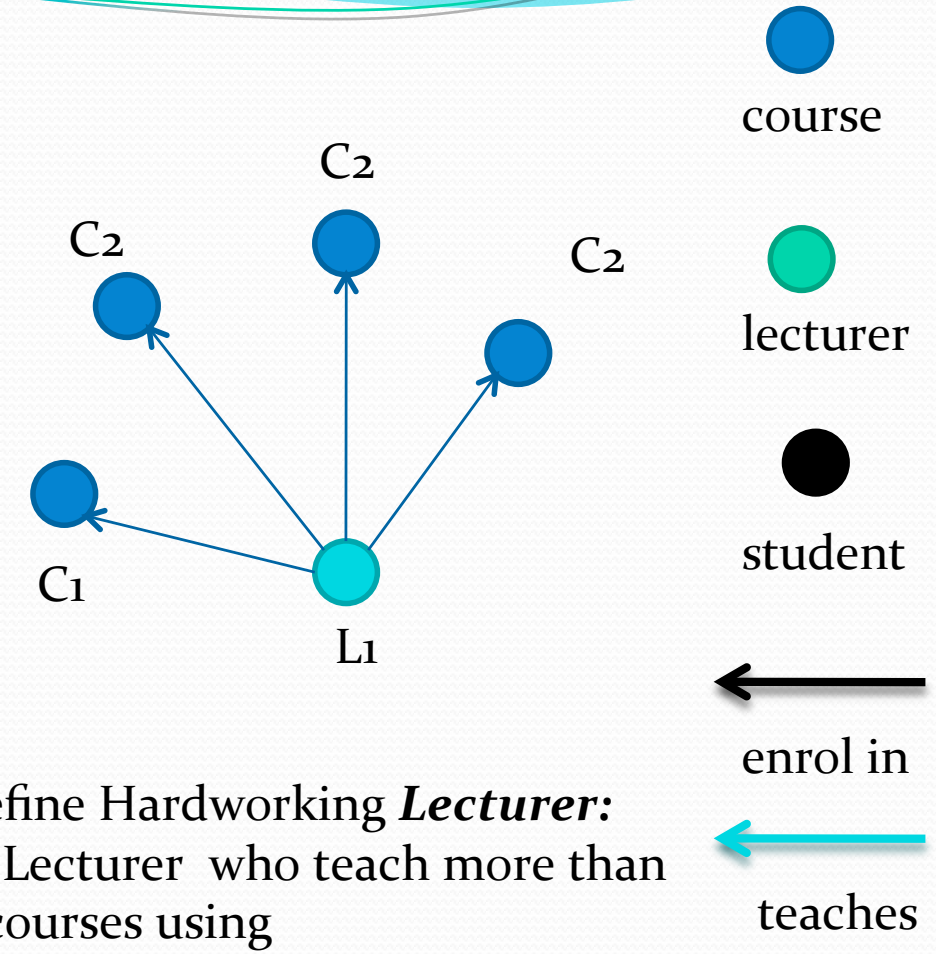
studied under the supervision of

Reasoning

Classification

Class hierarchy





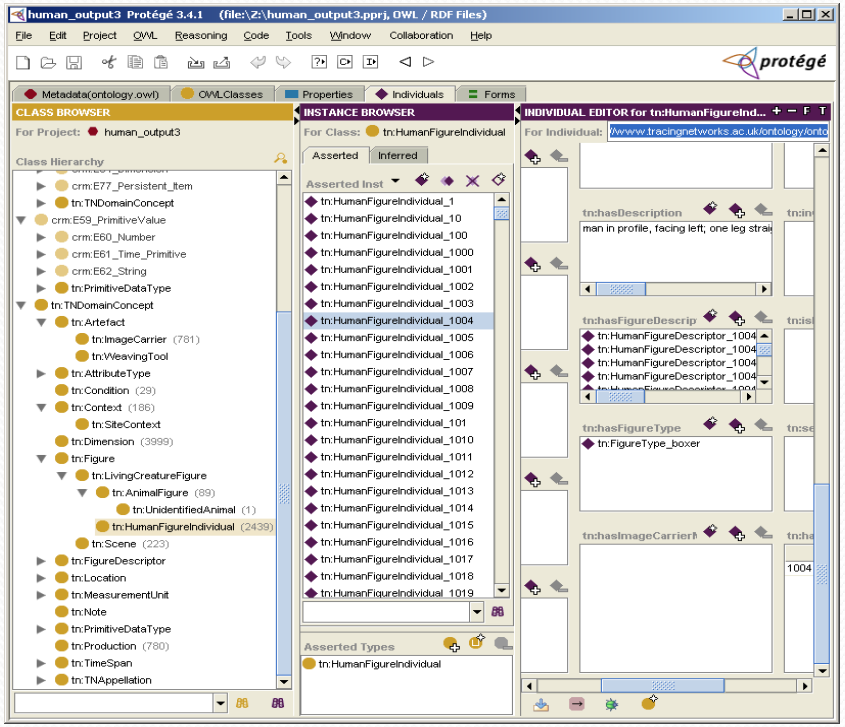
Define Hardworking ***Lecturer:***
A Lecturer who teach more than 4 courses using
Description Logic

Class(Hardworking Lecturer,
subClassOf(Restriction(hasCourse, min-cardinality(4))))

Reasoning

Protégé

Ontology Editor



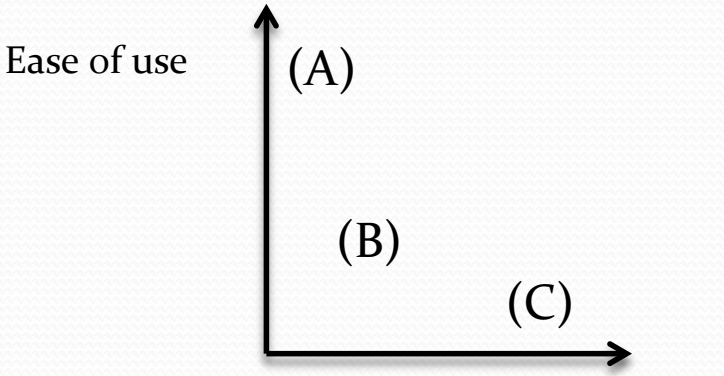
- Download Protége 3.4.7
  - http://protege.stanford.edu/download/protege/3.4/installanywhere/Web_Installers/

http://protege.stanford.edu/

Protége Tutorial

Query Ontology

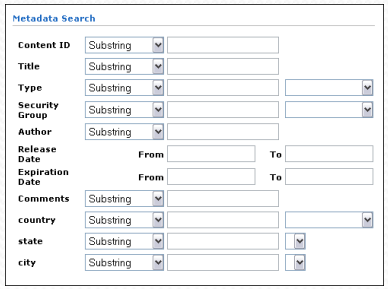
Traditional search interfaces



Single textbox
e.g. Google

Simple but with
limited expressivity

(A)



Advanced search
e.g. Library search

Not flexible
overly complicated
restricted expressivity

(B)

```
PREFIX tn:<http://www.tracingnetworks.ac.uk/loomweight.owl#>
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?motif (count(DISTINCT ?loomweight) as ?loomweight_count)
WHERE{
?loomweight rdf:type tn:Loomweight.
?loomweight tn:has_archaeological_context ?context.
?loomweight tn:has_dimension ?x.
?x tn:upper_bound_value ?val.
?x tn:dimension_type tn:height.
?loomweight tn:decoration ?decoration.
?decoration tn:decoration_technique ?technique.
?context tn:site_type ?site_type.
?decoration tn:decoration_motif ?motif.
FILTER (?val>0.1).
}
```

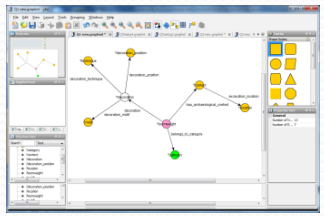
Native query language
e.g. SPARQL

Highly expressive
Complex syntax

(C)

Graph-based query tool

Draw queries (and rules) as graph patterns



Graph editor (e.g. yEd)

input

Graph-based Query Framework

Graph-to-Query Transformation engine

Generate Rules

Reasoning rules

Deductive reasoning rule engine

Build Query

Integrate Rules

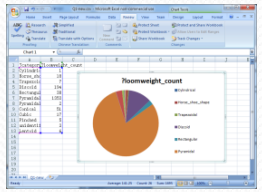
SPARQL query

SPARQL query engine

Ontology-based Database (RDF repository)

execute

output



e.g. Microsoft Excel (CSV)

Graph-based query tool

Graphic notations and colour scheme

Variables *(any nodes, starting with ? symbol)*
Should be included in the result set if node matches the graph pattern.

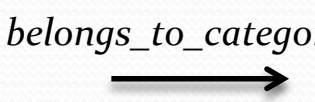
Intermediate variable *(unknown nodes, starts with ? symbol)*
Same as variable but will be excluded from the result set.

Named class or instance
A node that is already known to the user

Group results
Equivalent to "GROUP BY" clause

Count results
Equivalent to aggregate function COUNT()

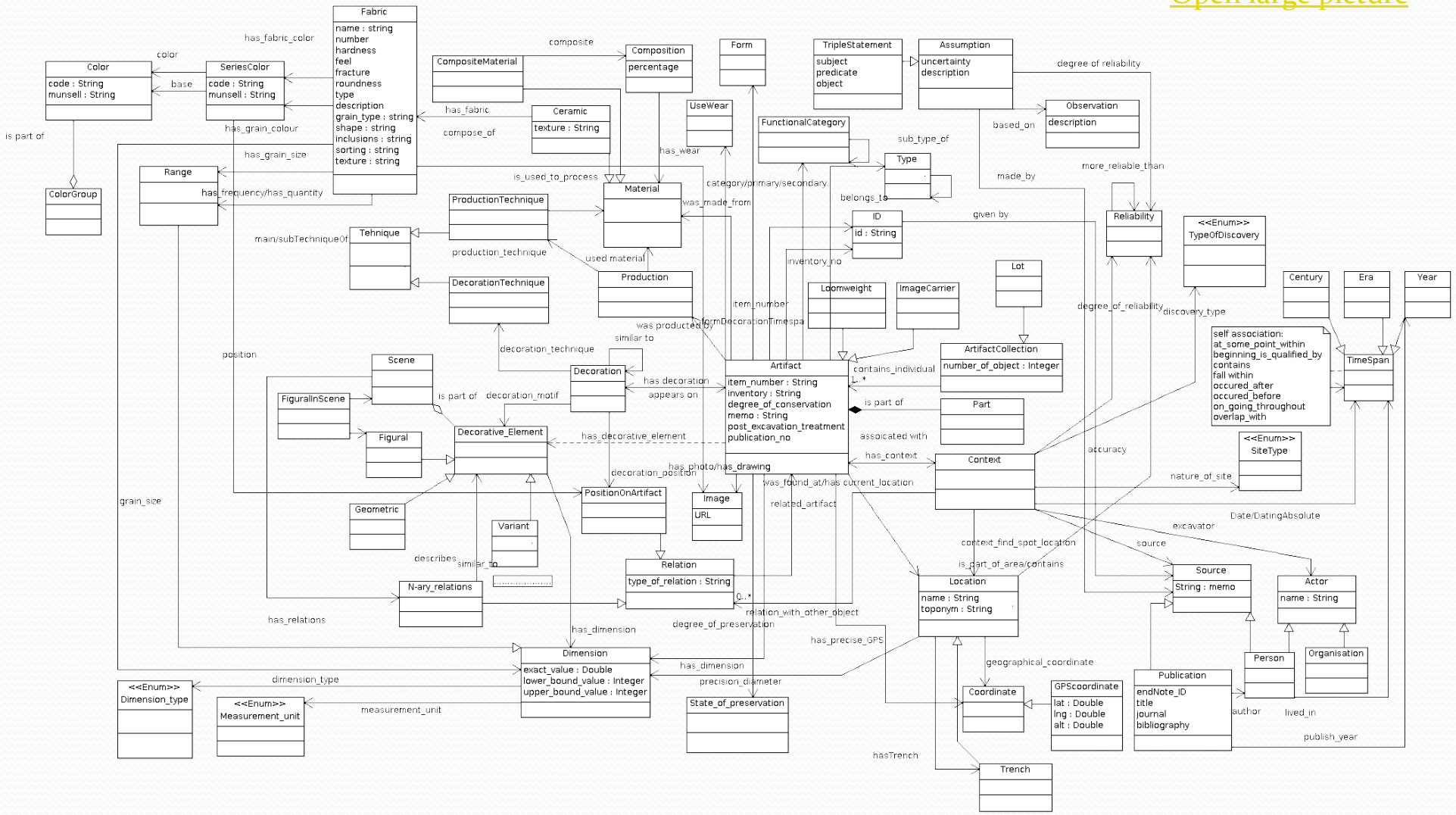
Numeric and string constant
(e.g. 10, 20.5, "abc" etc)



Property (predicate)
(e.g. *has_archaeological_context, excavation_location*, etc)

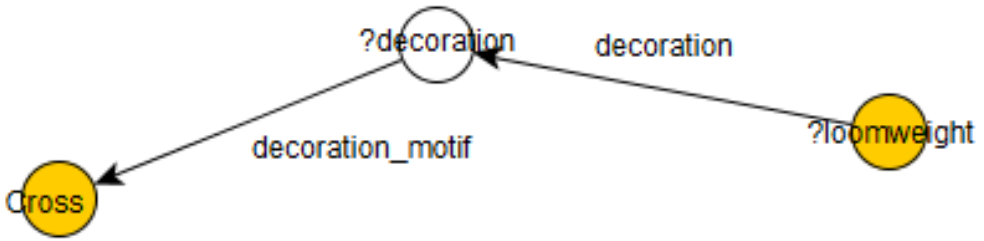
Tracing Networks Ontology

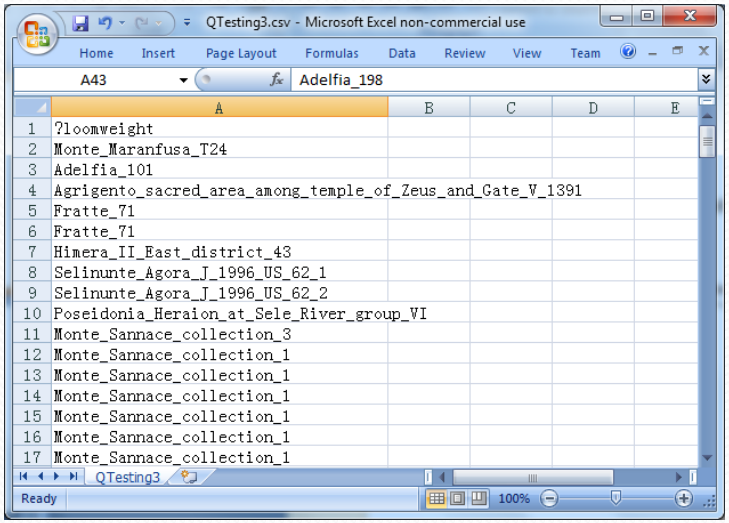
Open large picture



Example 1

Query:
Show all loom-weights with decoration motif: Cross




QTesting3.csv - Microsoft Excel non-commercial use

Home Insert Page Layout Formulas Data Review View Team

A43 Adelfia_198

	A
1	?loomweight
2	Monte_Maranfusa_T24
3	Adelfia_101
4	Agrigento_sacred_area_among_temple_of_Zeus_and_Gate_V_1391
5	Fratte_71
6	Fratte_71
7	Himera_II_East_district_43
8	Selinunte_Agora_J_1996_US_62_1
9	Selinunte_Agora_J_1996_US_62_2
10	Poseidonia_Heraion_at_Sele_River_group_VI
11	Monte_Sannace_collection_3
12	Monte_Sannace_collection_1
13	Monte_Sannace_collection_1
14	Monte_Sannace_collection_1
15	Monte_Sannace_collection_1
16	Monte_Sannace_collection_1
17	Monte_Sannace_collection_1

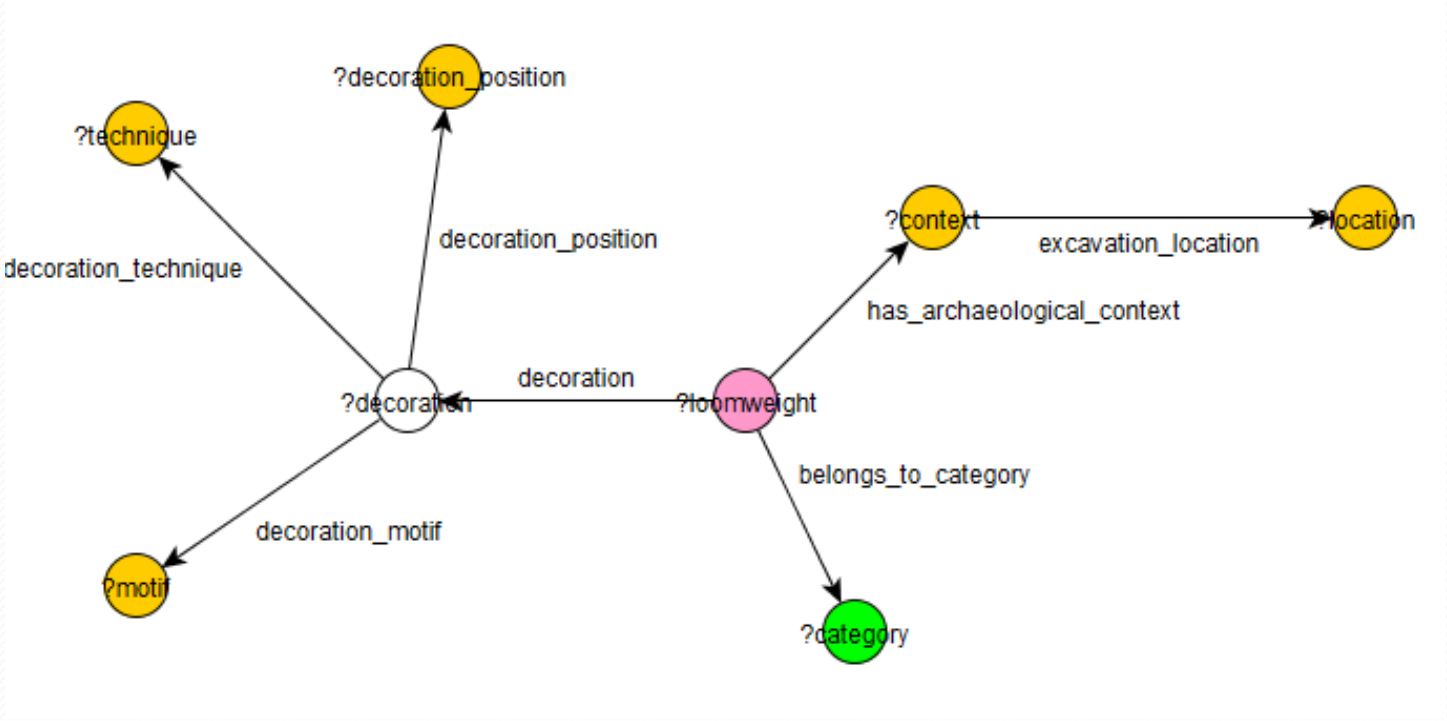
QTesting3

Ready 100%

Q1-Testing3.graphml

Example 2

Query:
Which is the most attested type of loom weight across all sites ?

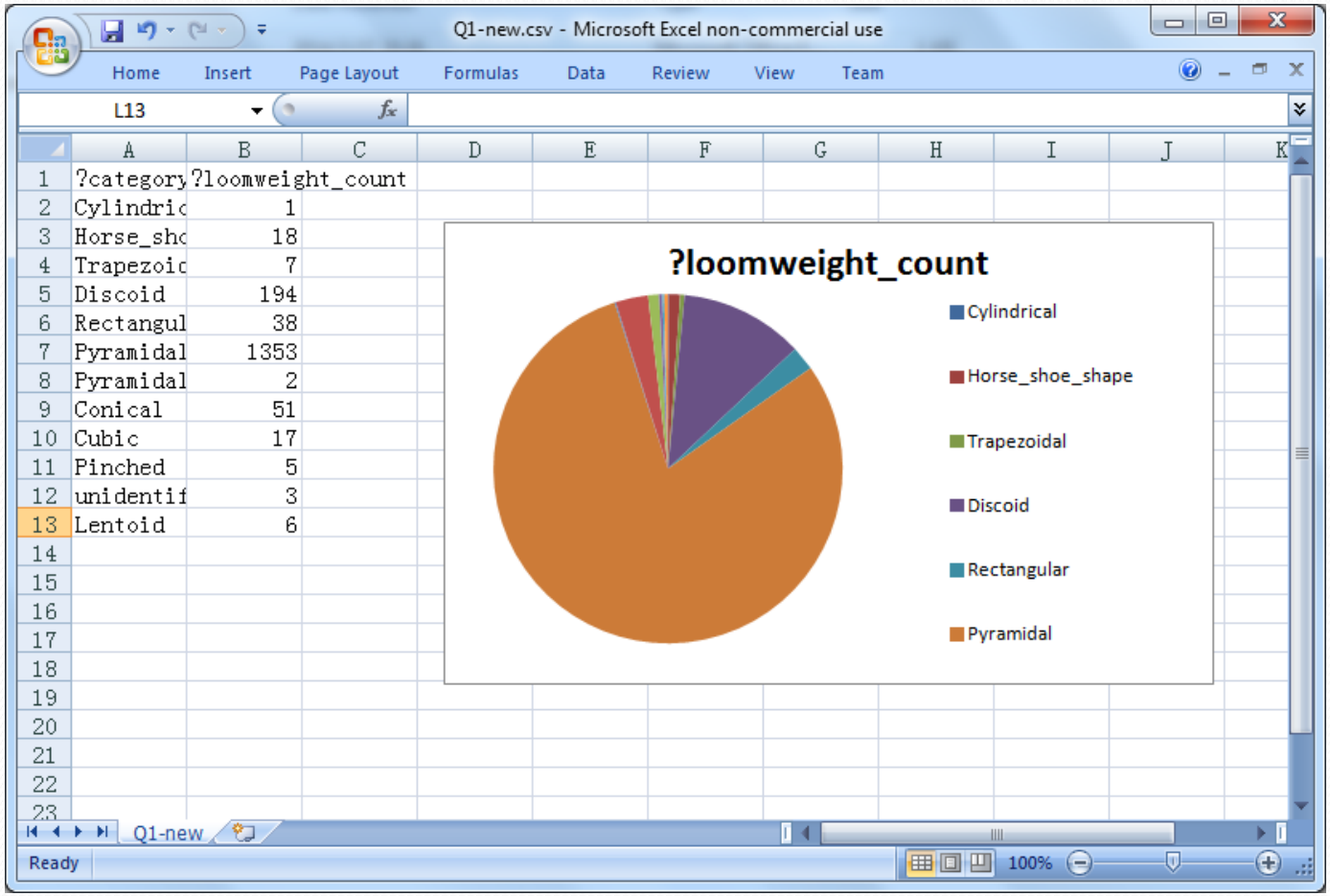


Q1-new.graphml

Example 2

Which is the most attested type of loom weight across all sites ?

Result:

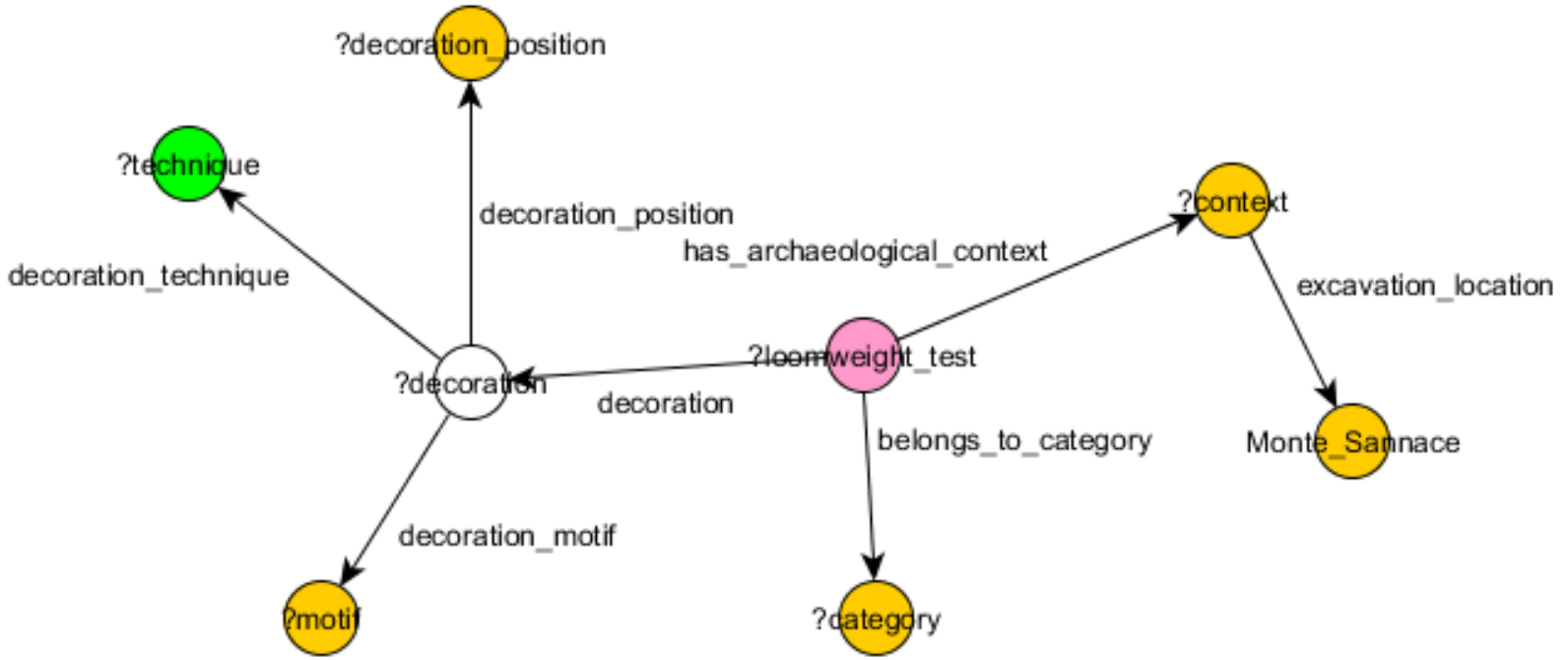


	A	B
1	?category	?loomweight_count
2	Cylindric	1
3	Horse_sho	18
4	Trapezoic	7
5	Discoid	194
6	Rectangul	38
7	Pyramidal	1353
8	Pyramidal	2
9	Conical	51
10	Cubic	17
11	Pinched	5
12	unidentif	3
13	Lentoid	6

Q1-new.graphml

Example 3

Query:
Which techniques were used in the decoration of loom-weights found in Monte_Sannace? (show percentage)

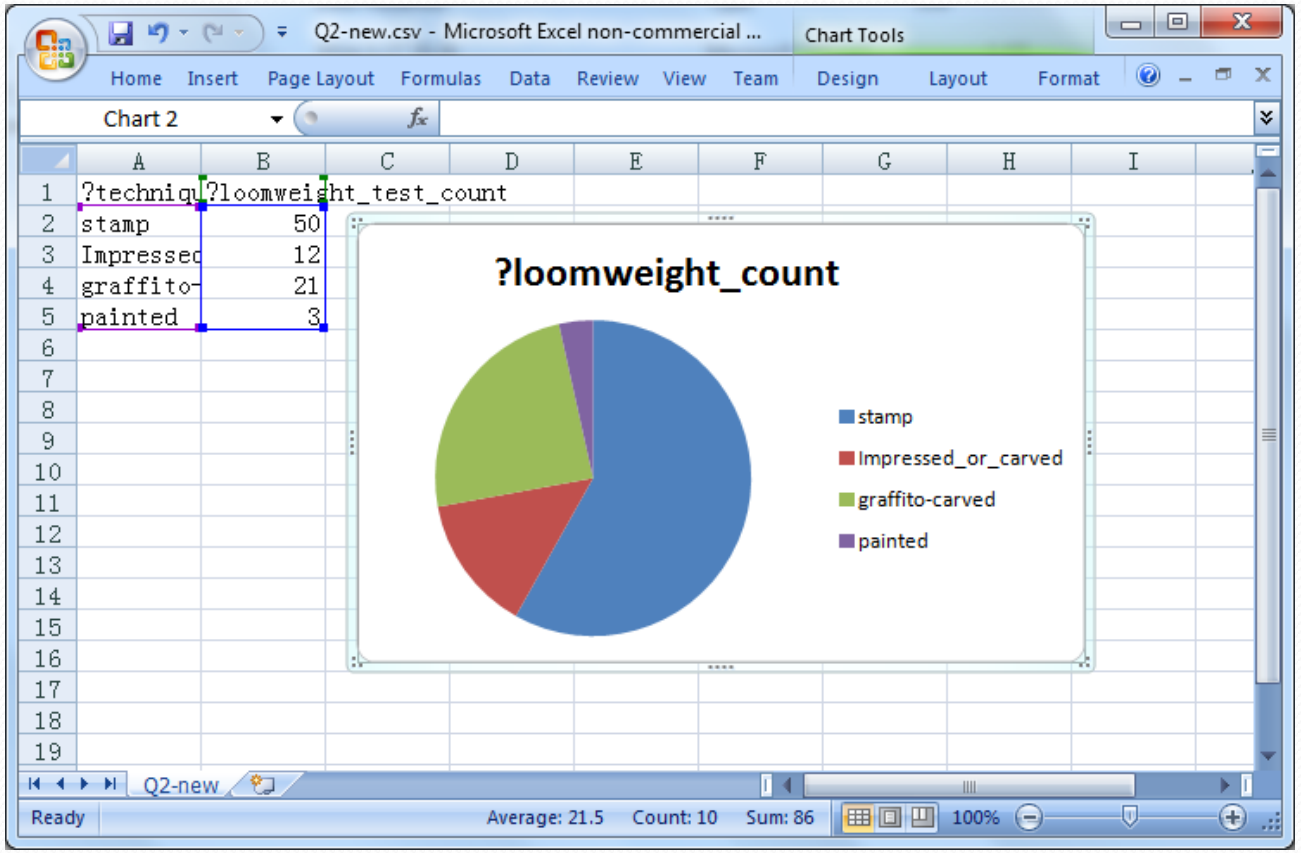


Q2-new.graphml

Example 3

Which techniques were used in the decoration of loom-weights found in Monte_Sannace? (show percentage)

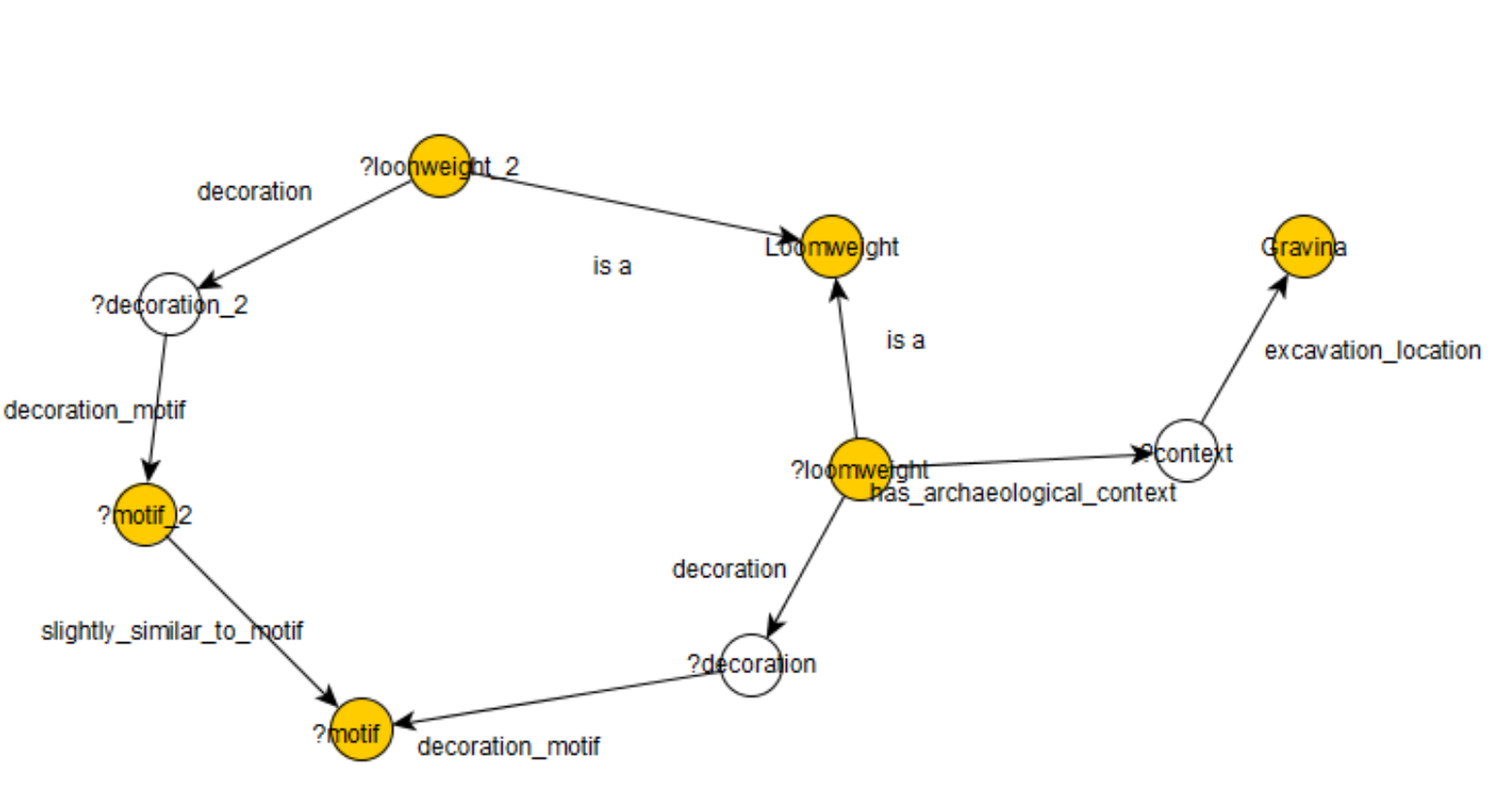
Result



Q2-new.graphml

Example 4

Query:
Show loom-weights found in other sites whose motifs are slightly similar to those found in Gravina

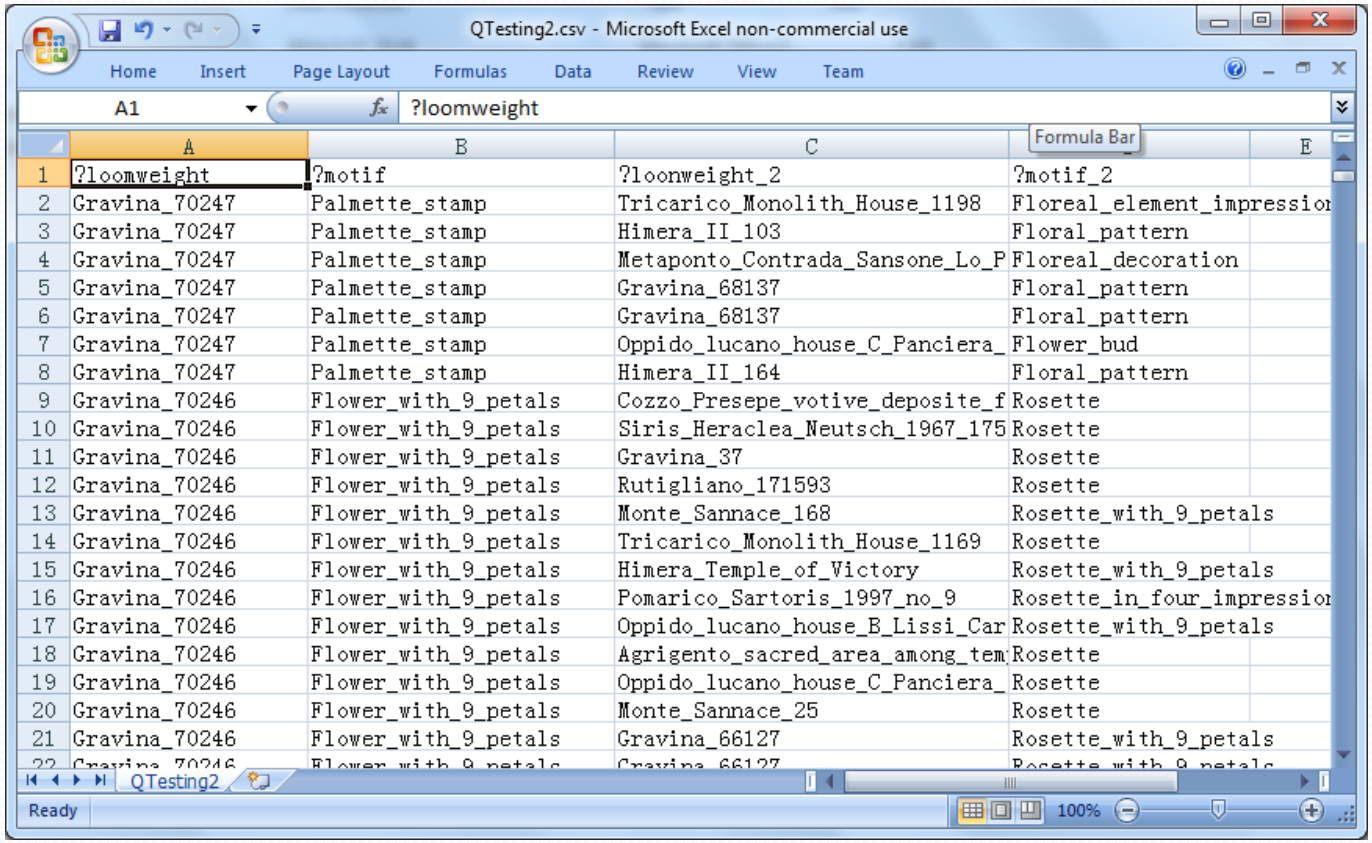


QTesting2.graphml

Example 4

Show loom-weights found in other sites whose motifs are slightly similar to those found in Gravina

Result

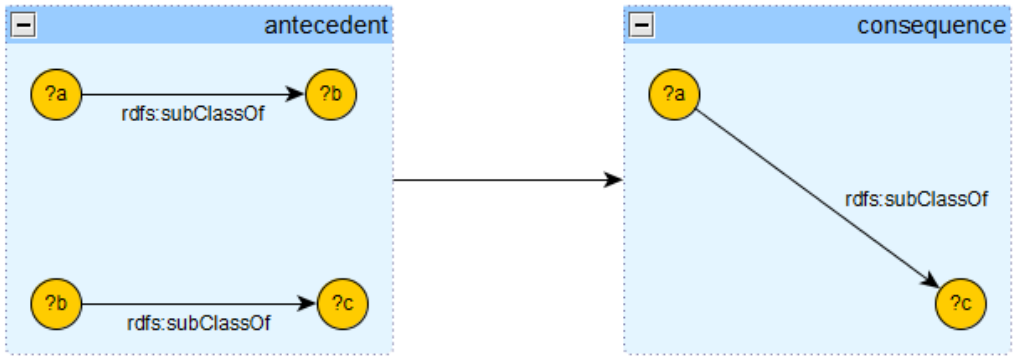


	A	B	C	D	E
1	?loomweight	?motif	?loonweight_2	?motif_2	
2	Gravina_70247	Palmette_stamp	Tricarico_Monolith_House_1198	Floreal_element_impressior	
3	Gravina_70247	Palmette_stamp	Himera_II_103	Floral_pattern	
4	Gravina_70247	Palmette_stamp	Metaponto_Contrada_Sansone_Lo_P	Floreal_decoration	
5	Gravina_70247	Palmette_stamp	Gravina_68137	Floral_pattern	
6	Gravina_70247	Palmette_stamp	Gravina_68137	Floral_pattern	
7	Gravina_70247	Palmette_stamp	Oppido_lucano_house_C_Panciera_	Flower_bud	
8	Gravina_70247	Palmette_stamp	Himera_II_164	Floral_pattern	
9	Gravina_70246	Flower_with_9_petals	Cozzo_Presepe_votive_deposite_f	Rosette	
10	Gravina_70246	Flower_with_9_petals	Siris_Heraclea_Neutsch_1967_175	Rosette	
11	Gravina_70246	Flower_with_9_petals	Gravina_37	Rosette	
12	Gravina_70246	Flower_with_9_petals	Rutigliano_171593	Rosette	
13	Gravina_70246	Flower_with_9_petals	Monte_Sannace_168	Rosette_with_9_petals	
14	Gravina_70246	Flower_with_9_petals	Tricarico_Monolith_House_1169	Rosette	
15	Gravina_70246	Flower_with_9_petals	Himera_Temple_of_Victory	Rosette_with_9_petals	
16	Gravina_70246	Flower_with_9_petals	Pomarico_Sartoris_1997_no_9	Rosette_in_four_impressior	
17	Gravina_70246	Flower_with_9_petals	Oppido_lucano_house_B_Lissi_Car	Rosette_with_9_petals	
18	Gravina_70246	Flower_with_9_petals	Agrigento_sacred_area_among_tem	Rosette	
19	Gravina_70246	Flower_with_9_petals	Oppido_lucano_house_C_Panciera_	Rosette	
20	Gravina_70246	Flower_with_9_petals	Monte_Sannace_25	Rosette	
21	Gravina_70246	Flower_with_9_petals	Gravina_66127	Rosette_with_9_petals	

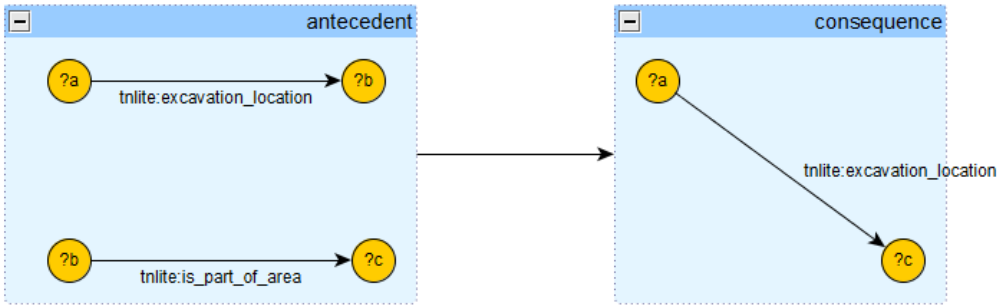
QTesting2.graphml

Rules

Generic rules (transparent to end users)



Customised rules



Installation

- All-in-one package
  - Including Protége 3.5, QueryMaker and yEd Graph editor

- Protége 3.5
  - http://protege.stanford.edu/download/protege/3.5/installanywhere/Web_Installers/
- yEd Graph editor
  - http://www.yworks.com/en/products_yed_download.html
- QueryMaker