

Methods and Tools for Developing Ontology- Based Data Management Solutions

Semantics 2018 Tutorial

HANDS-ON SESSION

Domenico Lembo, Valerio Santarelli, Domenico Fabio Savo



SAPIENZA
UNIVERSITÀ DI ROMA

Outline

1. Creation of the ontology
2. Creation of the mappings
3. Execution of a query

Outline

1. Creation of the ontology
2. Creation of the mappings
3. Execution of a query

Domain description

In this exercise, we focus on the world of DC comics, specifically its Characters and their Abilities. Among characters, there are Humans, Metahumans, and Extraterrestrials. These species are mutually disjoint, i.e., a human cannot be neither a metahuman nor an extraterrestrial, and so on. Every character has a name(`character_name`), that is a string, and extraterrestrials (and only extraterrestrials) are associated to the name (a string) of their planet of origin.

Some characters are Villains whereas others are Superheroes. A villain cannot be a superhero. Villains are enemies of superheroes. Notice that each villain is enemy of at least one superhero.

A character can have an Ability. Every ability has a name(`ability_name`), which is a string. Finally, we know that some abilities are Superpowers.

Eddy

1. Open Superheroes project
 1. Double-click on superheroes diagram from Project Explorer widget
2. Use Palette to draw Graphol ontology
3. Save project
4. Export to OWL

Outline

1. Creation of the ontology
- 2. Creation of the mappings**
3. Execution of a query

The datasource

We consider a PostgreSQL source database called "comics_db", which has a schema called "characters_db" containing the following three tables:

- **Characters:** which contains information about each character;
- **Abilities:** in which each character is associated to her own abilities;
- **Supercrimes:** which contains data about each (super)crime, the villain that has committed it and the superhero that has foiled it.

More details about each table are given in the next slides

The "characters" table (1/2)

id_ch	name	city	planet_of_origin	species
34	Zatanna			Human
35	Superman	Metropolis	Krypton	Kryptonian
38		GothamCity		Human
				Human

- **id_ch**: it indicates the code used to identify a character; this is the primary key of the table.
- **name**: it indicates the name of the character; it cannot be null.
- **city**: it indicates the city where the base of the character is collocated; it can be null.
- **planet_of_origin**: it indicates the name of the planet of origin of the character; if the character is a terrestrial, its value is NULL.
- **species**: it indicates the species which the character belongs to; it cannot be null.

The "characters" table (2/2)

id_ch	name	city	planet_of_origin	species
34	Zatanna			Human
35	Superman	Metropolis	Krypton	Kryptonian
38		GothamCity		Human
				Human

From the analysis of the DB we know that a character is Human if her species is "Human", an Extraterrestrial if a planet of origin is specified, and a Metahuman in all the other cases.

The "abilities" table

id_ch	superpower	ability_name
35	true	Flight
48	false	Master archer
67	true	Superstrength
		Superstrength

- **id_ch**: it denotes the identification code of the character who has the ability; it is part of the primary key of the table and references the **id_ch** column of the **characters** table (foreign key).
- **superpower**: it is **true** if the ability is a superpower, and **false** otherwise.
- **name**: it contains the name of the ability; it is part of the primary key of the table.

The "supercrimes" table (1/2)

criminal_id	crime_date	crime_description	foiled_by
289	21-July-2015	Puppies kidnapping	39
192	22-April-2016	Game of thrones spoiling	41
289		Adorable old woman snatching	35
			35

- **criminal_id**: it denotes the identification code of the character who has committed the crime; it is part of the primary key of the table and references the **id_ch** column of the **characters** table (foreign key).
- **crime_date**: it contains the date of the crime and is part of the primary key of the table.
- **crime_description**: it contains the description of the crime;
- **foiled_by**: it is the identification code of the character who has foiled the crime; it references the **id_ch** column of the **characters** table (foreign key), and cannot be NULL.

The "supercrimes" table (1/2)

criminal_id	crime_date	crime_description	foiled_by
289	21-July-2015	Puppies kidnapping	39
192	22-April-2016	Game of thrones spoiling	41
289		Adorable old woman snatching	35
			35

From the analysis of the DB we know that a character is a Villian if she has committed at least one crime, and is a Superhero if she has foiled at least one crime. Moreover, a Villian is enemy of every Superhero who has foiled at least one of her crimes.


Mastro Protégé plug-in

1. Open the superheroes ontology
2. Move to Mastro tab and from Configuration sub-tab
 1. Connect to DB
 2. Start Reasoner
3. Move to Mapping Views sub-tab and create SQL Views
4. Move to Mappings sub-tab and create mappings

Outline

1. Creation of the ontology
2. Creation of the mappings
3. Execution of a query

Mastro Protégé plug-in

1. Move to SPARQL Query sub-tab
 1. Write SPARQL query code in Query Editor
 2. Assign ID to query
 3. Run query ( button)
2. Try running other queries from catalog