Abilità informatiche

A.A. 2023/2024

05b - Modellazione dei Dati

Sebastian Barzaghi
sebastian.barzaghi2@unibo.it
https://orcid.org/0000-0002-0799-1527

Riassunto della lezione precedente (05b)

La nave di Teseo

Nave T.

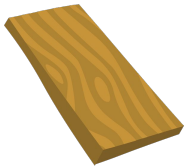
sostituisco 1 pezzo

sostituisco 1 pezzo

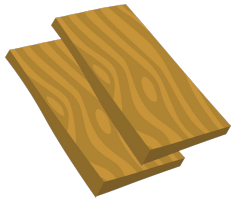
...

A

uso 1 pezzo



uso 1 pezzo



...

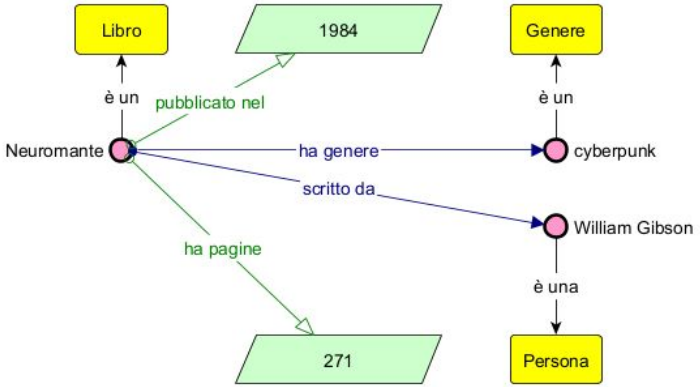
B

giorno 1

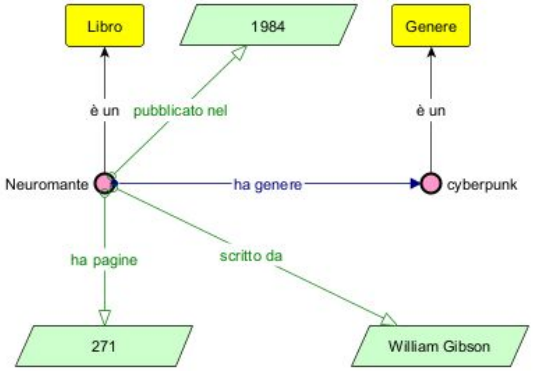
giorno 2

giorno 1000

Non esiste un unico modello

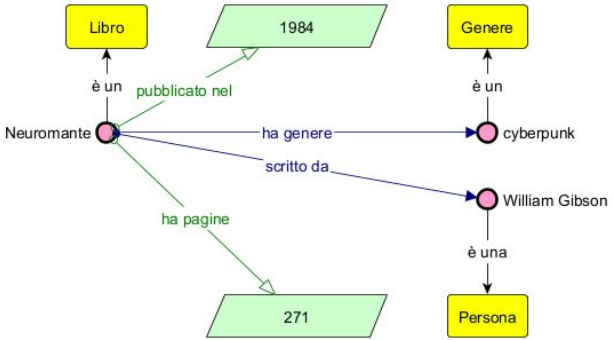


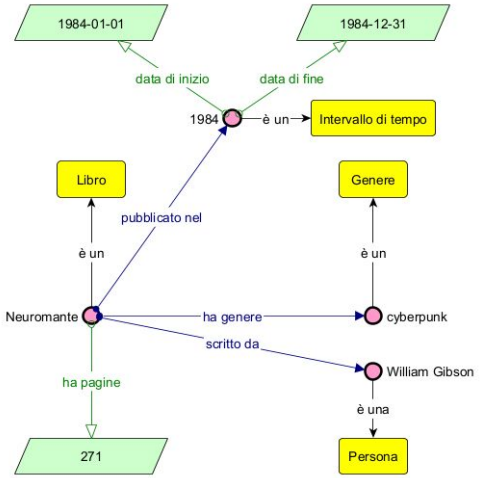
Ipotizziamo che non abbiamo interesse a modellare le persone come entità, ma ci basta sapere i nomi delle persone coinvolte nel ciclo di vita del libro



- Non abbiamo più la classe Persona
- “scritto da” diventa un attributo di Libro
- Modello più semplice, ma perdiamo espressività ed elasticità

Non esiste un unico modello

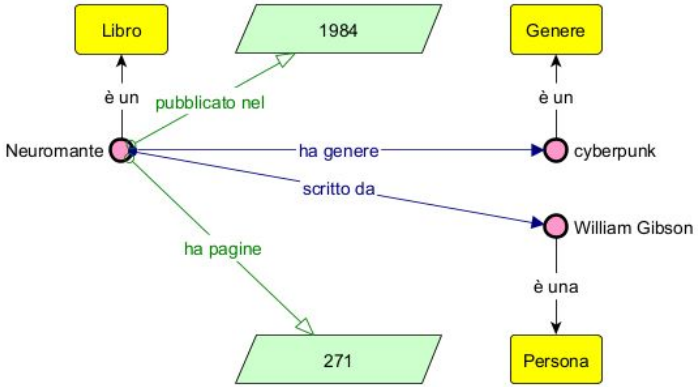


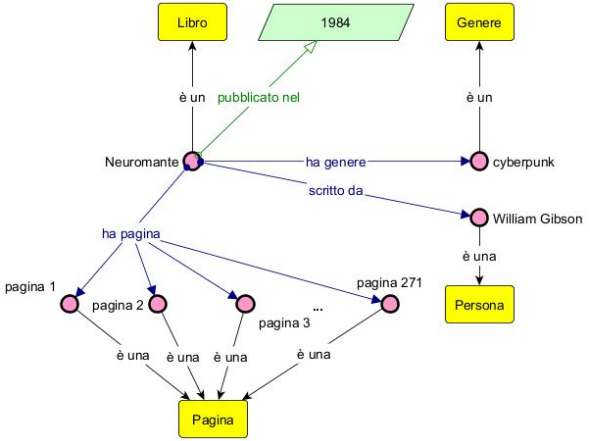


Ipotizziamo che abbiamo interesse a modellare il tempo come periodo caratterizzato da un inizio e una fine

- Abbiamo una nuova entità “1984” appartenente alla nuova classe “Intervallo di tempo” con due attributi “data di inizio” e “data di fine”
- “pubblicato nel” diventa una relazione
- Modello più complesso ed espressivo

Non esiste un unico modello





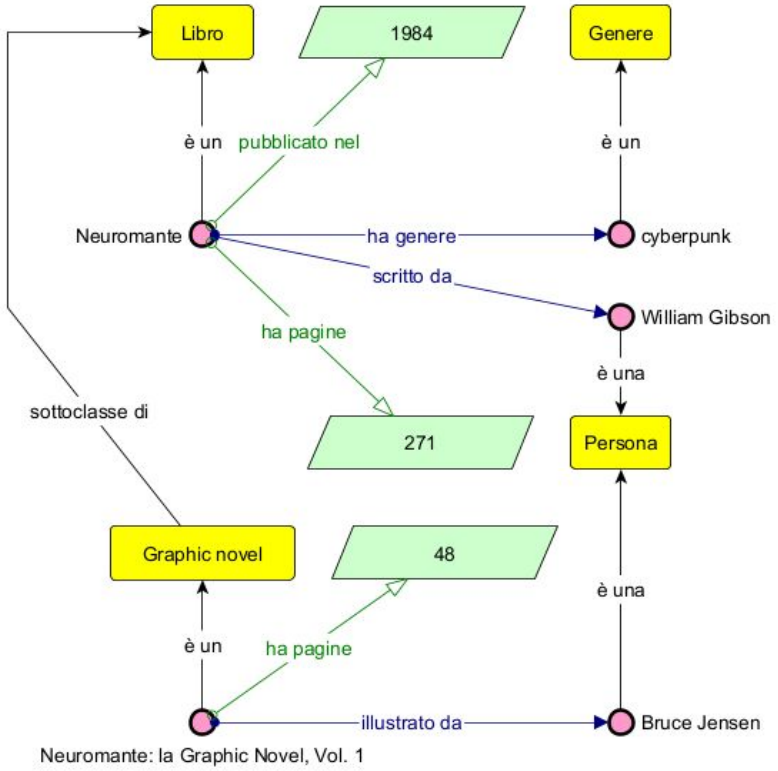
Ipotizziamo che abbiamo interesse a modellare le pagine singole

- Abbiamo 271 nuove entità appartenente alla nuova classe “Pagina”
- “ha pagine” diventa una relazione
- Modello più complesso ed espressivo

Ereditarietà

Una **sottoclasse eredita** tutte le proprietà della propria superclasse e (solitamente) ha proprietà aggiuntive che ne giustificano l'esistenza

"Graphic novel" eredita TUTTE le proprietà di "Libro" (in questo caso stiamo utilizzando solo "ha pagine"), e in aggiunta ha anche "illustrato da"



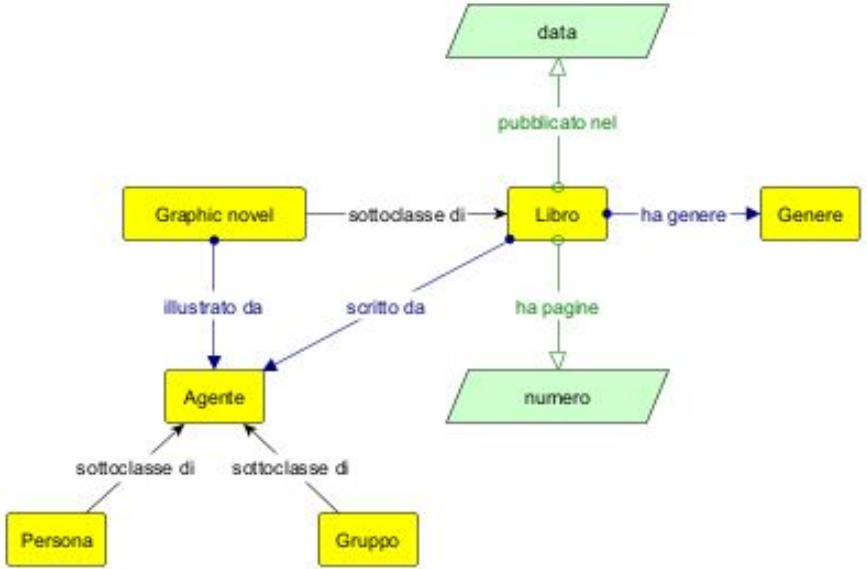
Neuromante: la Graphic Novel, Vol. 1

Ereditarietà

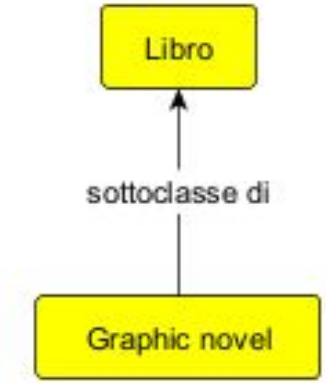
Aggiungiamo una nuova classe “Gruppo”

Però: “scritto da” e “illustrato da” possono applicarsi sia a persone singole che a gruppi

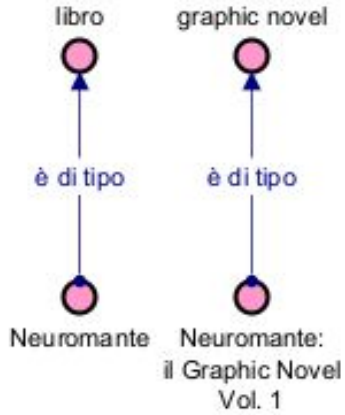
Invece di assegnare le stesse proprietà a due classi diverse, possiamo assegnarle ad una superclasse a loro comune: Agente



Proprietà o classe: tipizzazione



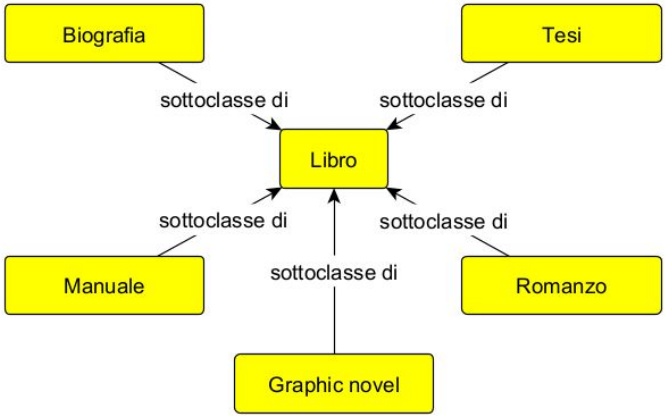
OPPURE

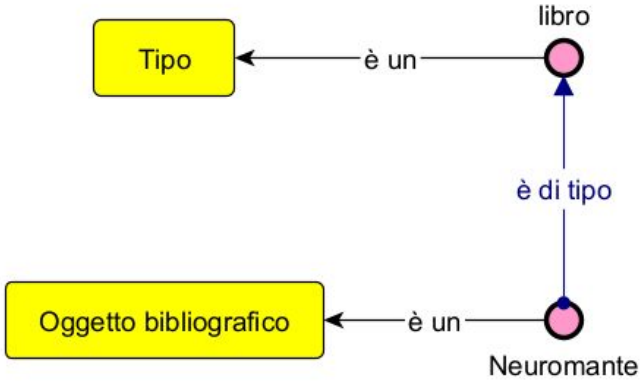


I concetti di Libro e Graphic novel sono Classi

I concetti di Libro e Graphic novel sono *Tipi*: ci servono per classificare oggetti bibliografici

Proprietà o classe: tipizzazione



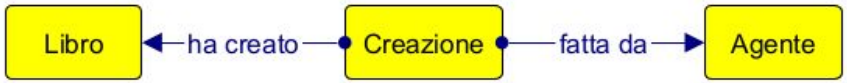


Molteplici classi particolari (maggiore espressività, molto difficile da mantenere)

Stessa differenziazione, ma con meno classi e nessuna informazione particolare (minore espressività, molto più facile da gestire)

Proprietà o classe: eventi

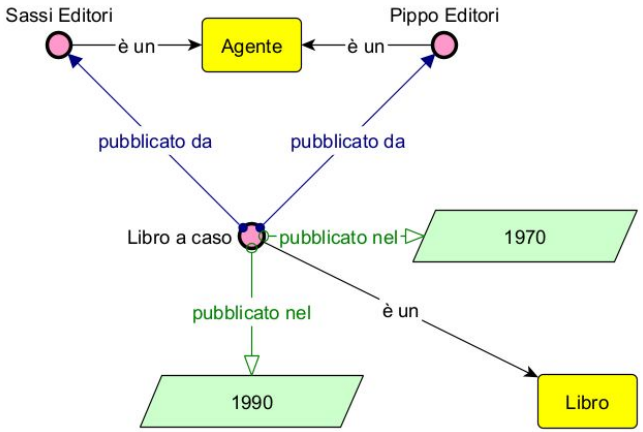
OPPURE



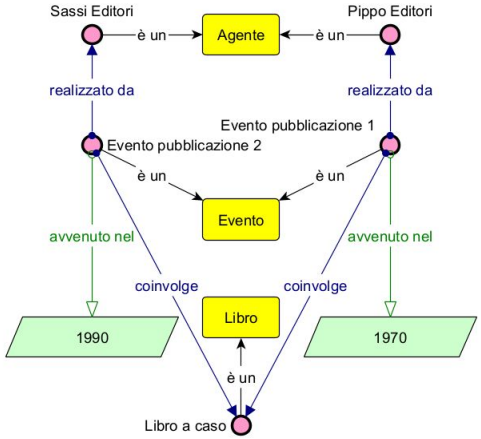
Il concetto di creazione di un Libro è reso da una relazione “scritto da” tra Libro e Agente

Il concetto di creazione di un Libro è una classe “Creazione” che rappresenta l’evento in cui un’istanza di libro è stata creata da un’istanza di Agente

Proprietà o classe: eventi



Impossibile capire chi ha pubblicato quando

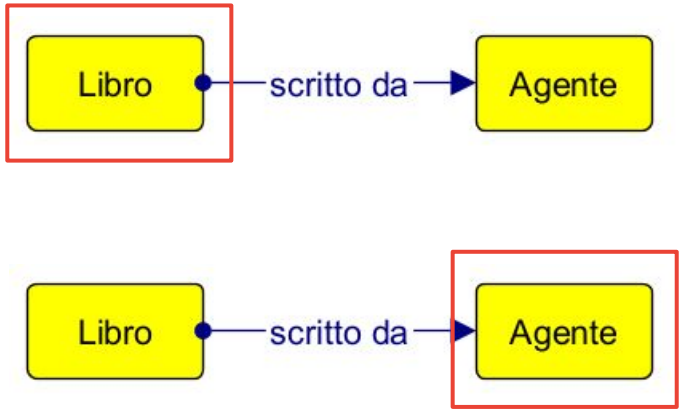


Il concetto di “Pubblicazione” (inteso come attività o evento) permette di disambiguare (rendendo il modello più complesso e più corretto)

Dominio e codominio

Dominio: classe del primo membro della proprietà

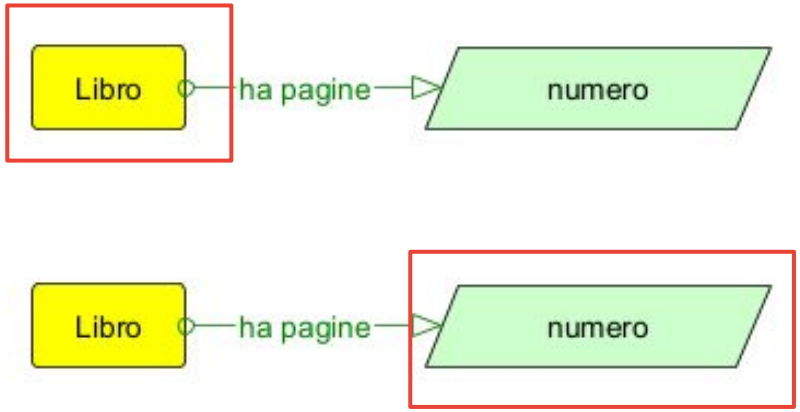
Codominio: classe del secondo membro della proprietà



An Introduction to Conceptual Modelling. In dariahTeach. https://teach.dariah.eu/course/view.php?id=31

Dominio e codominio

Vale anche per gli attributi!



Vincoli sulle proprietà

Opzionale vs **necessaria**: deve avere *almeno* un valore per ogni istanza del suo dominio (= *un Libro deve essere scritto da almeno un Agente*)

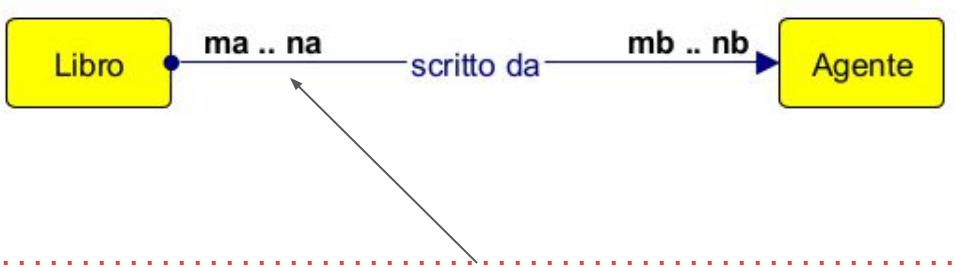
Monovalente vs **polivalente**: deve avere *al massimo* un valore per ogni istanza del suo dominio (= *un Libro può avere al massimo un numero di pagine*)

Dominio e/o codominio: ogni istanza del (co)dominio deve appartenere ad una/delle specifica/he classe/i (= *“ha pagina” può solo avere come dominio Libro e codominio Pagina*)

Vincoli e cardinalità

m_a: numero *minimo* di istanze del dominio (es. Libro) con cui un'istanza del codominio (es. Agente) *deve* avere una relazione

n_a: numero *massimo* di istanze del dominio (es. Libro) con cui un'istanza del codominio (es. Agente) *può* avere una relazione

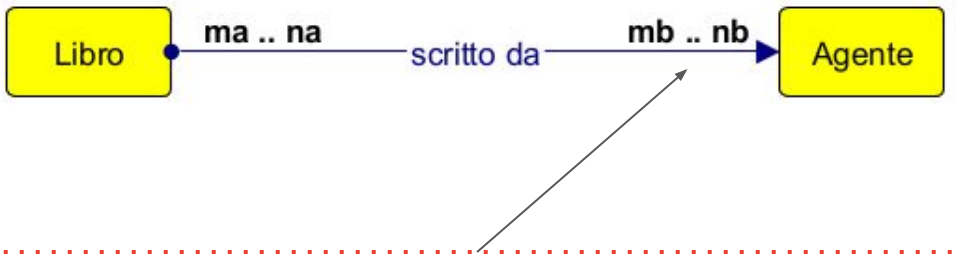


Un Libro deve essere scritto da almeno **m_a** e può essere scritto al massimo da **n_a** Agenti

Vincoli e cardinalità

m_b: numero *minimo* di istanze del codominio (Agente) con cui un'istanza del dominio (Libro) *deve* avere una relazione

n_b: numero *massimo* di istanze del codominio (Agente) con cui un'istanza del dominio (Libro) *può* avere una relazione

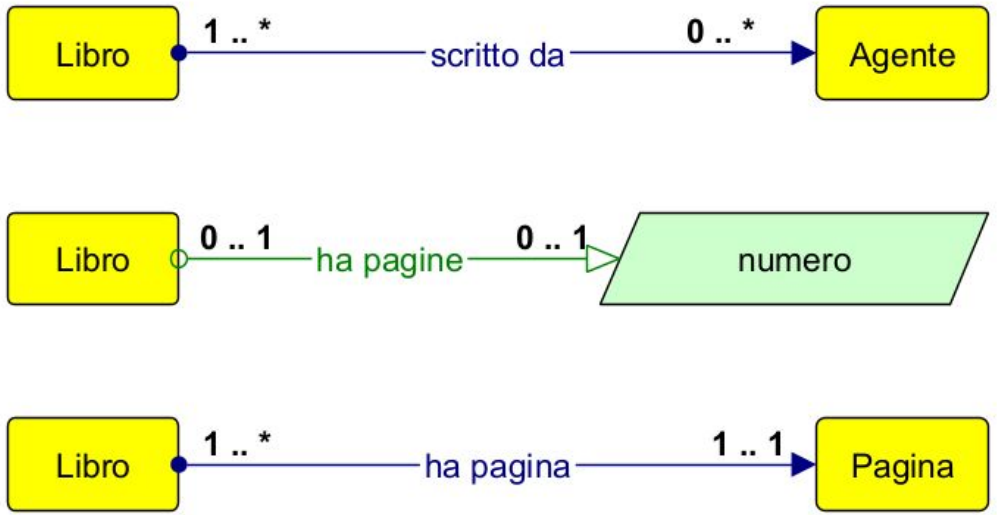


Un Agente deve aver scritto almeno **m_b** e può aver scritto al massimo **n_b** Libri

Vincoli e cardinalità

m_a, m_b:
- 0 → proprietà opzionale
- 1 → proprietà necessaria

n_a, n_b:
- 1 → proprietà monovalente ;
- * → proprietà polivalente



An Introduction to Conceptual Modelling. In dariahTeach. https://teach.dariah.eu/course/view.php?id=31

Cos'è un database?

Collezione organizzata di dati in modo da consentirne la gestione

Vari tipi, tra cui:

- Relazionali: i più utilizzati, basati su tabelle relazionali e adatti ai dati strutturati
- NoSQL: adatti ai dati semi- o non strutturati, basati su vari modelli (documenti, grafi, chiave-valore, ecc.)

```
dvdrental=# select title, release_year, length, replacement_cost from film
dvdrental-#   where length > 120 and replacement_cost > 29.50
dvdrental-#   order by title desc;
          title          | release_year | length | replacement_cost
-------------------------+--------------+--------+------------------
 West Lion               |         2006 |    159 |            29.99
 Virgin Daisy            |         2006 |    179 |            29.99
 Uncut Suicides          |         2006 |    172 |            29.99
 Tracy Cider             |         2006 |    142 |            29.99
 Song Hedwig             |         2006 |    165 |            29.99
 Slacker Liaisons        |         2006 |    179 |            29.99
 Sassy Packer            |         2006 |    154 |            29.99
 River Outlaw            |         2006 |    149 |            29.99
 Right Cranes            |         2006 |    153 |            29.99
 Quest Mussolini         |         2006 |    177 |            29.99
 Poseidon Forever        |         2006 |    159 |            29.99
 Loathing Legally        |         2006 |    140 |            29.99
 Lawless Vision          |         2006 |    181 |            29.99
 Jingle Sagebrush        |         2006 |    124 |            29.99
 Jericho Mulan           |         2006 |    171 |            29.99
 Japanese Run            |         2006 |    135 |            29.99
 Gilmore Boiled          |         2006 |    163 |            29.99
 Floats Garden           |         2006 |    145 |            29.99
 Fantasia Park           |         2006 |    131 |            29.99
 Extraordinary Conquerer |         2006 |    122 |            29.99
 Everyone Craft          |         2006 |    163 |            29.99
 Dirty Ace               |         2006 |    147 |            29.99
 Clyde Theory            |         2006 |    139 |            29.99
 Clockwork Paradise      |         2006 |    143 |            29.99
 Ballroom Mockingbird    |         2006 |    173 |            29.99
(25 rows)
```

Struttura di un DB relazionale

I dati in un database sono organizzati in **tabelle,** ognuna secondo un suo **schema**

Ogni **tabella** rappresenta una **classe**

Ogni **riga** rappresenta un'**entità** di quella classe

Ogni **colonna** rappresenta una **proprietà** di quella classe

	Name	Data type	Primary Key	Foreign Key
1	id	INTEGER		
2	document_id	INTEGER		
3	text	VARCHAR		
4	type	VARCHAR		

	id	document_id	text	type
1	1	3	10.1000/182	doi
2	2	3	948577574t4i	isbn

Struttura di un DB relazionale

Chiave Primaria: proprietà unica in una tabella che identifica in modo univoco ogni entità

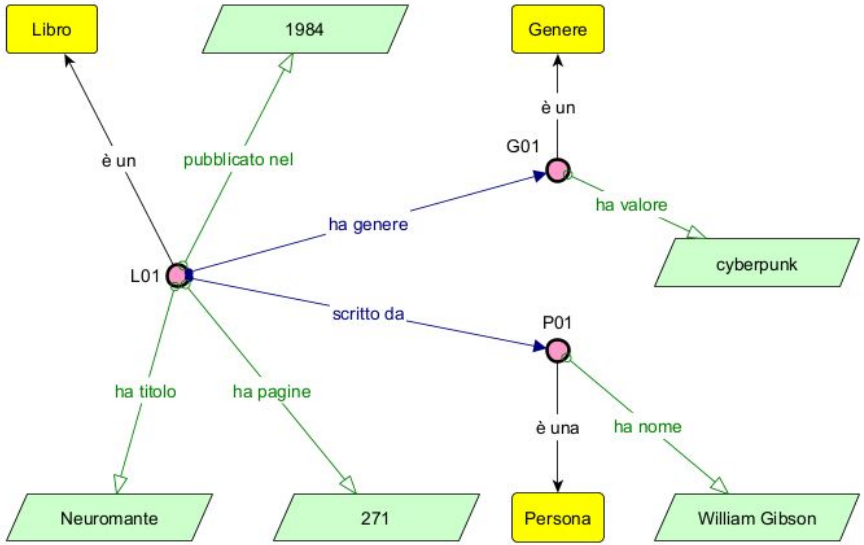
Chiave Esterna: proprietà in una tabella che si riferisce alla chiave primaria di un'altra tabella

	Name	Data type	Primary Key	Foreign Key
1	id	INTEGER		
2	document_id	INTEGER		
3	text	VARCHAR		
4	type	VARCHAR		

	id	document_id	text	type
1	1	3	10.1000/182	doi
2	2	3	948577574t4i	isbn

Usiamo il nostro modello per un DB

- 1 tabella per i libri, con colonne: “id” (chiave primaria), “ha titolo”, “scritto da” (chiave esterna), “pubblicato nel”, “ha pagine”, “ha genere” (chiave esterna)
- 1 tabella per i generi, con colonne: “id” (chiave primaria), “ha valore”
- 1 tabella per le persone, con colonne: “id” (chiave primaria), “ha nome”



Usiamo il nostro modello per un DB

id	ha titolo	scritto da	pubblicato nel	ha pagine	ha genere
L01	Neuromante	P01	1984	271	G01
L02	Il Signore degli Anelli	P02	1955	1178	G02
L03	…	…	…	…	…

1 tabella per i libri, con colonne: “id” (chiave primaria), “ha titolo”, “scritto da” (chiave esterna), “pubblicato nel”, “ha pagine”, “ha genere” (chiave esterna)

Usiamo il nostro modello per un DB

1 tabella per i generi, con colonne: “id” (chiave primaria), “ha valore”

id	**ha valore**
G01	cyberpunk
G02	high fantasy
G03	…

1 tabella per le persone, con colonne: “id” (chiave primaria), “ha nome”

id	**ha nome**
P01	William Gibson
P02	J.R.R. Tolkien
P03	…

Usiamo il nostro modello per un DB

id	ha titolo	scritto da	pubblicato nel	ha pagine	ha genere
L01	Neuromante	P01	1984	271	G01

id	ha nome
P01	William Gibson

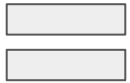
id	ha valore
G01	cyberpunk

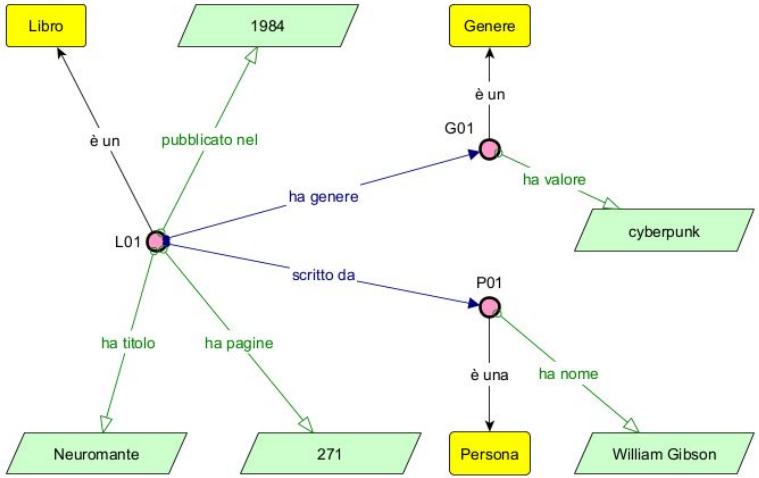
Usiamo il nostro modello per un DB

id	ha titolo	scritto da	pubblicato nel	ha pagine	ha genere
L01	Neuromante	P01	1984	271	G01

id	ha nome
P01	William Gibson

id	ha valore
G01	cyberpunk





Quiz 4

https://forms.gle/qxAiPtjUiQsdu1zS6

5.6 Semantic Web

Premesse
Definizione
Linked Open Data
Resource Description Framework
Modelli semantici di dati

I contenuti sul Web sono (solo) per noi

Contenuti leggibili e *comprensibili* dagli esseri umani

Ma le macchine?

HTML → *come* rappresentare (non *cosa*)

Alcuni tag sono semantici (es. <title>) ma il loro contenuto non è strutturato né standardizzato

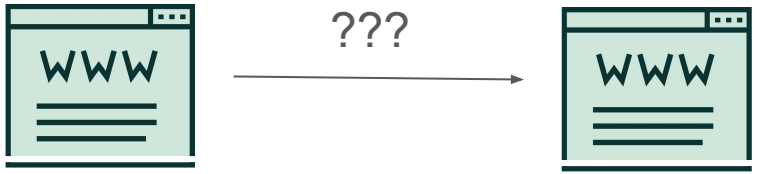
???

Verborgh,Ruben. The Semantic Web & Linked Open Data. In Web Fundamentals. https://rubenverborgh.github.io/WebFundamentals/semantic-web/

Link come pura funzione

Nessuna informazione su:

- cosa rappresenta il collegamento?
- che tipo di nesso esiste tra la risorsa A e la risorsa B?



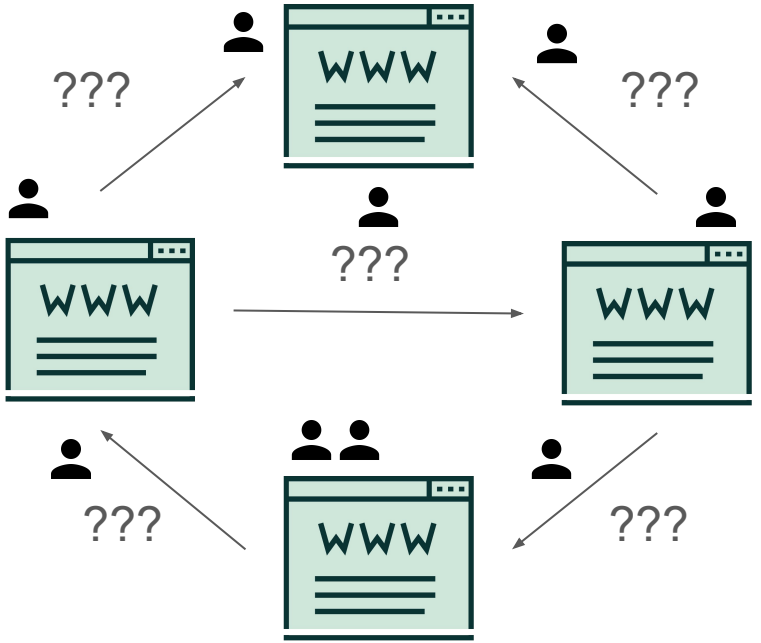
Verborgh,Ruben. The Semantic Web & Linked Open Data. In Web Fundamentals. https://rubenverborgh.github.io/WebFundamentals/semantic-web/

Il Web è universale

Qualunque pagina può collegarsi ad altre

Chiunque può pubblicare su qualsiasi argomento

- decentralizzazione
- inconsistenza dei dati
- incompletezza dei dati

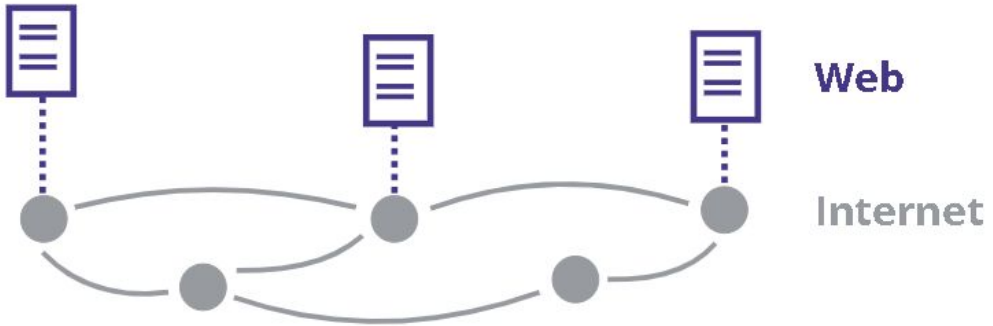


Verborgh,Ruben. The Semantic Web & Linked Open Data. In Web Fundamentals. https://rubenverborgh.github.io/WebFundamentals/semantic-web/

Cos'è il World Wide Web?

Un sistema documentale ipertestuale distribuito su Internet

- HTTP
- HTML
- URL
- …



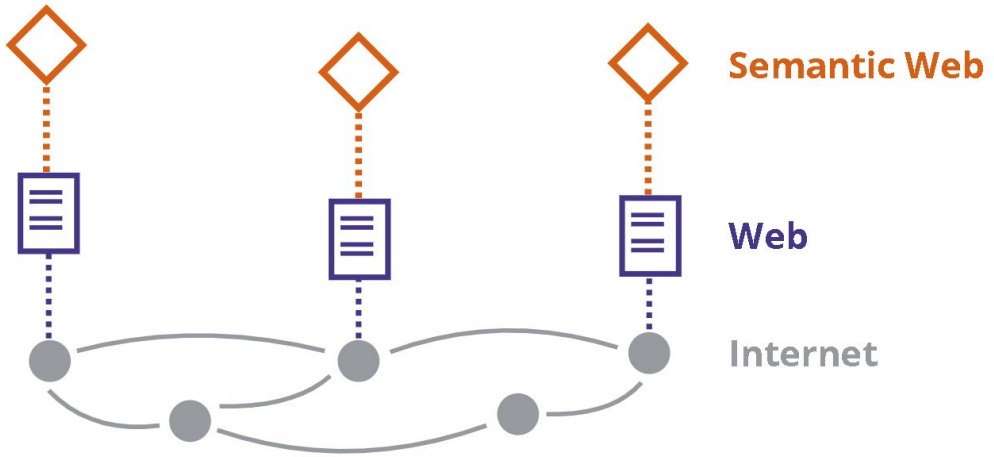
Fonte: https://rubenverborgh.github.io/WebFundamentals/

How the web works. In MDN Web Docs. https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/How_the_Web_works

Il Semantic Web è uno strato ulteriore

Proposto da Tim Berners Lee nel 2001

Ragionare sui dati disponibili sul Web in maniera automatica, estendendo il Web con informazioni semantiche

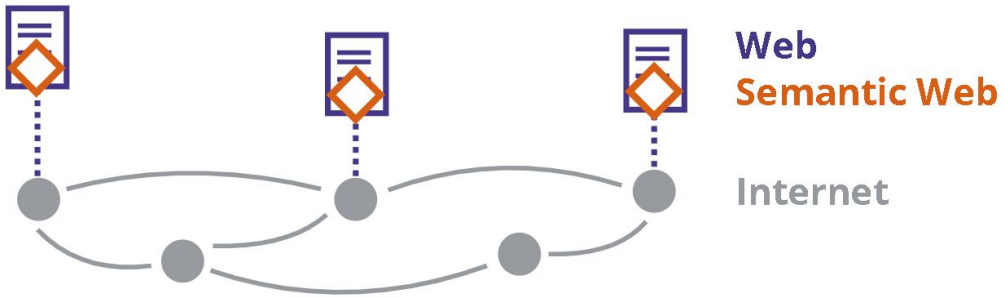


Verborgh,Ruben. The Semantic Web & Linked Open Data. In Web Fundamentals. https://rubenverborgh.github.io/WebFundamentals/semantic-web/

Il Semantic Web è integrato nel Web

Proposto da Tim Berners Lee nel 2001

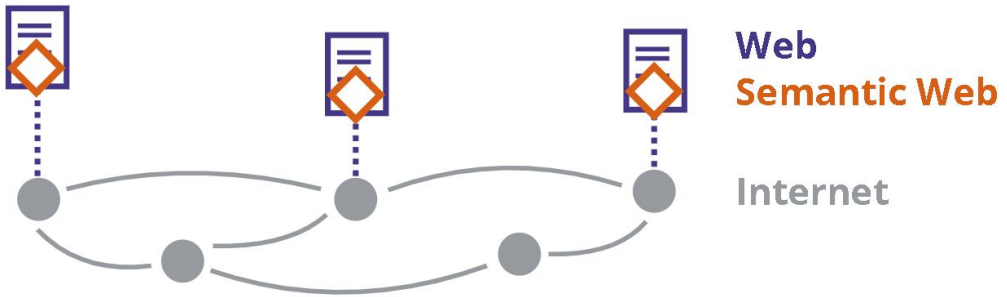
Ragionare sui dati disponibili sul Web in maniera automatica, estendendo il Web con informazioni semantiche



Verborgh,Ruben. The Semantic Web & Linked Open Data. In Web Fundamentals. https://rubenverborgh.github.io/WebFundamentals/semantic-web/

Il Semantic Web è integrato nel Web

- aggiungere informazioni
- aggiungere struttura
- permettere collegamenti semantici tra silos informativi
- permettere inferenze logiche (automatiche) sui dati



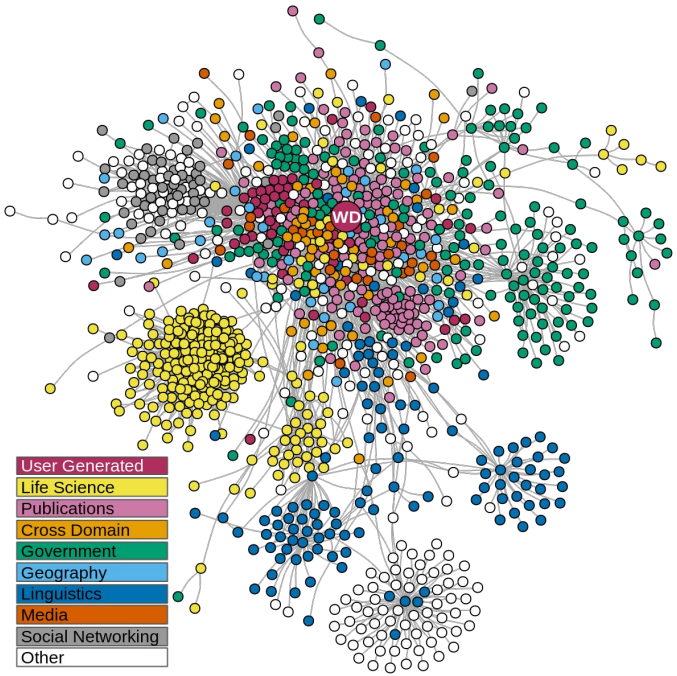
Verborgh,Ruben. The Semantic Web & Linked Open Data. In Web Fundamentals. https://rubenverborgh.github.io/WebFundamentals/semantic-web/

La base sono i Linked (Open) Data

Dati (semi-)strutturati in grafi, interpretabili dalle macchine (pubblicati in formato aperto)

1. Molteplici dataset con **licenza aperta**
2. Stessi formati **standard** di **riferimento** ai e **modellazione** dei dati

→ interrogazioni incrociate su dataset interoperabili



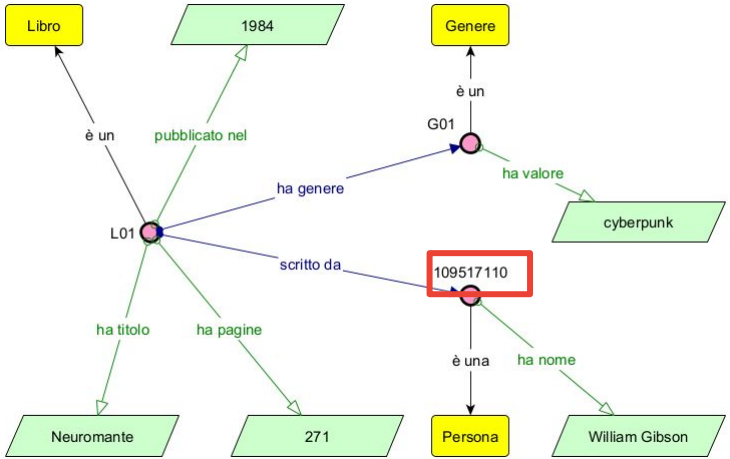
Jonathan Blaney, "Introduction to the Principles of Linked Open Data," *Programming Historian* 6 (2017), https://doi.org/10.46430/phen0068.

La base sono i Linked (Open) Data: esempio

La persona ~~William Gibson == G01~~

La persona William Gibson == *109517110*

Vantaggi: se tutti utilizzano lo stesso modo per riferirsi alla stessa cosa, possiamo con certezza disambiguare e ricercare informazioni su quella cosa *in qualsiasi dataset*



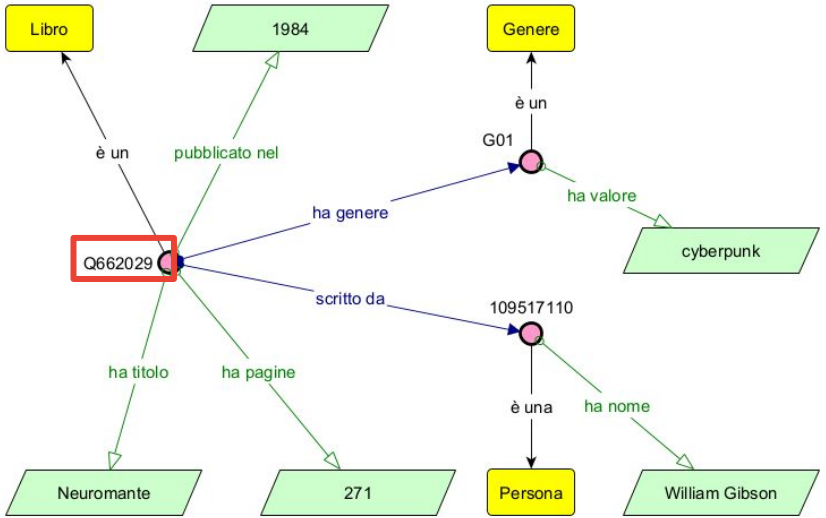
Jonathan Blaney, "Introduction to the Principles of Linked Open Data," *Programming Historian* 6 (2017), https://doi.org/10.46430/phen0068.

La base sono i Linked (Open) Data: esempio

Il romanzo [illegible] == L01

Il romanzo Neuromante == Q662029

In pratica: quello che fa un catalogo bibliografico (o un authority file)



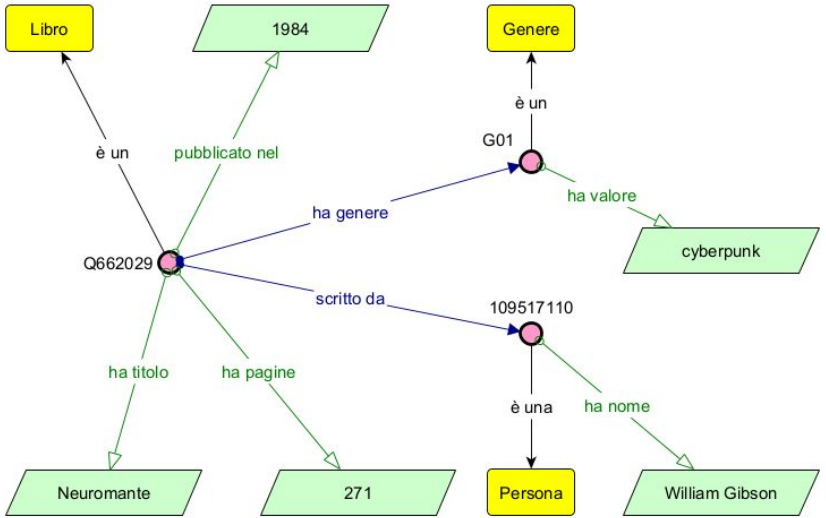
Jonathan Blaney, "Introduction to the Principles of Linked Open Data," *Programming Historian* 6 (2017), https://doi.org/10.46430/phen0068.

Gli identificatori possono esistere già...

In entrambi gli esempi, abbiamo utilizzato degli authority file (o risorse simili) per assegnare identificativi unici a William Gibson e a Neuromante

109517110 (William Gibson) è preso da VIAF

Q662029 (Neuromante) è preso da WikiData



Jonathan Blaney, "Introduction to the Principles of Linked Open Data," *Programming Historian* 6 (2017), https://doi.org/10.46430/phen0068.

… ma non sempre!

5-star rating system:

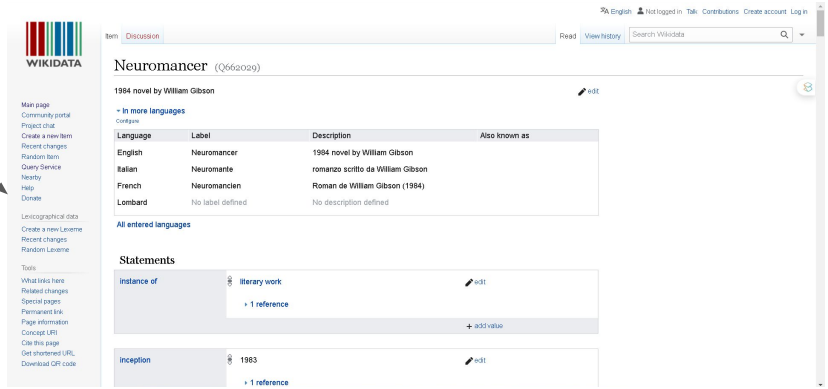
1. disponibile sul Web con una licenza aperta (es. Creative Commons)
2. 1 + formato strutturato e leggibile dalle macchine (es. Excel)
3. 2 + formato non proprietario (es. CSV)
4. 3 + usa standard aperti per identificare (es. RDF)
5. 4 + link a dati esterni per fornire ulteriore contesto

Jonathan Blaney, "Introduction to the Principles of Linked Open Data," *Programming Historian* 6 (2017), https://doi.org/10.46430/phen0068.

Quattro principi di pubblicazione LOD

1. usare **URI** come nomi per le cose
2. usare **HTTP** per permettere ai computer di cercare questi nomi
3. fornire **informazioni utili** al momento della ricerca
4. includere **link** ad altre cose

https://www.wikidata.org/entity/Q662029

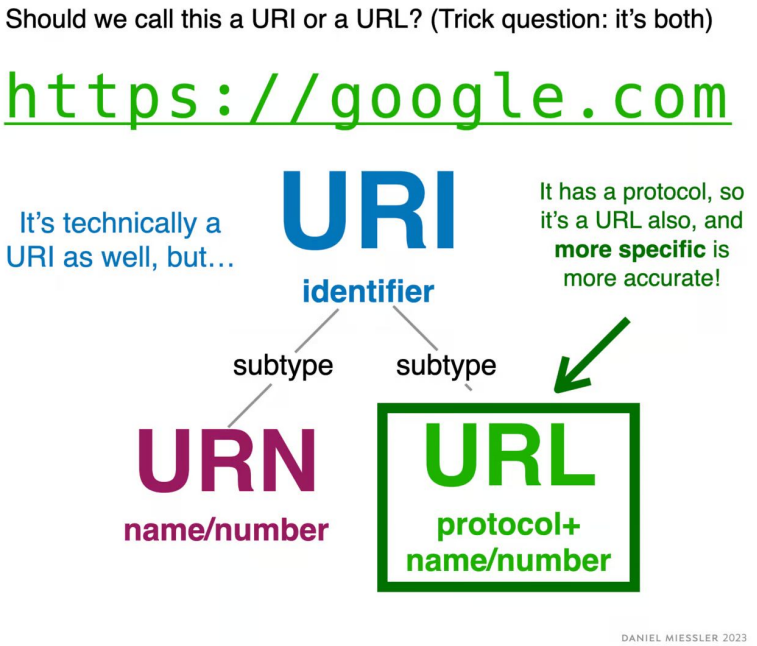


Jonathan Blaney, "Introduction to the Principles of Linked Open Data," *Programming Historian* 6 (2017), https://doi.org/10.46430/phen0068.

Usare URI per dare nomi alle cose

Le macchine hanno bisogno di identificativi unici per identificare i dati

Uniform Resource Identifier (**URI**): superclasse degli URL, si limita alla sola identificazione (localizzazione possibile ma non necessaria)

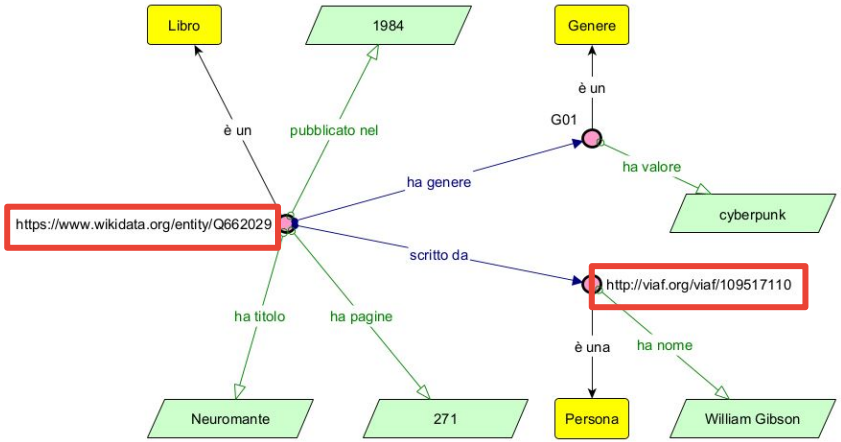


Jonathan Blaney, "Introduction to the Principles of Linked Open Data," *Programming Historian* 6 (2017), https://doi.org/10.46430/phen0068.

Usare URI per dare nomi alle cose: esempio

109517110 (William Gibson) e Q662029 (Neuromante) sono necessari ma non sufficienti

- http://viaf.org/viaf/109517110
- https://www.wikidata.org/entity/Q662029



Gli URI disambiguano su tutto il Web (perché i loro domini sono unici, perché registrati in maniera univoca nel DNS!)

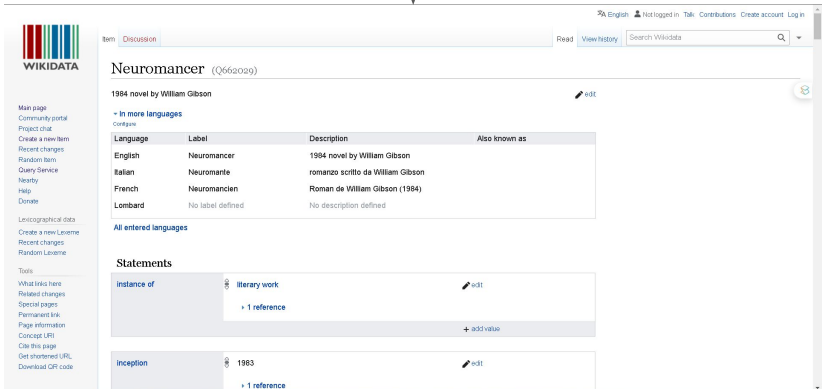
Jonathan Blaney, "Introduction to the Principles of Linked Open Data," *Programming Historian* 6 (2017), https://doi.org/10.46430/phen0068.

Usare HTTP per permettere la ricerca

Rendere l'URI **dereferenziabile** (cioé in grado di fornire una rappresentazione della risorsa che esso identifica)

In pratica: diventa un URL

https://www.wikidata.org/entity/Q662029

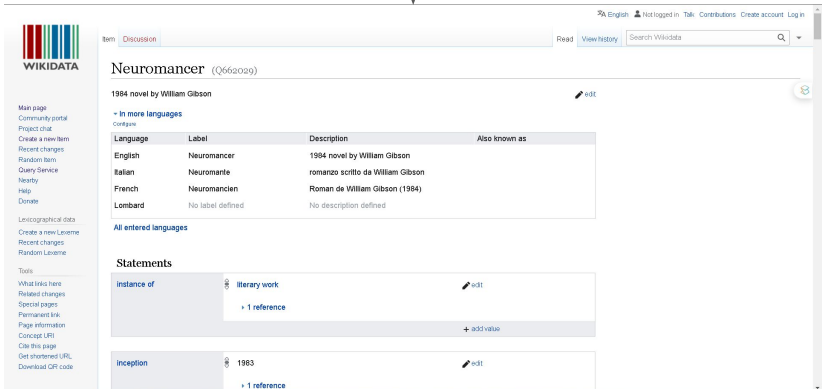


Jonathan Blaney, "Introduction to the Principles of Linked Open Data," *Programming Historian* 6 (2017), https://doi.org/10.46430/phen0068.

Fornire informazioni utili sulla risorsa

Un URI dereferenziato dovrebbe portare ad una rappresentazione (es. pagina HTML) che fornisca informazioni utili riguardanti la risorsa identificata dall'URI

https://www.wikidata.org/entity/Q662029



Jonathan Blaney, "Introduction to the Principles of Linked Open Data," *Programming Historian* 6 (2017), https://doi.org/10.46430/phen0068.

Includere link ad altre risorse

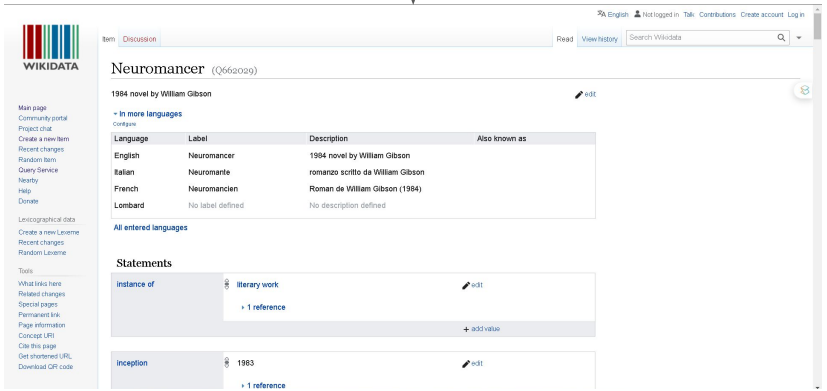
Tra le informazioni utili ci dovrebbero essere link ad altri dati

I link danno significato ai dati

I link permettono l'esplorazione del contesto e l'integrazione

I link creano un **Web di dati**

https://www.wikidata.org/entity/Q662029



Jonathan Blaney, "Introduction to the Principles of Linked Open Data," *Programming Historian* 6 (2017), https://doi.org/10.46430/phen0068.

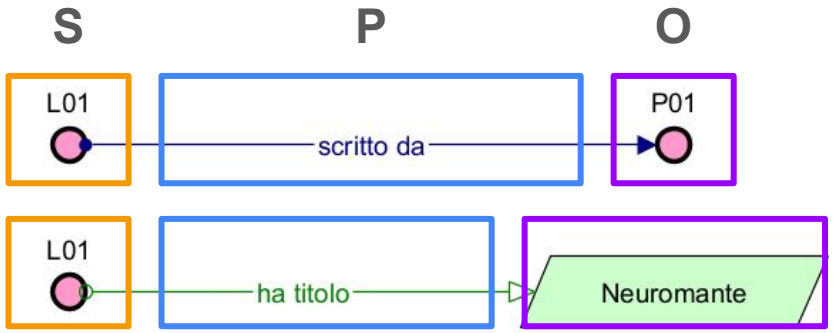
L’unità minima nei LOD: la tripla RDF

Costrutto astratto minimo di modellazione dei LOD

Soggetto → il dominio del predicato

Predicato → una caratteristica del soggetto

Oggetto → il codominio del predicato



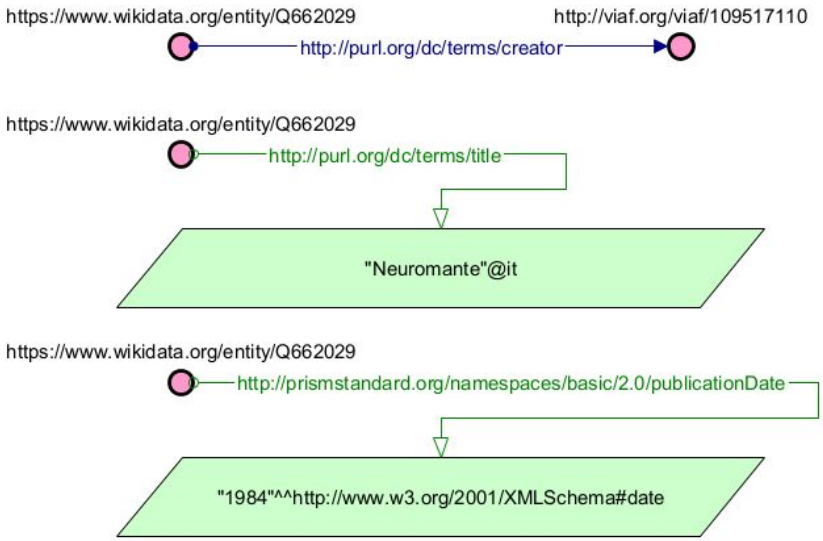
Jonathan Blaney, "Introduction to the Principles of Linked Open Data," *Programming Historian* 6 (2017), https://doi.org/10.46430/phen0068.

L’unità minima nei LOD: la tripla RDF

Soggetto e predicato sono sempre identificati da un URI

Anche l’oggetto, se il predicato è una relazione (e quindi se anche l’oggetto è un’entità)

Se il predicato è un attributo, l’oggetto è un semplice valore, con un’indicazione del tipo (es. “date”) o della lingua (es. “it”)

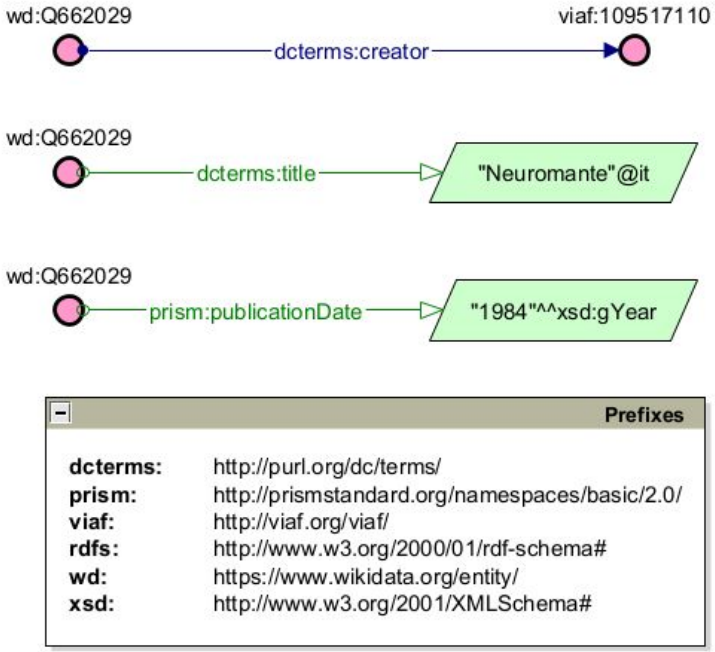


Jonathan Blaney, "Introduction to the Principles of Linked Open Data," *Programming Historian* 6 (2017), https://doi.org/10.46430/phen0068.

L’unità minima nei LOD: la tripla RDF

Per semplificare la loro visualizzazione, utilizziamo dei prefissi per abbreviare gli URI

Sono convenzioni (in teoria inventate, di fatto quelle più conosciute sono condivise da tutti)



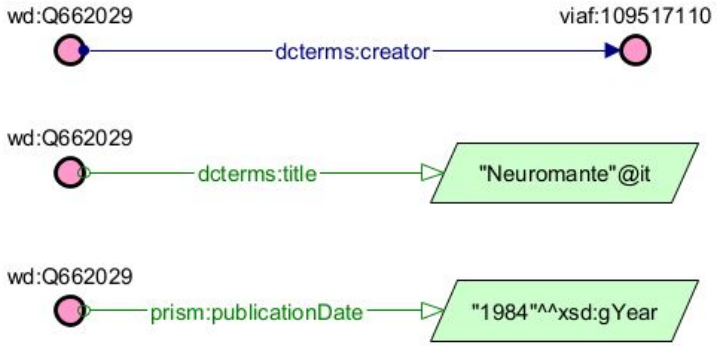
Jonathan Blaney, "Introduction to the Principles of Linked Open Data," *Programming Historian* 6 (2017), https://doi.org/10.46430/phen0068.

Resource Description Framework

Modello di dati standard che descrive i dati tramite triple SPO

Dice cosa fare, ma non:

- come scrivere triple → per questo ci sono le serializzazioni (sintassi concrete di RDF)
- come esprimere entità e proprietà → per questo ci sono vocabolari, ontologie, ecc.



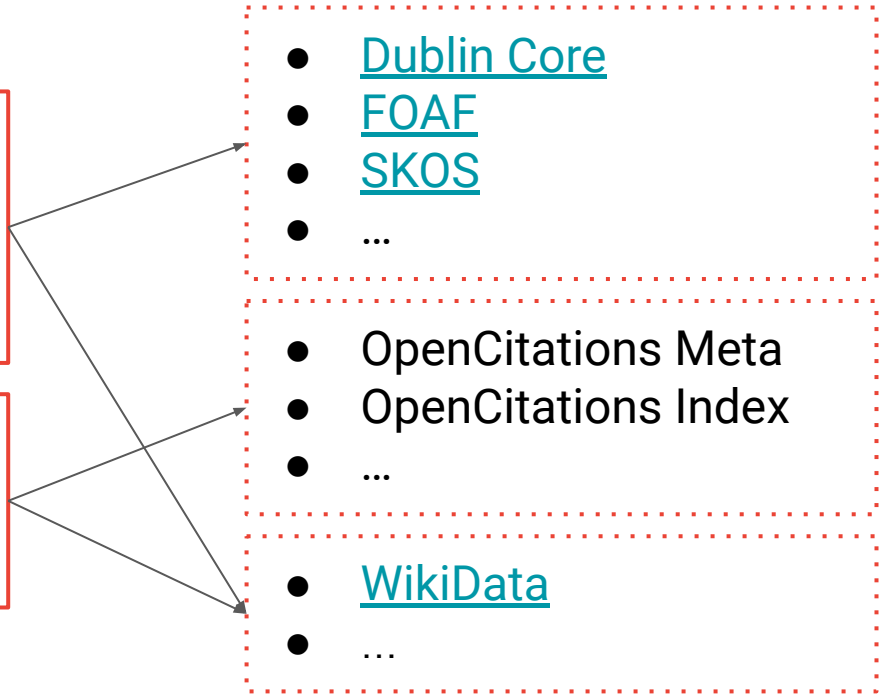

```
wd:Q662029 dcterms:creator viaf:109517110 ;
      dcterms:title “Neuromante”@it ;
      prism:publicationDate “1984”^^xsd:gYear .
```

Jonathan Blaney, "Introduction to the Principles of Linked Open Data," *Programming Historian* 6 (2017), https://doi.org/10.46430/phen0068.

Il Semantic Web ha tanti dati pronti al riuso

Livello strutturale: vocabolari (**modelli semantici** di dati) che forniscono classi e proprietà da riusare

Livello contenutistico: **dataset** che forniscono identificativi e dati delle entità



Verborgh,Ruben. The Semantic Web & Linked Open Data. In Web Fundamentals. https://rubenverborgh.github.io/WebFundamentals/semantic-web/

Perché non usare i DB relazionali, allora?

Assunzione del mondo aperto: *in un sistema logico, l'assenza di un fatto non lo rende falso*

I database relazionali usano strutture rigide, poiché aspirano a dati completi (quindi mondo chiuso)

In LOD, nessuna fonte ha tutti i dati

Tassonomie, tesauri, ontologie strutturano i LOD con un'apertura alla continua integrazione ed estensione

Dati, modelli pensati per essere

- espressivi
- **riutilizzati**
- **potenzialmente estesi**

Verborgh,Ruben. The Semantic Web & Linked Open Data. In Web Fundamentals. https://rubenverborgh.github.io/WebFundamentals/semantic-web/

Inferenza (o reasoning)

Abilità di un agente di verificare e scoprire fatti, combinarli a partire da diverse fonti e trarre conclusioni

I LOD forniscono un corpo di conoscenza su cui gli agenti possono fare reasoning

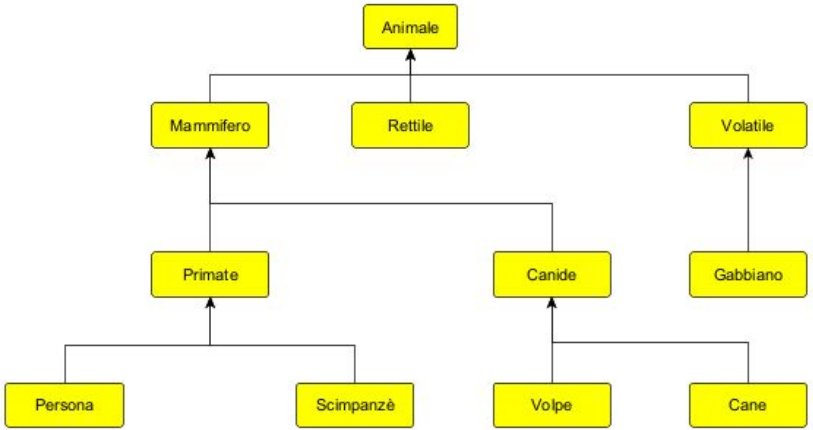
```
Sebastian - conosce - Cristian
.

conosce - ha dominio - Persona
.

Sebastian - è una - Persona .
```

Verborgh,Ruben. The Semantic Web & Linked Open Data. In Web Fundamentals. https://rubenverborgh.github.io/WebFundamentals/semantic-web/

Tassonomia

Insieme di termini organizzati in una gerarchia

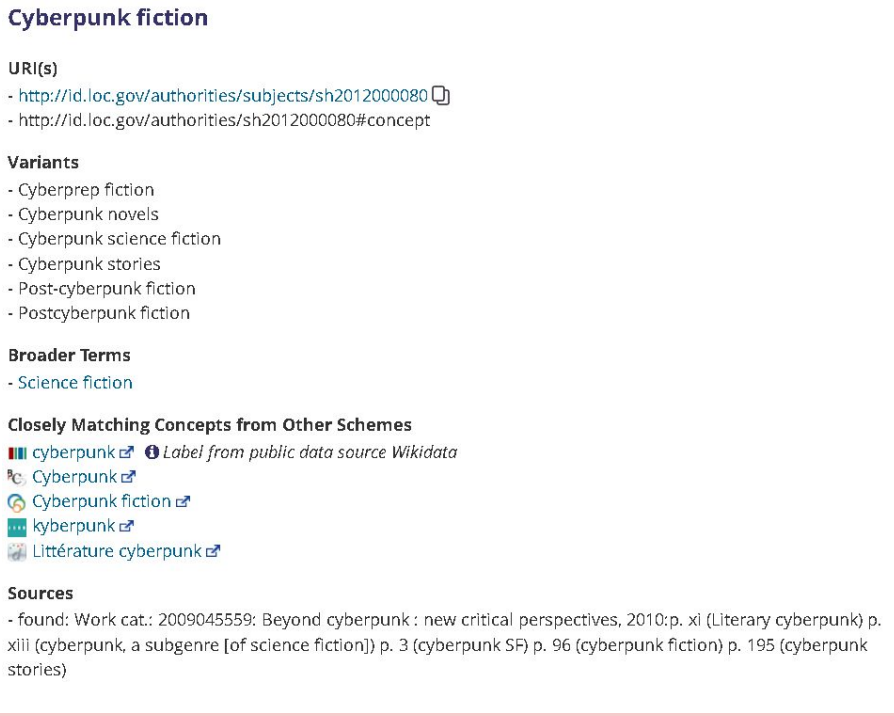
Esistono solo relazioni di sottoclasse / superclasse



Hjørland, Birger. 2017. “Classification”. *Knowledge Organization* 44, no. 2: 97-128. Also available in *ISKO Encyclopedia of Knowledge Organization*, eds. Birger Hjørland and Claudio Gnoli, https://www.isko.org/cyclo/classification

Tesauro

Vocabolario controllato nel quale sono presenti anche relazioni di varia natura, es. gerarchiche, associative (sinonimia, iperonimia, iponimia, olonimia, meronimia), ecc.



Cyberpunk fiction

URI(s)
- http://id.loc.gov/authorities/subjects/sh2012000080
- http://id.loc.gov/authorities/sh2012000080#concept

Variants
- Cyberprep fiction
- Cyberpunk novels
- Cyberpunk science fiction
- Cyberpunk stories
- Post-cyberpunk fiction
- Postcyberpunk fiction

Broader Terms
- Science fiction

Closely Matching Concepts from Other Schemes
- cyberpunk *Label from public data source Wikidata*
- Cyberpunk
- Cyberpunk fiction
- kyberpunk
- Littérature cyberpunk

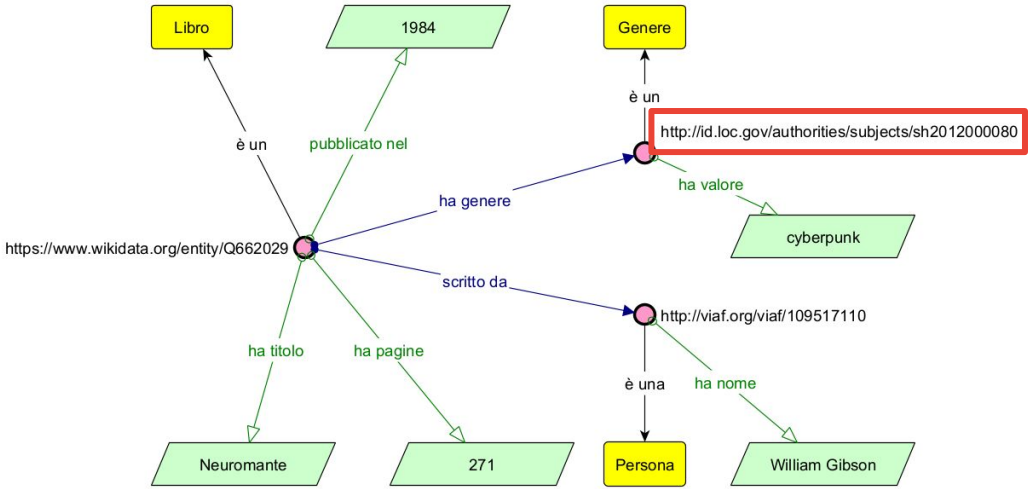
Sources
- found: Work cat.: 2009045559: Beyond cyberpunk : new critical perspectives, 2010:p. xi (Literary cyberpunk) p. xiii (cyberpunk, a subgenre [of science fiction]) p. 3 (cyberpunk SF) p. 96 (cyberpunk fiction) p. 195 (cyberpunk stories)

Hjørland, Birger. 2017. “Classification”. *Knowledge Organization* 44, no. 2: 97-128. Also available in *ISKO Encyclopedia of Knowledge Organization*, eds. Birger Hjørland and Claudio Gnoli, https://www.isko.org/cyclo/classification

Tesauro: esempio

Usiamo LC Subject Headings (LCSH) per controllare “cyberpunk”

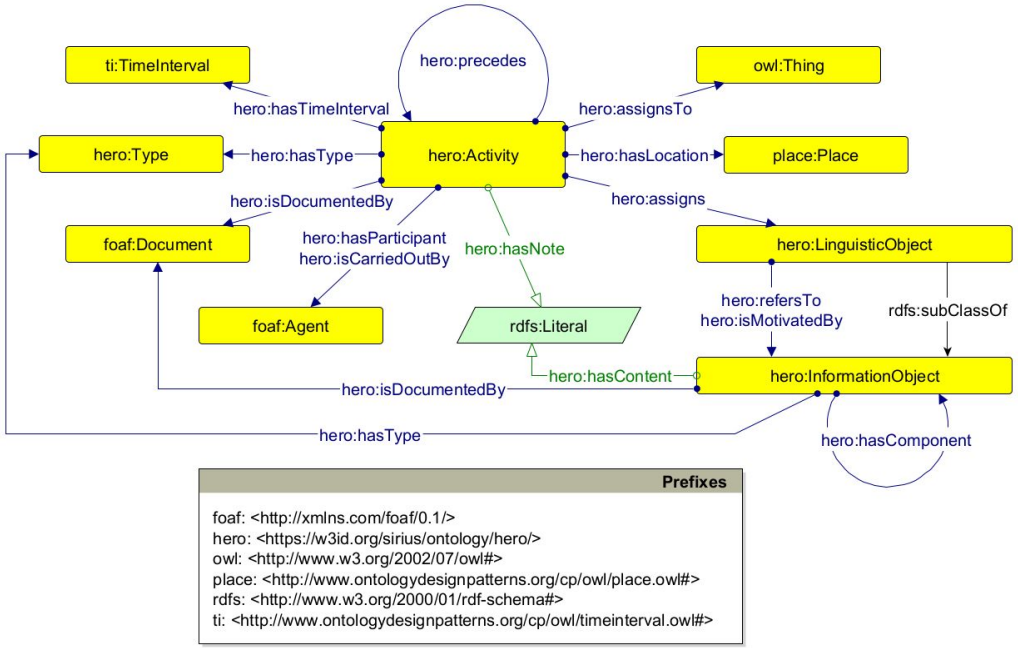
http://id.loc.gov/authorities/subjects/sh2012000080



Ontologia

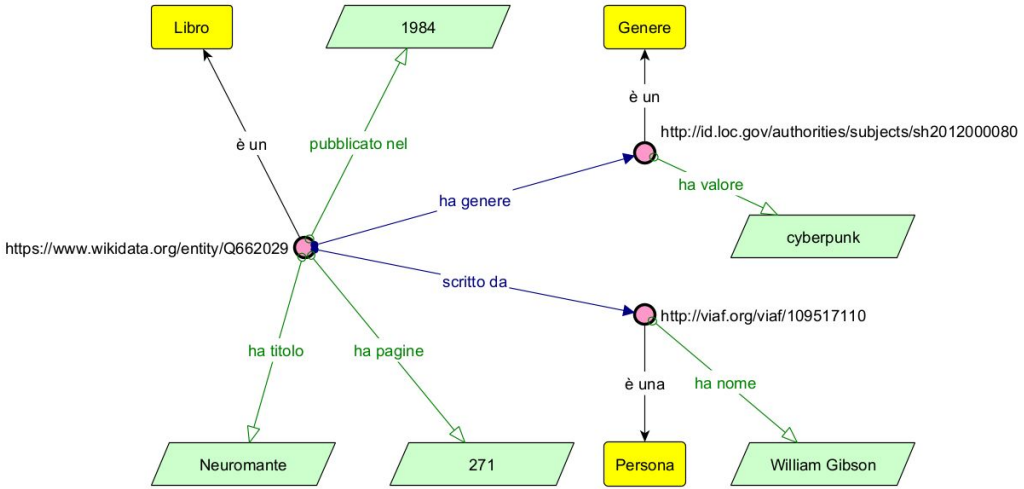
Modello di dati che descrive una particolare area di conoscenza definendo una terminologia comune per:

- entità
- proprietà (relazioni e attributi)
- vincoli logici e regole di inferenza



Riutilizziamo delle ontologie

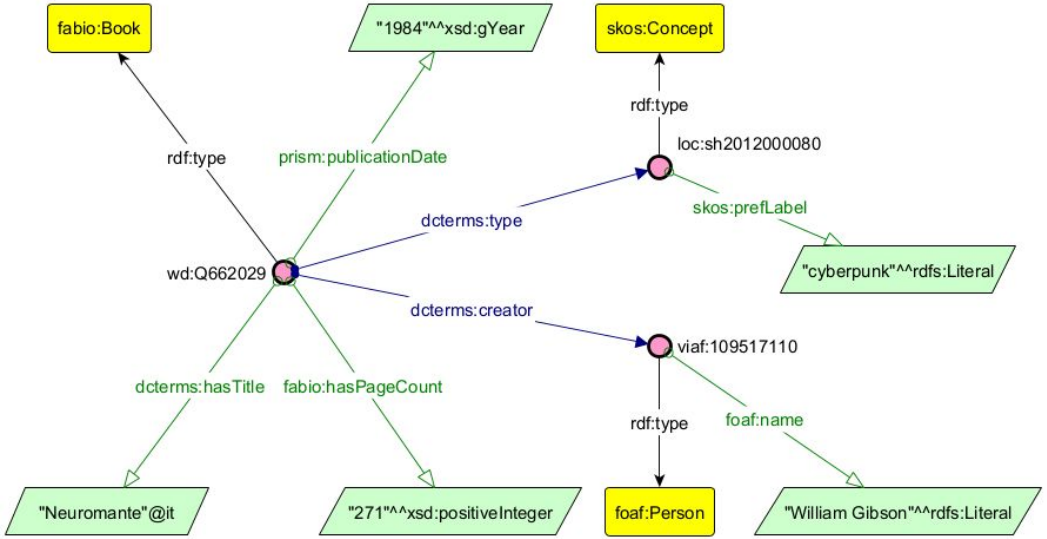
- Libro
- Genere
- Persona
- è un/una
- pubblicato nel
- ha genere
- scritto da
- ha pagine



Riutilizziamo delle ontologie

- Libro → http://purl.org/spar/fabio/Book
- Genere → http://www.w3.org/2004/02/skos/core#Concept
- Persona → http://xmlns.com/foaf/0.1/Person
- è un/una → http://www.w3.org/1999/02/22-rdf-syntax-ns#type
- pubblicato nel → http://prismstandard.org/namespaces/basic/2.0/publicationDate
- ha genere → http://purl.org/dc/terms/type
- scritto da → http://purl.org/dc/terms/creator
- ha pagine → http://purl.org/spar/fabio/hasPageCount

Riutilizziamo delle ontologie



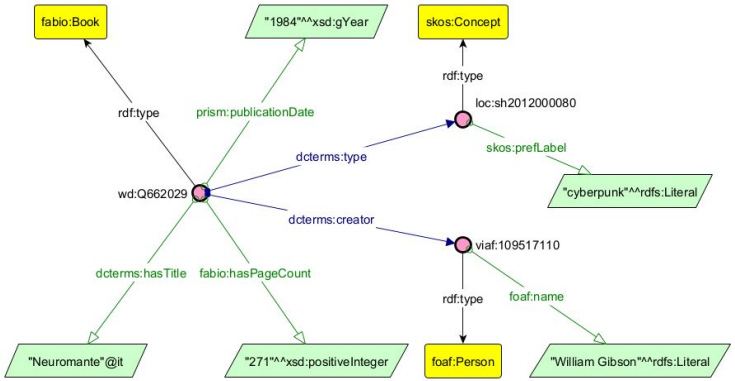
Prefixes

dcterms:	http://purl.org/dc/terms/
fabio:	http://purl.org/spar/fabio/
foaf:	http://xmlns.com/foaf/0.1/
loc:	http://id.loc.gov/authorities/subjects/
prism:	http://prismstandard.org/namespaces/basic/2.0/
rdf:	http://www.w3.org/1999/02/22-rdf-syntax-ns#
rdfs:	http://www.w3.org/2000/01/rdf-schema#
skos:	http://www.w3.org/2004/02/skos/core#
viaf:	http://viaf.org/viaf/
wd:	https://www.wikidata.org/entity/
xsd:	http://www.w3.org/2001/XMLSchema#

- dati (semi-)strutturati
- machine-readable
- semantici
- interoperabili
- riusabili

Riutilizziamo delle ontologie

```
wd:Q662029 rdf:type fabio:Book ;
      dcterms:creator viaf:109517110 ;
      dcterms:title “Neuromante”@it ;
      prism:publicationDate “1984”^^xsd:gYear ;
      fabio:hasPageCount “271”^^xsd:positiveInteger ;
      dcterms:type loc:sh2012000080 ;
      dcterms:creator viaf:109517110 .

loc:sh2012000080 a skos:Concept ;
      skos:prefLabel “cyberpunk”^^rdfs:Literal .

viaf:109517110 a foaf:Person ;
      foaf:name “William Gibson”^^rdfs:Literal .
```



- dati (semi-)strutturati
- machine-readable
- semantici
- interoperabili
- riusabili

Abilità informatiche

A.A. 2023/2024

05c - Fine

Sebastian Barzaghi
sebastian.barzaghi2@unibo.it
https://orcid.org/0000-0002-0799-1527