Abilità informatiche

A.A. 2023/2024

05b - Modellazione dei Dati

Sebastian Barzaghi

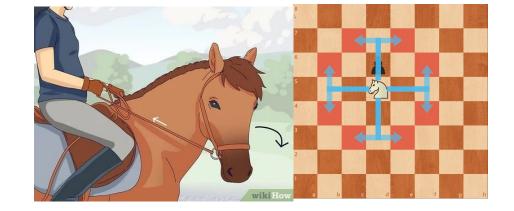
sebastian.barzaghi2@unibo.it
https://orcid.org/0000-0002-0799-1527

Riassunto della lezione precedente (05a)

Cos'è un linguaggio naturale?



Linguaggio caratterizzato da ricchezza espressiva, ambiguità e ridondanza, per cui un qualsiasi costrutto formulato è potenzialmente polisemico



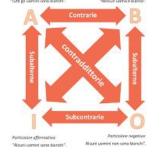
es. "sposta il cavallo"

Cos'è un linguaggio formale?



Per comunicare con un computer, l'ambiguità del linguaggio naturale è un problema (servono altre informazioni relative al contesto e alla pragmatica)

Linguaggio per formulare costrutti in modo preciso e non ambiguo



$$\Delta\left(n\right) = \left| \begin{array}{ccccc} 0 & 0 & 0 & \dots & 0 & 1 \\ 0 & 0 & 0 & \dots & 1 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 1 & 0 & \dots & \dots & 0 & 0 \\ 1 & 0 & 0 & \dots & 0 & 0 \end{array} \right|, \;\; n=2,3,4,\dots,+\infty$$

```
def quadrato(*numeri):
    c = [str(x**2) for x in numeri]
    print ' ; '.join(c)

def divisore(num, den):
    try:
        if den == 1:
             return num
        else:
             return float(num)/den
        except ZeroDivisionError:
             print 'Errore. Divisione per zero!'
```

L'informazione va codificata per il computer



Occorre rappresentare tale informazione in un formato facilmente manipolabile dal computer



Natura fisica a due stati dei componenti di un computer (progettazione e costruzione semplici dei circuiti elettronici) https://www.youtube.com/watch?v=h3 9SHE0SPM





Consideriamo un alfabeto binario



Bit (binary digit o cifra binaria)

Cifra binaria, ovvero uno dei due simboli del sistema numerico binario, classicamente chiamati zero (0) e uno (1)

- chiuso
- spento
- assente
- falso

- aperto
 - acceso
 - presente
 - vero

La binarietà è semplice



- 2 bit : 4 (2²) sequenze possibili:
 00, 01, 10, 11
- 3 bit: 8 (2³) sequenze possibili: 000, 001, 010, 100, 011, 101, 110, 111
- ...
- n bit: 2ⁿ sequenze

Byte

Una sequenza di 8 bit

Usato come unità di misura per indicare le dimensioni della memoria la velocità di trasmissione la "potenza" di un elaboratore

Il testo codificato: ASCII



L'alfabeto anglosassone ha circa 120 caratteri (maiuscole + minuscole + numeri + interpunzione ...



Bastano 7 bit = 2^7 = 128 simboli



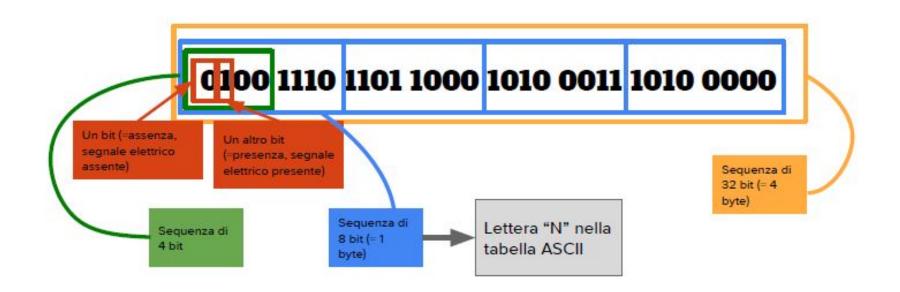
American Standard Code for Information Interchange (ASCII)

A → 1000001

B → 1000010

BABA → 1000010 1000001 1000010 1000001

Il testo codificato: due esempi



Il dilemma del porcospino



Quanto più due esseri umani si avvicinano tra loro per cercare calore e conforto, tanto più si feriranno a vicenda

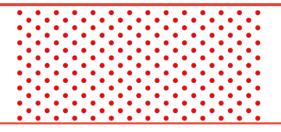


Metafora sulle complessità delle relazioni e dell'intimità umana





La mappa non è il territorio



Un'astrazione che rappresenta qualcosa non è quella cosa (ma può essere utile)

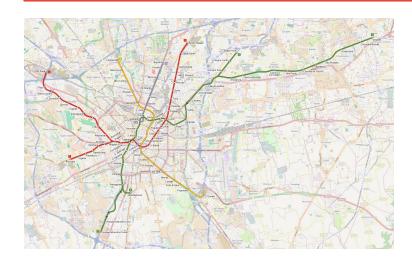


"(...) In quell'Impero, l'Arte della Cartografia giunse a una tal Perfezione che la Mappa di una sola Provincia occupava tutta una Città, e la mappa dell'impero tutta una Provincia. Col tempo, queste Mappe smisurate non bastarono più. I Collegi dei Cartografi fecero una Mappa dell'Impero che aveva l'Immensità dell'Impero e coincideva perfettamente con esso. Ma le Generazioni Seguenti, meno portate allo Studio della cartografia, pensarono che questa Mappa enorme fosse inutile e non senza Empietà la abbandonarono alle Inclemenze del Sole e degli Inverni. Nei deserti dell'Ovest rimangono lacere Rovine della Mappa, abitate da Animali e Mendichi; in tutto il Paese non c'è altra reliquia delle Discipline Geografiche. (Suárez Miranda, Viajes de varones prudentes, libro IV, cap. XIV, Lérida, 1658)".

Jorge Luis Borges, Del rigore della scienza

Astrazione

Processo di rimozione dei dettagli trascurabili di una situazione in modo da semplificarla e focalizzare l'attenzione sulle sue caratteristiche principali





Cos'è un modello?



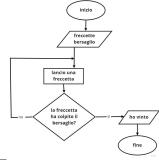
Una rappresentazione semplificata di un sistema o fenomeno



Cattura le caratteristiche fondamentali di quel sistema o fenomeno, permettendo di comprendere, analizzare o predire tale comportamento



$$\left\{egin{aligned}
abla^2\phi - rac{1}{c^2}rac{\partial^2\phi}{\partial t^2} &= -rac{
ho}{arepsilon} \
abla^2A_x - rac{1}{c^2}rac{\partial^2A_x}{\partial t^2} &= -\mu
ho v_x \
abla^2A_y - rac{1}{c^2}rac{\partial^2A_y}{\partial t^2} &= -\mu
ho v_y \
abla^2A_z - rac{1}{c^2}rac{\partial^2A_z}{\partial t^2} &= -\mu
ho v_z \end{aligned}
ight.$$







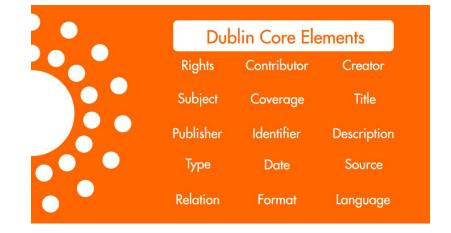
Cos'è un modello?



Un linguaggio formale per costruire rappresentazioni astratte



Es. convenzioni cartografiche, schemi di metadati, equazioni differenziali...



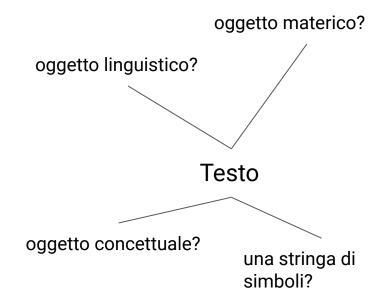


Cos'è la modellazione?



Attività epistemologica che consiste nella creazione e utilizzo di una somiglianza, una semplificazione di qualcosa, per comprendere meglio quel qualcosa

Atto interpretativo che esplicita le nostre assunzioni sulla natura della cosa che stiamo modellando



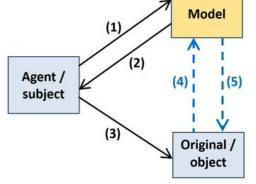
Cos'è un modello?

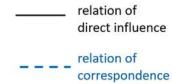


Può essere astratto o concreto, descrittivo o prescrittivo

Modellare è trovare un equilibrio tra due estremità:

- standardizzazione
- precisione / espressività





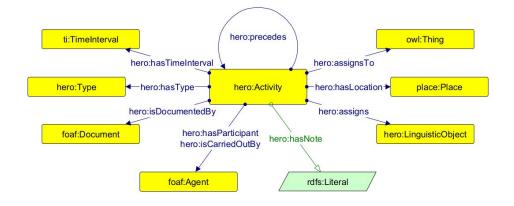
- (1) Agent creates/uses model
- (2) Model represents original for agent
- (3) Agent creates or adapts original by model
- (4) Original is similar to model
- (5) Model is similar to original



Cos'è la modellazione dei dati?

L'insieme di attività di rappresentazione formale di un segmento di realtà in modo tale da renderlo computabile

Es. creazione di database, schemi XML, ontologie, ecc.



foaf: http://xmlns.com/foaf/0.1/>

hero: <https://w3id.org/sirius/ontology/hero/>

owl: ">

place: http://www.ontologydesignpatterns.org/cp/owl/place.owl#>

Prefixes

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

ti: "> ti: "> ti: "> ti: "> ti: ti: <a href="http://www.ontologydesi

Un classification task

Definizione di classi

Classe: categoria di *entità* con proprietà e comportamenti simili **Entità**: oggetto che fa parte della realtà osservata

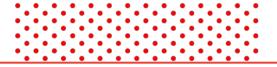
.

.

Attributo: proprietà espressa come associazione logica tra entità **Relazione**: proprietà espressa come valore attribuito ad un'entità

- Classe "Libro": ha un titolo, un autore e un numero di pagine
- Entità: "Il Signore degli Anelli" di J.R.R. Tolkien
- Attributi: "Titolo" (con valore "Il Signore degli Anelli") e "Numero di pagine" (con valore "1178").
- Relazione: "Autore" (collegato a "J.R.R Tolkien", che è un'istanza della classe "Persona")

Formuliamo un modello



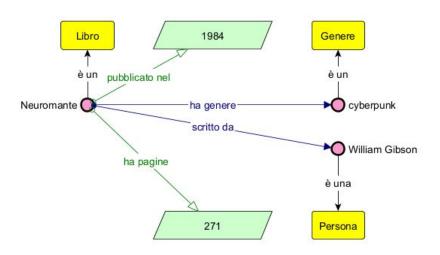
Neuromante, scritto da William Gibson e pubblicato nel 1984, è un libro di 271 pagine e di genere cyberpunk

Individuiamo <mark>classi</mark>, <mark>entità</mark>, <mark>attributi</mark> e relazioni

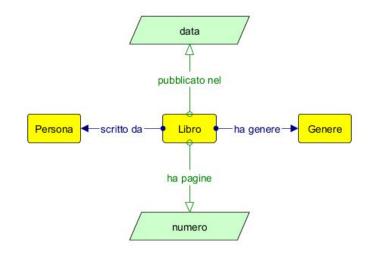
- Neuromante è libro
- (William Gibson è persona)
- (cyberpunk è genere)
- Neuromante scritto da William Gibson
- Neuromante pubblicato nel 1984
- Neuromante ha pagine 271
- Neuromante ha genere cyberpunk

Formuliamo un modello

Modello - istanza



Modello - schema



5.4 Approfondimenti di modellazione

Non unicità Ereditarietà Questioni complesse Dominio e codominio

La nave di Teseo



Esperimento mentale sul mantenimento dell'identità nello spaziotempo



Se ogni parte della nave viene sostituita gradualmente con nuove parti, è ancora la stessa nave?

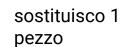






La nave di Teseo







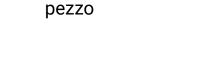




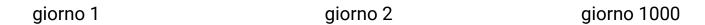
Α



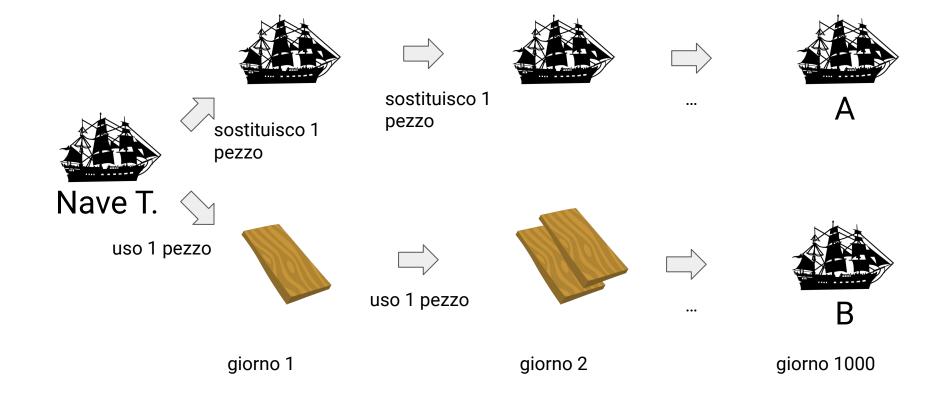
Nave T.

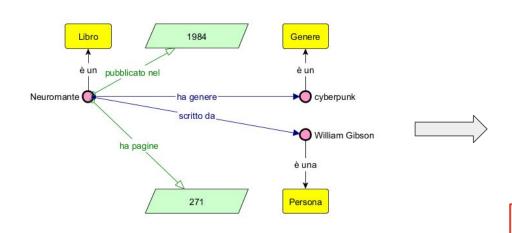


sostituisco 1

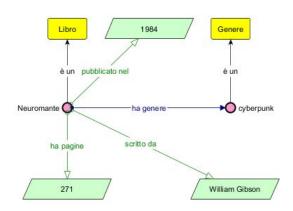


La nave di Teseo

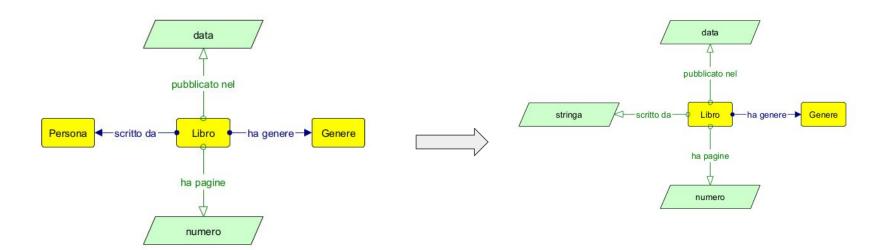




Ipotizziamo che non abbiamo interesse a modellare le persone come entità, ma ci basta sapere i nomi delle persone coinvolte nel ciclo di vita del libro

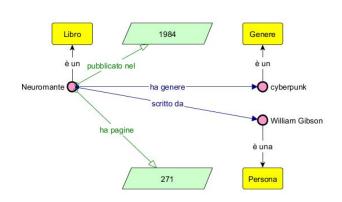


- Non abbiamo più la classe Persona
- "scritto da" diventa un attributo di Libro
- Modello più semplice, ma perdiamo espressività ed elasticità

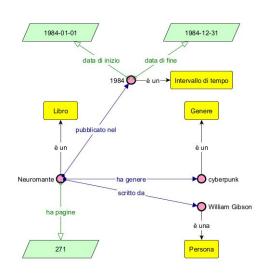


Ipotizziamo che non abbiamo interesse a modellare le persone come entità, ma ci basta sapere i nomi delle persone coinvolte nel ciclo di vita del libro

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

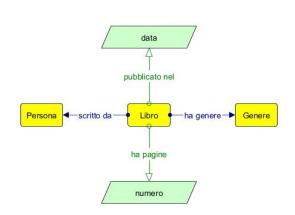


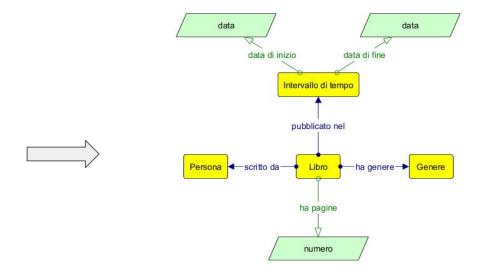




Ipotizziamo che abbiamo interesse a modellare il tempo come periodo caratterizzato da un inizio e una fine

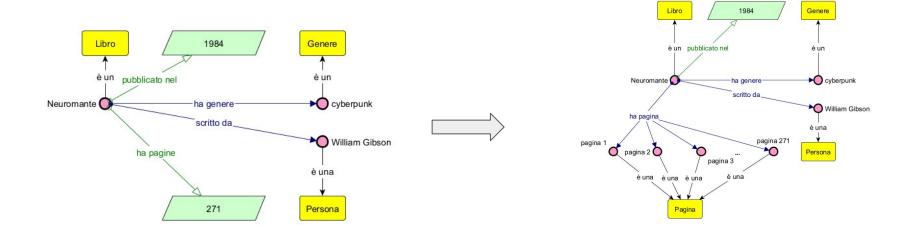
- Abbiamo una nuova entità "1984" appartenente alla nuova classe "Intervallo di tempo" con due attributi "data di inizio" e "data di fine"
- "pubblicato nel" diventa una relazione
- Modello più complesso ed espressivo





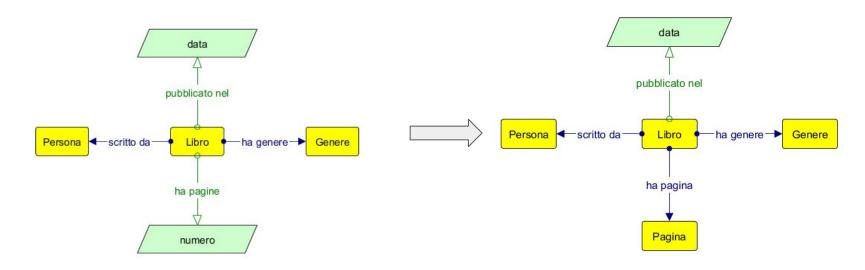
Ipotizziamo che abbiamo interesse a modellare il tempo come periodo caratterizzato da un inizio e una fine

- Abbiamo una 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



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



Ipotizziamo che abbiamo interesse a modellare le pagine singole

- Abbiamo una nuova classe "Pagina"
- "ha pagine" diventa una relazione
- Modello più complesso ed espressivo



Esiste anche una graphic novel di Neuromante

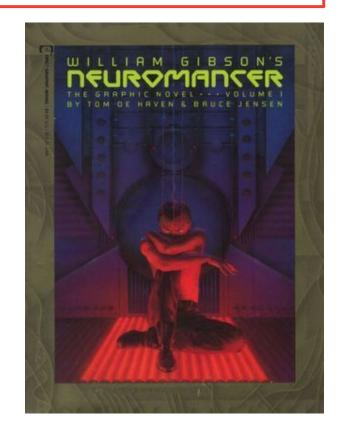


Siamo interessati a modellare anche questa entità e le sue caratteristiche



Quale classe le assegnamo?



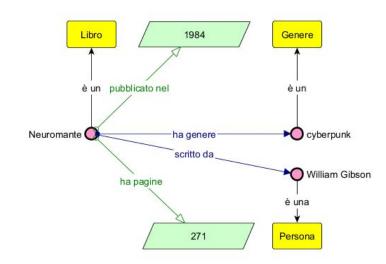




Potremmo trattarla come istanza della classe "Libro", dato che ci interessano le stesse caratteristiche



Ma ci interessa modellare anche una caratteristica particolare della graphic novel: chi l'ha illustrata (Bruce Jensen, una nuova entità appartenente a "Persona")





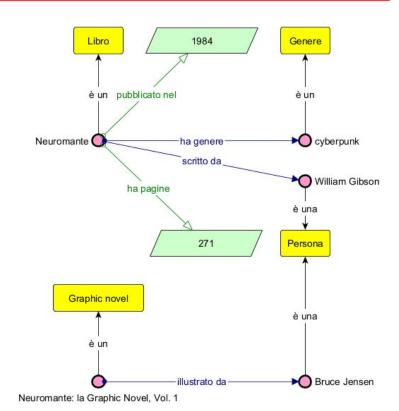
Neuromante: la Graphic Novel, Vol. 1



Creiamo una classe "Graphic novel", le assegnamo la nuova entità "Neuromante: la Graphic Novel, Vol. 1", stabiliamo una relazione "illustrato da" tra la graphic novel e la nuova entità "Bruce Jensen"



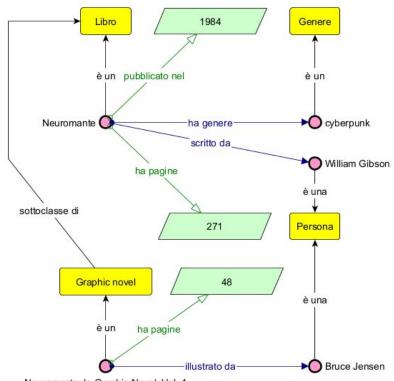
Ma ricordiamoci che ci interessano anche altre caratteristiche, tipiche di un Libro: chi l'ha scritta? quante pagine ha? di quale genere fa parte?





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"

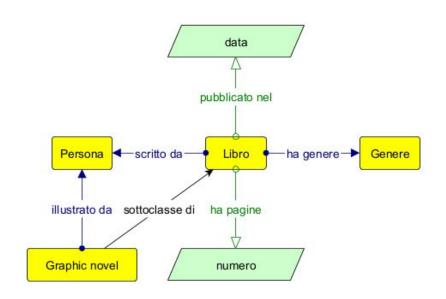


Neuromante: la Graphic Novel, Vol. 1



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"

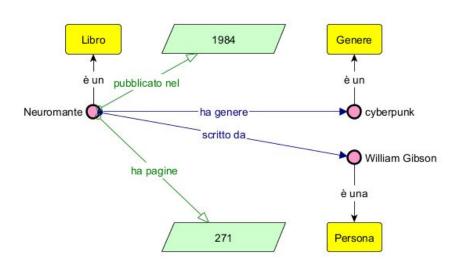


Altro esempio: Persona

Generalmente ciò che stiamo rappresentando va bene, ma esistono eccezioni (es. <u>Luther Blissett / Wu Minq</u>)

Come facciamo?





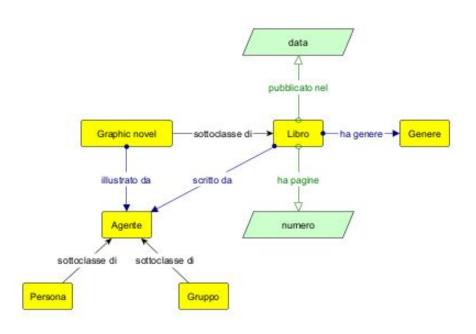
Ereditarietà



Aggiungiamo una nuova classe "Gruppo"

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

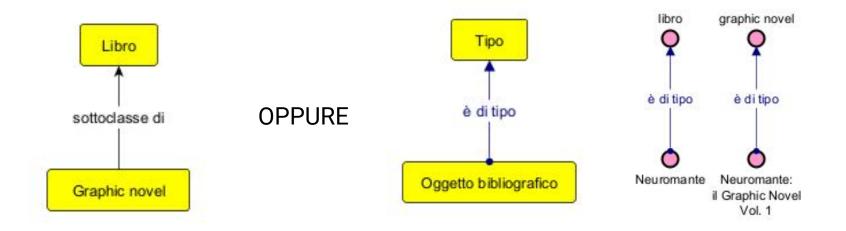
Invece di assegnare le stesse proprietà a due classi diverse, possiamo assegnarle ad una superclasse a loro comune: Agente



Tassonomia Animale Mammifero Rettile Volatile Struttura informativa gerarchica di classi e sottoclassi basata Primate Canide Gabbiano sull'ereditarietà Volpe Persona Scimpanzè Cane

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

Proprietà o classe: tipizzazione



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: eventi

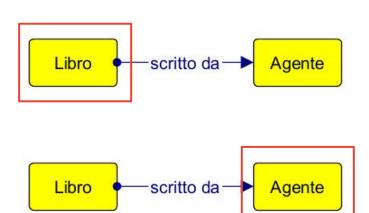


Il concetto di creazione di un Libro è reso da una relazione "scritto da" tra Libro e Agente Il concetto di creazione di un Libro è una classe "Creazione" che rappresenta l'evento in cui un'istanza di libro è stata creata da un'istanza di Agente

Dominio e codominio

Dominio: classe del primo membro della proprietà

Codominio: classe del secondo membro della proprietà

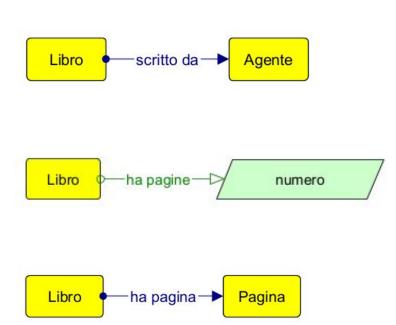


Vincoli sulle proprietà

Opzionale vs **necessaria**: deve avere almeno un valore per ogni istanza del suo dominio

Monovalente vs **polivalente**: deve avere *al massimo* un valore per ogni istanza del suo dominio

Dominio e/o codominio: ogni istanza del (co)dominio deve appartenere ad una/delle specifica/he classe/i



Vincoli e cardinalità

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

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



Vincoli e cardinalità

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

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



Vincoli e cardinalità

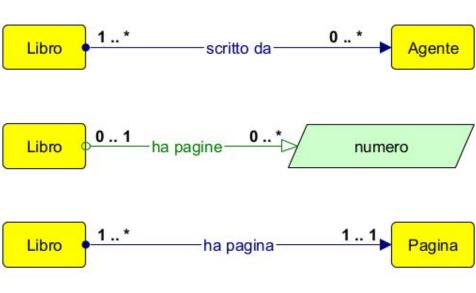
ma, mb:

- $0 \rightarrow \text{proprietà opzionale}$
- $1 \rightarrow \text{propriet}$ à necessaria

na, nb:

- $1 \rightarrow \text{propriet} \hat{a}$ monovalente;
- * → proprietà polivalente







5.5 Database

Dati strutturati
Definizione
Struttura di un DB relazionale
Modello per DB relazionale

Dati strutturati



I dati strutturati sono dati organizzati in uno schema predefinito (es. tabella)

- Hanno una struttura rigorosa (es. colonne e righe ben definite)
- Solitamente, sono facili da gestire e analizzare con l'uso di database relazionali e linguaggi di interrogazione
- Esempi: tabelle di database, fogli di calcolo, ecc.

Structured Data

ID	Name	Age	Program			
160801	Shaun	18	Communication			
160802	Linh	20	Accounting			
160803	Ahmed	19	Psychology			

Fonte:

https://gleematic.com/differences-between-structured-semi-structured-and-unstructured-data

/



Dati semi-strutturati



I dati semi-strutturati sono dati con una struttura parziale e flessibile

- Possono essere organizzati in formati come XML, JSON, documenti HTML, ecc.
- Sono più flessibili dei dati strutturati ma richiedono spesso una strutturazione minima che segua determinati schemi di codifica e vocabolari
- Esempi: documenti XML, documenti JSON, documenti HTML, dataset RDF, ecc.

Semi-Structured Data

Fonte:

https://gleematic.com/differences-between-structured-semi-structured-and-unstructured-data

/



Dati semi-strutturati



I dati non strutturati sono dati che mancano di una struttura organizzata

- Sono spesso in forma libera, come testo naturale, audio o video
- L'analisi richiede tecniche avanzate, come l'elaborazione del linguaggio naturale (NLP)
- Esempi: documenti di testo, registrazioni audio, immagini, ecc.

Unstructured Data

The university has 5600 students. Shaun (ID Number: 160801), 18 years old Communication study. Linh with ID number 160802, majoring in Accounting and is 20 years old. Ahmed from Psychology study program, 19 years old, ID number 160803.

Fonte:

https://gleematic.com/differences-between-structured-semi-structured-and-unstructured-data

/



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, re	placement_cost from film
dvdrental-# where lengtl	n > 120 and rep	lacement_d	ost > 29.50
dvdrental-# order by ti	tle 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)			

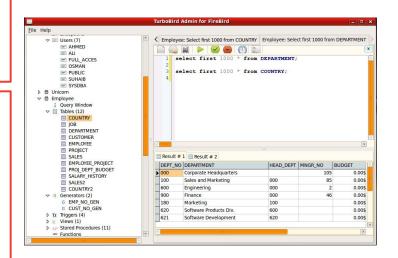
Cos'è un database?

• • • • • • • • • • • • • • • • •

La gestione di un DB è solitamente affidata a un **sistema di gestione di database** (**DBMS**), di solito eseguito su un server (es. Oracle, Microsoft SQL Server, MySQL, ...)

Un DBMS fornisce linguaggi e altri meccanismi come **SQL** (Structured Query Language) per

- definire le strutture che dovranno contenere i dati
- modificare il contenuto del DB (es. inserimento, modifica, cancellazione, interrogazione)



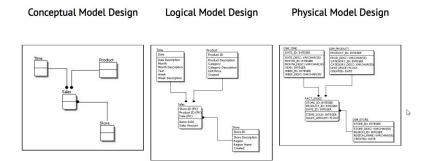
Fasi di costruzione di un DB



Modello concettuale: livello di astrazione più alto, rappresenta i concetti del mondo reale

Modello logico: livello di astrazione inferiore, rappresenta il modello concettuale in termini di istanze, struttura e regole

Modello fisico: livello di astrazione più basso: progettazione fisica del DB, considerando hardware e software



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

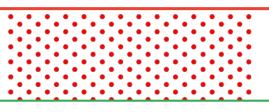
Ogni **colonna** rappresenta una **proprietà** di quella classe

	Name	Data type	Primary Key	Foreign Key
1	id	INTEGER	8	
2	document_id	INTEGER		1/4
3	text	VARCHAR		
4	type	VARCHAR		

Schema di una tabella del DB

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

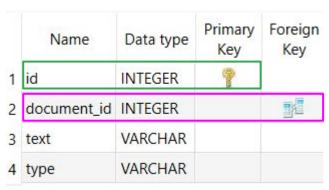
Struttura di un DB relazionale



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





Schema di una tabella del DB

id		document_id	text	type
	1	3	10.1000/182	doi
	2	3	948577574t4i	isbn
		Una tab	ella del DB	

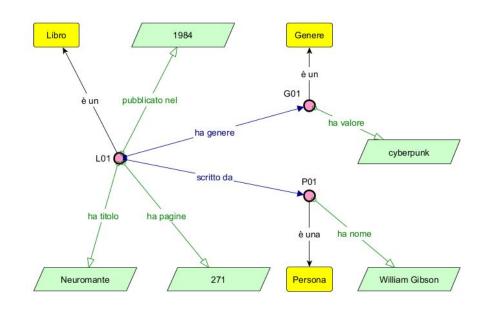
Prima: identifichiamo univocamente



Aggiungiamo un nuovo attributo a ogni classe contenente un valore per noi leggibile

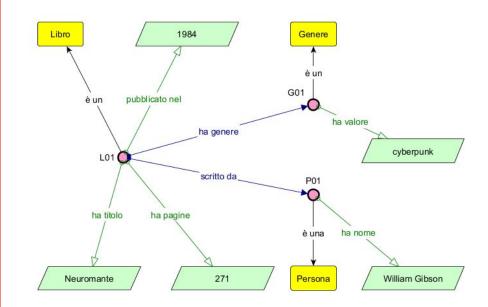
- "ha titolo" per Libro
- "ha nome" per Persona
- "ha valore" per Genere

Ci servono perché nel frattempo abbiamo assegnato degli identificatori univoci alle entità



1-1-1-1-1-1-1-1-1-1-1-1-1-1

- 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"

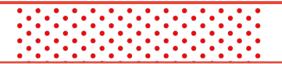


														•
•	•	•	•	•	•	•	•	•	-	•	•	•		•
													٠.	••
													•	•

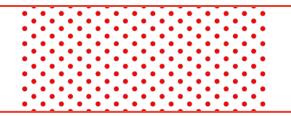
id	ha titolo	scritto da	pubblicato nel	ha pagine	ha genere
L01	Neuromante	<u>P01</u>	1984	271	<u>G01</u>
L02	II Signore degli Anelli	<u>P02</u>	1955	1178	<u>G02</u>
L03					



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



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

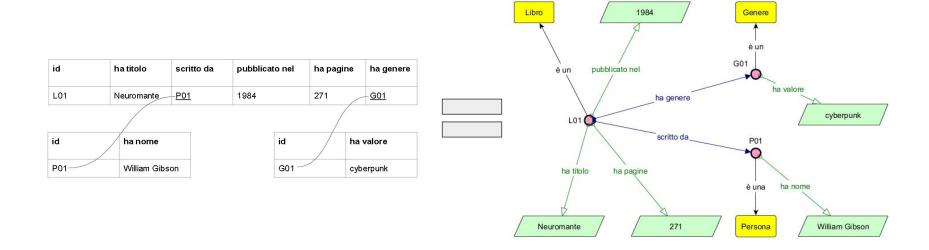


1 tabella per le persone, con colonne: "id" (chiave primaria), "ha nome"

id	ha valore
G01	cyberpunk
G02	high fantasy
G03	

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

id	ha titolo	scritto da	pubblicat	o nel	ha pagin	ne ha genere
L01	Neuromante	<u>P01</u>	1984		271	<u>G01</u>
id	ha nome			id		ha valore
P01	William Gibso	on		G01		cyberpunk



Abilità informatiche

A.A. 2023/2024

05b - Fine

Sebastian Barzaghi

<u>sebastian.barzaghi2@unibo.it</u> <u>https://orcid.org/0000-0002-0799-1527</u>