

Basi di Dati - II

Corso di Laurea in Informatica
Anno Accademico 2021/2022

Alessandra Raffaetà

raffaeta@unive.it

- Progettare una base di dati significa progettare la
 - struttura dei dati
 - applicazioni
- La progettazione della struttura dei dati è l'attività fondamentale
- Richiede di specificare un modello della realtà di interesse (**universo del discorso**) quanto più possibile fedele
- Per questo ci concentreremo sulla modellazione:
 - cosa significa definire un modello?
 - cosa si modella?
 - come si modella (quale formalismo)?

Modello dei Dati e Progettazione

Definizione: Un **modello astratto** è la rappresentazione formale di idee e conoscenze relative a un fenomeno.

- Aspetti di un modello:
 - il modello è la **rappresentazione di certi fatti**;
 - la rappresentazione è data con un **linguaggio formale**;
 - il modello è il risultato di un **processo di interpretazione**, guidato dalle idee e conoscenze possedute dal soggetto che interpreta.



- Ciascuna di queste fasi è centrata sulla modellazione
- La modellazione verrà discussa quindi con riferimento alla problematica della progettazione delle basi di dati

Modellazione concettuale

- Quale conoscenza del dominio del discorso si rappresenta?
 - (aspetto ontologico)
- Con quali meccanismi di astrazione si modella?
 - (aspetto logico)
- Con quale linguaggio formale si definisce il modello?
 - (aspetto linguistico)
- Come si procede per costruire un modello?
 - (aspetto pragmatico)

Cosa si modella?

- **Conoscenza concreta**
 - I fatti
- **Conoscenza astratta**
 - Struttura e vincoli sulla conoscenza concreta
- **Conoscenza procedurale**
 - Le operazioni di base
 - Le operazioni degli utenti
- **Comunicazioni**
 - Come si comunicherà con il sistema informatico

Nel seguito l'attenzione sarà sulla conoscenza concreta e astratta.

- Fatti specifici che si vogliono rappresentare:
 - le **entità** con le loro **proprietà**,
 - le **collezioni** di entità omogenee e
 - le **associazioni** fra entità.

- Le **entità** sono ciò di cui interessa rappresentare alcuni fatti (o **proprietà**):
oggetti concreti, oggetti astratti, eventi
 - Es.: un libro, una descrizione bibliografica, un prestito
- Le **proprietà** si distinguono dalle entità poiché sono fatti che interessano solo in quanto descrivono caratteristiche di determinate entità
 - Es.: indirizzo che interessa solo in quanto indirizzo di un utente
- **Nota:** Un'entità non coincide con i valori delle sue proprietà

- Una **proprietà** è una coppia <Attributo, valore di un certo tipo>.
- Classificazione delle proprietà
 - **atomica o strutturata**
 - **univoca / multivalore**
 - **totale / parziale**
- Esempi:
 - **nome** (atomica, univoca, totale)
 - **residenza = [indirizzo, cap, città]** (strutturata)
 - **recapiti telefonici** (multivalore, parziale)

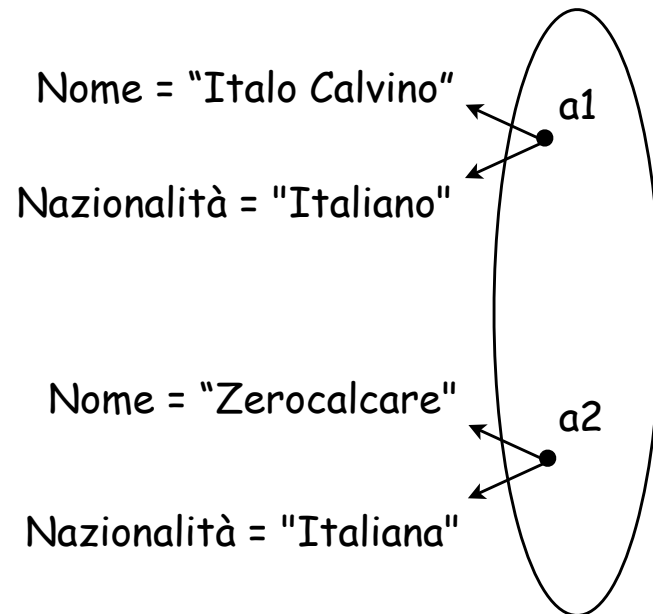
- **Tipi di entità**: ogni entità ha un **tipo** che ne specifica la natura (identifica caratteristiche: proprietà e dominio relativo)
 - Ad es. **Antonio** ha tipo **Persona** con proprietà
 - Nome: `string`
 - Indirizzo: `string`
- **Collezione (classe)**: un insieme variabile nel tempo di entità omogenee (dello stesso tipo).
 - Ad es. `Studenti`: insieme di tutti gli studenti nel dominio del discorso.

Conoscenza concreta: esempi di tipi di entità 14

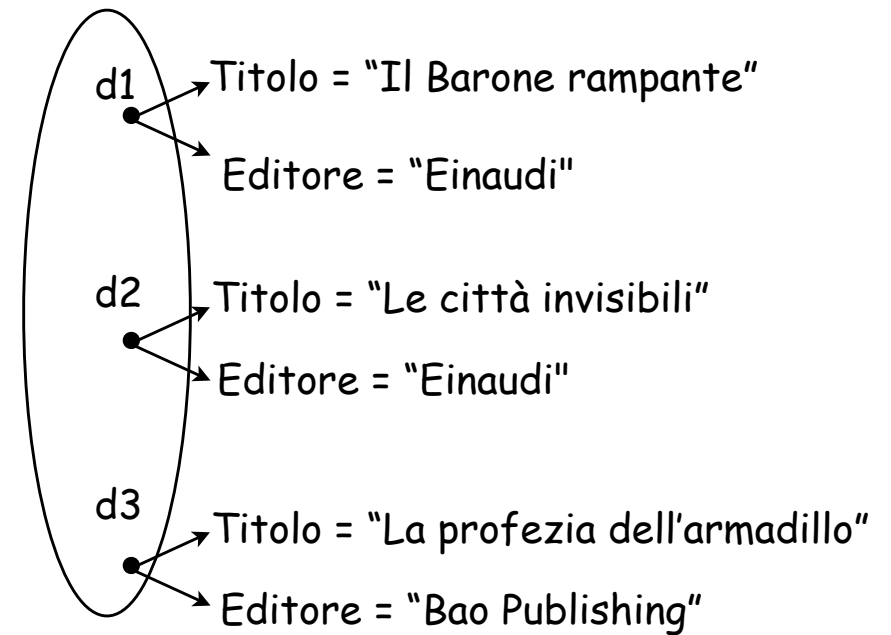
Tipo Entità	Proprietà
Studente	Nome, AnnoNascita, Matricola, e-mail, ...
Esame	Materia, Candidato, Voto, Lode, ...
Auto	Modello, Colore, Cilindrata, Targa, ...
Descrizione bibliografica	Autori, Titolo, Editore, Anno, ...

Conoscenza concreta: Esempio di Collezioni 15

Autori



Descrizioni
Bibliografiche



- Certi fatti possono essere interpretati come proprietà in certi contesti e come entità in altri, per esempio:

- **Descrizione bibliografica** con proprietà

Autori, Titolo, Editore, LuogoEdizione, Anno

oppure

- **Autore** con proprietà **Nome, Nazionalità, AnnoNascita, ...**
- **Editore** con proprietà **Nome, Indirizzo, e-mail, ...**
- **Descrizione** bibliografica con proprietà **Titolo, Anno,...**

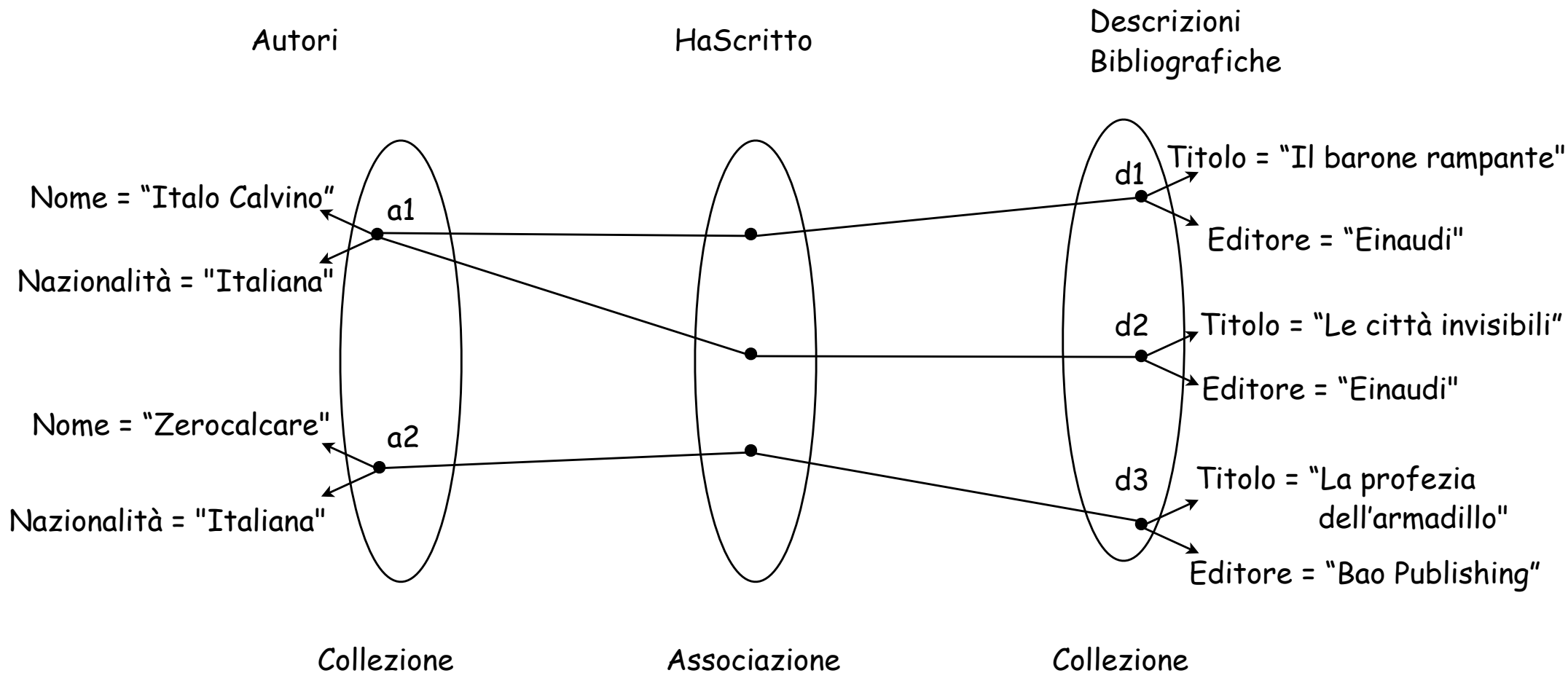
- Spesso le collezioni di entità sono organizzate in una gerarchia di **specializzazione/generalizzazione** (si parla anche di **sottoclassi** e **superclassi**)
- **Es:** nella BD della biblioteca la collezione degli `Utenti` può essere considerata una generalizzazione di `Studenti` e `Docenti`
- Due importanti caratteristiche delle gerarchie:
 - **ereditarietà** delle proprietà
 - **inclusione**: se la collezione `C1` specializza `C2`, gli elementi di `C1` sono un sottoinsieme degli elementi di `C2`

- la classe degli **studenti** universitari è una generalizzazione delle classi:
 - **matricole e laureandi,**
 - **studenti in corso e studenti fuori corso,**
 - **studenti veneziani e studenti fuori sede,**
- **Attenzione:**
 - “un laureando è **uno** (is-a) studente”
è diverso da
“Mario è **uno** studente!”

- Un'**istanza di associazione** è un fatto che correla due o più entità, stabilendo un legame logico tra di loro.
 - la descrizione bibliografica con titolo "Basi di Dati" **riguarda** il documento con collocazione "D3-55-2"
 - l'utente "Tizio" **ha in prestito** una copia della "Divina Commedia"
- Un'**associazione** $R(X, Y)$ fra due collezioni di entità X ed Y è un insieme di istanze di associazione tra elementi di X e Y , che varia in generale nel tempo. Il prodotto cartesiano $(X \times Y)$ è detto **dominio** dell'associazione.

Associazioni: Esempio

20



- Un'associazione è caratterizzata dalle seguenti proprietà strutturali:
 - **molteplicità** (o cardinalità)
 - **totalità.**

- **Vincolo di univocità:**

Un'associazione $R(X, Y)$ è **univoca da X a Y** se per ogni elemento x di X esiste al più un elemento di Y che è associato ad x ; se non vale questo vincolo, l'associazione è **multivalore da X a Y**.

- **Cardinalità**

- $R(X, Y)$ è (1:N) se essa è multivalore da X a Y ed univoca da Y a X
- $R(X, Y)$ è (N:1) se essa è univoca da X a Y e multivalore da Y a X
- $R(X, Y)$ è (N:M) se essa è multivalore da X a Y e multivalore da Y a X
- $R(X, Y)$ è (1:1): se essa è univoca su da X a Y e univoca da Y a X.

- `Frequenta(Studenti, Corsi)`
ha molteplicità **(N:M)**,
- `Insegna(Professori, Corsi)`
ha molteplicità **(1:N)**,
- `SuperatoDa(Esami, Studenti)`
ha molteplicità **(N:1)**,
- `Dirige(Professori, Dipartimenti)`
ha molteplicità **(1:1)**.

Vincolo di totalità: Un'associazione $R(X, Y)$ è **totale** da X a Y se per ogni elemento x di X esiste almeno un elemento di Y che è associato ad x ; se non vale questo vincolo, l'associazione è **parziale** da X a Y .

Esempio: `Insegna(Professori, Corsi)` è totale su `Corsi` in quanto non può esistere un corso del piano di studi senza il corrispondente docente che lo tiene, parziale su `Professori`, in quanto un professore potrebbe non tenere corsi.

Tipi di associazioni fra Persone e Città:

NataA(Persone, Città)

ha cardinalità (N:1), totale su Persone e parziale su Città

HaVisitato(Persone, Città)

ha cardinalità (N:M), parziale su Persone e Città

ÈSindacoDi(Persone, Città)

ha cardinalità (1:1), parziale su Persone e Città

- Fatti generali che descrivono:
 - la **struttura della conoscenza concreta**
 - **collezioni**: nomi, tipo degli elementi (nome, dominio, caratteristiche delle proprietà), ...
 - **associazioni**: nomi, collezioni correlate, proprietà strutturali
 - restrizioni sui **valori possibili** della conoscenza concreta e sui modi in cui essi possono **evolvere** nel tempo (**vincoli d'integrità**),
 - regole per **derivare** nuovi fatti da altri noti.

Conoscenza astratta: Vincoli di Integrità ²⁷

- **Vincoli di integrità statici:** definiscono delle condizioni sui valori della conoscenza concreta che devono essere soddisfatte indipendentemente da come evolve l'universo del discorso
 - Es. `Stipendio` deve essere positivo, `Matricola` è una chiave, ...
- **Vincoli di integrità dinamici:** definiscono delle condizioni sul modo in cui la conoscenza concreta può evolvere nel tempo.
 - Es.: `DataNascita` non può cambiare, uno studente iscritto ad un corso di laurea non può iscriversi nuovamente, ...

- Fatti derivabili da altri:
 - L'età di una persona, ricavabile per differenza fra l'anno attuale e il suo anno di nascita
 - La media dei voti degli esami superati da uno studente

Come si modella?

Come si modella? Modello dei dati a oggetti 30

- Un **modello dei dati** è un insieme di meccanismi di astrazione per descrivere la struttura della conoscenza concreta (**schema**)
- Uno schema verrà dato usando una notazione grafica, variante dei cosiddetti **diagrammi ER** (Entità-Relazione)
- Nozioni fondamentali:
 - Oggetto, Tipo di oggetto, Classe
 - Ereditarietà, Gerarchia fra tipi, Gerarchia fra classi

dominio del discorso

entità

tipo entità

collezione

associazione

modello informatico

oggetto (entity instance)

tipo oggetto (entity type)

classe (entity)

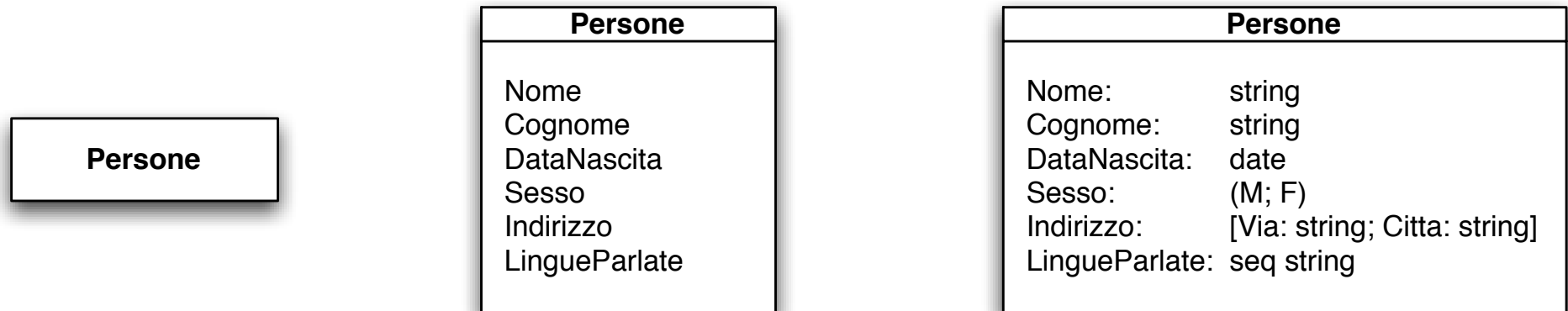
associazione o relazione

- Ad ogni **entità** del dominio corrisponde un **oggetto** del modello informatico.
- Un oggetto è un'entità software con **stato**, **comportamento** e **identità**.
 - Lo **stato** è modellato da un insieme di **costanti o variabili** con valori di qualsiasi complessità.
 - Il **comportamento** è modellato da un insieme di procedure locali (con parametri) chiamate **metodi**.
 - L'**identità** è associata all'oggetto dalla creazione e non viene modificata da aggiornamenti dello stato.
- Un oggetto può rispondere a richieste, dette **messaggi**, restituendo valori memorizzati nello stato o calcolati con una procedura locale.

- Il primo passo nella costruzione di un modello consiste nella **classificazione delle entità del dominio** con la identificazione di collezioni omogenee e la definizione dei **tipi degli oggetti** che le rappresentano.
- Un **tipo oggetto** definisce:
 - componenti dello stato
 - metodi

- Noi useremo solo un paio di concetti elementari del modello a oggetti:
- Oggetto:
 - identità
 - stato (insieme di attributi)
- no metodi, no incapsulamento, ecc.

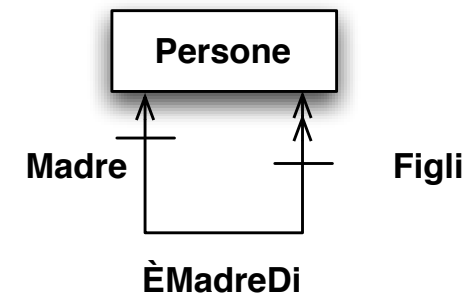
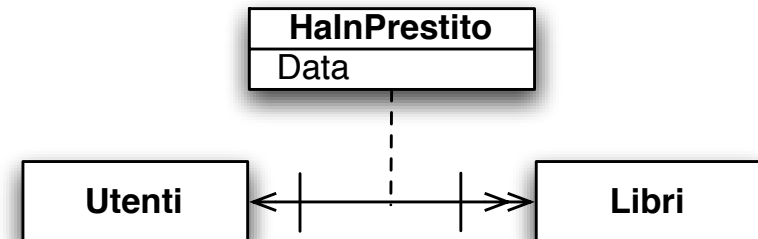
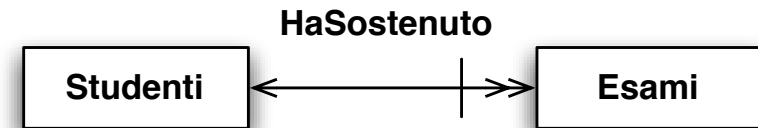
- Una classe è un insieme di oggetti dello stesso tipo, modificabile con operatori per includere o estrarre elementi dall'insieme, associabile a vincoli di integrità.



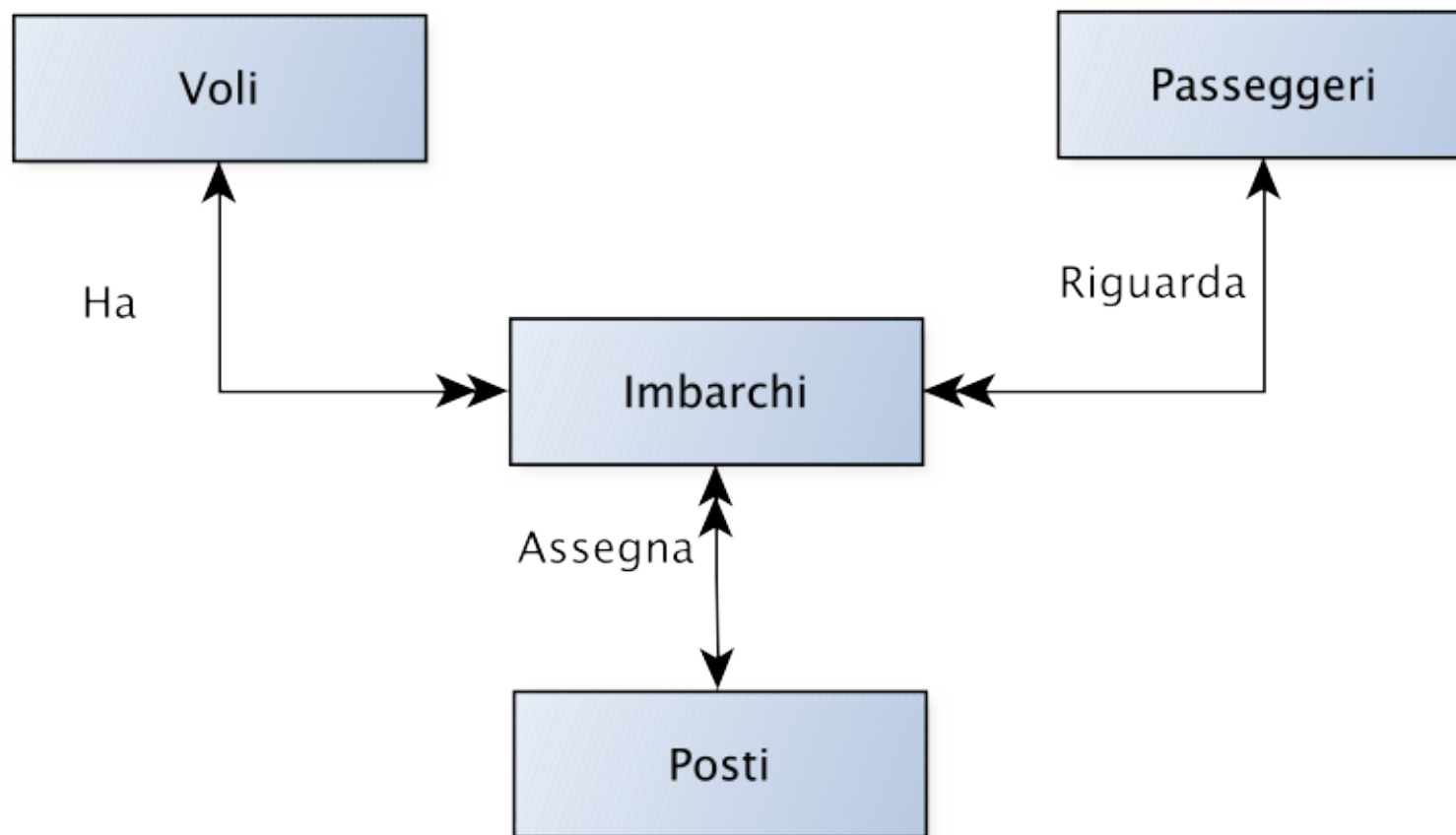
Una classe **Persone** a diversi livelli di specifica

- I tipi degli attributi possono essere
 - **primitivi** (`int`, `real`, `bool`, `date`, `string`)
 - **non primitivi**
- **Tipi non primitivi**: ottenuti applicando i seguenti operatori ad altri tipi:
 - **tipo record**
`[A1:T1; ..., An:Tn]`
 - **tipo enumerazione**
`(Val1; ...; Valn)`
 - **tipo sequenza**
`seq T`

- Le associazioni si modellano con un costrutto apposito
- Le associazioni possono avere delle proprietà
- Le associazioni possono essere ricorsive
- Associazioni n-arie



- **Esempio:** Si vuole rappresentare l'associazione tra Voli, Passeggeri e Posti. Per ogni volo, al momento dell'imbarco, viene assegnato un posto a ciascun passeggero.

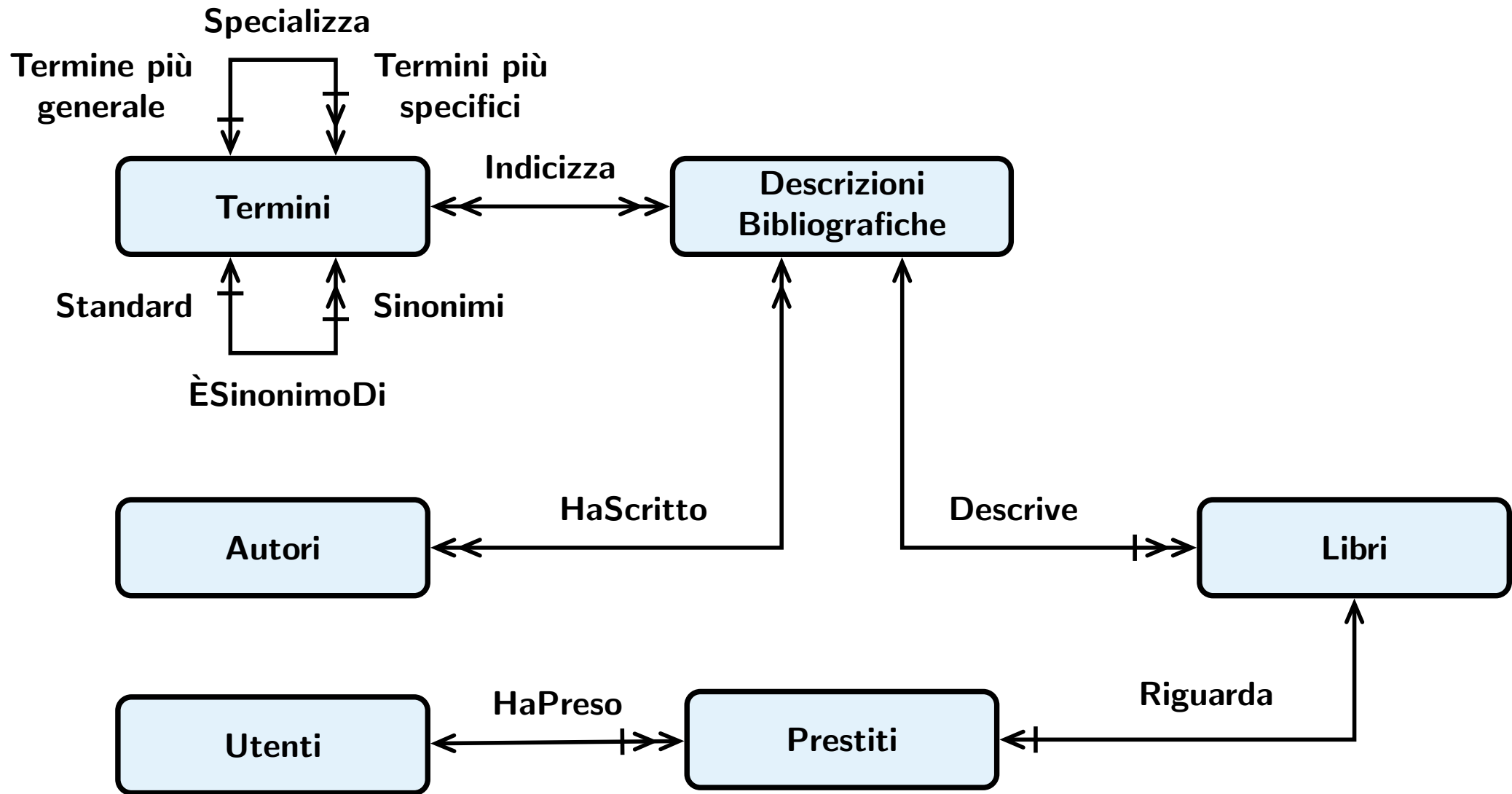


Si vogliono modellare alcuni fatti riguardanti una biblioteca universitaria:

- le **descrizioni bibliografiche** dei libri, opere con un solo volume,
- i **termini del thesaurus** (parole chiave),
- le **copie dei libri** disponibili che corrispondono ad una descrizione bibliografica,
- gli **autori** dei libri,
- gli **utenti** della biblioteca,
- i **prestiti** in corso.

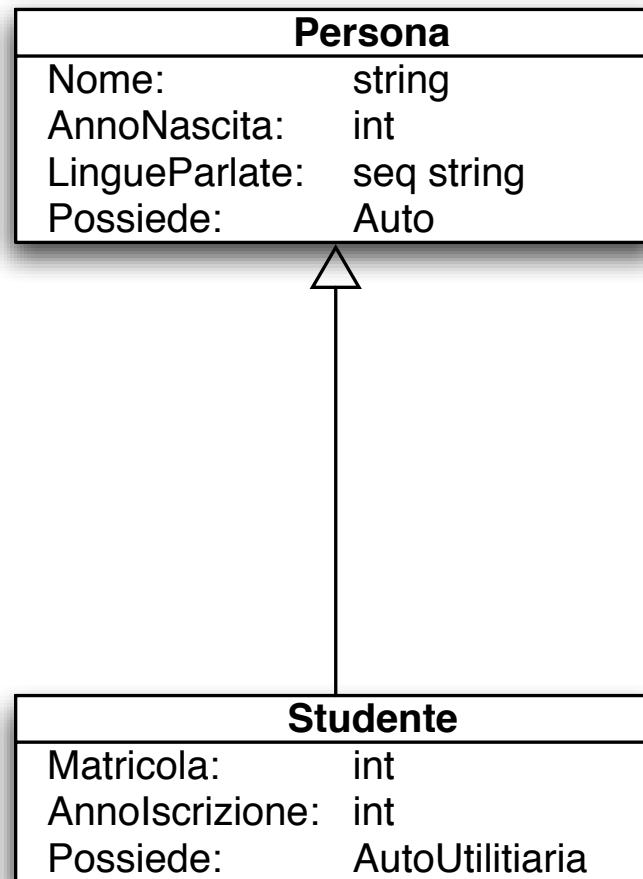
- Le Descrizioni Bibliografiche dei libri a volume unico, sia già disponibili che quelli ordinati ma non ancora consegnati alla biblioteca, sono caratterizzati dal Codice ISBN (International Standard Book Number) che le identifica, titolo del libro, autori, editore, anno di pubblicazione e termini che le descrivono.
- Degli autori dei libri interessano il codice fiscale, il nome e cognome, la nazionalità e la data di nascita.
- I libri, disponibili in una o più copie ognuna identificata da un Codice, una stringa con la loro collocazione e numero.
- Quando un utente prende un libro in prestito, si registrano i dati dell'utente, se non sono già presenti, la data del prestito e la data di restituzione. Di un utente interessano il codice fiscale, il nome, il cognome, l'indirizzo e i recapiti telefonici. Un utente può avere più libri in prestito. I dati su un prestito interessano fino al momento della restituzione del libro.

- Il thesaurus è un insieme di termini, e di associazioni fra di loro, che costituiscono il lessico specialistico da usare per descrivere il contenuto dei libri.
- Fra i termini del thesaurus interessano le seguenti relazioni, fra le tante possibili:
 - **Preferenza**, per rimandi da termini standard a termini non standard e viceversa. Per esempio:
 - Elaboratore **Standard (vedi)** Calcolatore;
 - Calcolatore **Sinonimi (UsatoPer)** Elaboratore, Calcolatrice, Stazione di lavoro.
 - **Gerarchia**, per mettere in evidenza il rapporto specificità-generalità tra due termini. Per esempio:
 - Felino **PiùSpecifico** Gatto Leone Tigre;
 - Gatto **PiùGenerale** Felino;

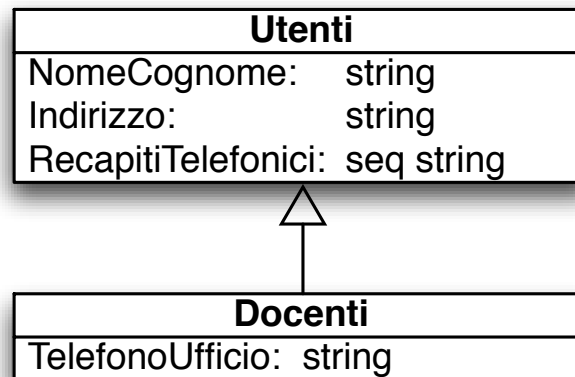


- Relazione di **sottotipo** \leq , tra i tipi oggetto (ordine parziale)
- Se **t è sottotipo di t' ($t \leq t'$)**
 - gli elementi di tipo t possono essere usati in ogni contesto in cui possano apparire elementi di tipo t' (**sostitutività**).
- In particolare:
 - gli elementi di t hanno tutti gli attributi degli elementi di t'
 - per ogni attributo A in t', il suo tipo in t è sottotipo del suo tipo in t'.

- **Ereditarietà (inheritance)**: permette di definire un tipo oggetto a partire da un altro “per differenza”
 - aggiunta di attributi
 - ridefinizione di attributi esistenti
- Normalmente l’ereditarietà tra tipi si usa solo per definire sottotipi (**ereditarietà stretta**); in questo caso:
 - gli attributi possono essere aggiunti
 - gli attributi possono essere ridefiniti solo specializzandone il tipo

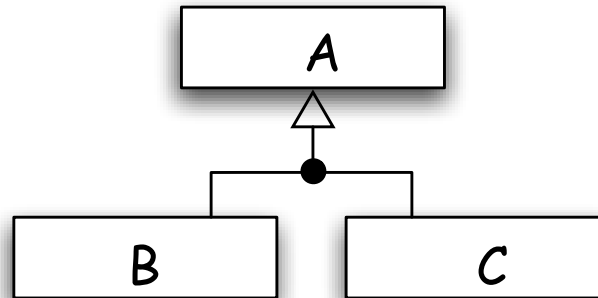


- Fra le classi può essere definita una relazione di **sottoclasse** (talvolta detta sottoinsieme), con le seguenti proprietà:
 - riflessiva, antisimmetrica e transitiva (ordine parziale).
 - Se C è sottoclasse di C' , allora le entità in C sono un sottoinsieme delle entità in C' (**vincolo estensionale**).
 - Se C è sottoclasse di C' , allora il tipo delle entità in C è sottotipo del tipo degli elementi di C' (**vincolo intensionale**)



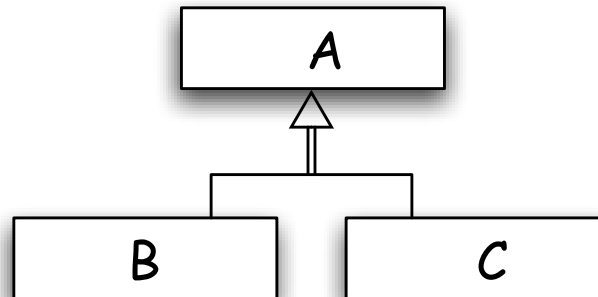
- Vincolo di disgiunzione

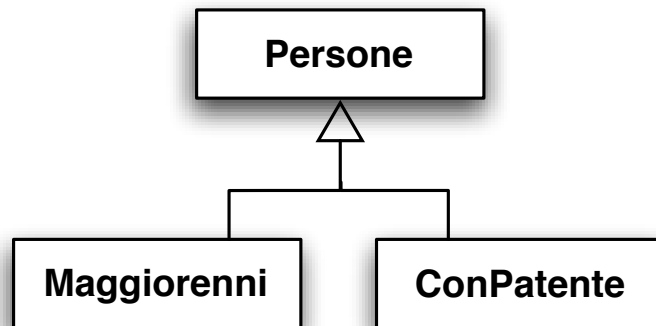
$$B \cap C = \emptyset$$



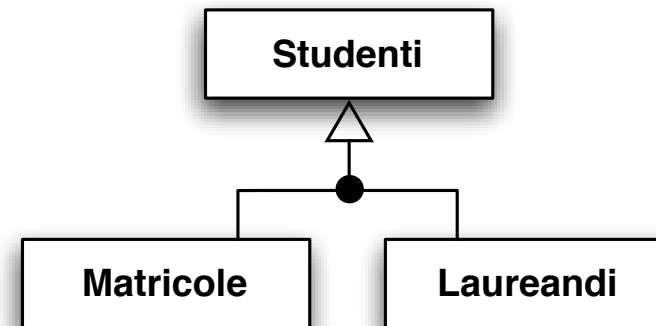
- Vincolo di copertura

$$B \cup C = A$$

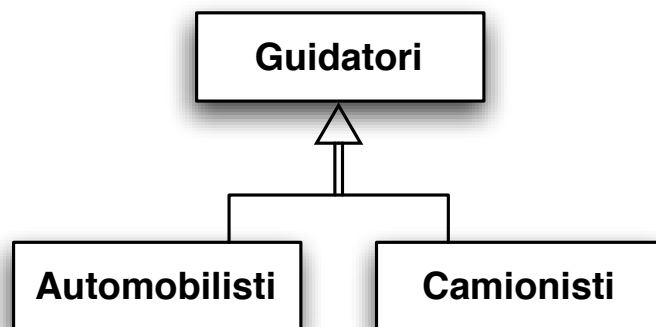




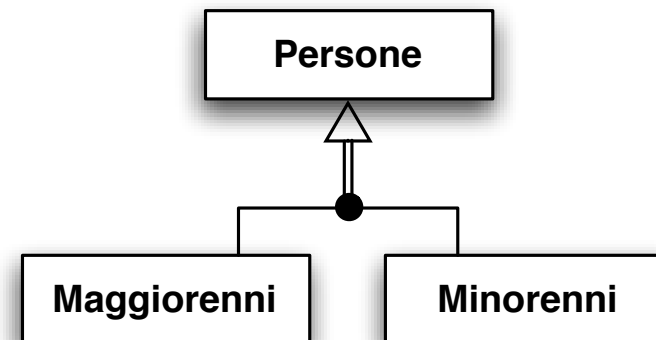
Sottoclassi scorrelate



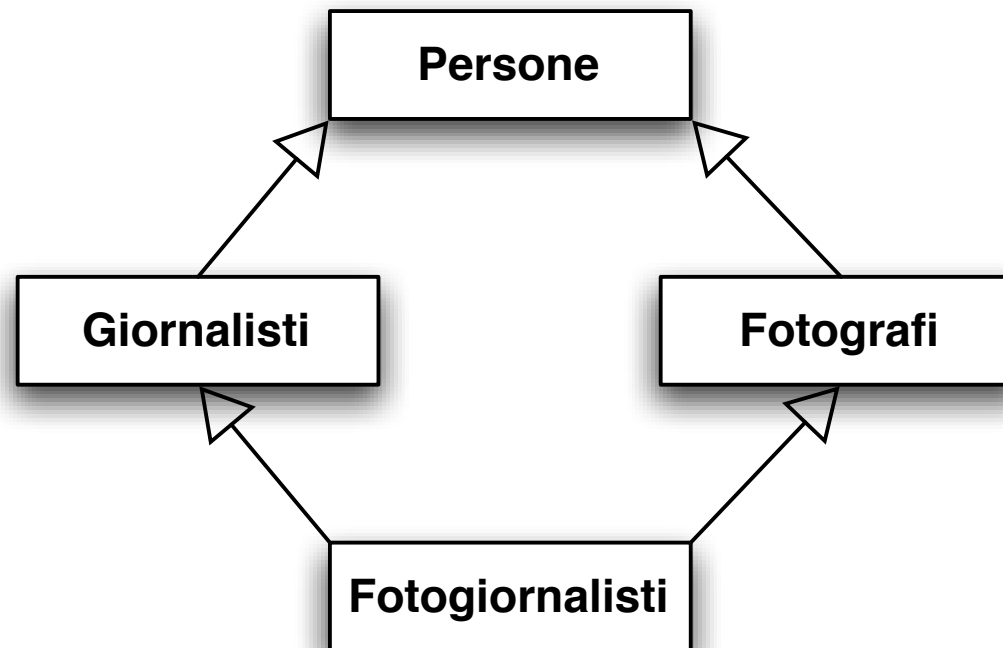
Sottoclassi disgiunte



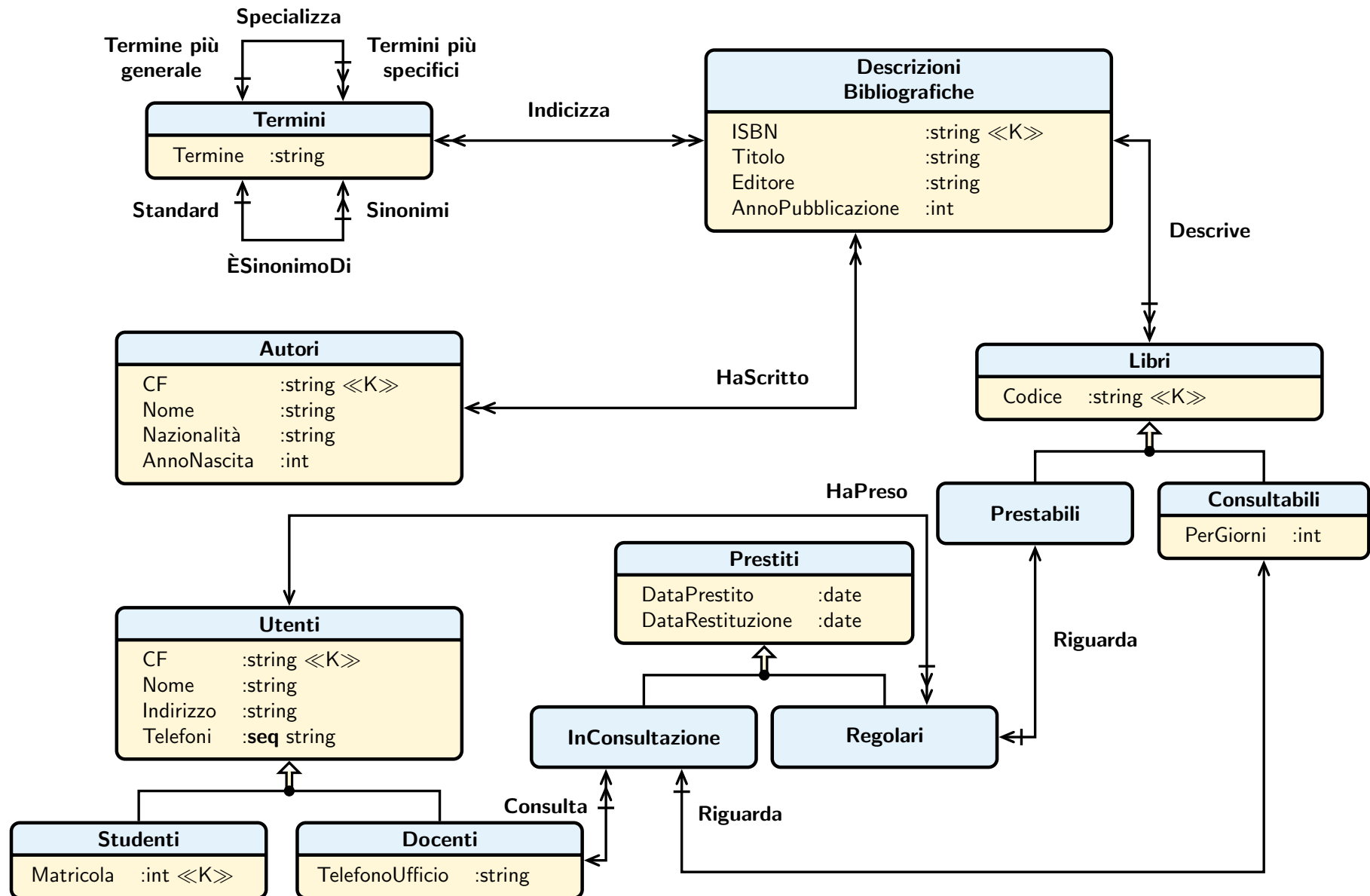
Sottoclassi copertura

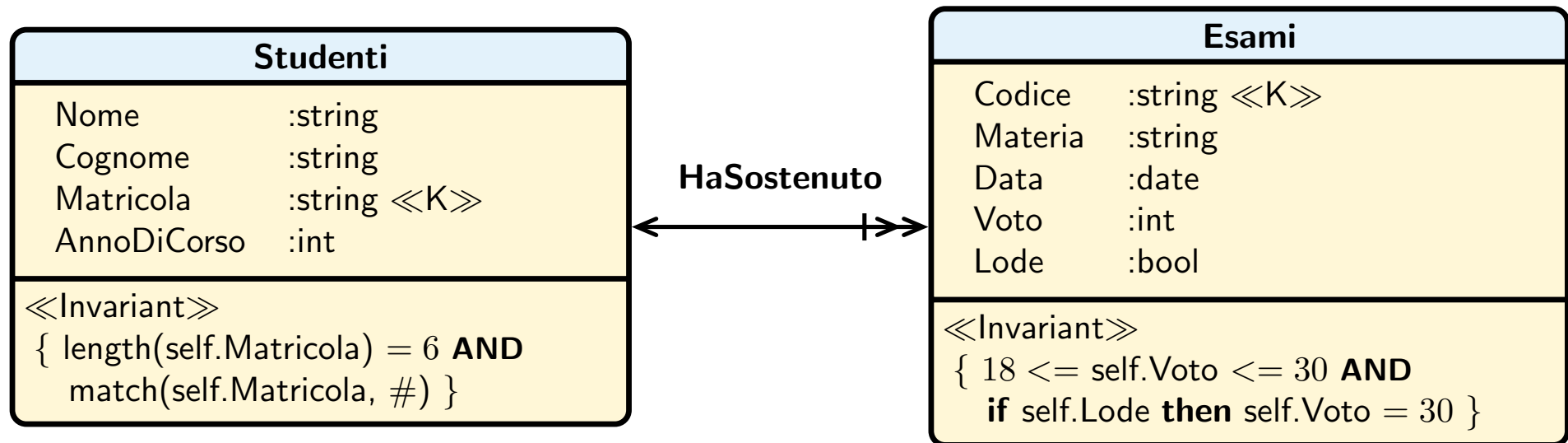


Sottoclassi partizione



- Gli utenti possono essere studenti o docenti. Di uno studente interessa anche la matricola e di un docente anche il telefono dell'ufficio.
- Alcuni libri sono per la sola consultazione e possono essere presi in prestito per un numero prefissato di giorni solo dagli utenti che sono docenti.





`<<K>>` **chiave**: sottoinsieme minimale di attributi che identifica l'oggetto

`<<NOT NULL>>` **totalità**

`self.Nome` = attributo Nome dell'oggetto stesso

In `Esami` potremmo usare `self.HaSostenuto.Matricola`

