

# Tutorato di Base di Dati

## Lezione 1

Andrea Gasparetto

# Materiale didattico

- ▶ Libro di testo:
  - A. Albano, G. Ghelli, R. Orsini. *Fondamenti di Basi di Dati*. Zanichelli, 2005.  
(<http://www.fondamentidibasididati.it/>)
- ▶ Slide del tutorato
  - TBA
- ▶ Slide del corso (da cui queste sono tratte)
  - <http://moodle.unive.it/course/view.php?id=227>
  - Enrolment Key: BD\_online

# Prerequisiti lezioni

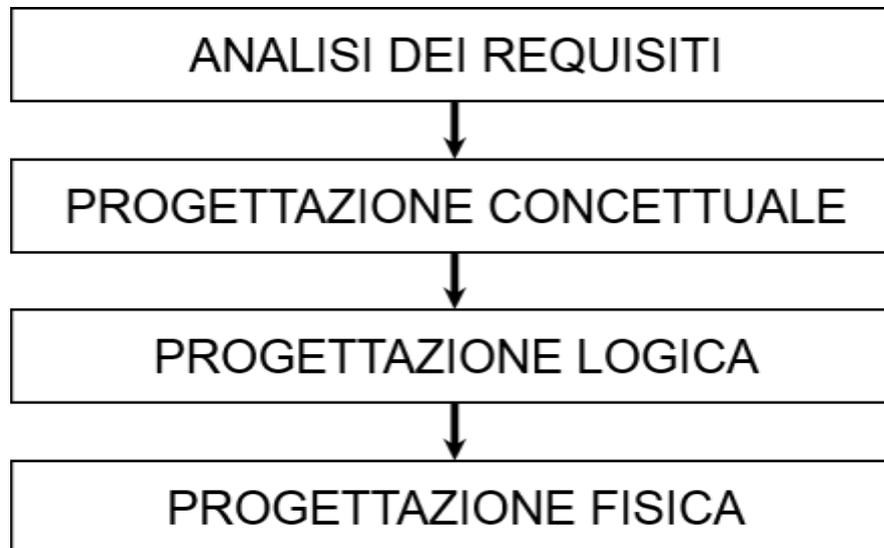
## ▶ **Lezione 2**

- **Capitolo 2: paragrafo 2.4.4**
- **Capitolo 4: il modello relazionale**

# Progettazione di una BD

- ▶ Significa:
  - **Progettare la struttura dati**
    - Richiede di specificare un *modello* della realtà (*universo del discorso*) quanto più fedele possibile
  - Progettare l'applicazione
- ▶ 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.

# Progettazione di una BD



# 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

# Conoscenza concreta

- ▶ 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

# Conoscenza concreta

- ▶ Una proprietà è una coppia  $\langle$ Attributo, valore di un certo tipo $\rangle$ .
- ▶ Classificazione delle proprietà
  - atomica o strutturata
  - univoca / multi-valore
  - totale / parziale
- ▶ Esempi:
  - nome (atomica, univoca, totale)
  - residenza = [indirizzo, CAP, città] (strutturata)
  - recapiti telefonici (multi-valore, parziale)

# Collezioni di entità

- ▶ 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.
  - Esempio →
    - *Studenti*: insieme di tutti gli studenti nel dominio del discorso.

# Scelta delle entità e delle proprietà

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

# Gerarchie

- ▶ Organizzazione delle entità in una gerarchia di specializzazione / generalizzazione
  - 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

# Associazioni

- ▶ 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

- ▶ Un'associazione è caratterizzata da:
  - molteplicità (o cardinalità)
  - totalità

# Associazioni

## ▶ Molteplicità (o cardinalità)

- 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  $Y$ .
- Molteplicità
  - $R(X, Y)$  è (1:N) se essa è multivalore da  $X$  a  $Y$  ed univoca da  $Y$  a  $X$ 
    - *Insegna(Professori, Corsi)* ha molteplicità (1:N)
  - $R(X, Y)$  è (N:1) se essa è univoca da  $X$  a  $Y$  e multivalore da  $Y$  a  $X$ 
    - *SuperatoDa(Esami, Studenti)* ha molteplicità (N:1),
  - $R(X, Y)$  è (N:M) se essa è multivalore da  $X$  a  $Y$  e multivalore da  $Y$  a  $X$ 
    - *Frequenta(Studenti, Corsi)* ha molteplicità (N:M)
  - $R(X, Y)$  è (1:1): se essa è univoca su da  $X$  a  $Y$  e univoca da  $Y$  a  $X$ .
    - *Dirige(Professori, Dipartimenti)* ha molteplicità (1:1).

# Associazioni

## ▶ Totalità

- Vincolo di totalità: Un'associazione  $R(X, Y)$  è totale da  $X$  a  $Y$  (surgettiva su  $X$ ) 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.

# Vincoli d'integrità

- ▶ 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
  - 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

# Vincoli d'integrità

- ▶ 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, ...

# Modello ad oggetti

## DOMINIO DEL DISCORSO

Entità

Tipo Entità

Collezione

Associazione

## MODELLO INFORMATICO

Oggetto (Entity Instance)

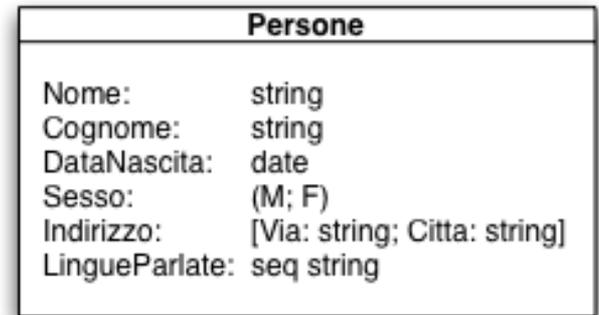
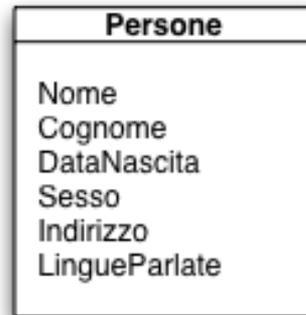
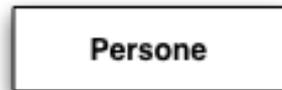
Tipo Oggetto (Entity Type)

Classe (Entity)

Associazione O Relazione

# Classi

- ▶ 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.



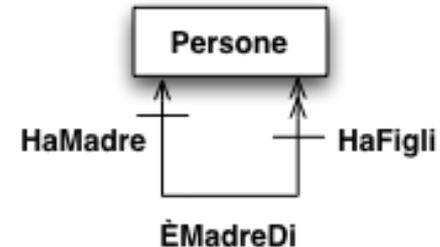
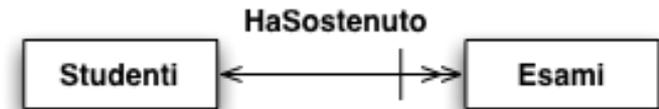
Una classe **Persone** a diversi livelli di specifica

# Classi (cont.)

- ▶ 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
$$[A_1:T_1; \dots, A_n:T_n]$$
  - tipo enumerazione
$$(Val_1; \dots; Val_n)$$
  - tipo sequenza  
seq T

# Associazioni

- ▶ Le associazioni si modellano con un costrutto apposito
- ▶ Le associazioni possono avere delle proprietà
- ▶ Le associazioni possono essere ricorsive



# Descrizione di un caso

- ▶ 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.

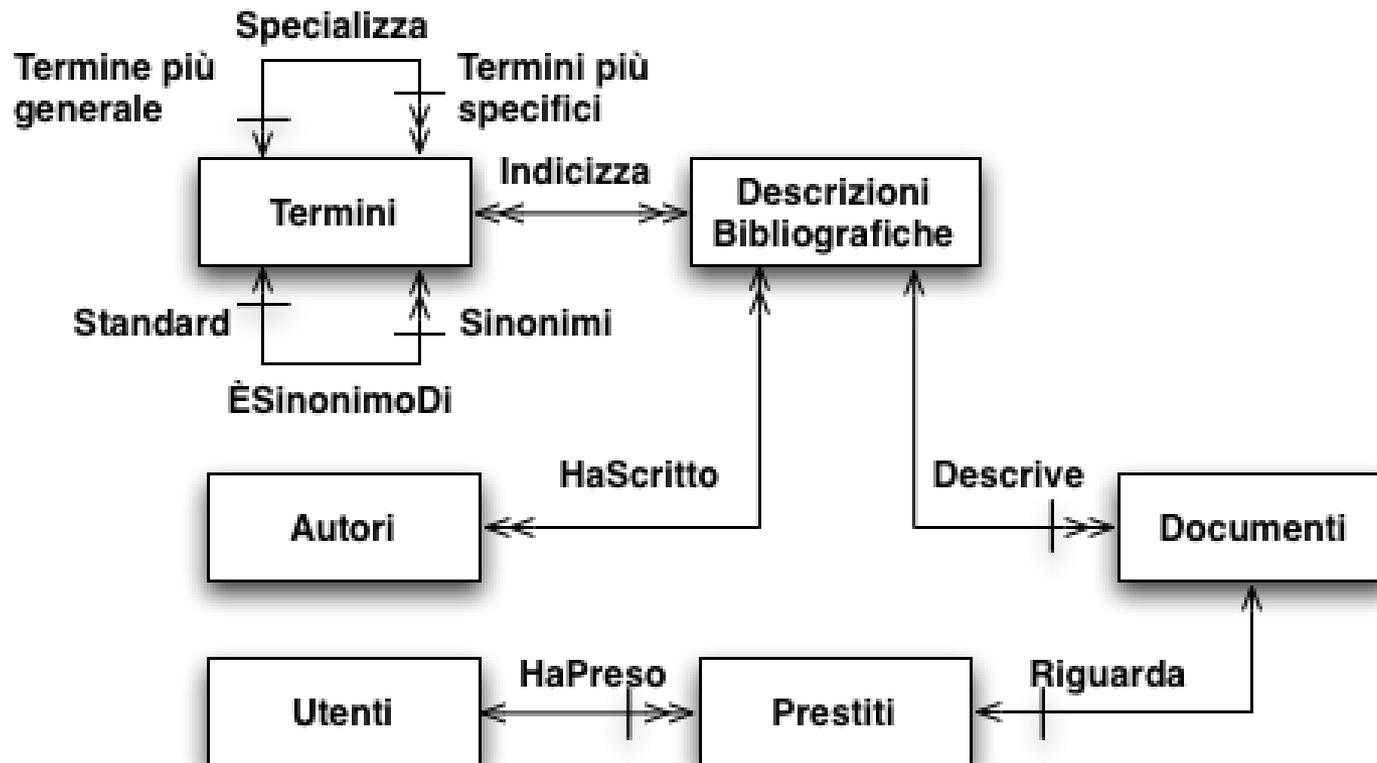
# Descrizione di un caso (cont.)

- ▶ Le descrizioni bibliografiche, che riguardano sia i libri già acquistati che quelli in ordine, sono caratterizzate dal codice ISBN, che le identifica, il titolo dell'opera, gli autori, l'editore, i termini del thesaurus che la indicizzano e l'anno di pubblicazione. Degli autori delle opere interessano il nome e cognome, la nazionalità e la data di nascita.
- ▶ Le opere disponibili (o documenti fisici) sono caratterizzate da un unico codice di collocazione e il numero della copia.
- ▶ 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 nome, il cognome, l'indirizzo e i recapiti telefonici. Un utente può avere più opere in prestito. I dati su un prestito interessano fino al momento della restituzione dell'opera. Gli utenti della biblioteca vengono sospesi dal servizio se non rispettano le regole del prestito.

# Descrizione di un caso (cont.)

- ▶ 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. Di ogni termine interessa anche una descrizione.
- ▶ 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;

# Esempio della Biblioteca



# Descrizione di un caso (cont.)

- ▶ Gli utenti possono essere studenti o docenti. Di uno studente interessa anche la matricola e di un docente anche il telefono dell'ufficio.
- ▶ Alcune opere sono per la sola consultazione e possono essere presi in prestito solo da docenti. L'accesso in sola consultazione è limitato ai primi 6 mesi dopo l'acquisizione.

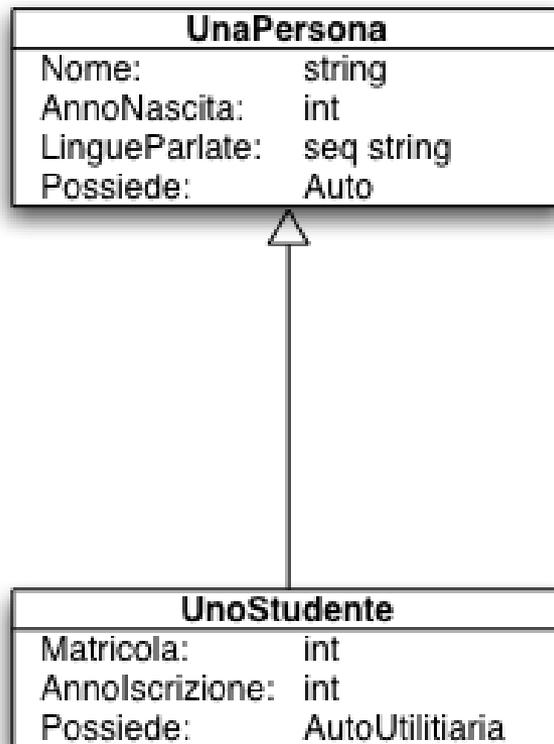
# Gerarchia tra tipi oggetto

- ▶ 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  $T'$  (**sostituibilità**).
- ▶ 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  $T'$ .
- ▶ Esempio
  - Tipi: Utente, Studente, Docente
  - $\text{Docente} \leq \text{Utente}$ ,  $\text{Studente} \leq \text{Utente}$   $\rightarrow$  posso usare uno Studente o un Docente considerandolo un Utente.

# Ereditarietà

- ▶ **Ereditarietà**: 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 (usando un sottotipo del tipo dell’attributo)

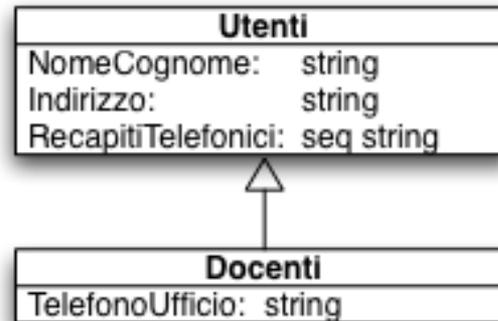
# Tipi definiti per ereditarietà



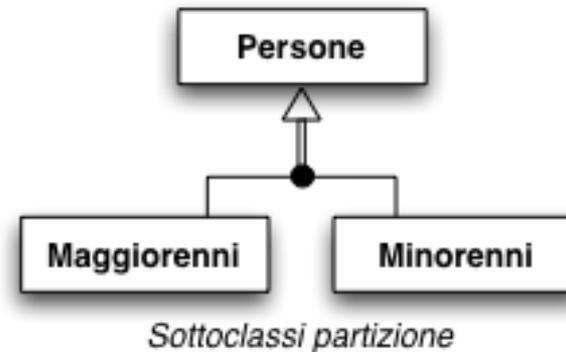
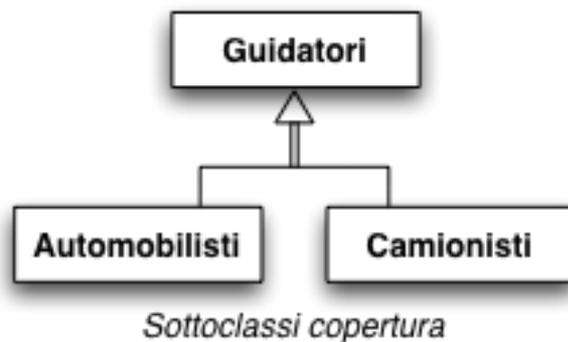
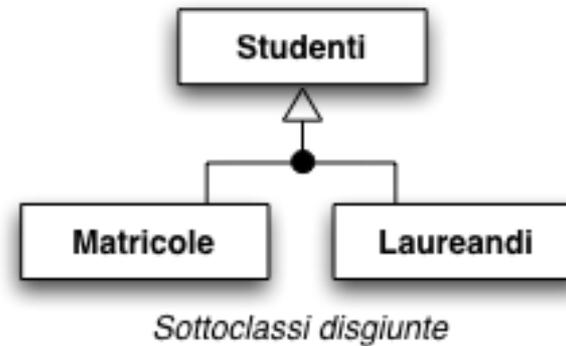
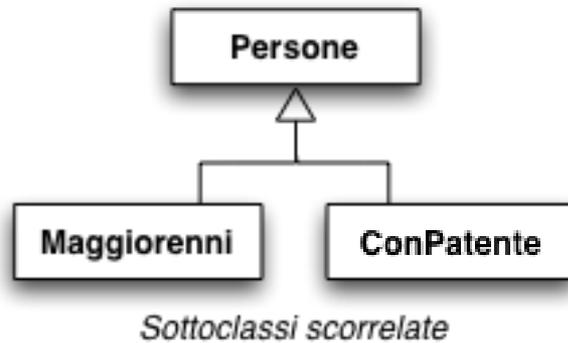
# Gerarchia tra classi

- ▶ Fra le classi può essere definita una relazione di **sottoclasse** (talvolta detta sottoinsieme), con le seguenti proprietà:
  - riflessiva, asimmetrica 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**)

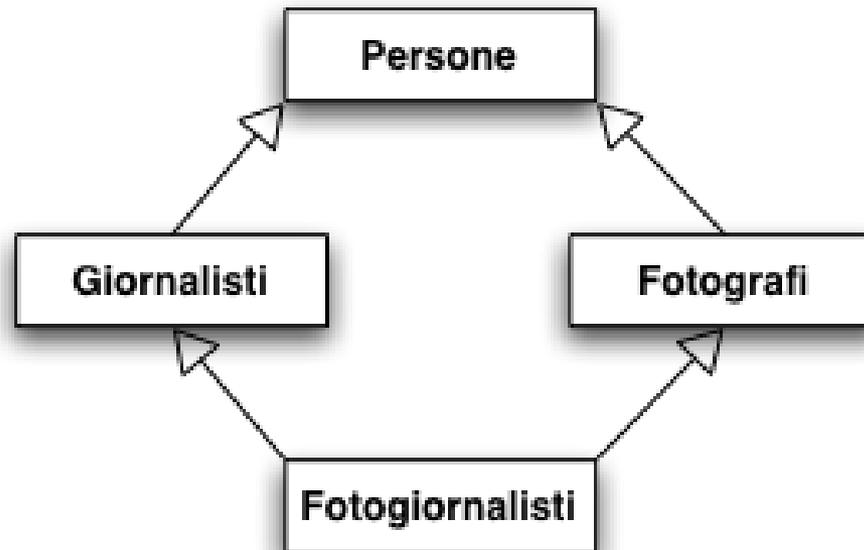
# Esempio



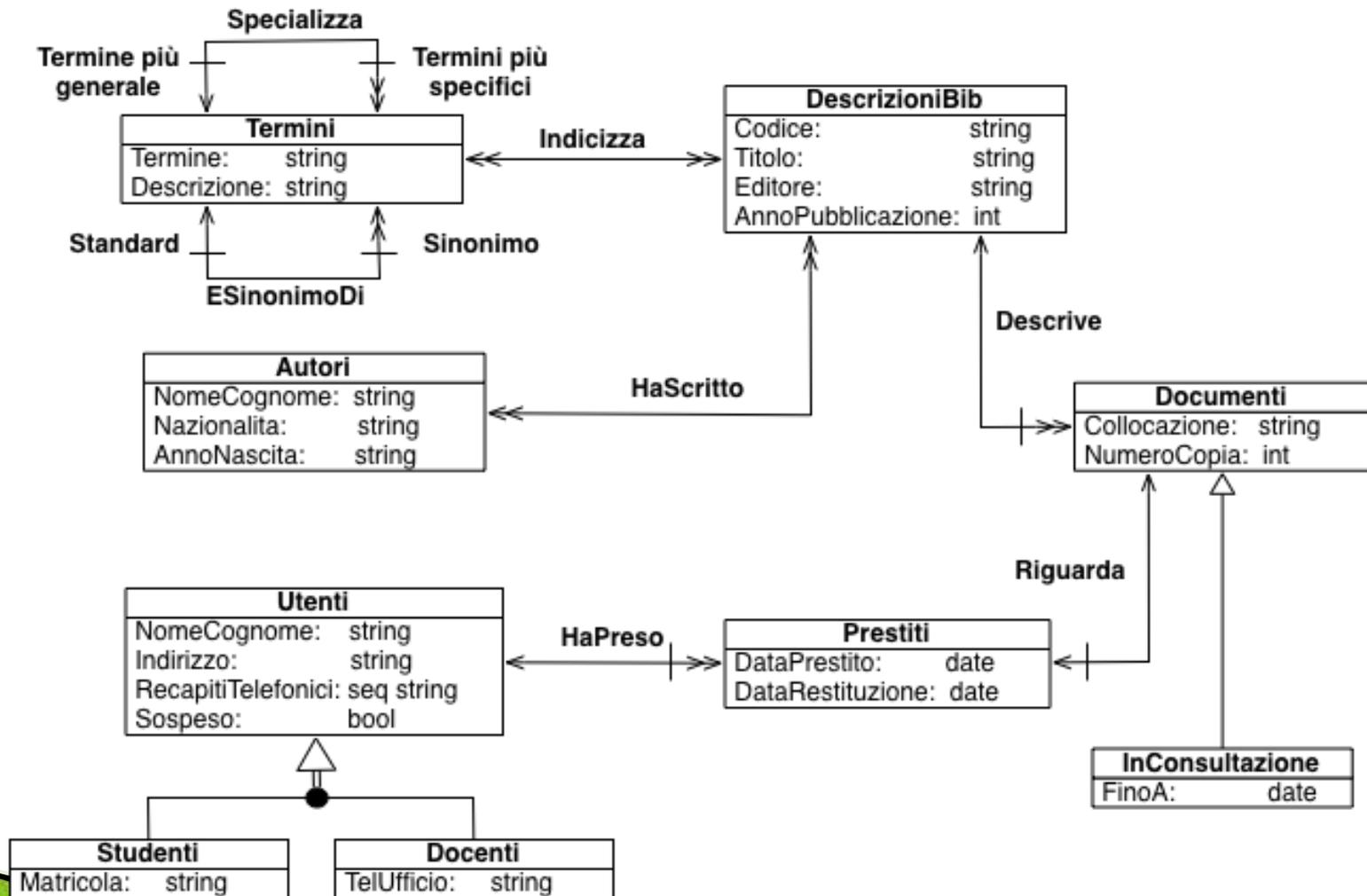
# Vincoli su sottoclassi: Esempi



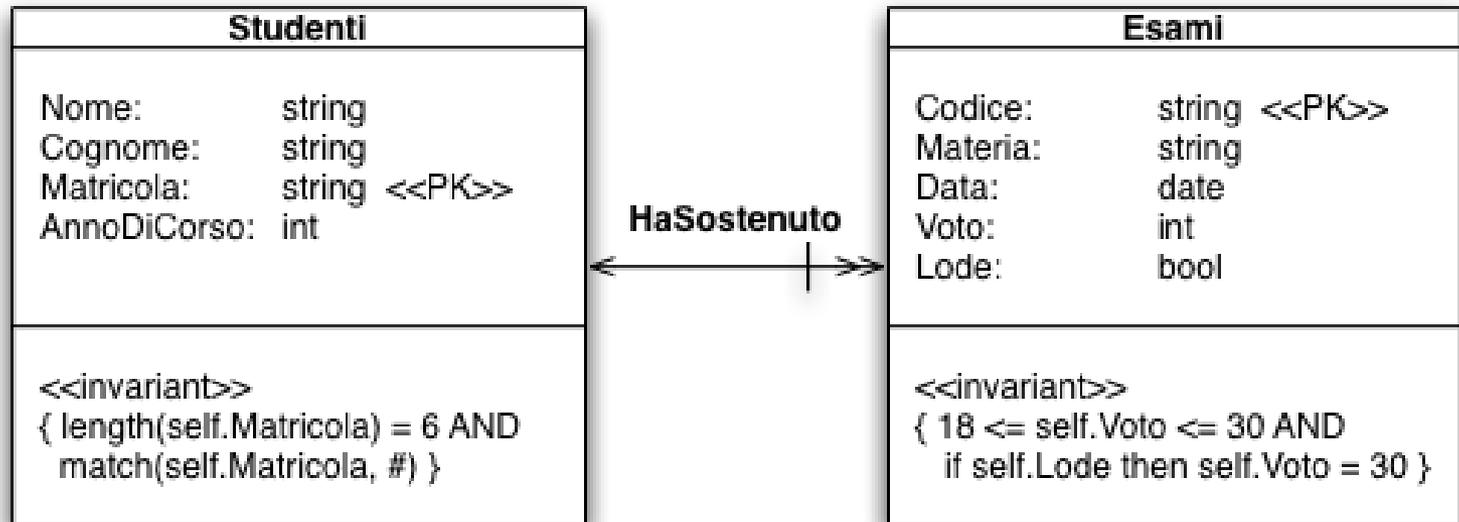
# Gerarchia Multipla



# Esempio con sottoclassi



# Descrittore di classe con vincoli



<<PK>> **chiave primaria**: sottoinsieme di attributi che identifica l'oggetto

<<NOT NULL>> **totalità**

`self.nome` = **attributo** nome **dell'oggetto stesso**

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

# Esercizio: Regate

Si vogliono gestire informazioni relative alle regate. Di una regata interessa il nome, la data e la città in cui si svolge e le barche partecipanti. Di una barca si vuole conoscere il nome della barca, il tipo di imbarcazione, il numero di remi, l'equipaggio. Dei rematori interessa il nome e cognome, l'indirizzo, la remiera di appartenenza, il luogo di nascita. Per le città interessa la provincia e il numero di abitanti. Per ogni regata si vuol inoltre conoscere la barca vincitrice e il tempo impiegato per la regata.

# Esercizio: Regate

Si vogliono gestire informazioni relative alle regate. Di una **REGATA** interessa il nome, la data e la città in cui si svolge e le barchE partecipanti. Di una barca si vuole conoscere il nome della barca, il tipo di imbarcazione, il numero di remi, l'equipaggio. Dei rematori interessa il nome e cognome, l'indirizzo, la remiera di appartenenza, il luogo di nascita. Per le città interessa la provincia e il numero di abitanti. Per ogni regata si vuol inoltre conoscere la barca vincitrice e il tempo impiegato per la regata.

# Esercizio: Regate

Si vogliono gestire informazioni relative alle regate. Di una **REGATA** interessa il nome, la data e la città in cui si svolge e le barchE partecipanti. Di una **BARCA** si vuole conoscere il nome della barca, il tipo di imbarcazione, il numero di remi, l'equipaggio. Dei rematori interessa il nome e cognome, l'indirizzo, la remiera di appartenenza, il luogo di nascita. Per le città interessa la provincia e il numero di abitanti. Per ogni regata si vuol inoltre conoscere la barca vincitrice e il tempo impiegato per la regata.

# Esercizio: Regate

Si vogliono gestire informazioni relative alle regate. Di una **REGATA** interessa il nome, la data e la città in cui si svolge e le barchE partecipanti. Di una **BARCA** si vuole conoscere il nome della barca, il tipo di imbarcazione, il numero di remi, l'equipaggio. Dei **REMATORI** interessa il nome e cognome, l'indirizzo, la remiera di appartenenza, il luogo di nascita. Per le città interessa la provincia e il numero di abitanti. Per ogni regata si vuol inoltre conoscere la barca vincitrice e il tempo impiegato per la regata.

# Esercizio: Regate

Si vogliono gestire informazioni relative alle regate. Di una **REGATA** interessa il nome, la data e la città in cui si svolge e le barchE partecipanti. Di una **BARCA** si vuole conoscere il nome della barca, il tipo di imbarcazione, il numero di remi, l'equipaggio. Dei **REMATORI** interessa il nome e cognome, l'indirizzo, la remiera di appartenenza, il luogo di nascita. Per le **CITTA'** interessa la provincia e il numero di abitanti. Per ogni regata si vuol inoltre conoscere la barca vincitrice e il tempo impiegato per la regata.

# Esercizio: Regate

Si vogliono gestire informazioni relative alle regate. Di una **REGATA** interessa il nome, la data e la città in cui si svolge e le barchE partecipanti. Di una **BARCA** si vuole conoscere il nome della barca, il tipo di imbarcazione, il numero di remi, l'equipaggio. Dei **REMATORI** interessa il nome e cognome, l'indirizzo, la remiera di appartenenza, il luogo di nascita. Per le **CITTA'** interessa la provincia e il numero di abitanti. Per ogni **regata** si vuol inoltre conoscere la barca vincitrice e il tempo impiegato per la regata.