



# Basi di Dati 2

Tipi di dato specializzati:

- Intero  $> 0$ , Reale  $\leq 0$ , etc., tipo intervallo (di interi): 18..30, 0..100, etc.

## ▼ Progetti ed Esercitazioni

La mia implementazione delle esercitazioni e progetti del corso



[Azienda 1](#)



[Voli Aerei 1](#)



[Università 1](#)

## Appunti Teoria

## Unified Modeling Language

### Diagramma delle classi e degli oggetti per l'analisi

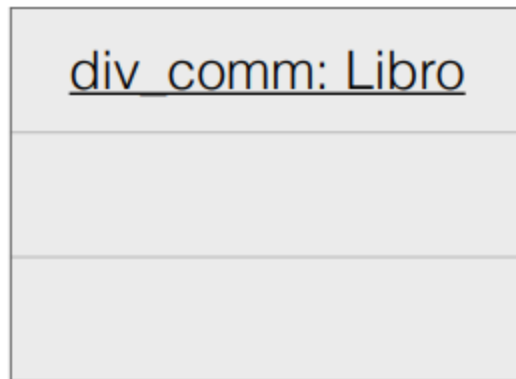
Nella fase di analisi ci si concentra sulle classi più che sugli oggetti  
Gli oggetti servono essenzialmente per descrivere elementi singoli

particolarmente significativi oppure per descrivere esempi.

## Oggetti in UML

Un **oggetto** in UML modella un elemento del dominio di analisi che:

- Ha "vita propria"
- è identificato univocamente mediante l'identificatore di oggetto
- è istanza di una classe (la cosiddetta classe più specifica – vedremo che, in determinate circostanze, un oggetto è istanza di più classi, ma in ogni caso, tra le classi di cui un oggetto è istanza, esiste sempre la classe più specifica)



*div\_comm* è l'identificatore di oggetto (scelto dall'analista per potersi riferire all'oggetto nello schema concettuale)

Libro è la classe più specifica di cui l'oggetto è istanza

Si noti la

sottolineatura

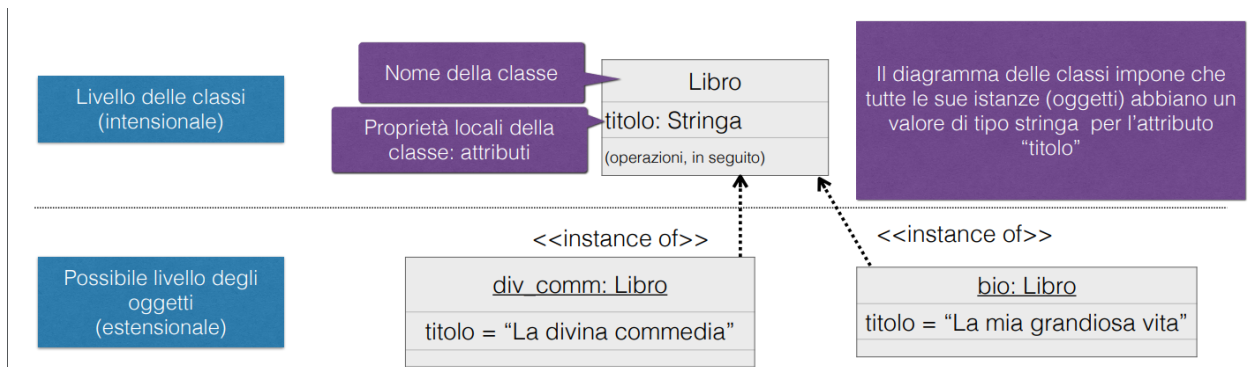
## Classi

Una **classe** modella un insieme di oggetti omogenei (le istanze della classe) ai quali sono

associate proprietà statiche (attributi) e dinamiche.

Ogni classe è descritta da:

- un nome
- un insieme di proprietà (astrazioni delle proprietà comuni degli oggetti che sono istanze delle classi)



Un **oggetto** (istanza di una classe) è **identificato** da un **identificatore univoco**  
 Un

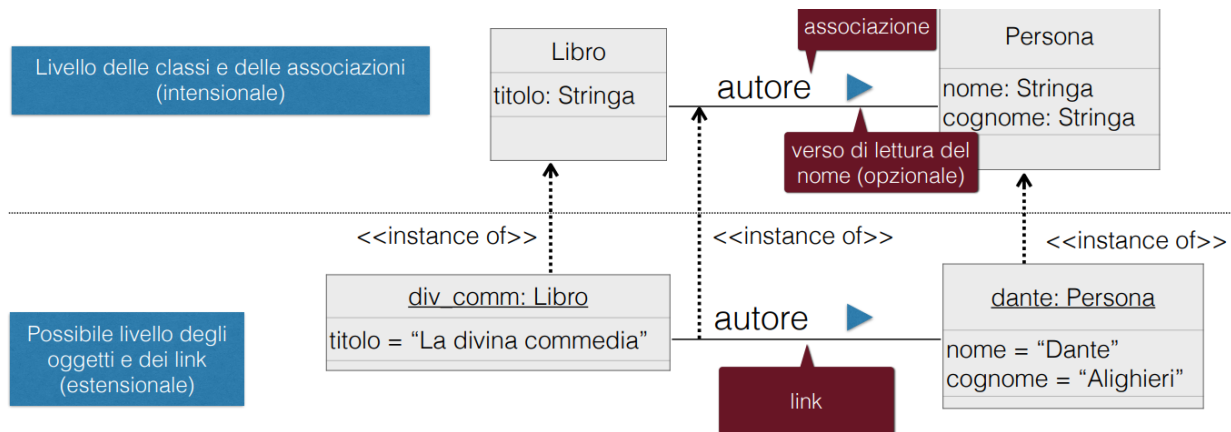
**diagramma** delle classi in generale **permette la coesistenza di oggetti identici**

## Associazioni e Link

Una **associazione** modella la possibilità che oggetti di due (o più classi) abbiano dei legami

Le

**istanze di associazioni si chiamano link**: se A è una associazione tra le classi C1 e C2, una istanza di A è un link tra due oggetti, uno della classe C1 e l'altro della classe C2.

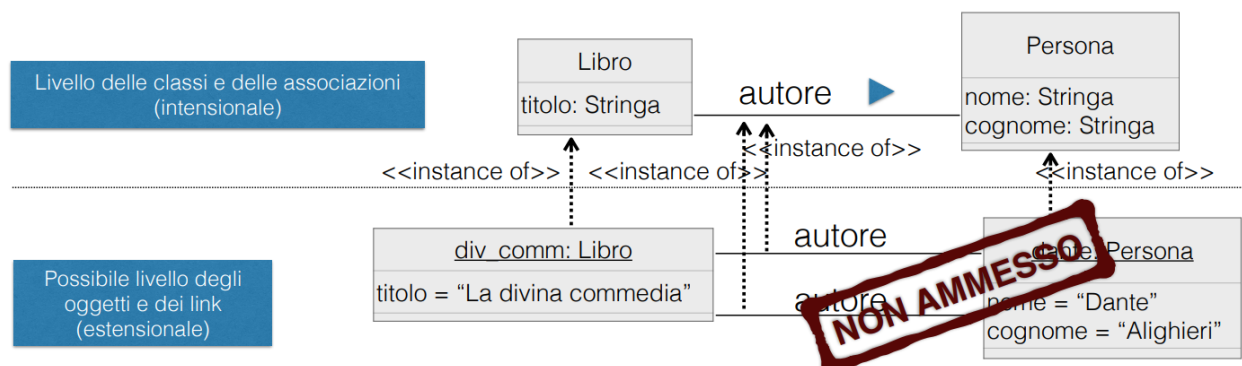


Come gli oggetti sono istanze delle classi, così i **link sono istanze delle associazioni** (gli archi <<instance of>> non sono necessari)

Al contrario degli oggetti, però,

**i link non hanno identificatori espliciti**: un link è **implicitamente identificato dalla coppia** (o in generale dalla ennupla, v. seguito) di oggetti che esso rappresenta  
Ciò implica, ad esempio, che

**il seguente diagramma degli oggetti non è ammesso** dal diagramma delle classi

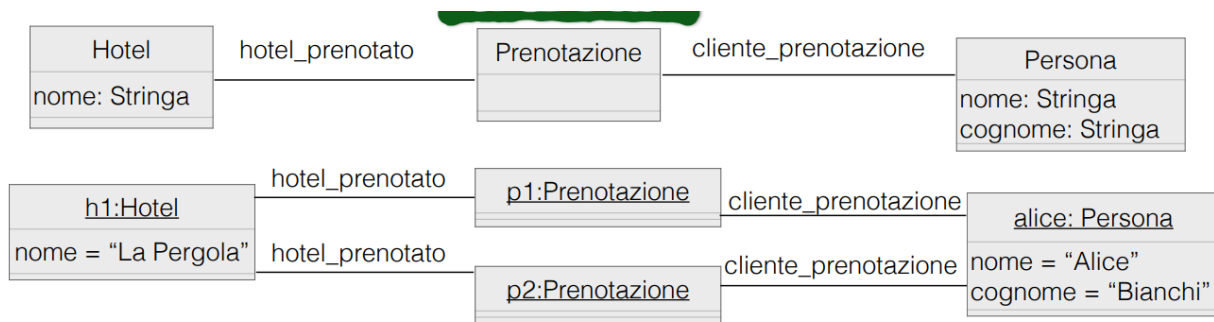


Una associazione tra le classi Libro e Persona definisce la possibilità che il le istanze (oggetti) della classe Libro siano in relazione con istanze (oggetti) di classe Persona (la relazione si chiama "autore").

Aver modellato la possibilità di tali legami mediante una associazione implica che, per l'analista, non ha concettualmente senso il fatto che un libro abbia "più volte lo stesso autore"

**Esempio:**

Si vuole progettare un'applicazione che permetta ai clienti di prenotare hotel via web



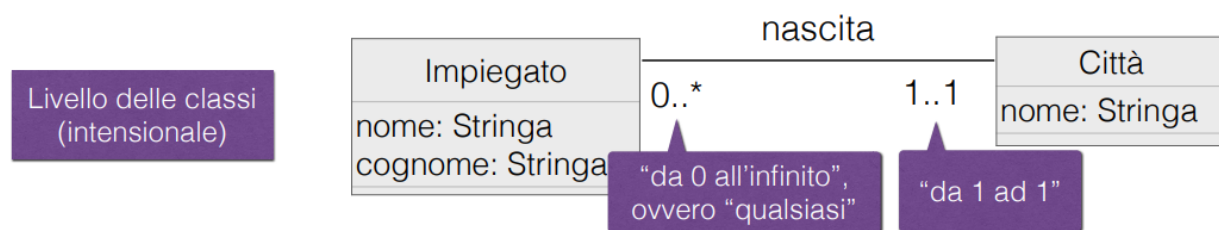
Tra le stesse classi possono essere **definite più associazioni**, che modellano **legami di natura diversa**

## Vincoli di molteplicità sulle associazioni e sugli attributi

UML permette di definire vincoli di integrità in un diagramma delle classi

Un vincolo di integrità impone ulteriori restrizioni (oltre quelle strutturali imposte dal diagramma) sui livelli estensionali ammessi

Vediamo ora i vincoli di molteplicità sulle associazioni

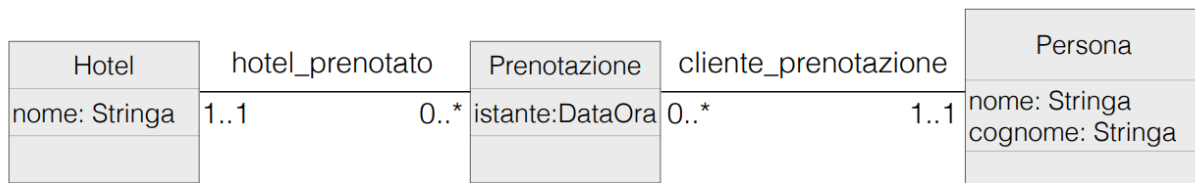


Vediamo ora i due casi:

- **1..1:** Ogni istanza di Impiegato deve essere coinvolta in un numero di link dell'associazione "nascita" che va "da 1 ad 1"  
Dato che non possiamo avere più link tra la stessa coppia di oggetti, questo è equivalente a: ogni istanza di Impiegato deve essere legata ad una ed una sola istanza di Città (tramite link dell'associazione "nascita").

- **0..:** Ogni istanza di Città deve essere coinvolta in un numero di link dell'associazione "nascita" che va "da 0 all'infinito"  
È equivalente a dire: ogni istanza di Città può essere legata ad un numero qualunque (0..) di istanze di Impiegato (tramite link dell'associazione "nascita")

Riprendendo l'esempio di prima:

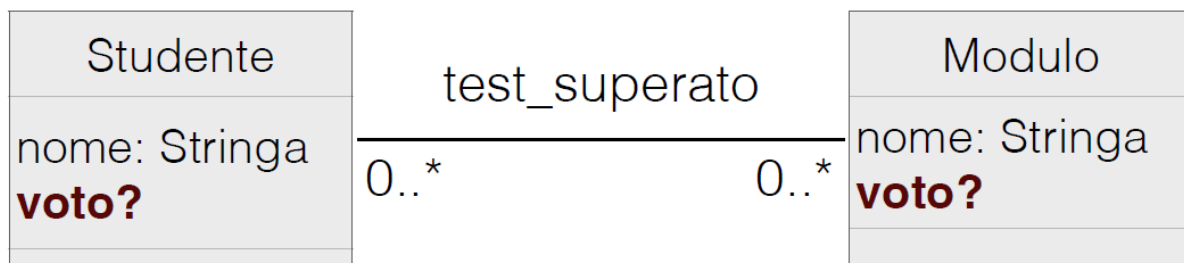


## Associazioni con attributi

Si vuole progettare un sistema che gestisca gli esiti (voti in trentesimi) dei test superati dagli studenti di un corso.

Il corso è diviso in moduli e uno studente può superare il test di ogni modulo al più una volta.

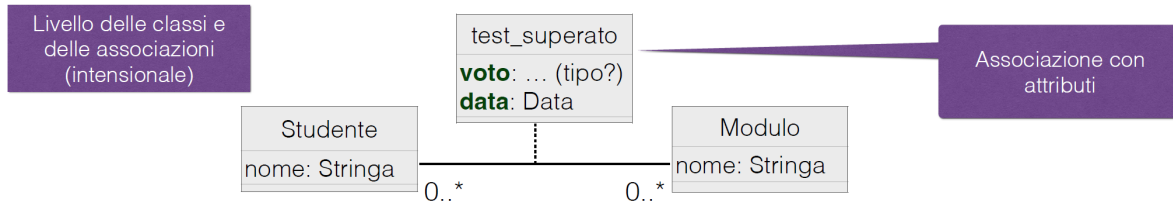
Si definisca un diagramma delle classi concettuale per l'applicazione



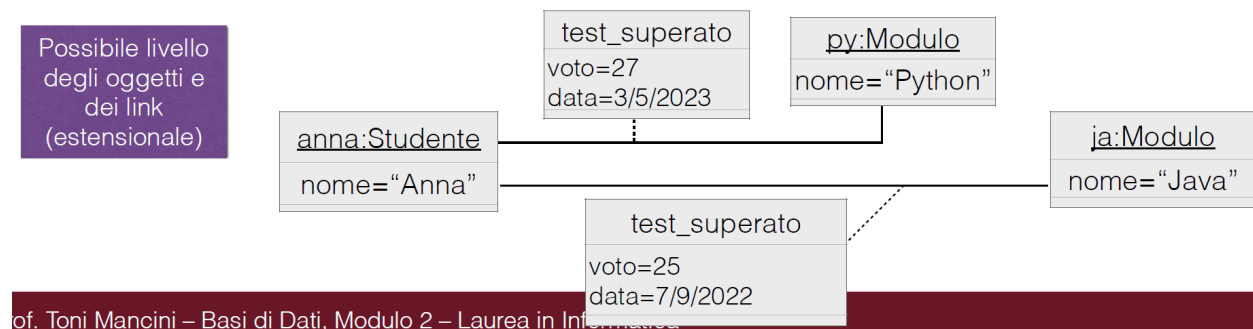
Come rappresentare i voti conseguiti dagli studenti nei test dei diversi moduli? Non possiamo aggiungere un attributo "voto" né nella classe Studente né nella classe Modulo!

In effetti, un voto non è una proprietà locale di uno studente, né di un modulo, ma è una proprietà del legame tra uno studente ed un modulo... ovvero è una **proprietà dell'associazione**

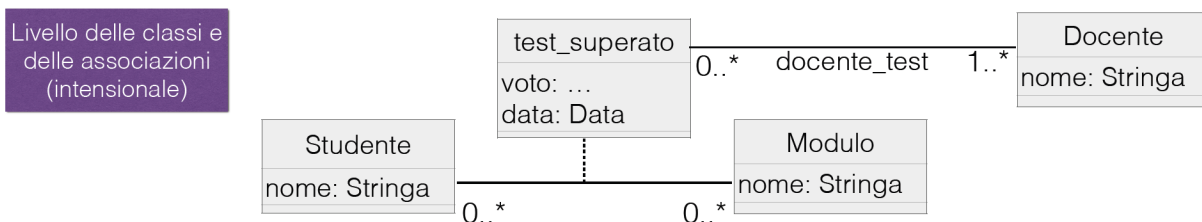
## Livello delle classi



## Livello delle istanze



Attenzione: anche in presenza di attributi, **la natura dell'associazione non cambia**:  
Il diagramma  
**permette ad una stessa coppia di oggetti di formare al più un link dell'associazione**



## Tipi di dato concettuali

Vogliamo utilizzare tipi di dato concettuali, che siano facilmente realizzabili con qualsivoglia tecnologia informatica (Python, Java, sistemi di gestione di basi di dati, etc.)

Tipi base:

- Intero, Reale, Booleano, Data, Ora, DataOra

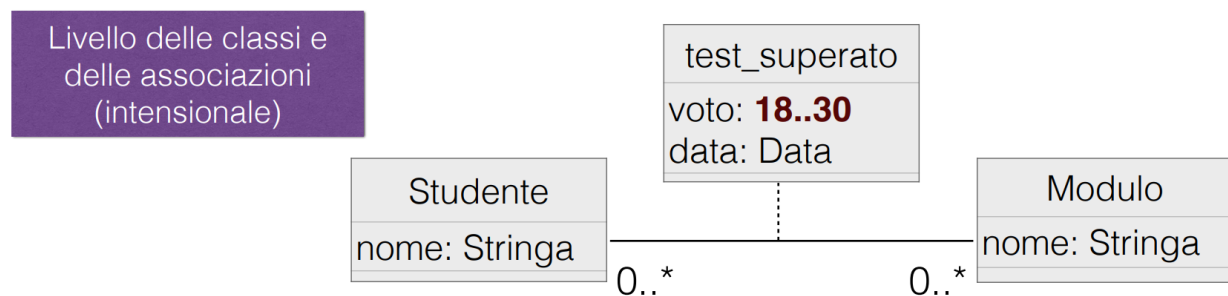
Vogliamo però anche realizzare una modellazione più simile alla realtà possibile, per questo useremo i tipi di dato specializzati:

Per esempio:

- Intero > 0, Reale <= 0, etc., tipo intervallo (di interi): 18..30, 0..100, etc. (ESTREMI INCLUSI)

## NB

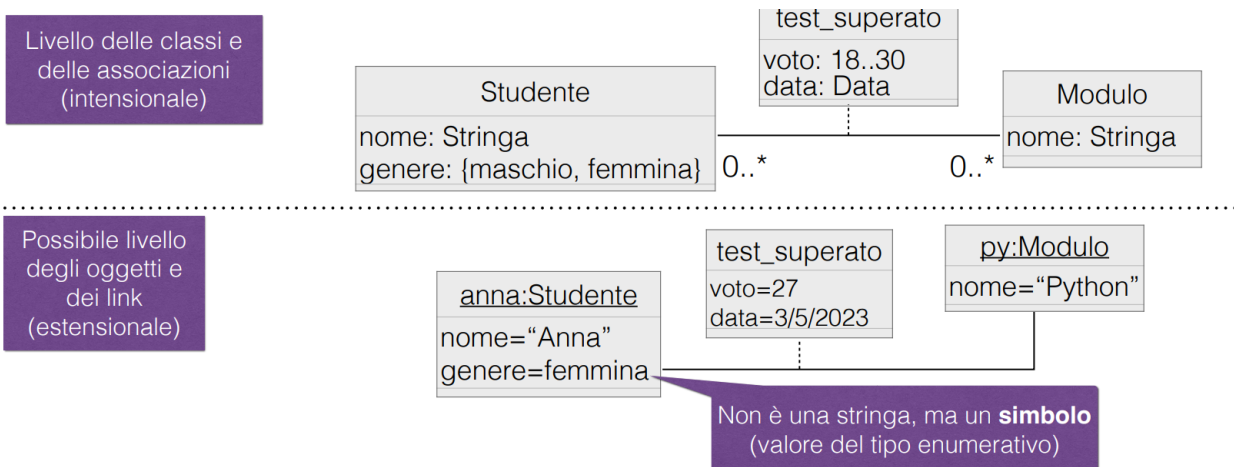
Per UML tutti gli intervalli sono da definizione di intervalli di **interi**



Quando l'insieme dei possibili valori di un attributo è **finito e piccolo**, possiamo usare un tipo di **dato enumerativo**, che definisce esplicitamente e completamente l'insieme dei valori possibili per l'attributo.

Ad esempio:

- {maschio, femmina}, {Africa, America, Antartide, Asia, Europa, Oceania}, etc.



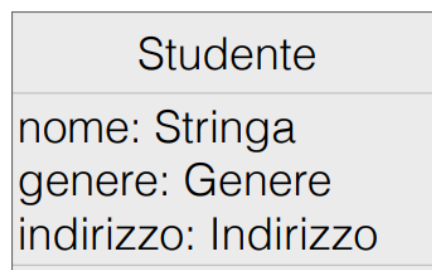


UML consente all'analista di definire anche **tipi di dato composti da più campi**: tipi record

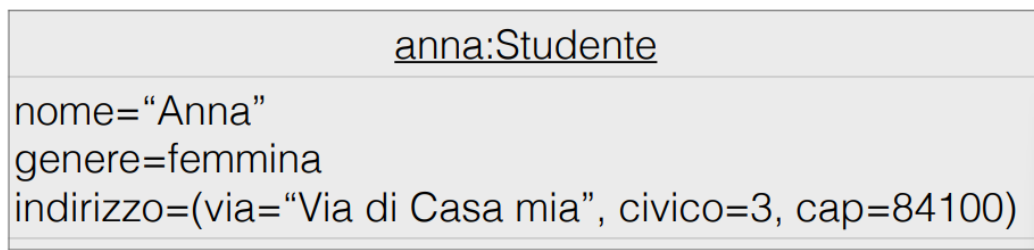
Esempio:

- Tipo Genere = {maschio, femmina}
- Tipo Indirizzo = (via:stringa, civico:intero>0, cap:intero>0)

Classe:



Istanza:



Anche gli attributi di classe e associazione possono avere **vincoli di molteplicità (default: 1..1)**

Esempio:

- Ogni studente ha associato esattamente un nome e un genere  
Ogni studente ha associato uno o più indirizzi email  
Ogni studente ha associato al più un indirizzo

Classe:

Studente
nome: Stringa genere: Genere email: Stringa [1..*] indirizzo: Indirizzo [0..1]

Istanza:

<u>anna:Studente</u>
nome="Anna" genere=femmina email={"anna@kmail.com", "anna2002@yahuu.com"} indirizzo={}