



Informatica **Umanistica**

Basi di Dati

Definizione del Modello Concettuale dei Dati:

Concetti Fondamentali



UNIVERSITÀ DI PISA

Concetti Fondamentali

- ◆ **Introduzione**
 - Raccolta dei Requisiti
- ◆ **Diagramma Concettuale delle Classi**
 - Classe
 - Associazione
 - Cardinalità
 - Generalizzazione
- ◆ **Linee Guida per la Modellazione**

Introduzione

- ◆ **Modellazione concettuale**
 - attività svolta durante l'analisi
- ◆ **Obiettivo**
 - definire il modello concettuale dei dati
- ◆ **Funzioni nel processo di sviluppo**
 - analizzare le relazioni tra i dati
 - definire il nucleo delle classi (componenti)
 - guidare la progettazione della base di dati

Introduzione

◆ Il punto di partenza

- la raccolta dei requisiti

◆ Raccolta dei Requisiti

- i requisiti iniziali sono normalmente imprecisi
- è necessario precisarli e approfondirli
- in sintesi, si tratta di comprendere come funziona la realtà a cui l'applicazione si riferisce

Raccolta dei Requisiti

◆ Fonti per la raccolta

- utenti dell'applicazione, attraverso interviste e questionari
- documentazione esistente (leggi e normative, regolamenti interni)
- modulistica usata dall'organizzazione

◆ Il problema delle fonti

- fonti diverse possono fornire informazioni contrastanti

Raccolta dei Requisiti

◆ **Attenzione**

- è un'attività difficile e poco standardizzabile

◆ **Suggerimenti**

- effettuare iterazioni ripetute con gli utenti
- tenere presente la prospettiva dell'utente sull'applicazione
- accertarsi di adottare un linguaggio comune
- identificare i casi d'uso e discuterne le attività in dettaglio, individuando le priorità

Raccolta dei Requisiti

- ◆ **Caso fortunato**
 - gli utenti conoscono UML
 - è possibile basare la comunicazione sugli schemi concettuali intermedi
- ◆ **Al termine del processo i requisiti sono descritti da**
 - un documento di sintesi
 - lo schema concettuale prodotto

Il Diagramma delle Classi di UML

- ◆ **Nel nostro approccio**
 - diagramma delle classi di UML
 - ci sono altri approcci molto simili; esempio: modello Entità-Relazione (ER) >>

- ◆ **Modello concettuale**
 - classi con attributi
 - associazioni
 - cardinalità
 - generalizzazioni

Il Diagramma delle Classi di UML

◆ **Attenzione**

- siamo in fase di analisi
- è necessario adottare un livello di “astrazione” opportuno

◆ **In particolare**

- non è necessario che ci siano tutti gli attributi
- non è rilevante il tipo degli attributi
- non sono rilevanti i metodi

Un Esempio: Il S.I. Universitario

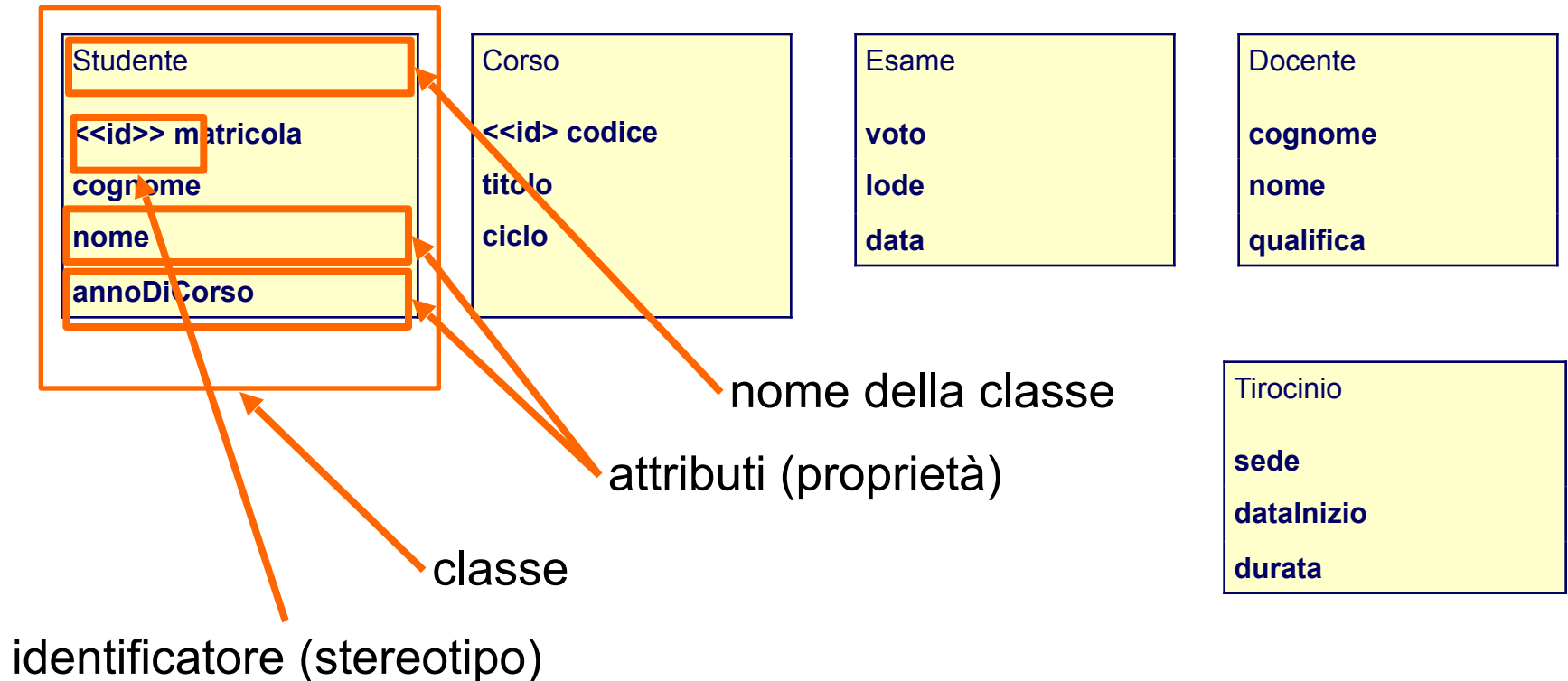
- ◆ E' necessario sviluppare un frammento del S.I. utilizzato dalla segreteria studenti per la gestione dei dati dei corsi di laurea in Informatica dell'Università della Basilicata
- ◆ Il sistema deve gestire i dati degli studenti della laurea triennale e specialistica. Per ciascuno studente è necessario gestire i dati relativi agli esami sostenuti
- ◆ Il sistema deve inoltre gestire i dati relativi agli insegnamenti offerti, e i dati relativi agli esami sostenuti per ciascun insegnamento
- ◆ Per ciascun corso è necessario tenere traccia dei docenti, che possono essere uno o più, e che sono interni o supplenti esterni;
- ◆ Per ogni docente si tiene traccia dei recapiti telefonici, per poter contattare il docente in caso di problemi relativi agli esami sostenuti
- ◆ Per gli studenti, è necessario tenere traccia del docente relatore della prova finale e dei dati del tirocinio svolto. Gli studenti della laurea triennale possono chiedere un relatore solo se sono iscritti al terzo anno
- ◆ Infine, il sistema deve tenere traccia delle attività di tutorato svolte dagli studenti della laurea sp. nei confronti degli studenti della laurea triennale

Classi

- ◆ **“Concetti” della realtà di interesse**
 - fatti, persone, cose, con esistenza autonoma
 - esempio: studente, corso, esame, docente
- ◆ **Istanza di una classe**
 - insieme di oggetti della realtà di interesse
- ◆ **Le classi hanno attributi**
 - proprietà rilevanti per l'applicazione
 - stereotipo <<id>> per gli identificatori
 - stereotipo: notazione per indicare che un costrutto ha un ruolo ben identificabile

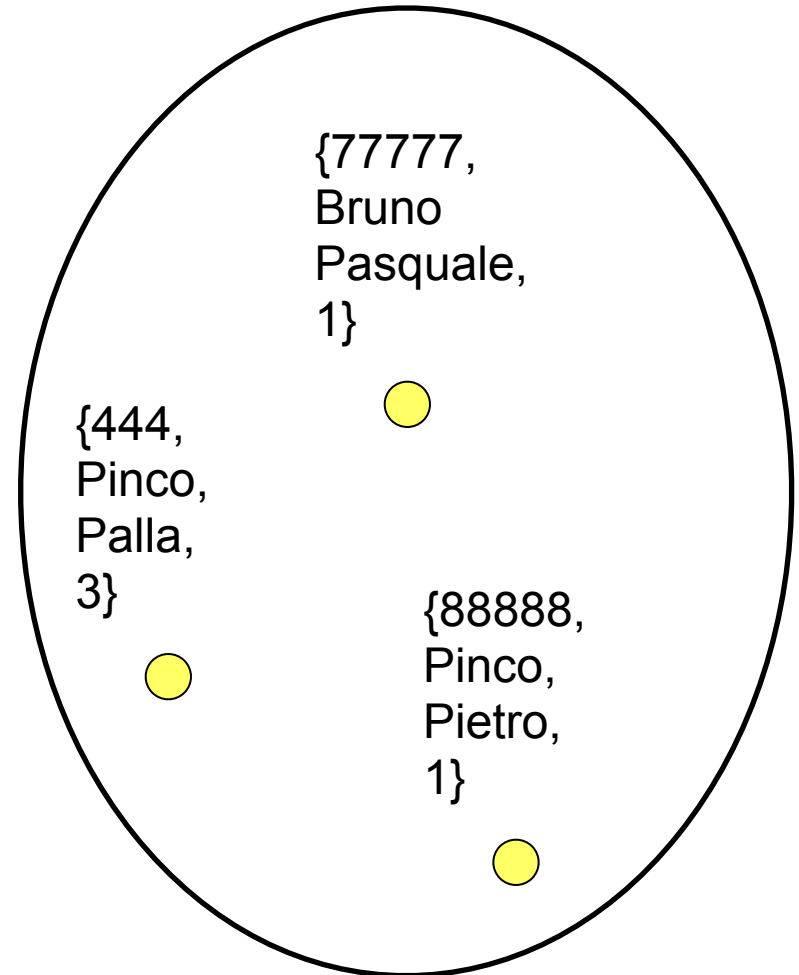
Classi

◆ Esempi:



Classi

◆ Istanze delle classi



Associazioni

◆ Relazione tra classi

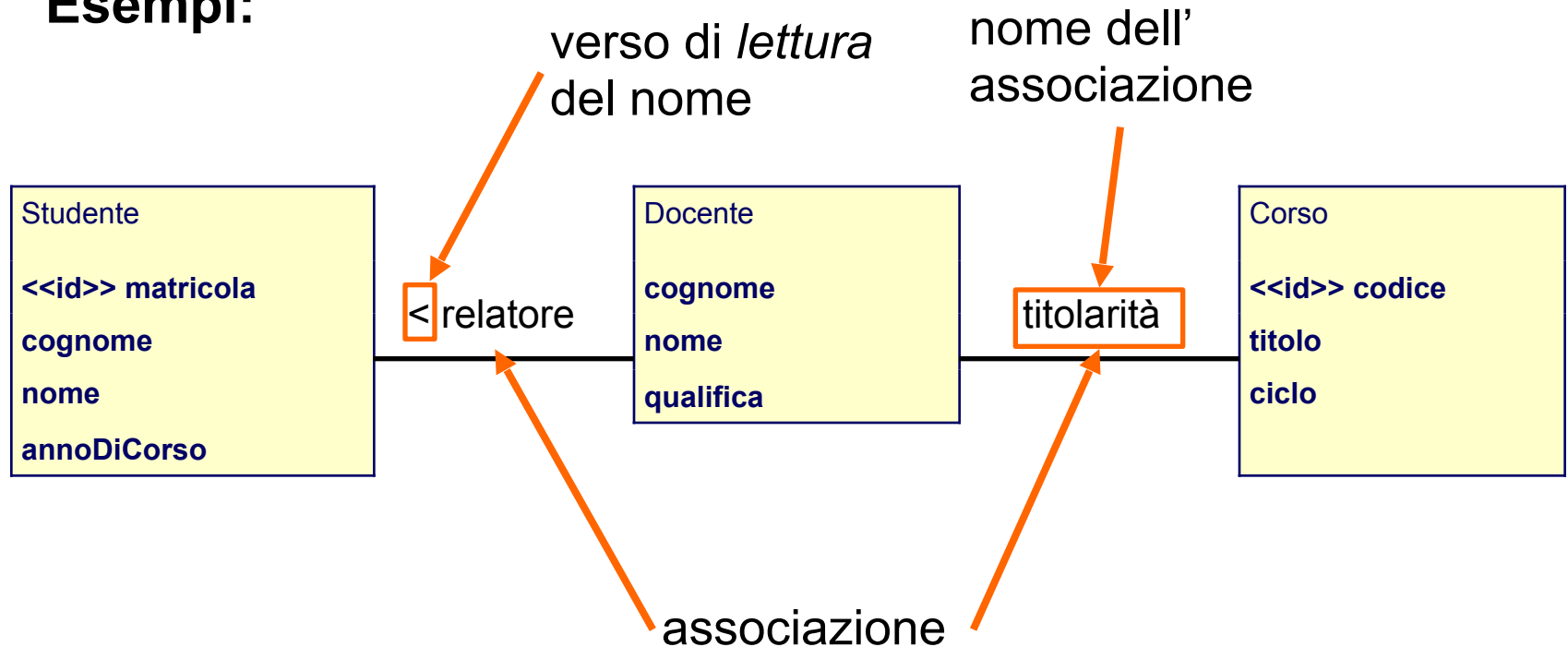
- legame logico rilevante per l'applicazione
- es: superamento tra studente ed esame
- es: titolarità tra docente e corso

◆ Istanza di un'associazione

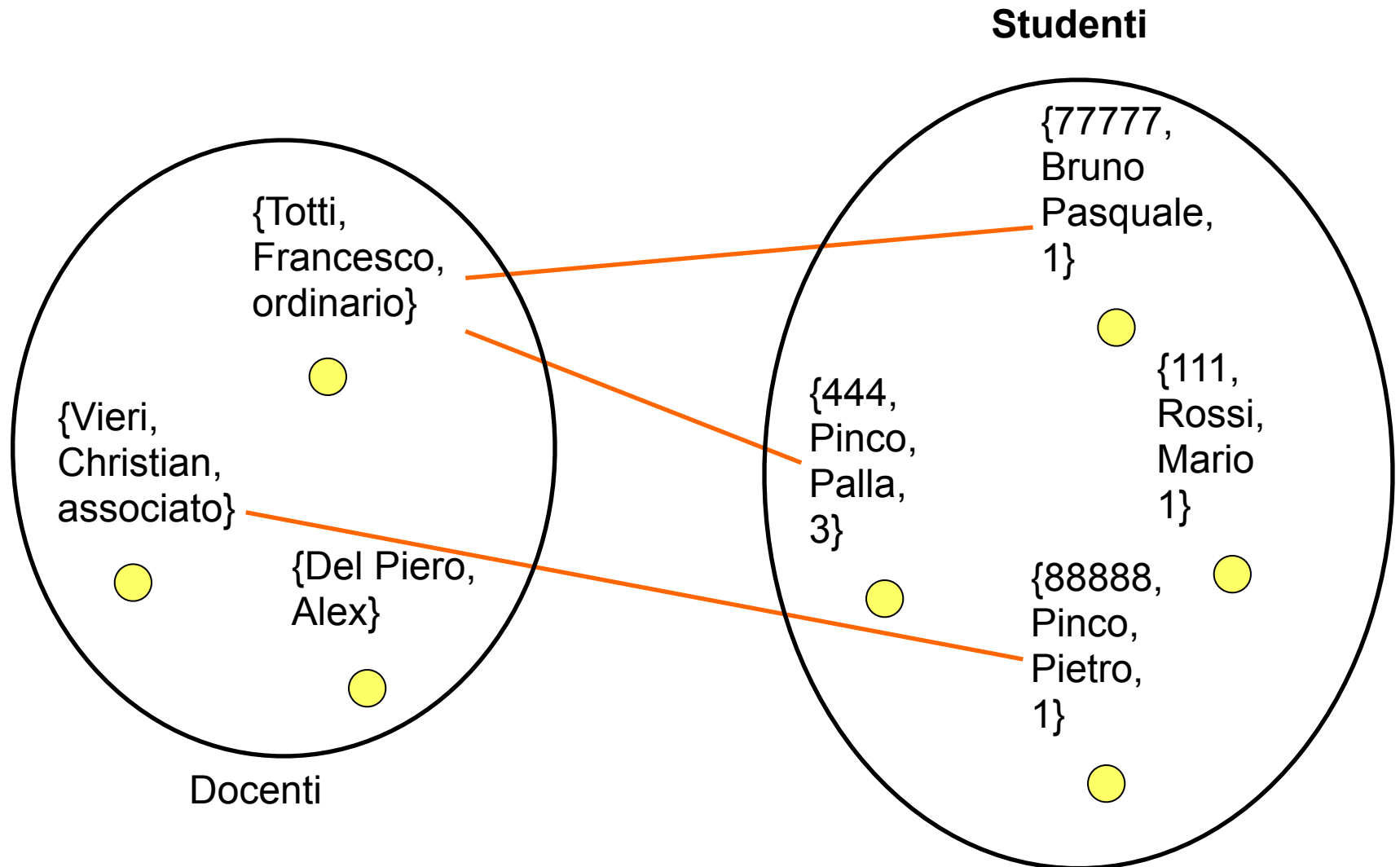
- insieme di archi tra le istanze delle classi coinvolte

Associazioni

◆ Esempi:



Associazioni: Istanze



Cardinalità

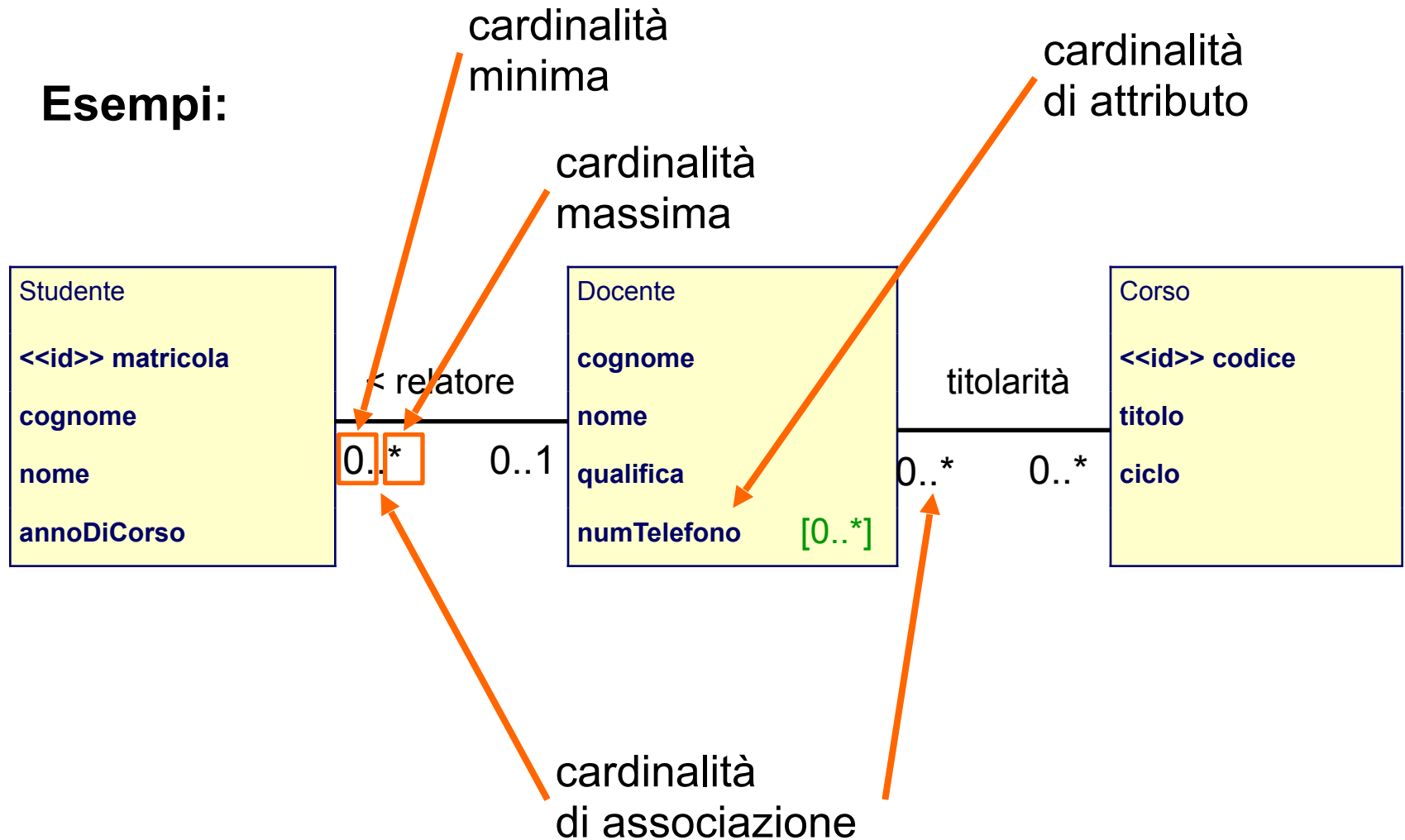
- ◆ **Vincoli sulle associazioni**
 - vincoli sul numero di archi tra gli oggetti
- ◆ **Vincoli sugli attributi**
 - numero di valori dell'attributo
- ◆ **Cardinalità minima**
 - normalmente 0 oppure 1
- ◆ **Cardinalità massima**
 - normalmente 1 oppure * (n) (ma anche 3 o 5)

Cardinalità

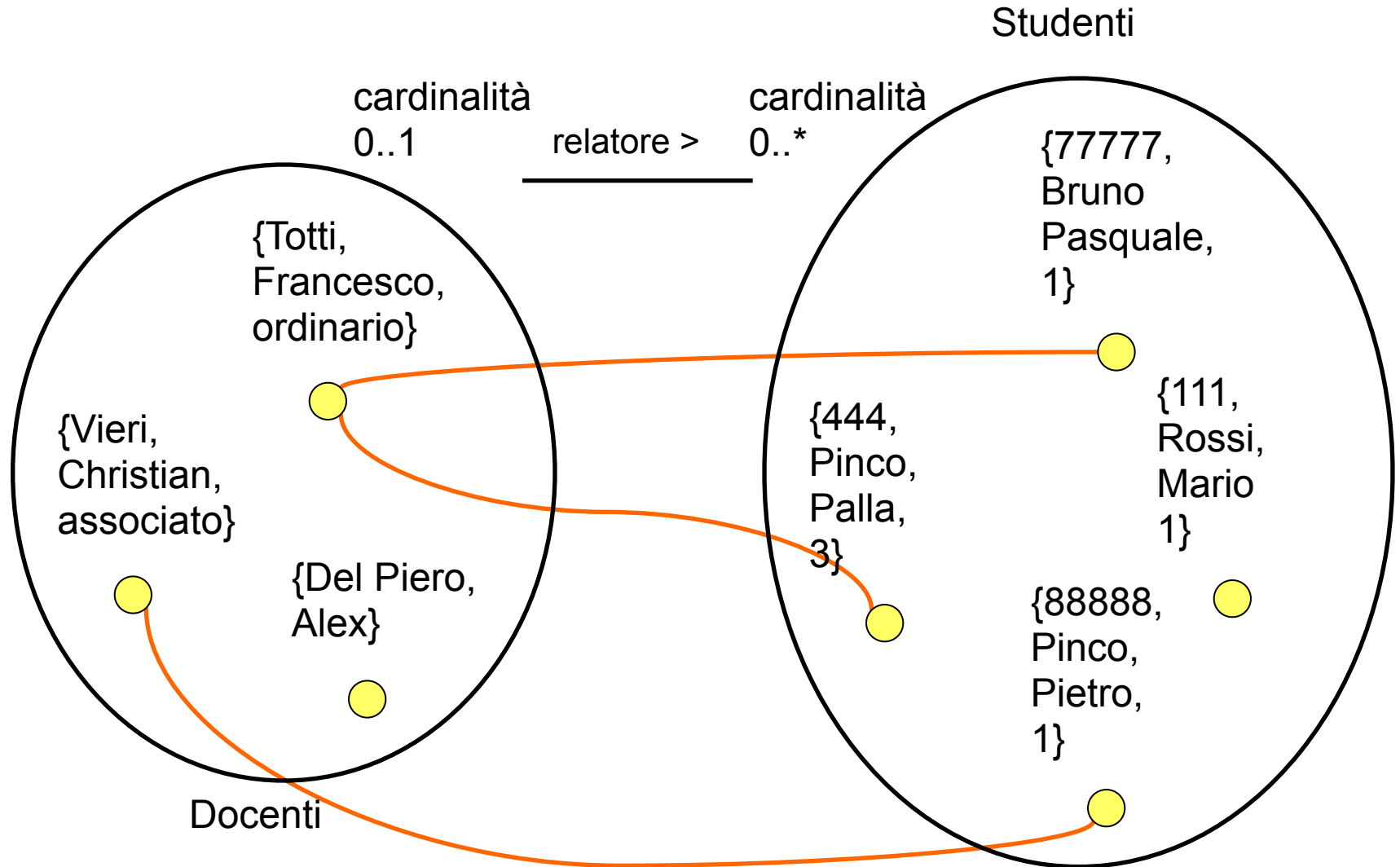
- ◆ **Cardinalità di una associazione**
 - espresse per entrambe le classi
 - quattro cardinalità (ma alcune possono essere omesse)
- ◆ **Cardinalità di una classe in un'associazione**
 - numero di minimo e massimo di oggetti della classe associabili ad un oggetto dell'altra

Cardinalità

◆ Esempi:



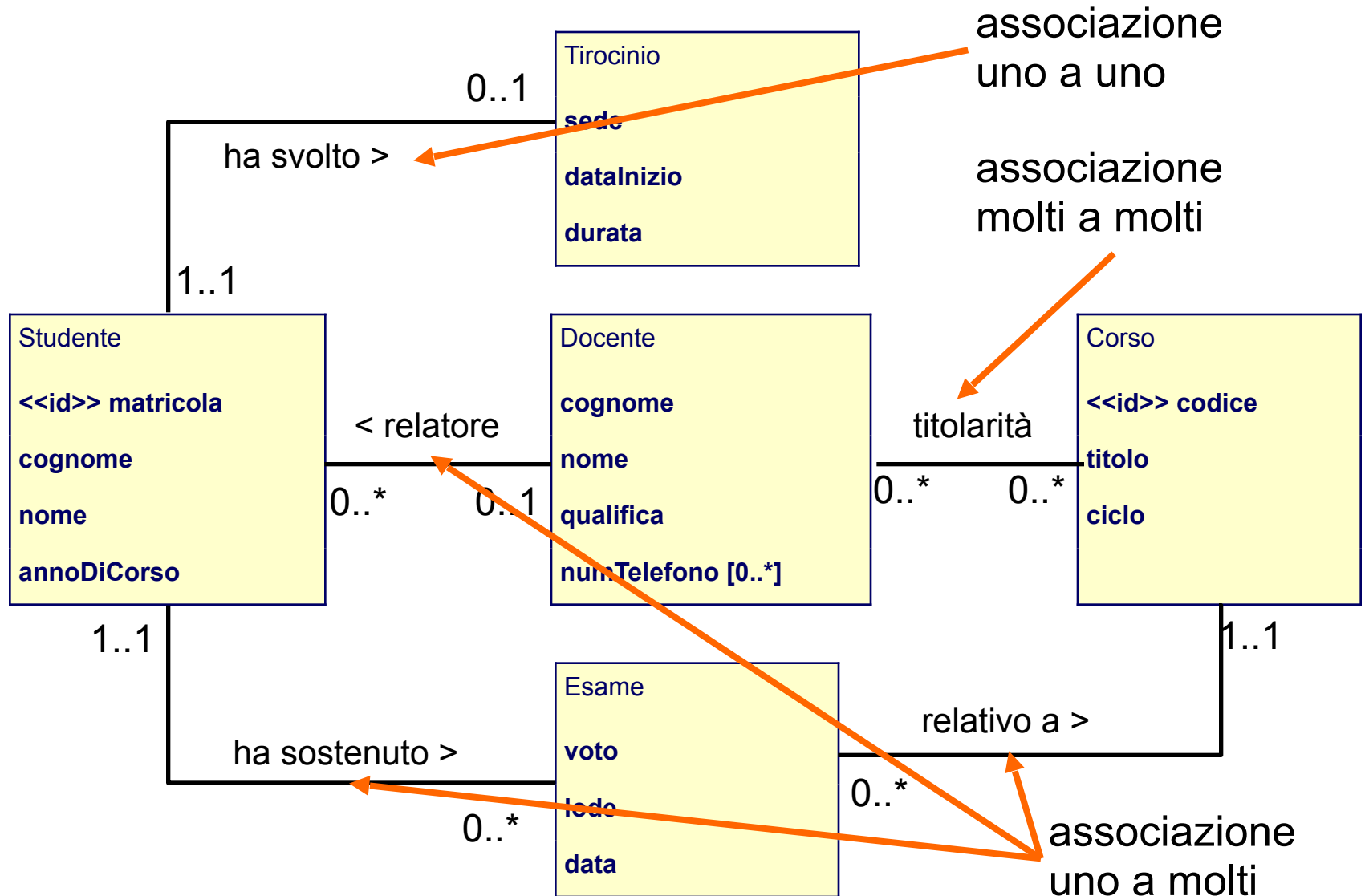
Cardinalità



Cardinalità

- ◆ **Classificazione delle associazioni rispetto alla cardinalità**
 - uno a uno: cardinalità massima 1 da tutte e due le parti
 - uno a molti: cardinalità massima 1 da una parte e * dall'altra
 - molti a molti: cardinalità massima * da tutte e due le parti

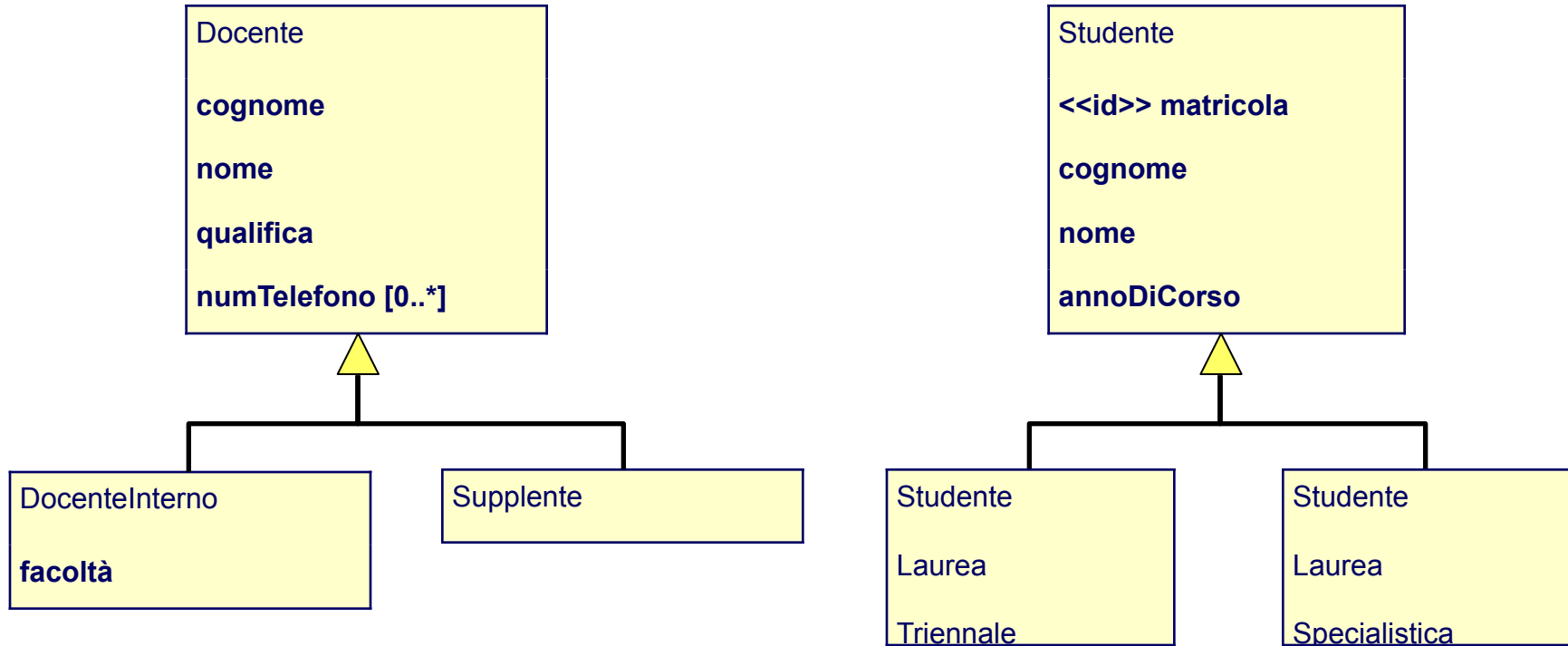
Cardinalità



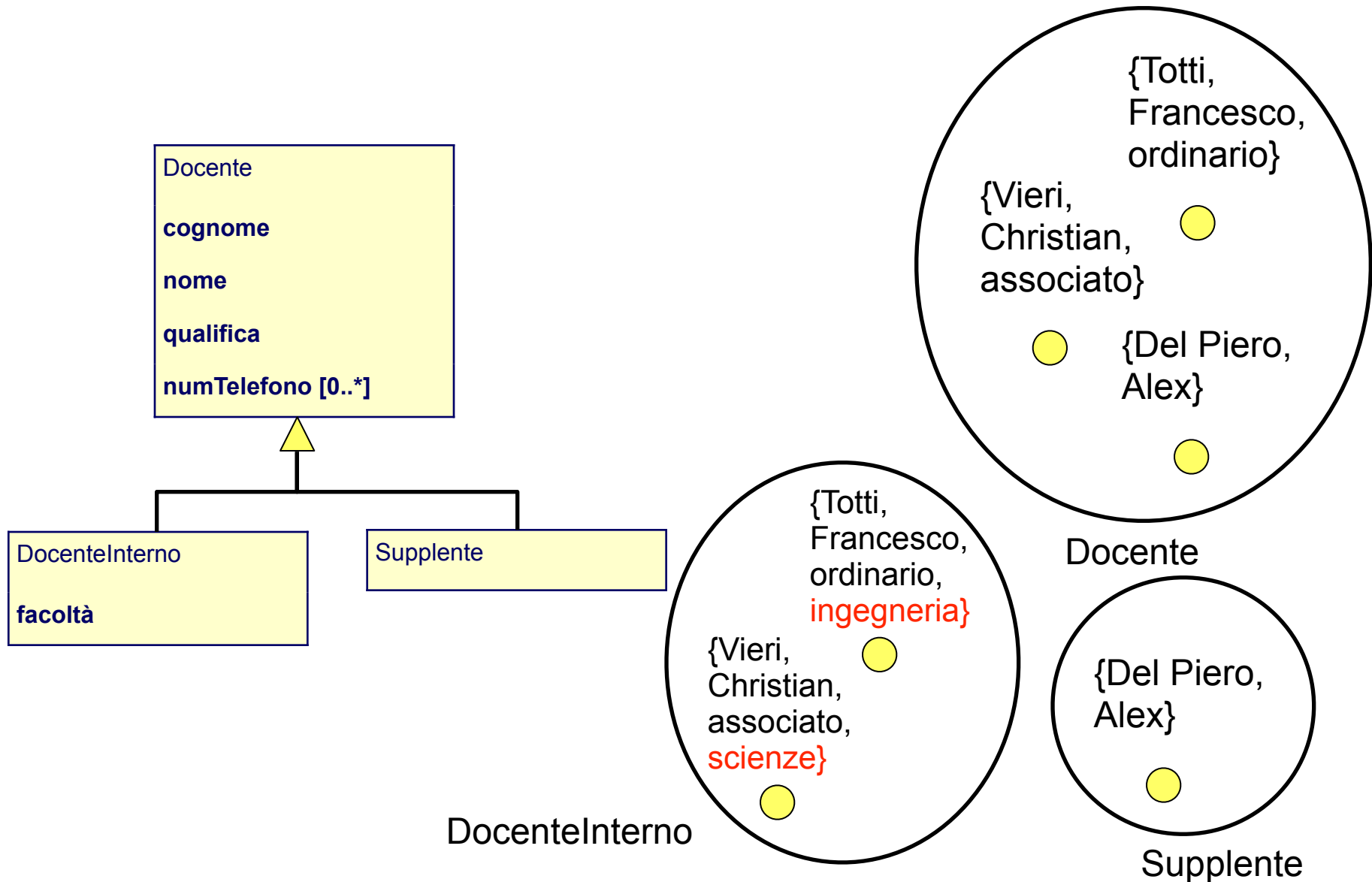
Generalizzazioni

- ◆ **Relazioni tra i concetti rappr. dalle classi**
 - superclasse (padre): concetto più generale
 - sottoclasse (figlio): concetto più specifico
- ◆ **Implicano la semantica dell'ereditarietà**
 - le proprietà e le associazioni del padre sono anche proprietà e associazioni dei figli
 - le istanze dei figli sono anche istanze del padre

Generalizzazione



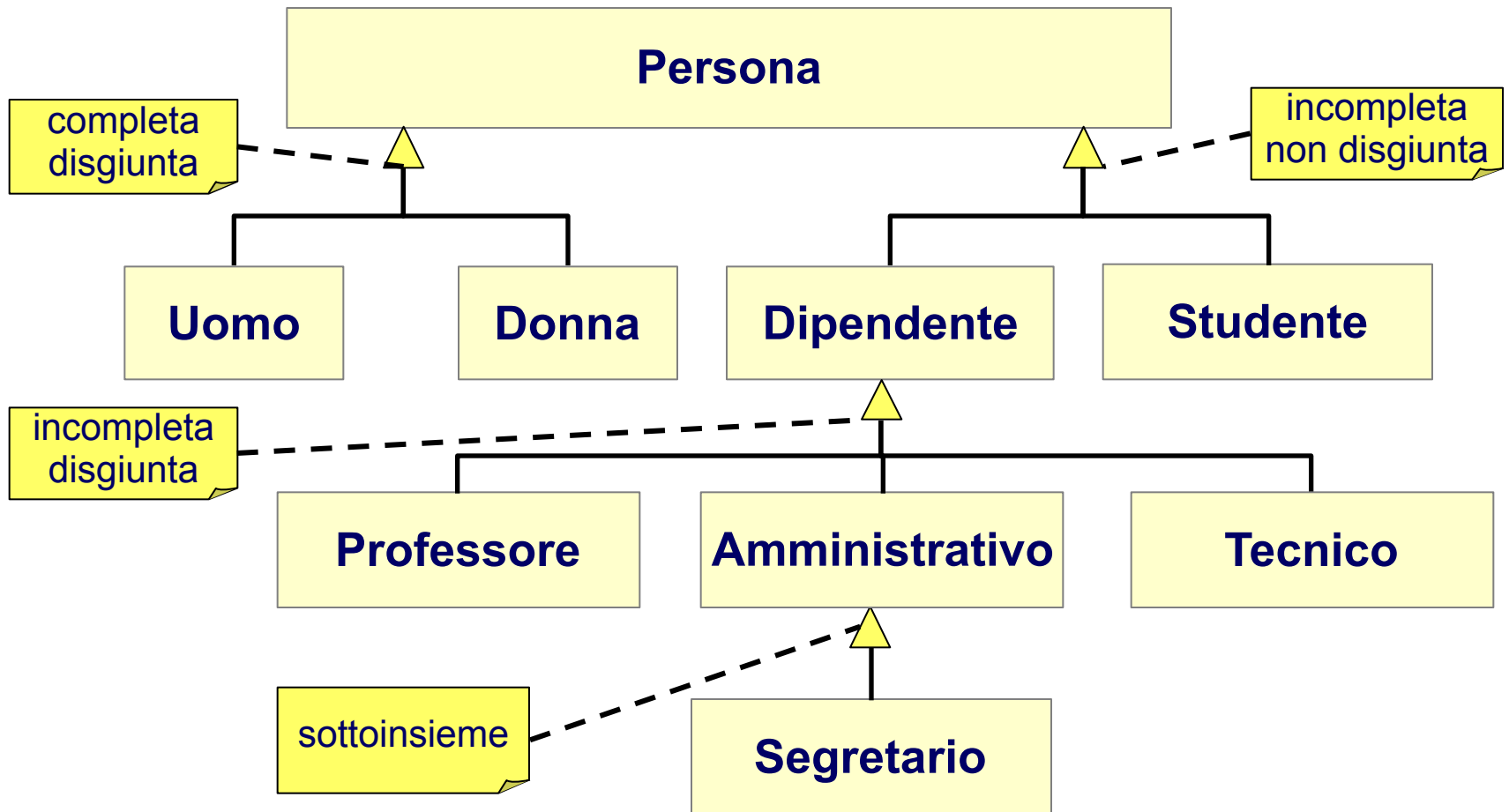
Generalizzazione: Istanze



Generalizzazioni

- ◆ **Relazioni di ereditarietà tra i concetti**
 - consentono di descrivere gerarchie
- ◆ **Possono essere di vario tipo**
 - completa o incompleta: tutte le istanze della superclasse sono classificate o meno nelle sottoclassi
 - disgiunte o non disgiunte: le sottoclassi hanno intersezione vuota o meno
 - sottoinsiemi: un'unica sottoclasse

Generalizzazioni

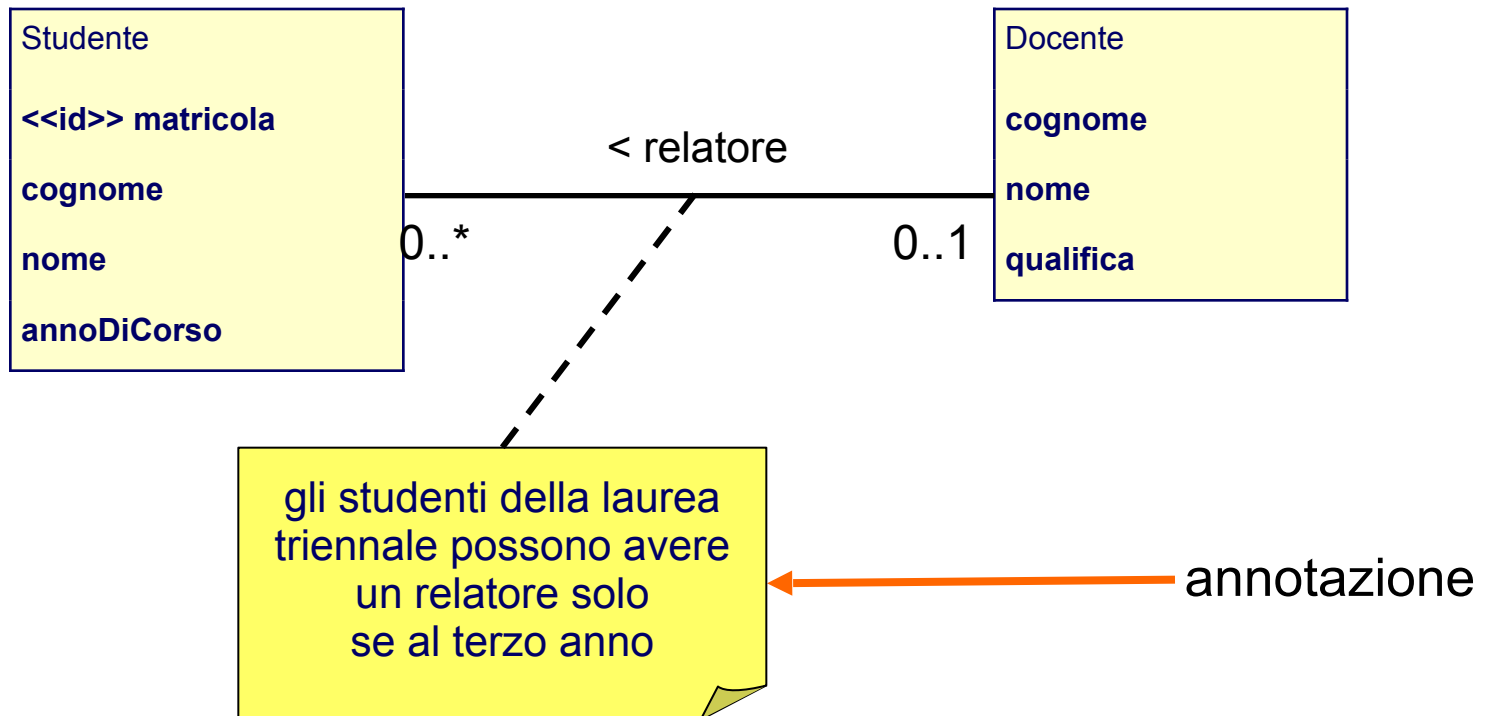


Il Diagramma Completo

- ◆ **Il diagramma completo contiene tutti i costrutti visti**
- ◆ **E' possibile aggiungere annotazioni**
 - utili per commentare i costrutti
 - e per esprimere vincoli altrimenti non esprimibili; es: gli studenti della laurea tr. possono chiedere un relatore solo se sono iscritti al terzo anno

Il Diagramma Completo

◆ Esempi:



Ancora sulle Classi

◆ Identificatori

- non tutte le classi hanno identificatori espliciti
- ma gli oggetti della classe sono comunque distinti (es: due mele nella classe “Mele”)
- nell’applicazione, tutti gli oggetti hanno un identificatore implicito (nascosto): OID
- gli identificatori espliciti (es: matricola, codice fiscale, targa) esistono per la catalogazione
- sono rilevanti per la base di dati

Classi

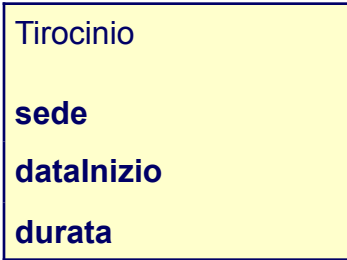
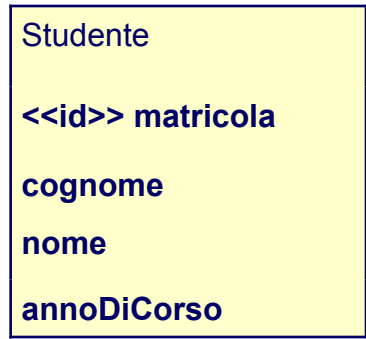
◆ **Identificatori esterni**

- alcuni oggetti sono identificabili sulla base degli oggetti a cui sono associati
- esempio: un tirocinio può essere identificato con lo studente che lo ha sostenuto
- per identificarlo è necessario attraversare un arco di una associazione
- la cardinalità all'altro estremo deve essere 1

Classi

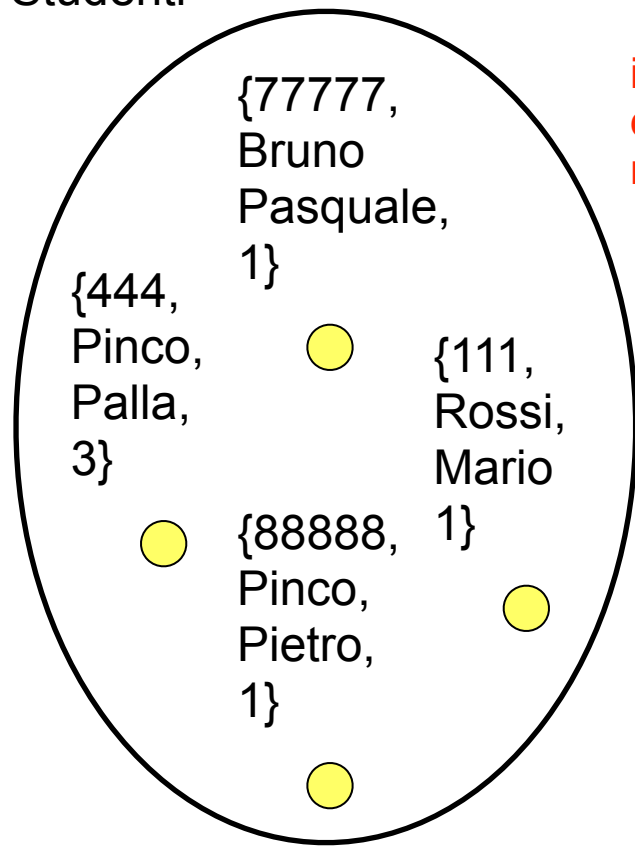
- ◆ **Identificatore esterno**
 - indicato con un'annotazione in cui viene indicata l'associazione da attraversare
- ◆ **Esempio: tirocinio universitario**



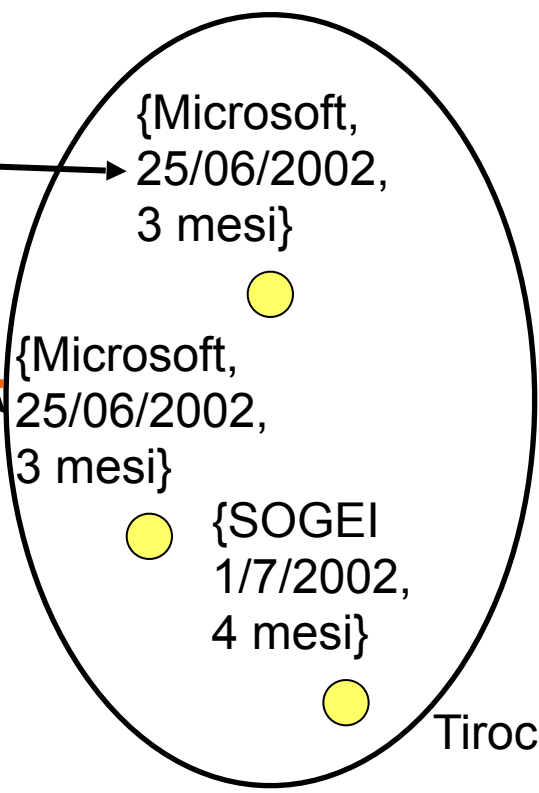


identificatore esterno "ha svolto"

Studenti



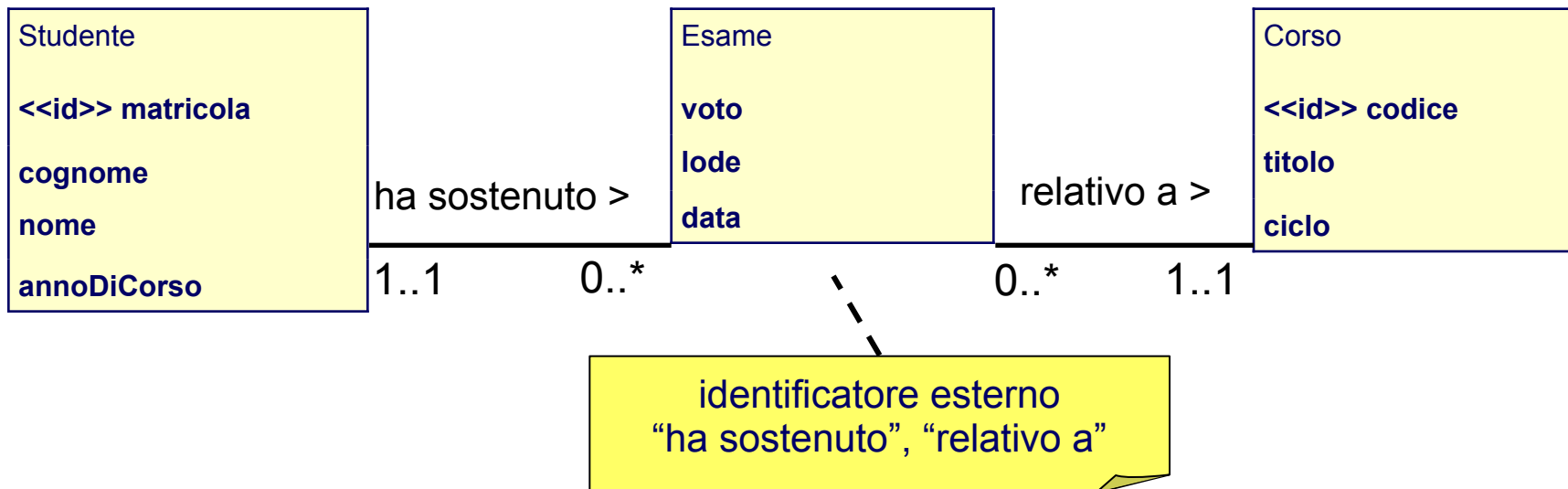
i due tirocini sono distinguibili usando la matricola dello studente



Tirocini

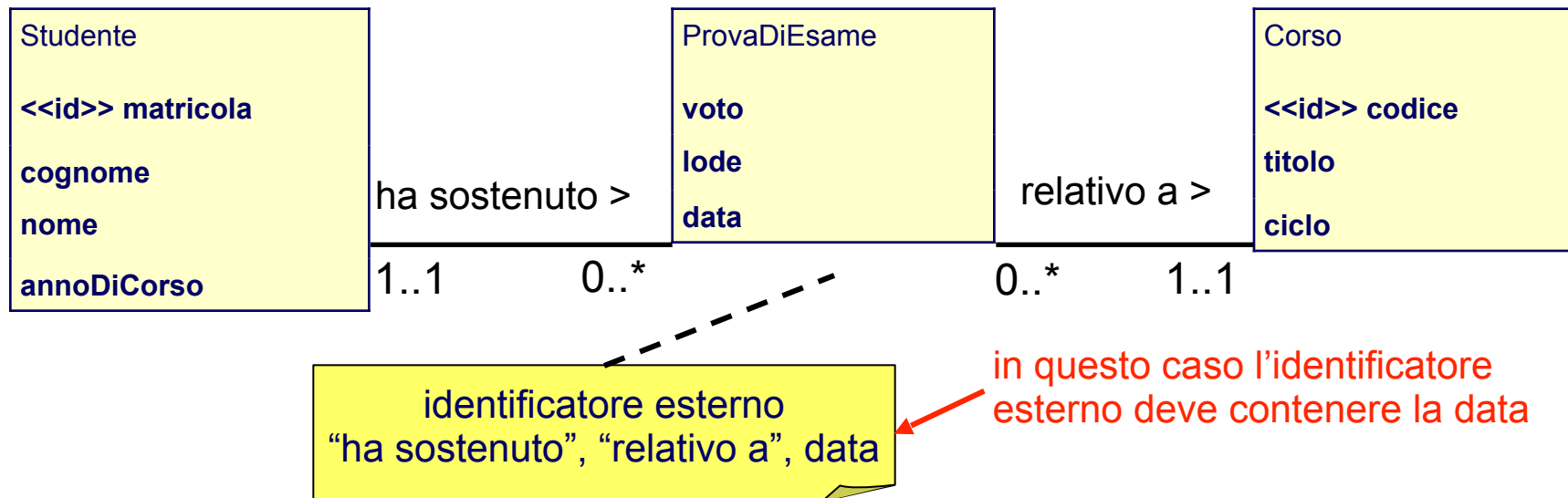
Classi

- ◆ **Identificatore esterno complesso**
 - una o più associazioni, uno o più attributi int.
- ◆ **Esempio: esame universitario**



Classi

- ◆ Nel caso volessimo tenere traccia anche degli esami non andati a buon fine
 - lo studente può sostenere più volte l'esame



Associazioni

- ◆ **Relazioni logiche tra le classi**
- ◆ **Istanza dell'associazione**
 - arco che collega due oggetti delle classi
- ◆ **Cardinalità**
 - vincoli sul numero di archi per un oggetto
- ◆ **Classificazione rispetto alle cardinalità**
 - uno a uno: cardinalità massime pari a 1
 - uno a molti: cardinalità massime 1 e *
 - molti a molti: cardinalità massime * e *

Associazioni

◆ Normalmente

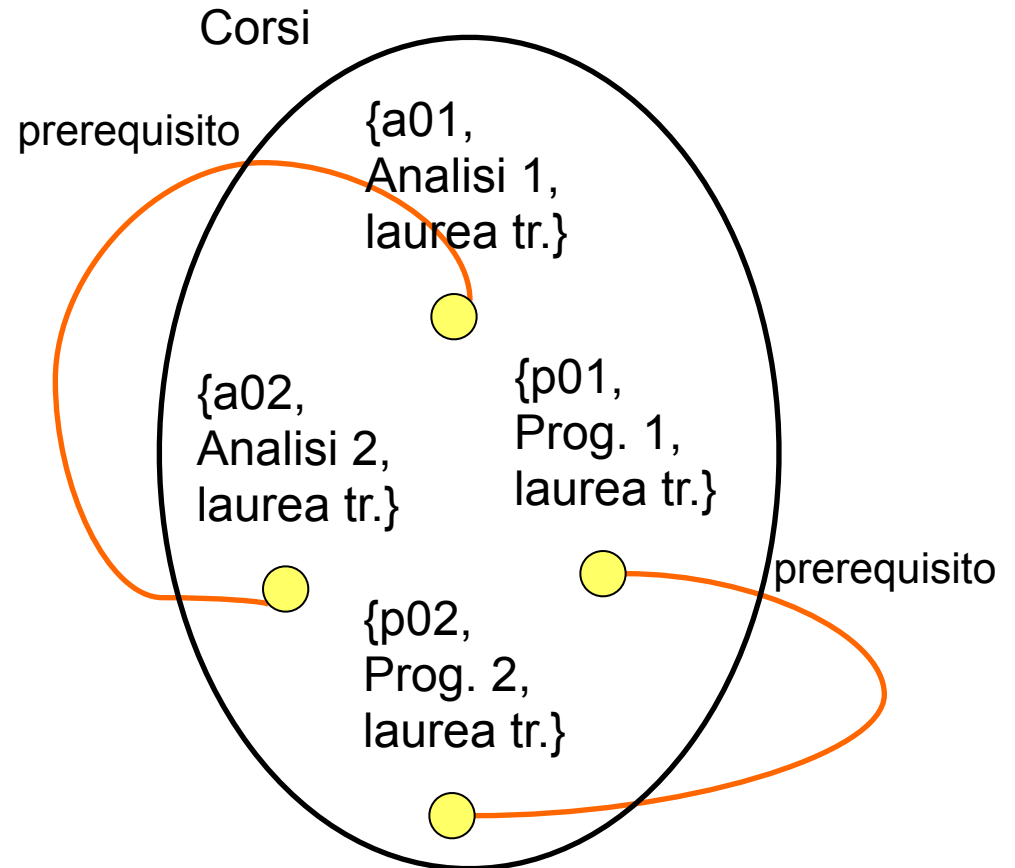
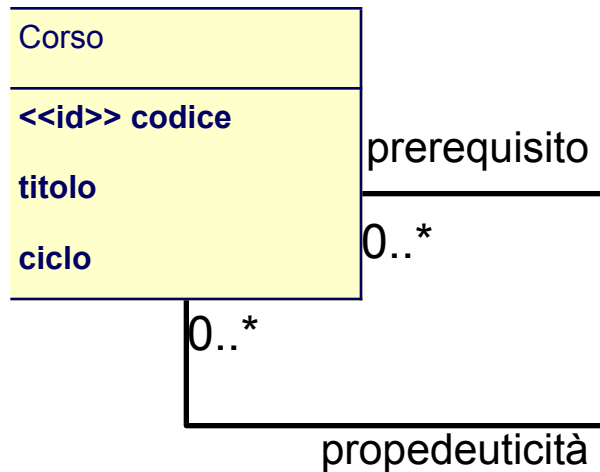
- associazioni binarie

◆ Ma ci sono altre forme

- associazioni ricorsive: associazione tra oggetti della stessa classe
- associazioni n-arie (poco usate): associazioni che coinvolgono oggetti di n classi (3 o più)

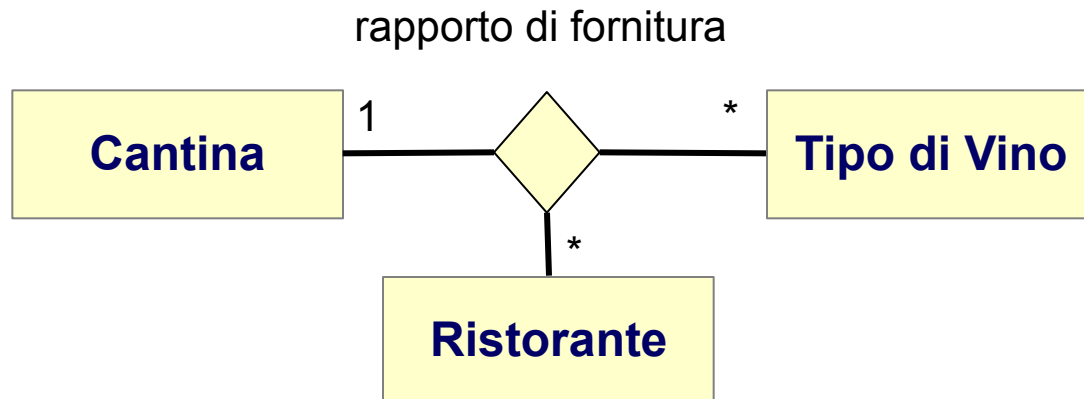
Associazioni Ricorsive

◆ Esempio

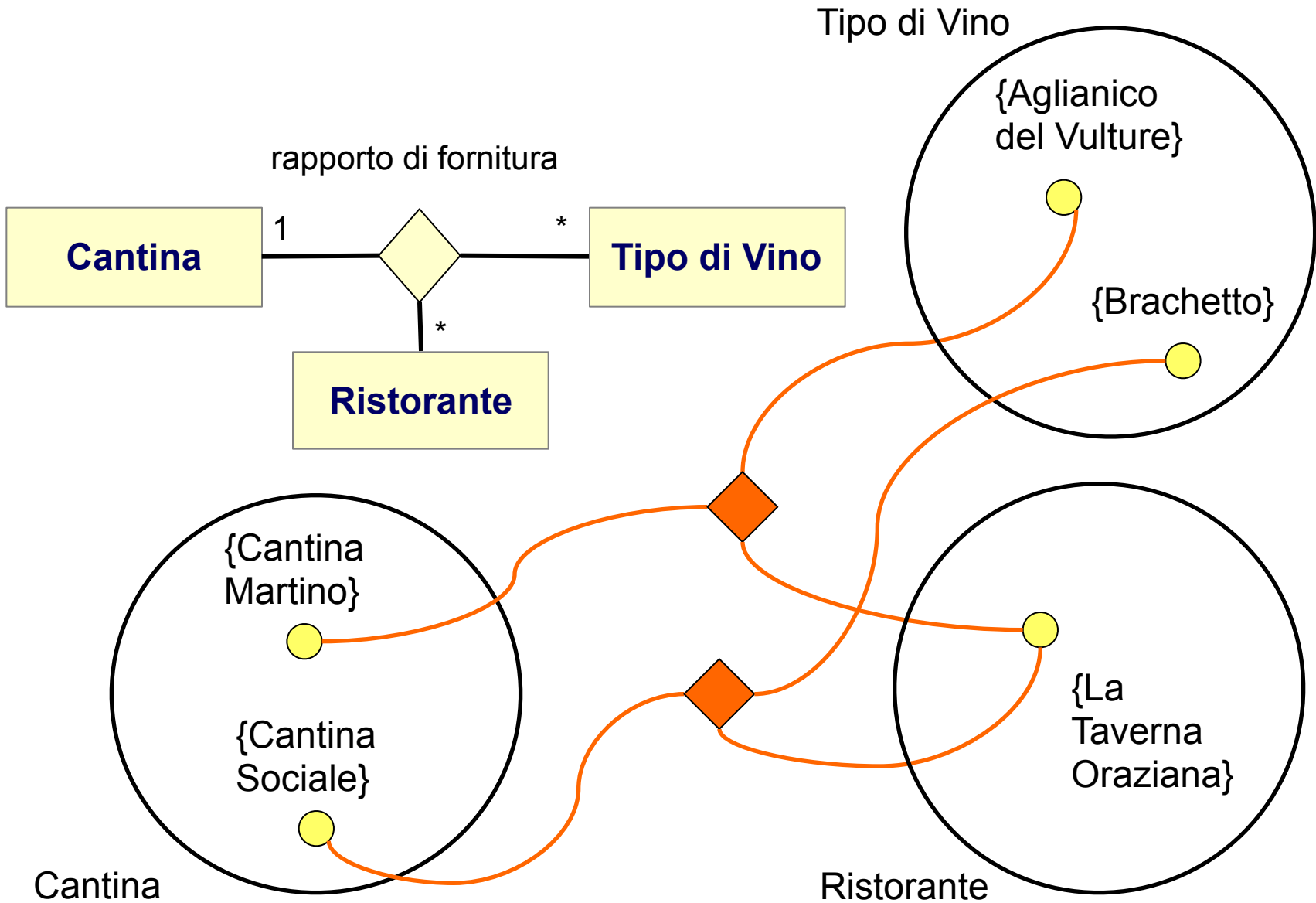


Associazioni Ternarie

- ◆ **Si utilizza il simbolo del rombo**
 - associano elementi di tre classi
- ◆ **Esempio: forniture di vini ai ristoranti**

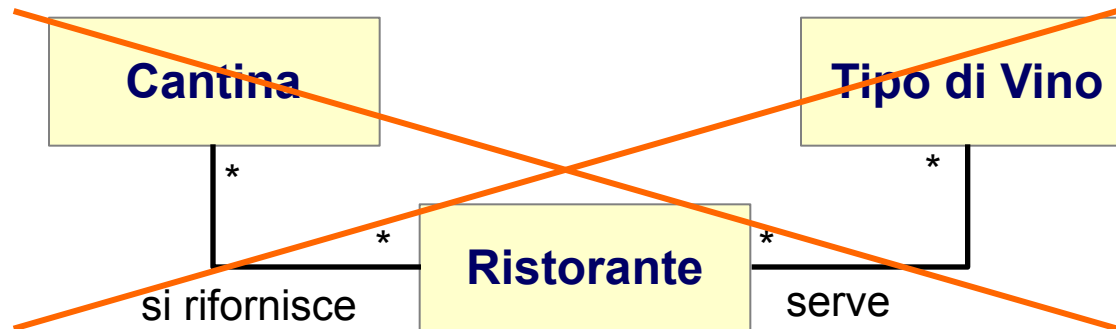


Associazioni Ternarie

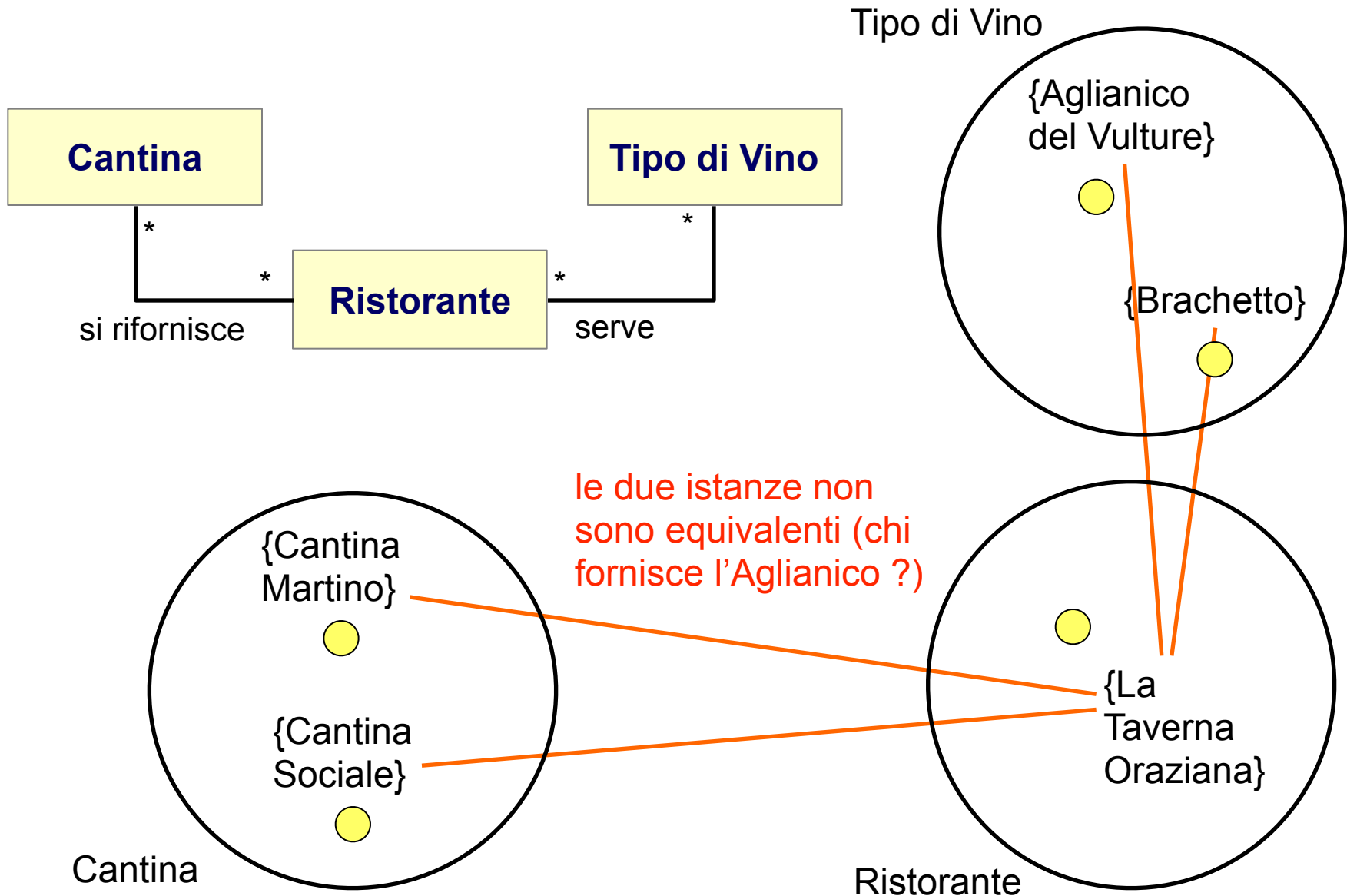


Associazioni Ternarie

- ◆ **Sono di difficile gestione**
 - sarebbe opportuno sostituirle con più binarie
- ◆ **Attenzione agli errori. Esempio:**
 - Ristorante si rifornisce da Cantina
 - Ristorante serve Tipo di Vino

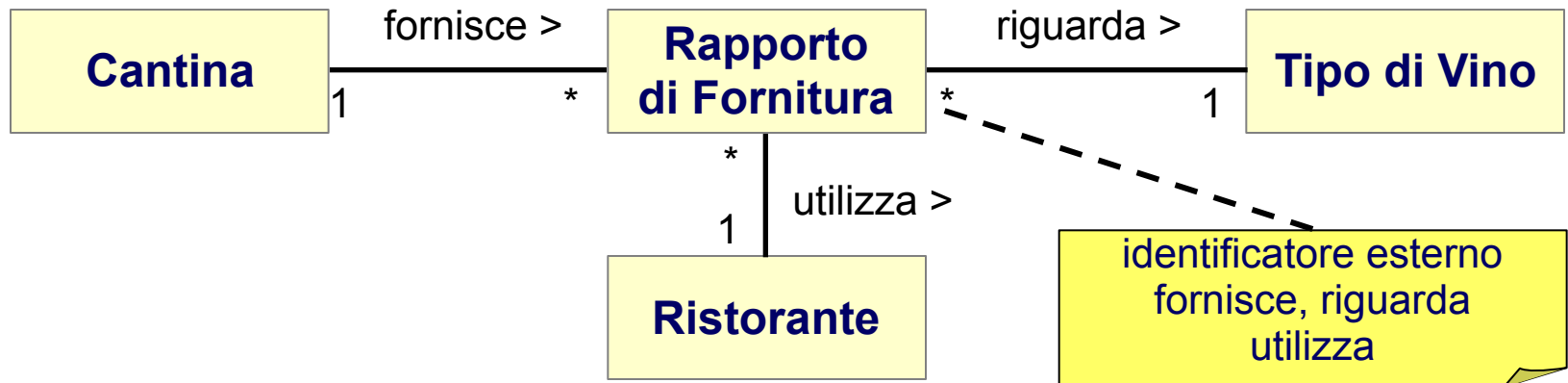


Associazioni Ternarie



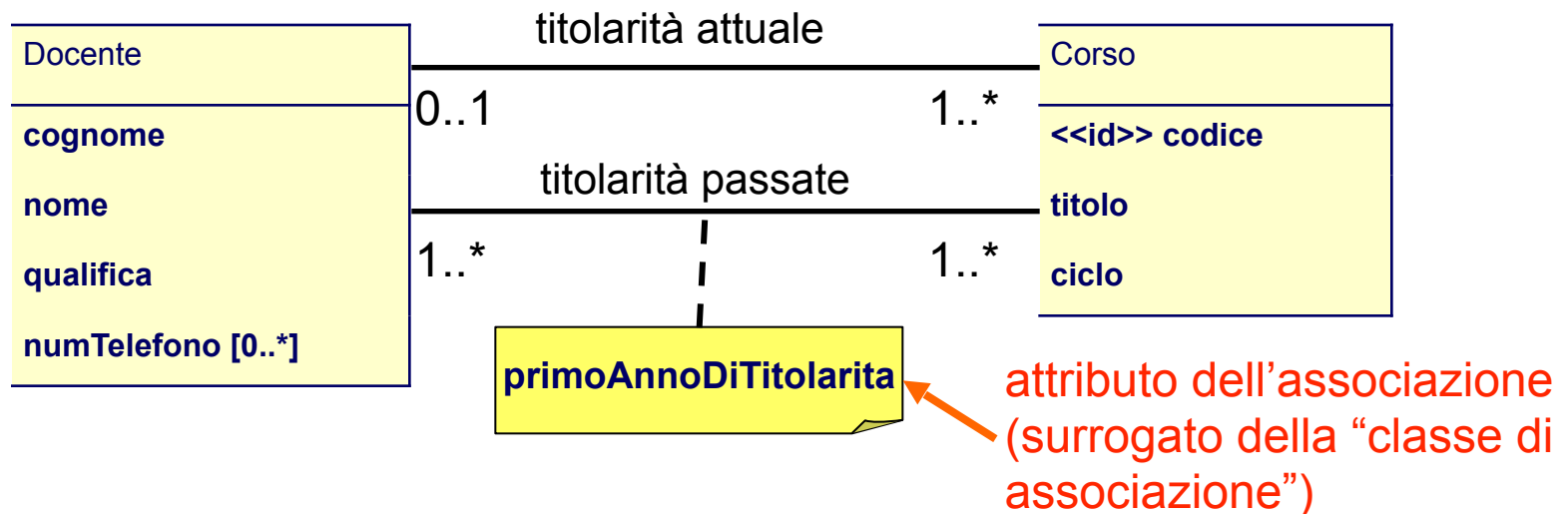
Associazioni Ternarie

- ◆ **Possono essere sostituite da**
 - una classe aggiuntiva
 - tre associazioni binarie



Attributi di un'Associazione

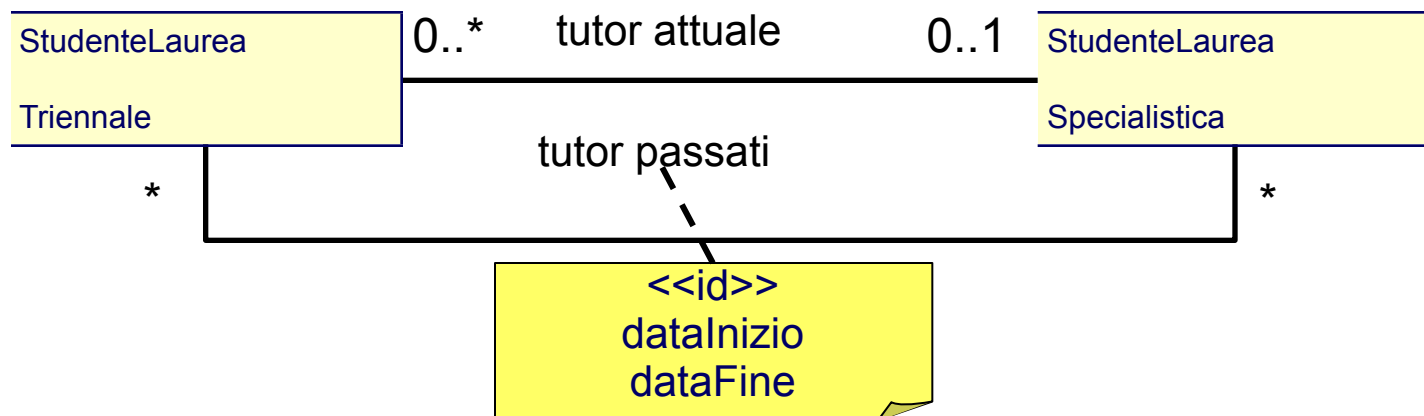
- ◆ Le associazioni possono avere attributi
- ◆ Esempio: corsi tenuti nel passato
 - tengo traccia del primo anno di titolarità



Attributi di un'Associazione

◆ Identificazione dell'associazione

- gli attributi possono essere identificatori per le istanze dell'associazione
- tra gli stessi nodi possono esserci più archi
- è consentito solo per associazioni molti a m.



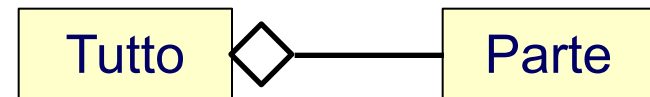
Altre forme di Associazione – Cenni

◆ Esistono altre forme di associazione

- aggregazione e composizione

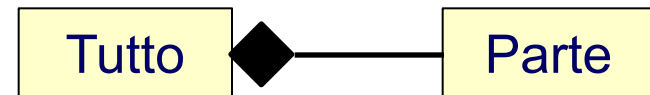
◆ Aggregazione

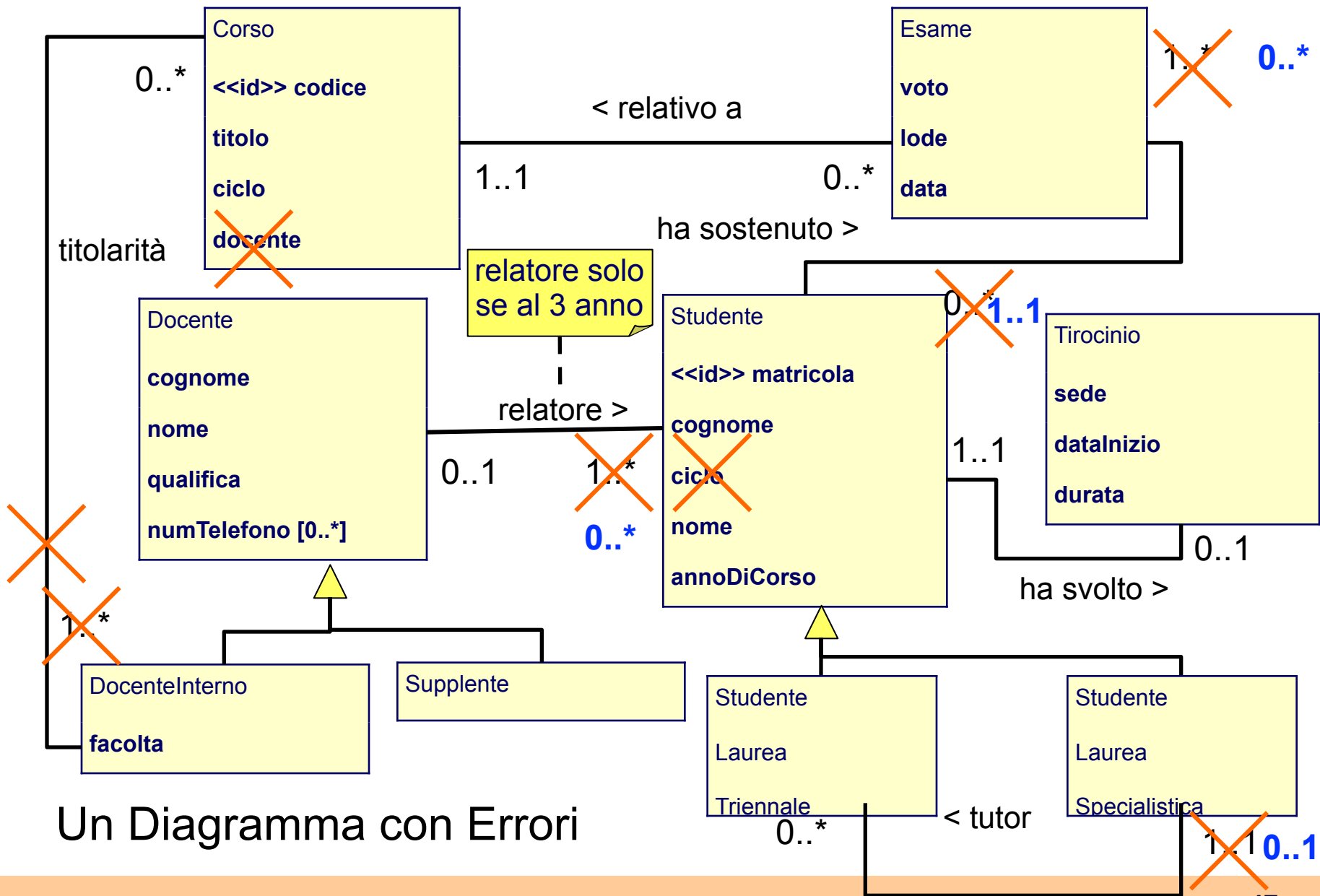
- associazione tra un tutto e le sue parti
- simbolo rombo vuoto dalla parte del tutto
- esempio: dipartimento e impiegati

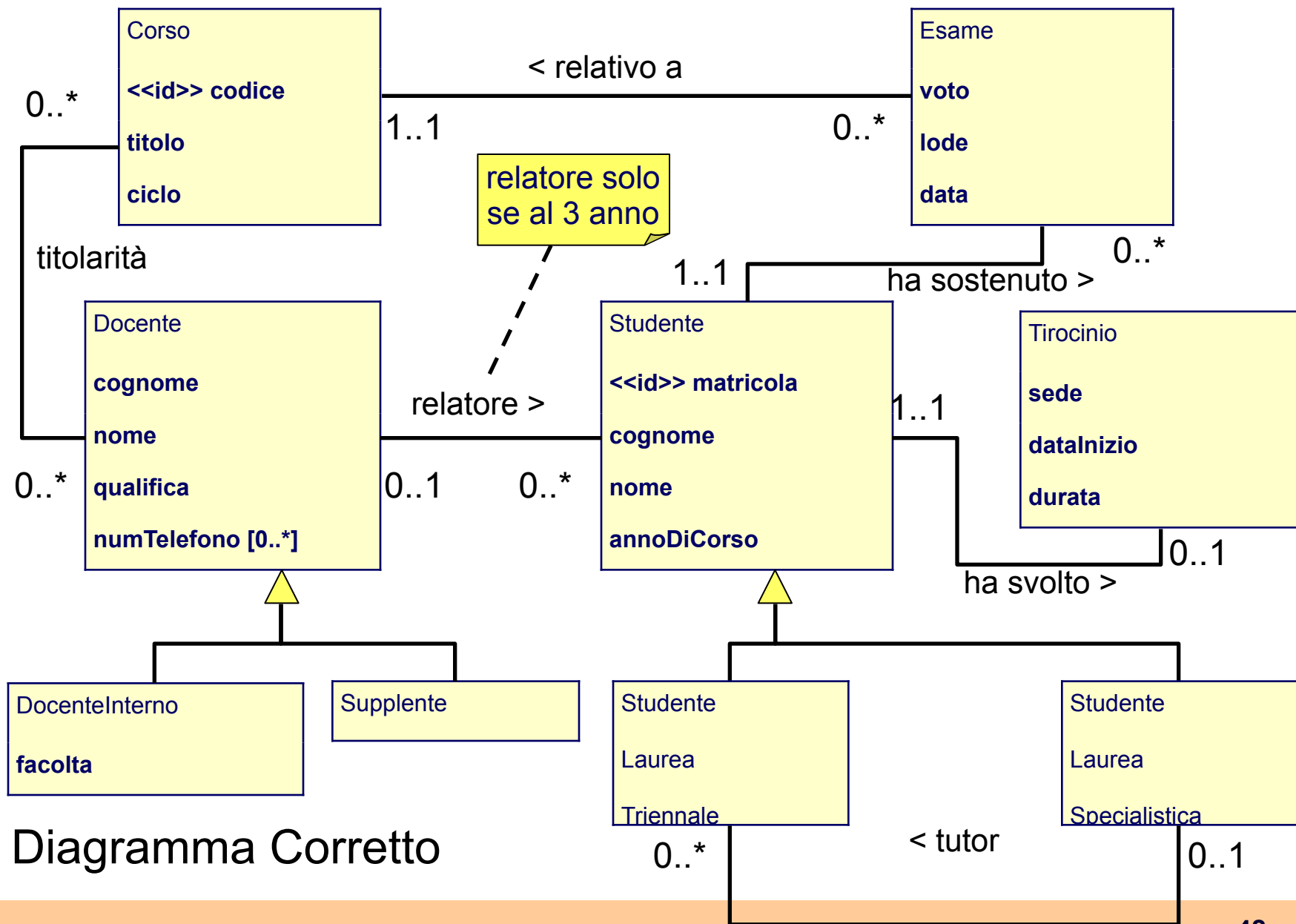


◆ Composizione

- aggregazione e le parti non esistono senza il tutto
- simbolo rombo pieno dalla parte del tutto
- esempio: libro e capitoli







Linee Guida per la Modellazione

◆ Punto di partenza

- requisiti
- l'attività di modellazione concettuale e l'analisi dei requisiti procedono parallelamente

◆ Processo iterativo

- prototipo del modello
- approfondimento dei requisiti
- nuovo prototipo ecc.

Linee Guida per la Modellazione

- ◆ **Qualità dello schema concettuale**
 - **correttezza**: deve rispecchiare correttamente le specifiche (si verifica disegnando le istanze)
 - **completezza**: deve descrivere tutti gli aspetti delle specifiche
 - **leggibilità**: deve essere facilmente interpretabile (notazione per i nomi)
 - **minimalità**: deve evitare ridondanze

Linee Guida per la Modellazione

◆ Per cominciare

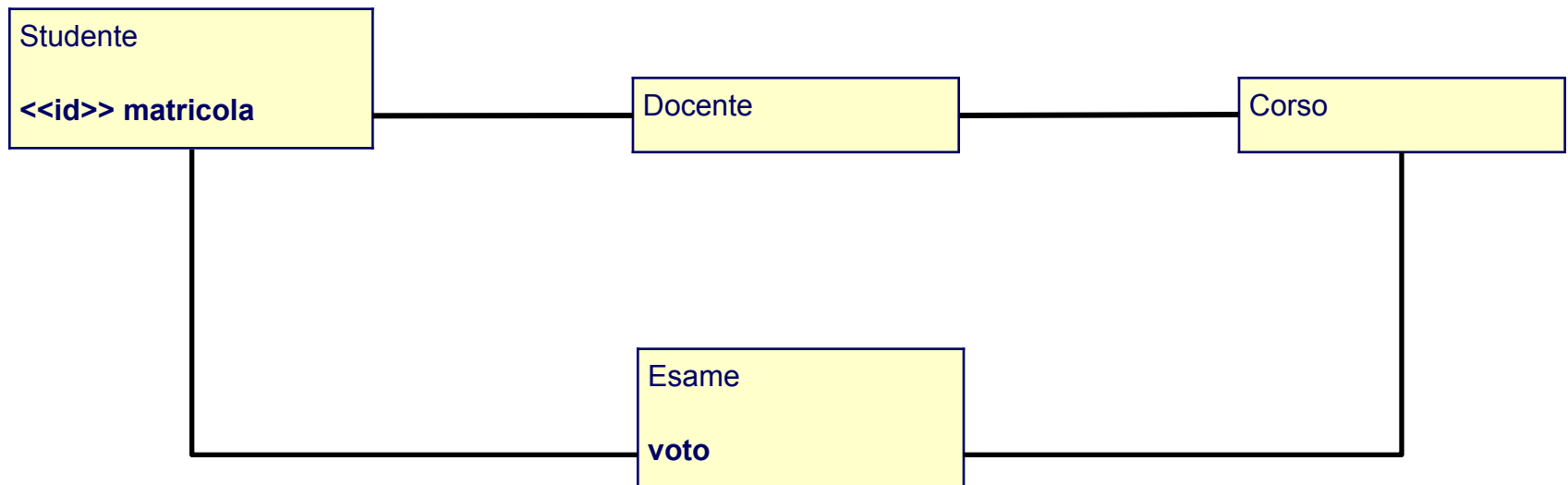
- divisione dei requisiti in sezioni
- ogni sezione rappresenta la specifica di una parte dell'applicazione

◆ Schema scheletro complessivo

- schema delle classi corrispondenti ai concetti fondamentali (es: docente, studente, esame, corso)

Schema Scheletro

◆ Esempio:



Linee Guida per la Modellazione

- ◆ **Approfondimento delle sezioni**
 - per ciascuna sezione viene prodotto uno schema di dettaglio
 - esempio: gerarchia di docenti
 - esempio: gerarchia di studenti e tutorato
 - esempio: relazione tra studenti ed esami

- ◆ **Integrazione degli schemi**
 - gli schemi delle sezioni vengono integrati sulla base dello schema scheletro

Modello Entità-Relazione

- ◆ **Sintassi grafica alternativa per il modello concettuale**
 - precedente ad UML
 - tradizionalmente usato per l'analisi e la modellazione concettuale delle basi di dati
 - UML è in parte ispirato all'ER
 - meno generale del diagramma delle classi
 - non include funzionalità dinamiche

Modello Entità-Relazione

- ◆ **Per la modellazione concettuale**
 - sostanzialmente la semantica è equivalente
- ◆ **Costrutti dei due modelli**

Diagramma delle classi	Modello ER
classe	entità
attributo	attributo
associazione	relazione
cardinalità	cardinalità
generalizzazione	generalizzazione

Modello Entità-Relazione: Sintassi

- ◆ **Entità: rettangolo**
- ◆ **Attributi: due sintassi**
 - linee esterne all'entità
 - sintassi equiv. a quella UML
- ◆ **Relazione: rombo**
- ◆ **Cardinalità: posizione invertita**
- ◆ **Generalizzazione: freccia**

