Basi di Dati

Definizione del Modello Concettuale dei Dati:

Concetti Fondamentali

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” (verranno introdotti più avanti)

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 e oggetti

- **Un *oggetto* è una "cosa" del mondo reale che è caratterizzato da alcuni attributi in corrispondenza dei quali assume valori.**

- **Una ennupla è un oggetto?**
  - Un oggetto è qualcosa di più: ha una identità (*identificatore*) a prescindere dall'esistenza o meno di una chiave ed ha anche un comportamento, cioè dei metodi che gli permettono di modificare i valori dei suoi attributi e di interagire con altri oggetti.

- **Ma per questo corso non ci saranno grosse differenze tra tuple ed oggetti.**

Classificazione

- **Una *classe* è la definizione di oggetti simili, cioè che hanno gli stessi attributi anche se con valori differenti.**
  - Uno schema di relazione è una classe semplificata, senza descrizione dei comportamenti.

- **Nella modellazione della conoscenza un meccanismo fondamentale di astrazione è la *classificazione*:**
  - passare dalla osservazione di oggetti alla definizione di una classe che descrive le proprietà comuni (attributi) e i domini dei valori che possono essere assunti.
  - Cioè la classificazione è la capacità di descrivere oggetti mettendo in risalto le proprietà comuni e ignorando le altre.

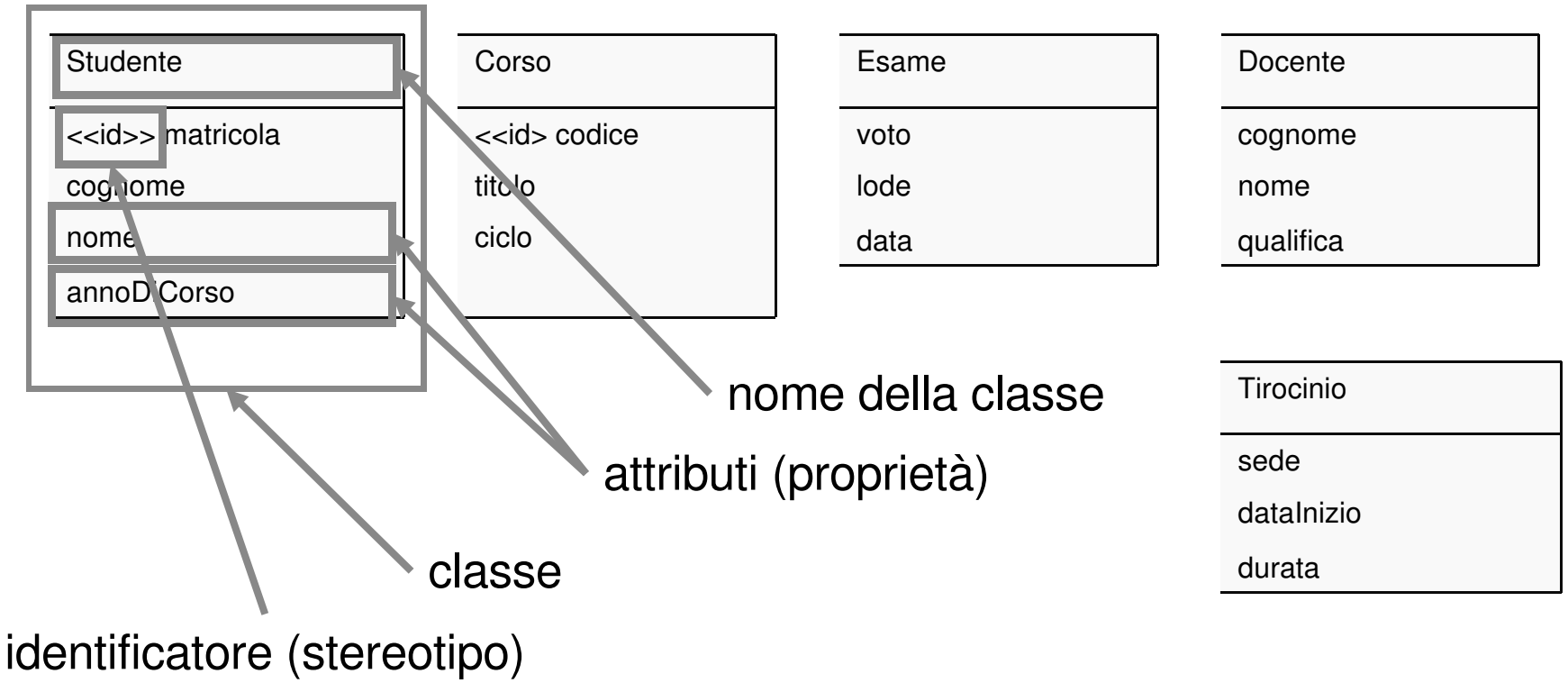
- **Il meccanismo inverso è l'*instanziazione***

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

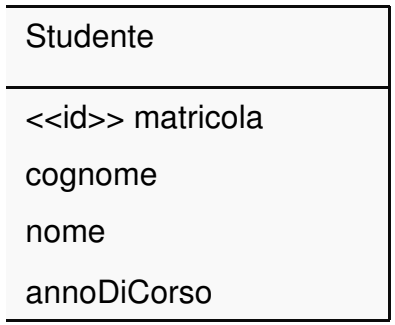
Classi

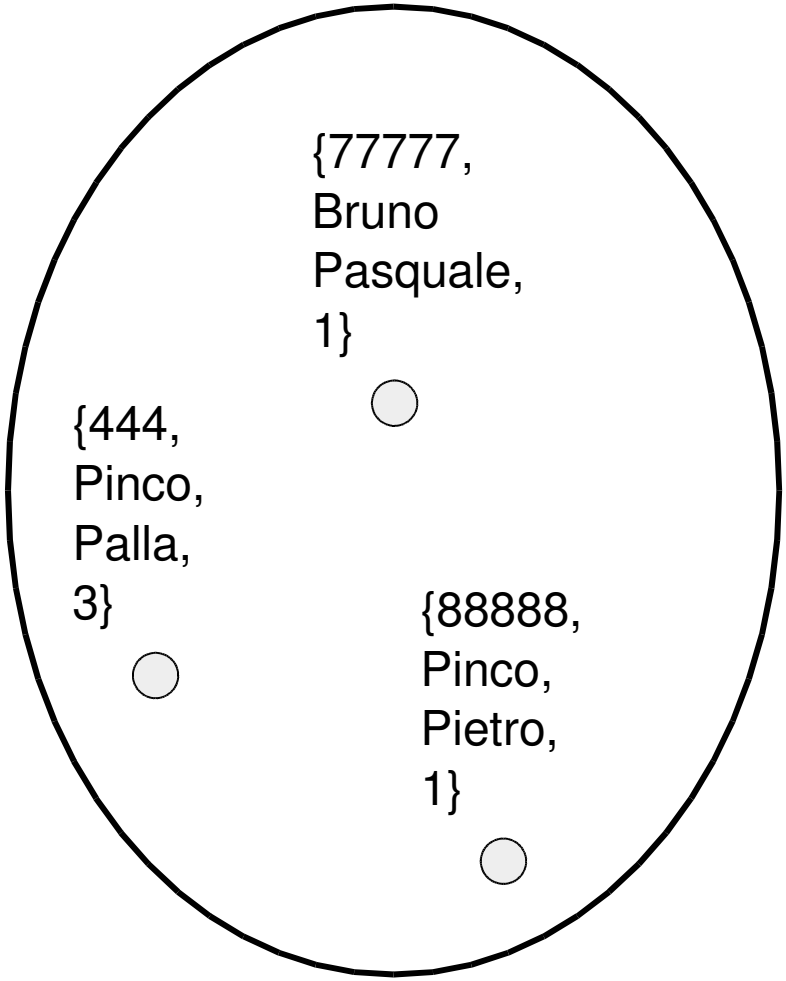
- **Esempi:**



Classi

- **Istanze delle classi**





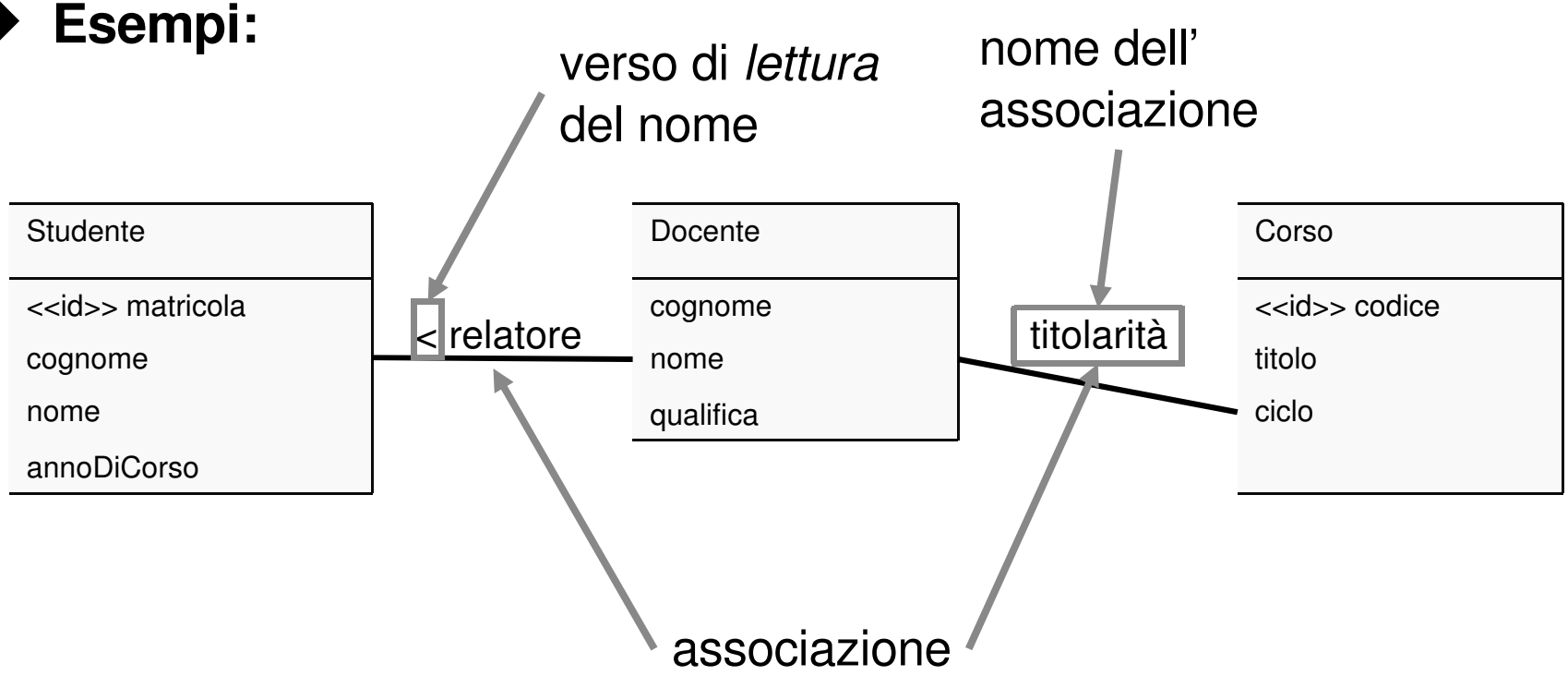
Associazioni

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

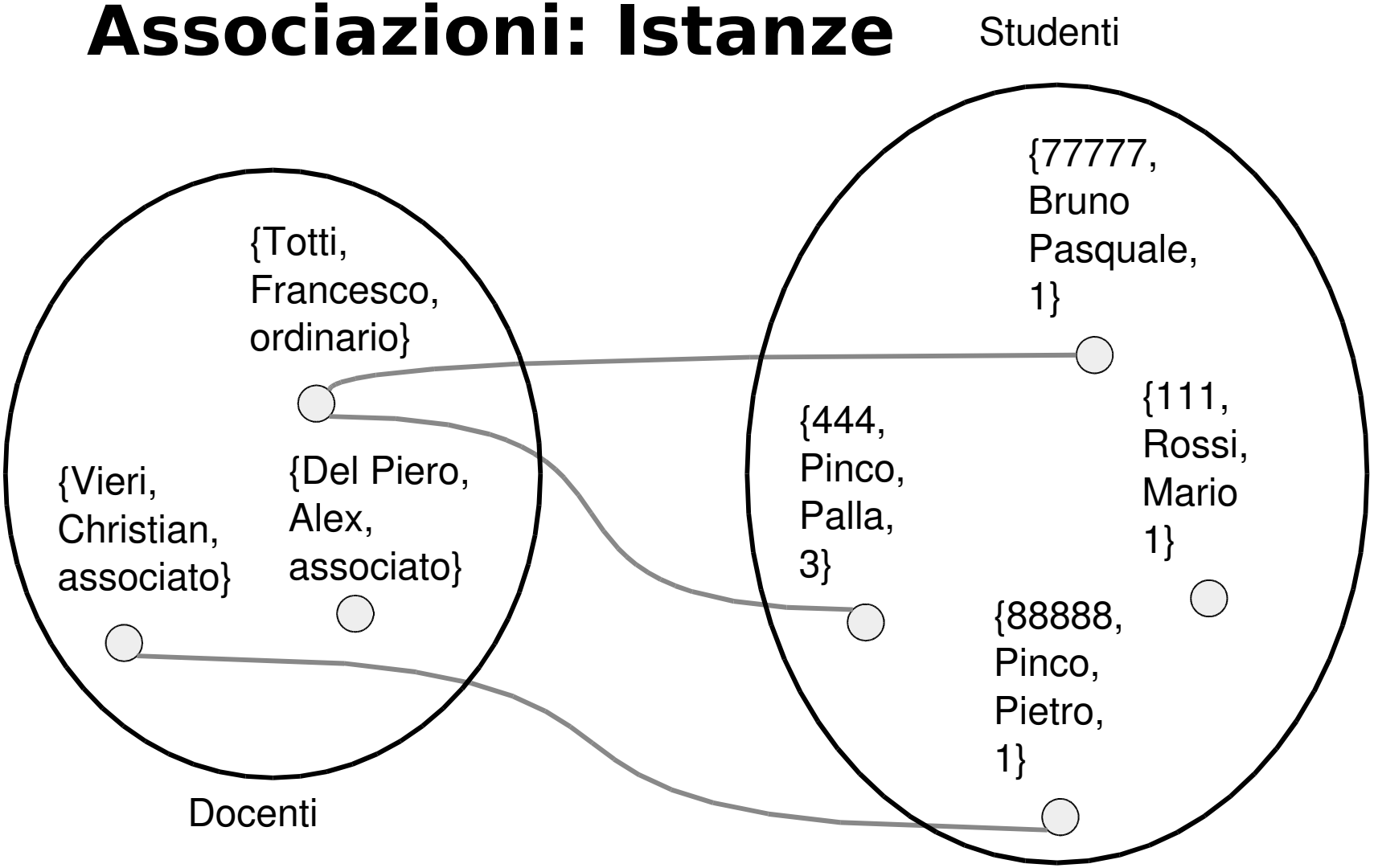
- **Istanza di un'associazione**
  - insieme di archi tra le istanze delle classi coinvolte

Associazioni

- **Esempi:**



Associazioni: Istanze



Cardinalità

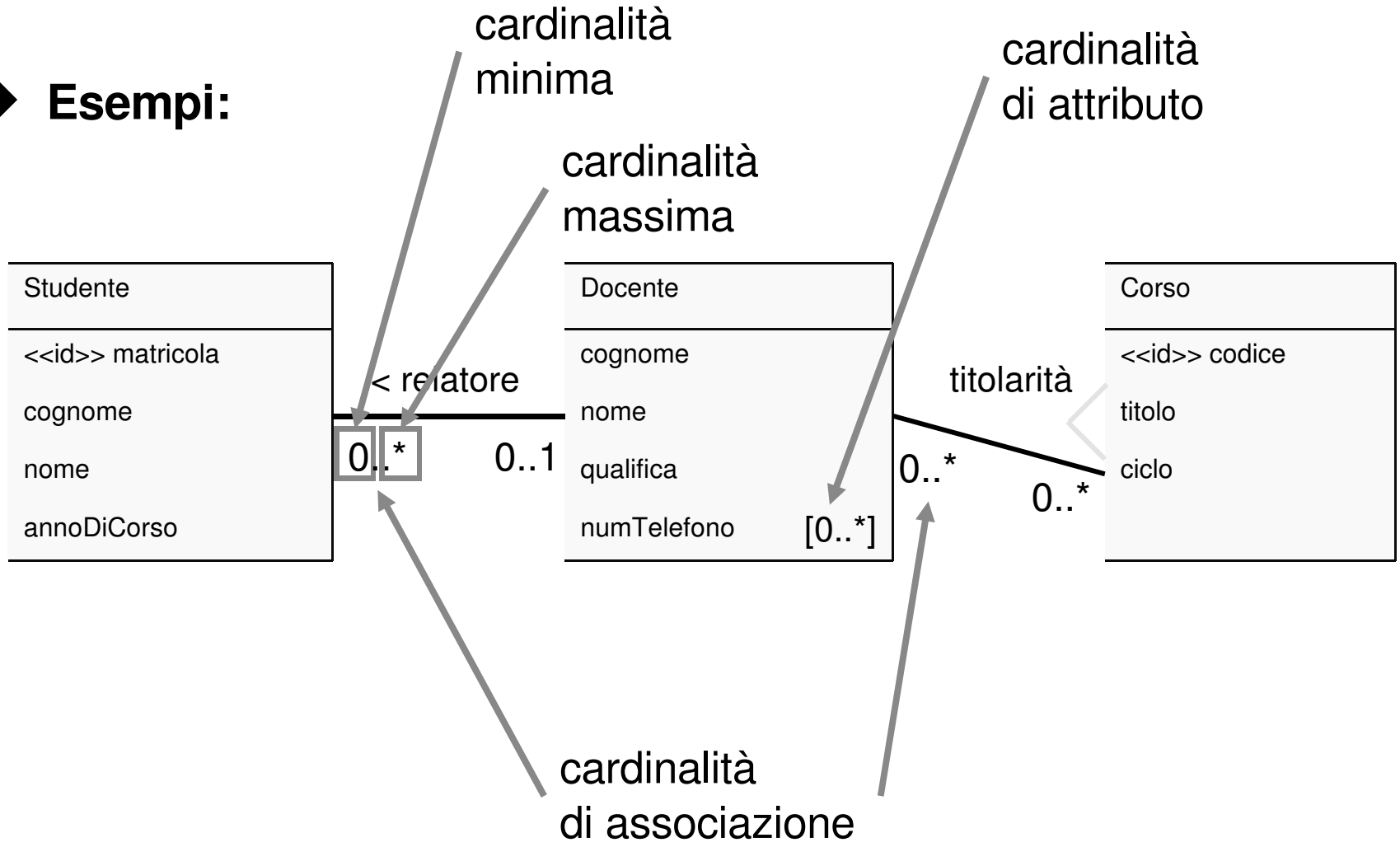
- **Vincoli sulle associazioni**
  - vincoli sul numero di archi tra gli oggetti
- **Vincoli sugli attributi**
  - numero di valori dell'attributo
- **Cardinalità minima**
  - normalmente 0 oppure 1
- **Cardinalità massima**
  - normalmente 1 oppure * (n) (ma anche 3 o 5)

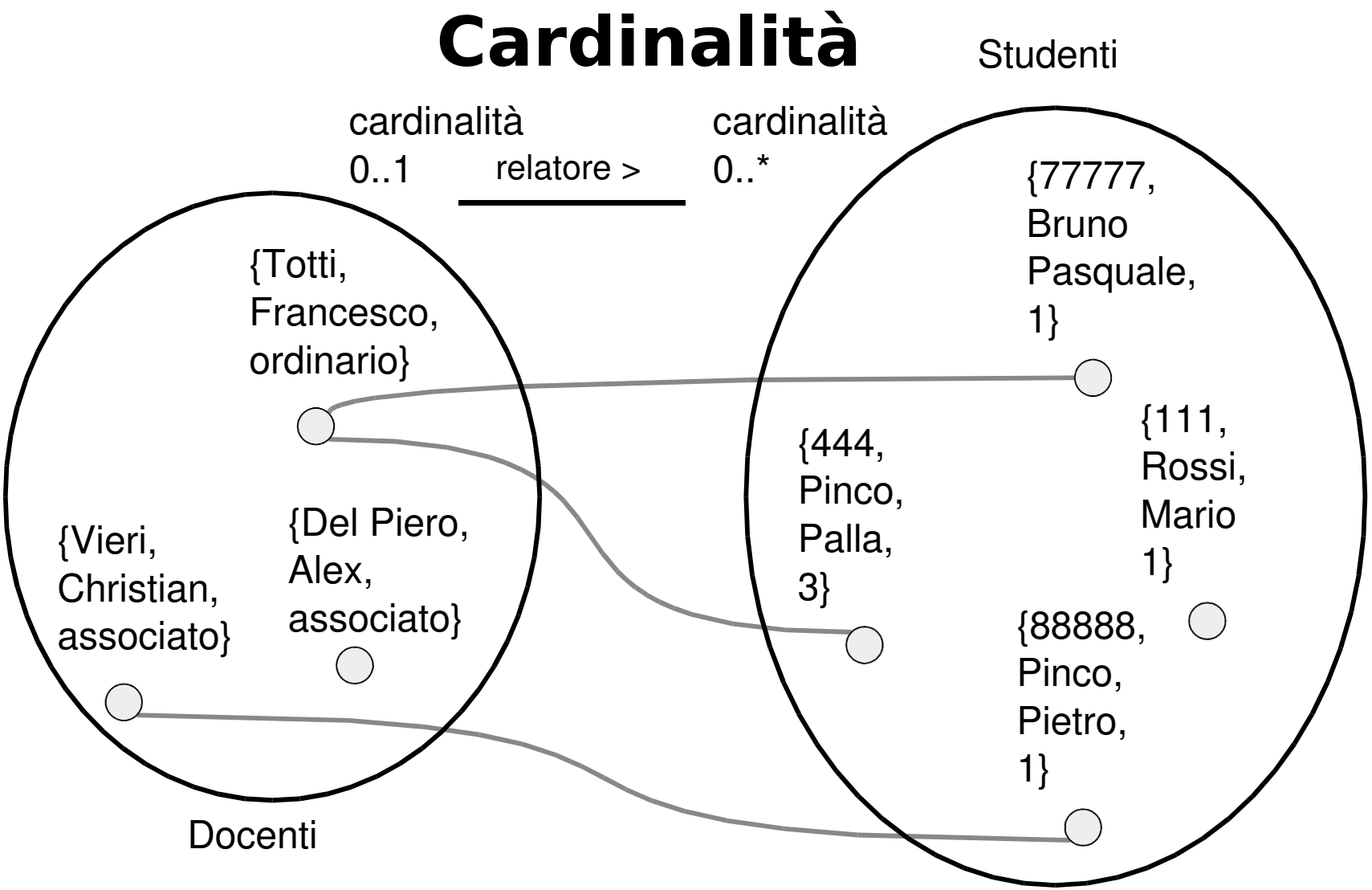
Cardinalità

- **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:**

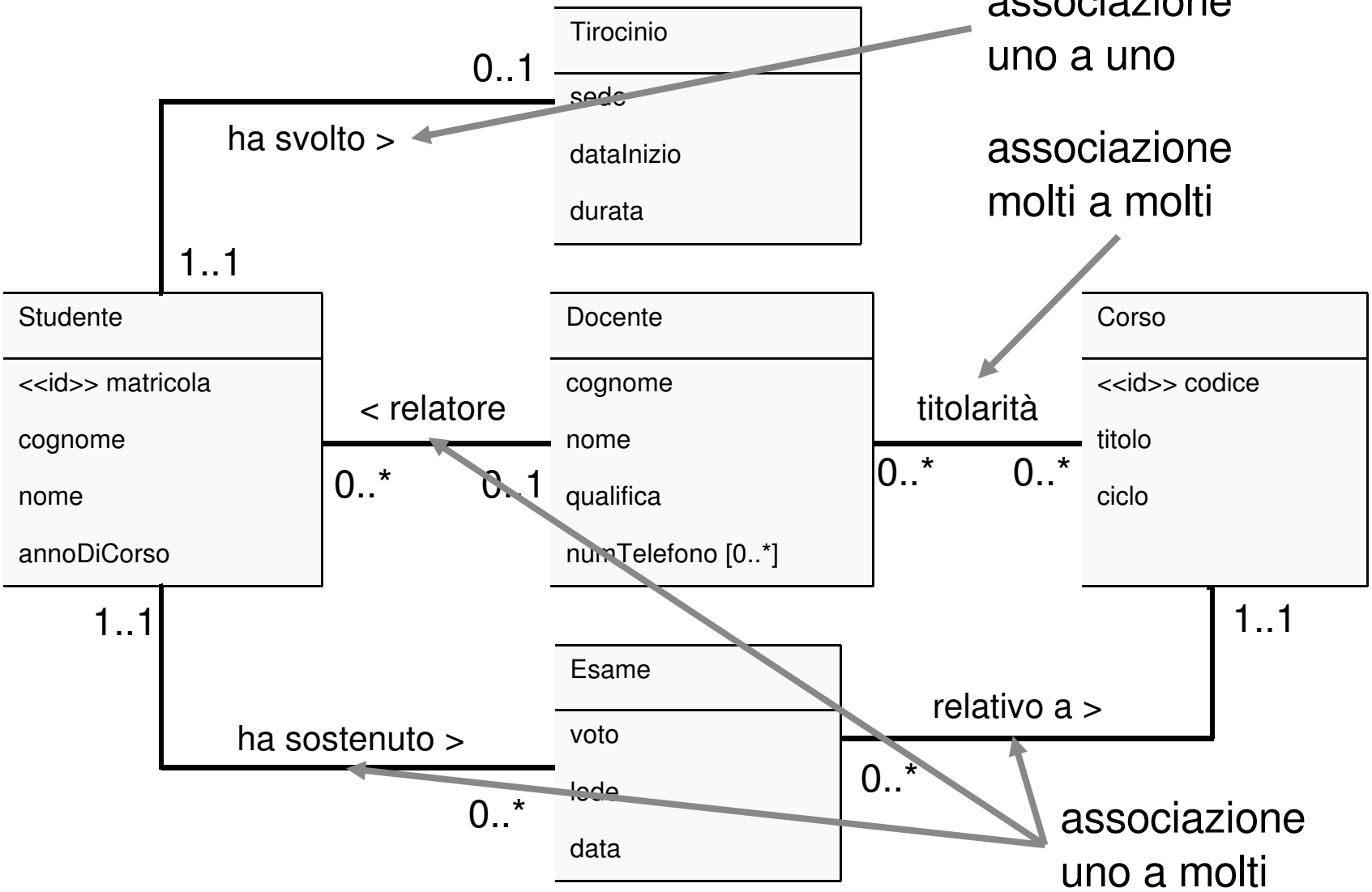



Cardinalità
Studenti
cardinalità
0..1
relatore >
cardinalità
0..*
{Totti,
Francesco,
ordinario}
{77777,
Bruno
Pasquale,
1}
{111,
Rossi,
Mario
1}
{444,
Pinco,
Palla,
3}
{88888,
Pinco,
Pietro,
1}
{Vieri,
Christian,
associato}
{Del Piero,
Alex,
associato}
Docenti

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



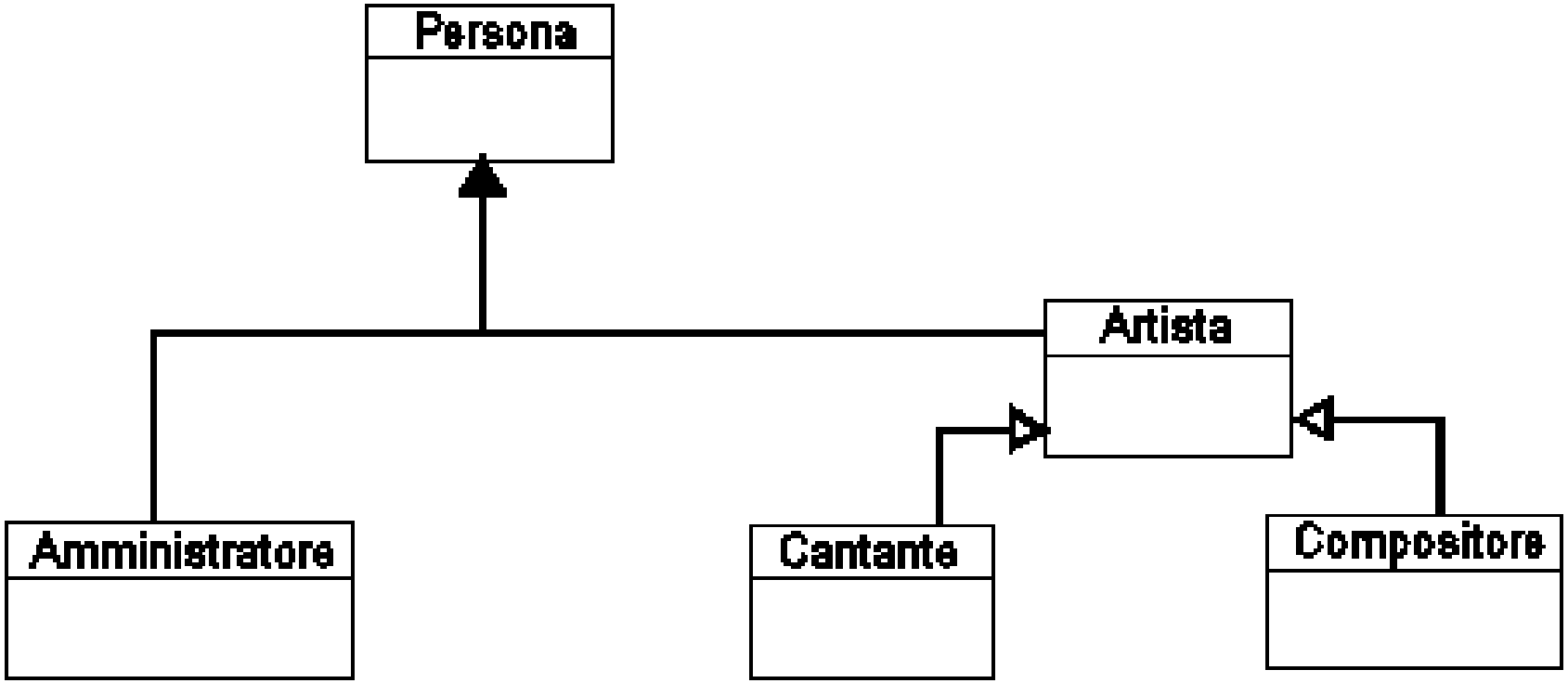
Generalizzazione e specializzazione

- **Relazioni gerarchiche tra i concetti rappr. dalle classi**
  - superclasse (padre): concetto più generale
  - sottoclasse (figlio): concetto più specifico

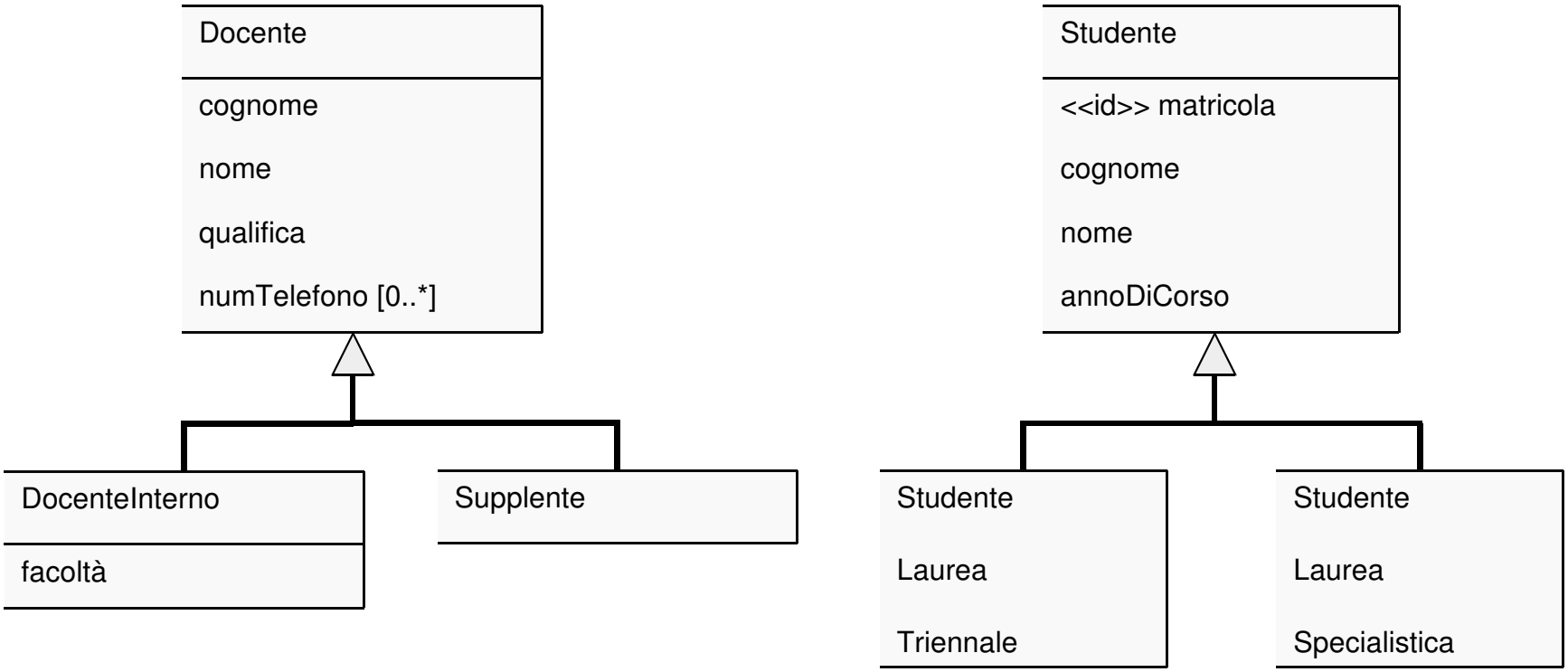
- **Implicano la semantica dell'ereditarietà**
  - le proprietà e le associazioni del padre sono anche proprietà e associazioni dei figli
  - le istanze dei figli sono anche istanze del padre

- **Meccanismo di astrazione: a livello della superclasse ci si concentra sulle proprietà a comune fra le diverse sottoclassi, dimenticandoci le proprietà specifiche.**

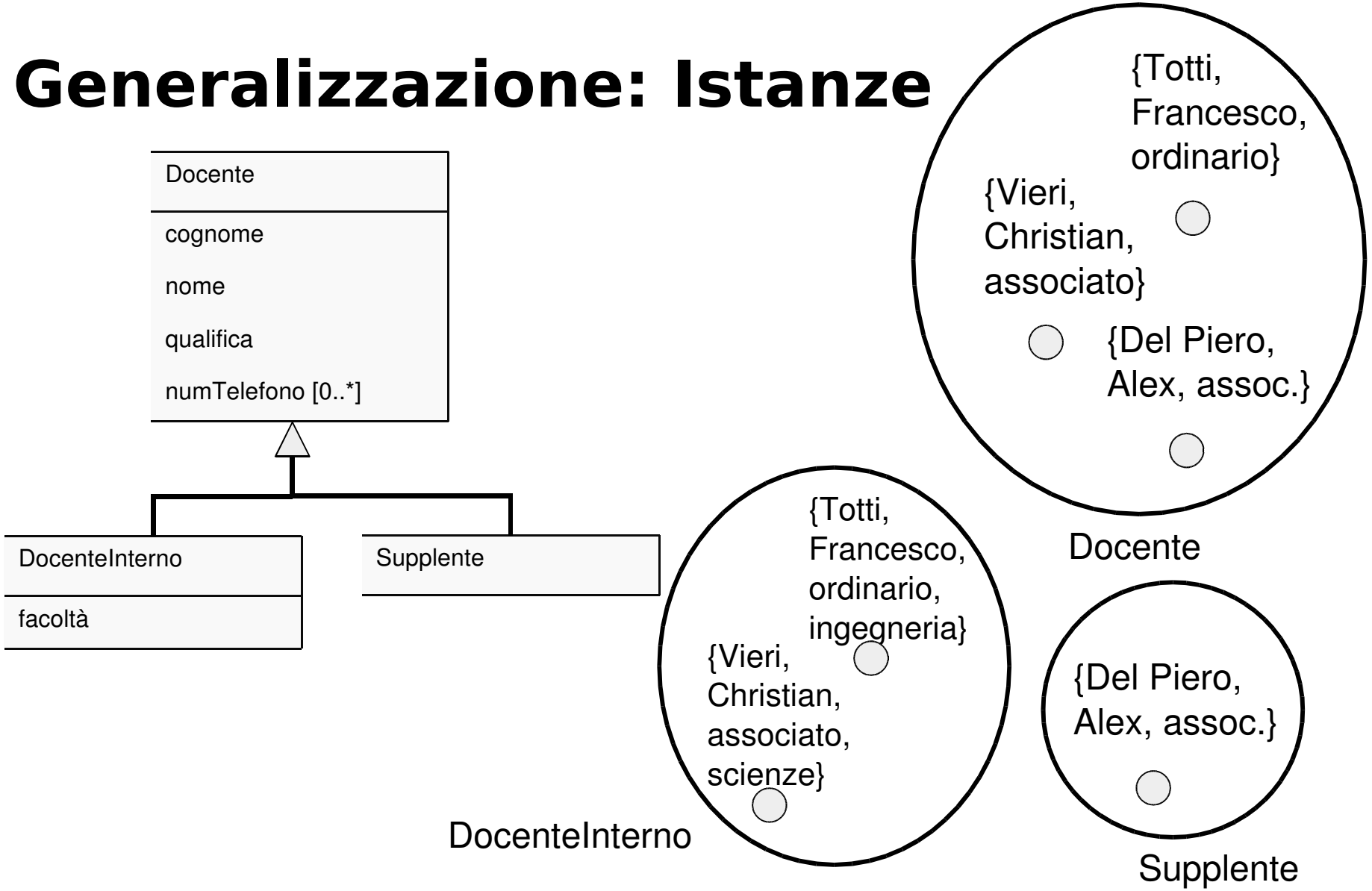
Esempio di gerarchia (tassonomia)



Generalizzazione



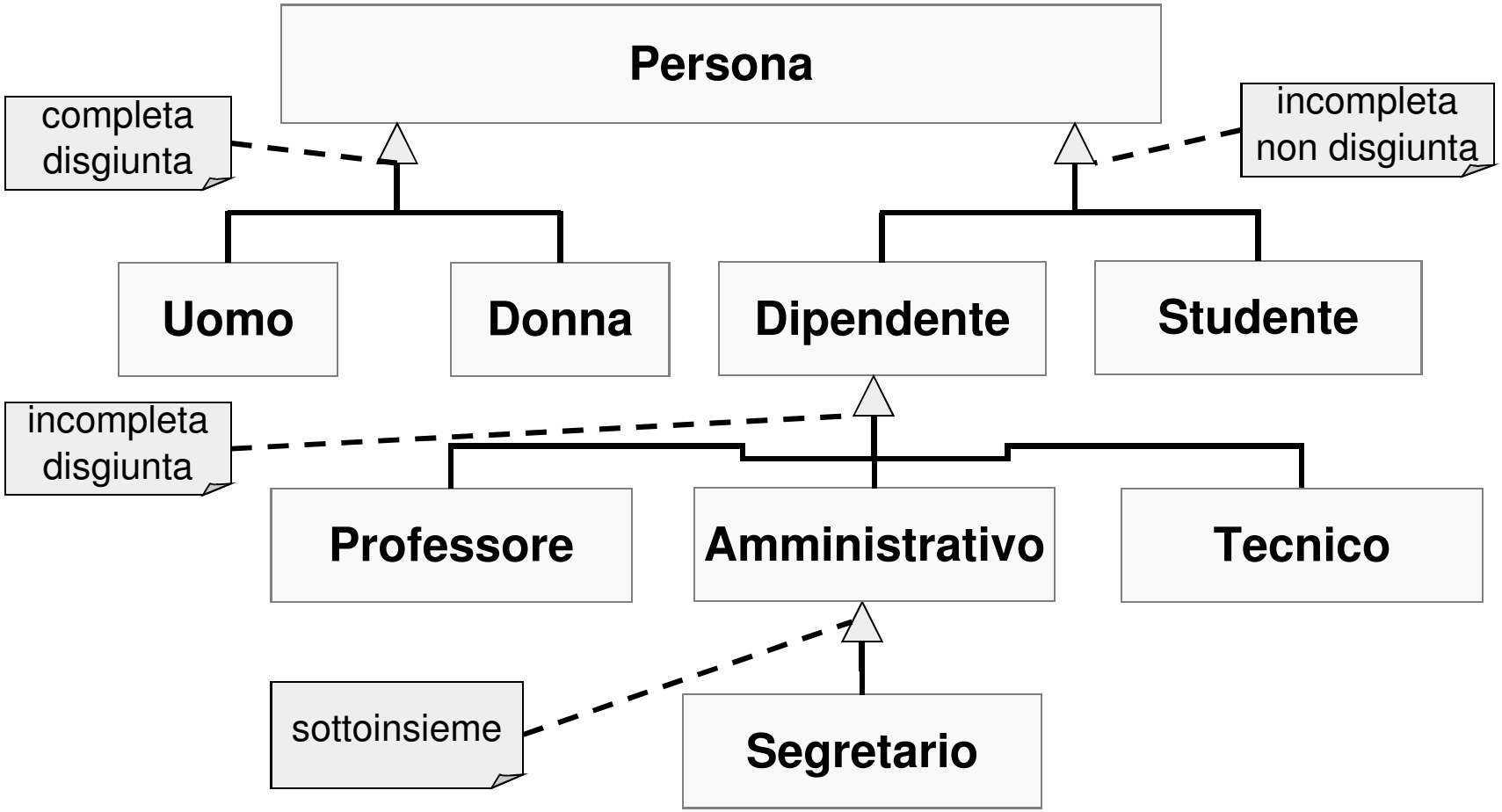
Generalizzazione: Istanze



Generalizzazioni

- **Relazioni di ereditarietà tra i concetti**
  - consentono di descrivere gerarchie, o tassonomie, o ontologie
- **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

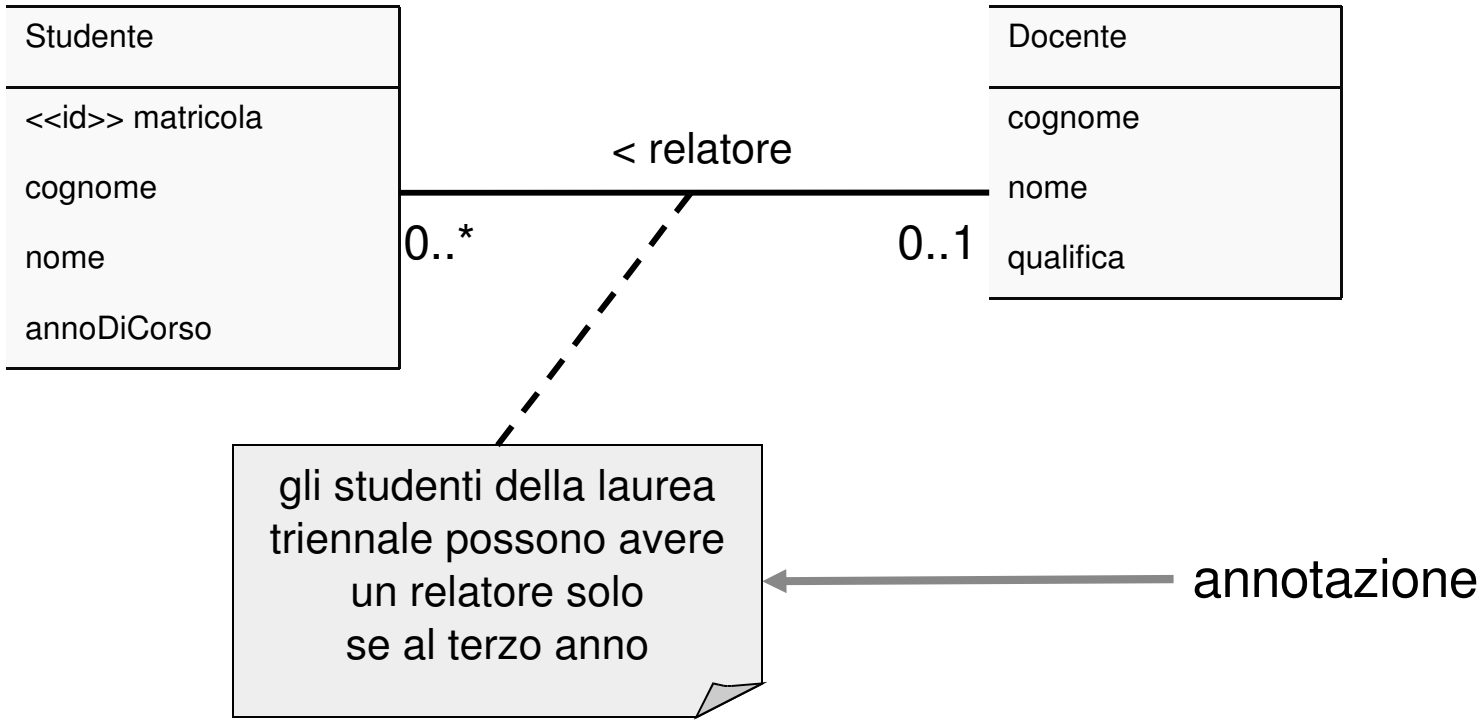


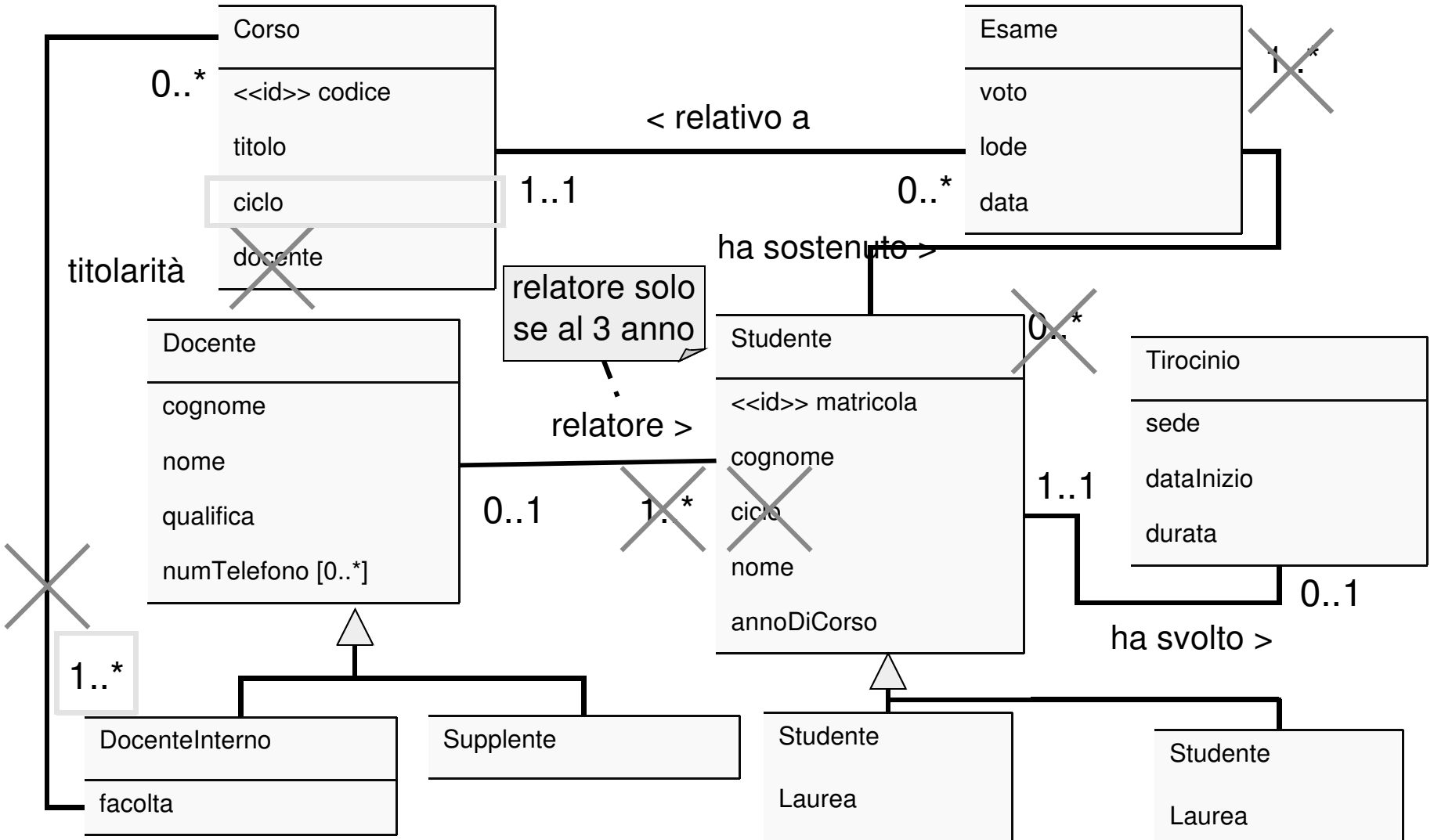
Il Diagramma Completo

- **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:**





Un Diagramma con Errori

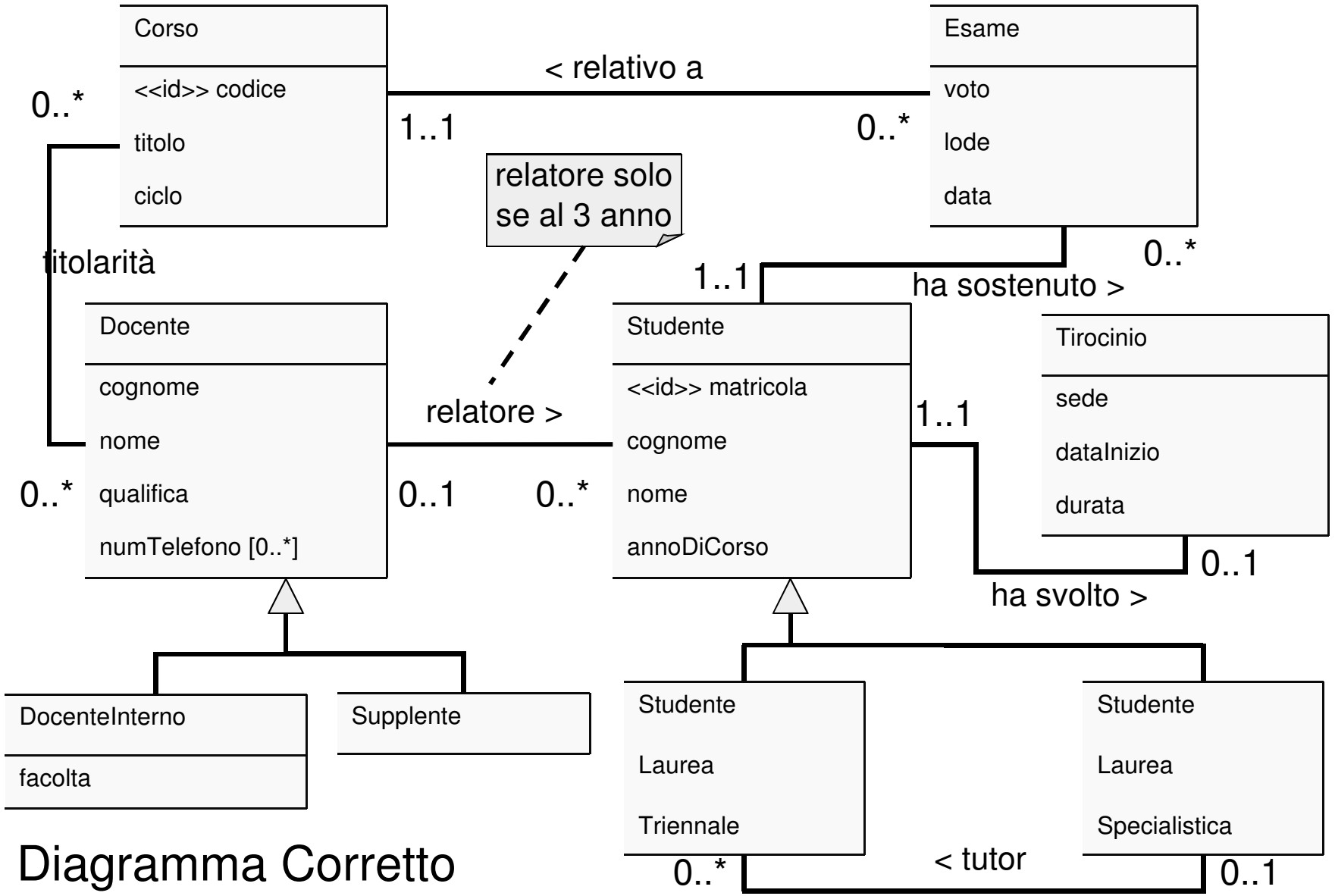


Diagramma Corretto

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

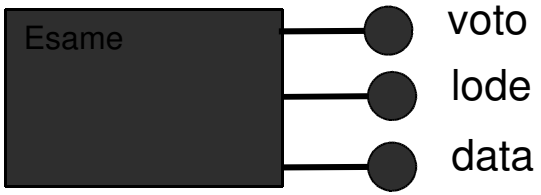
- **Per la modellazione concettuale**
  - sostanzialmente la semantica è equivalente

- **Costrutti dei due modelli**

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

Modello Entità-Relazione: Sintassi

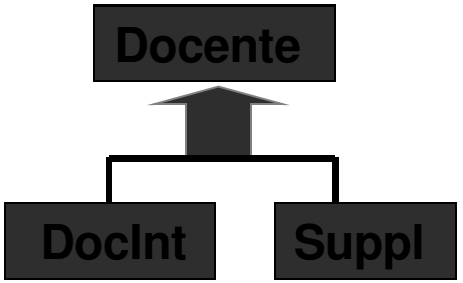
- **Entità: rettangolo**



- **Attributi: due sintassi**
  - linee esterne all'entità
  - sintassi equiv. a quella UML

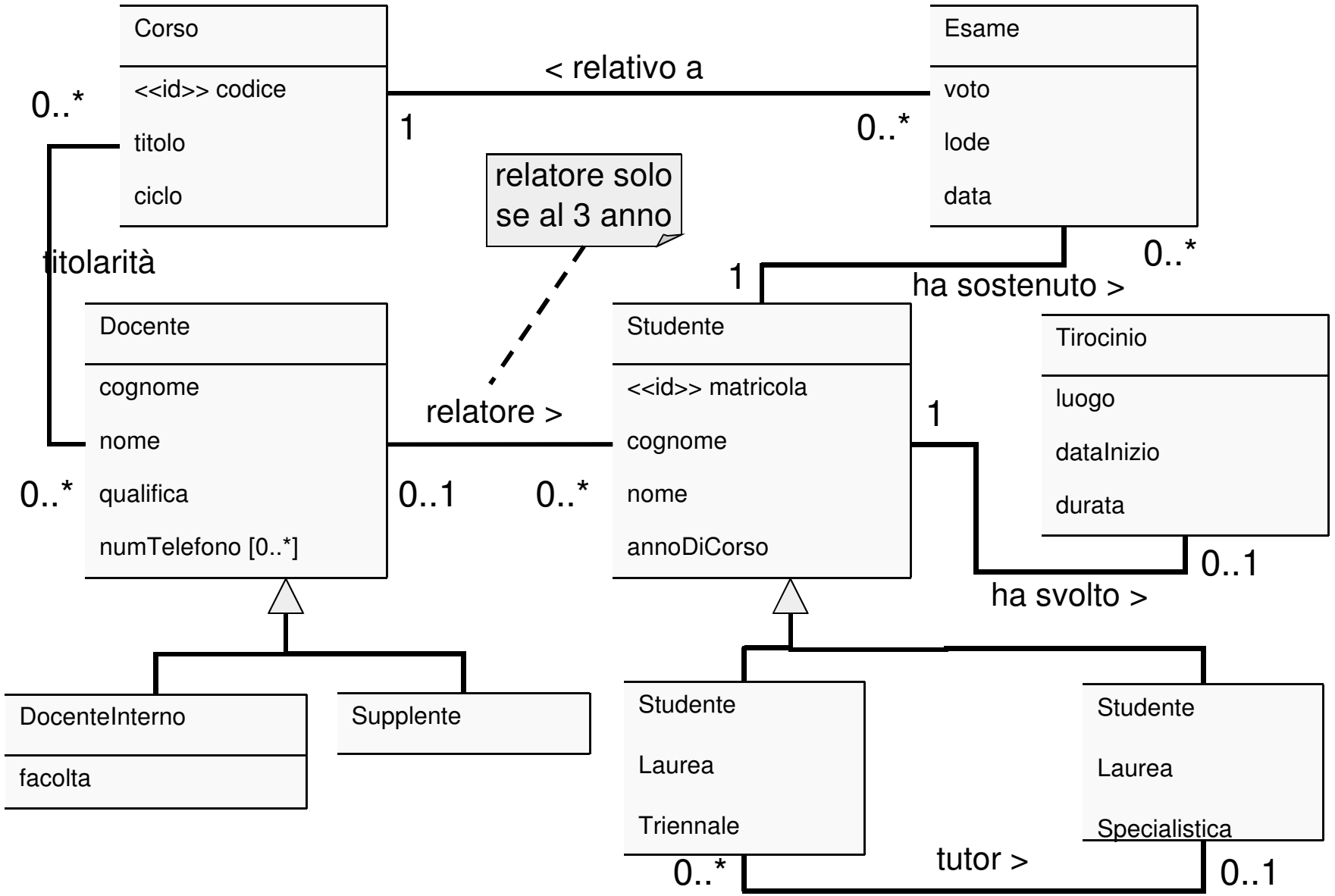
- **Relazione: rombo**

- **Cardinalità: posizione invertita**

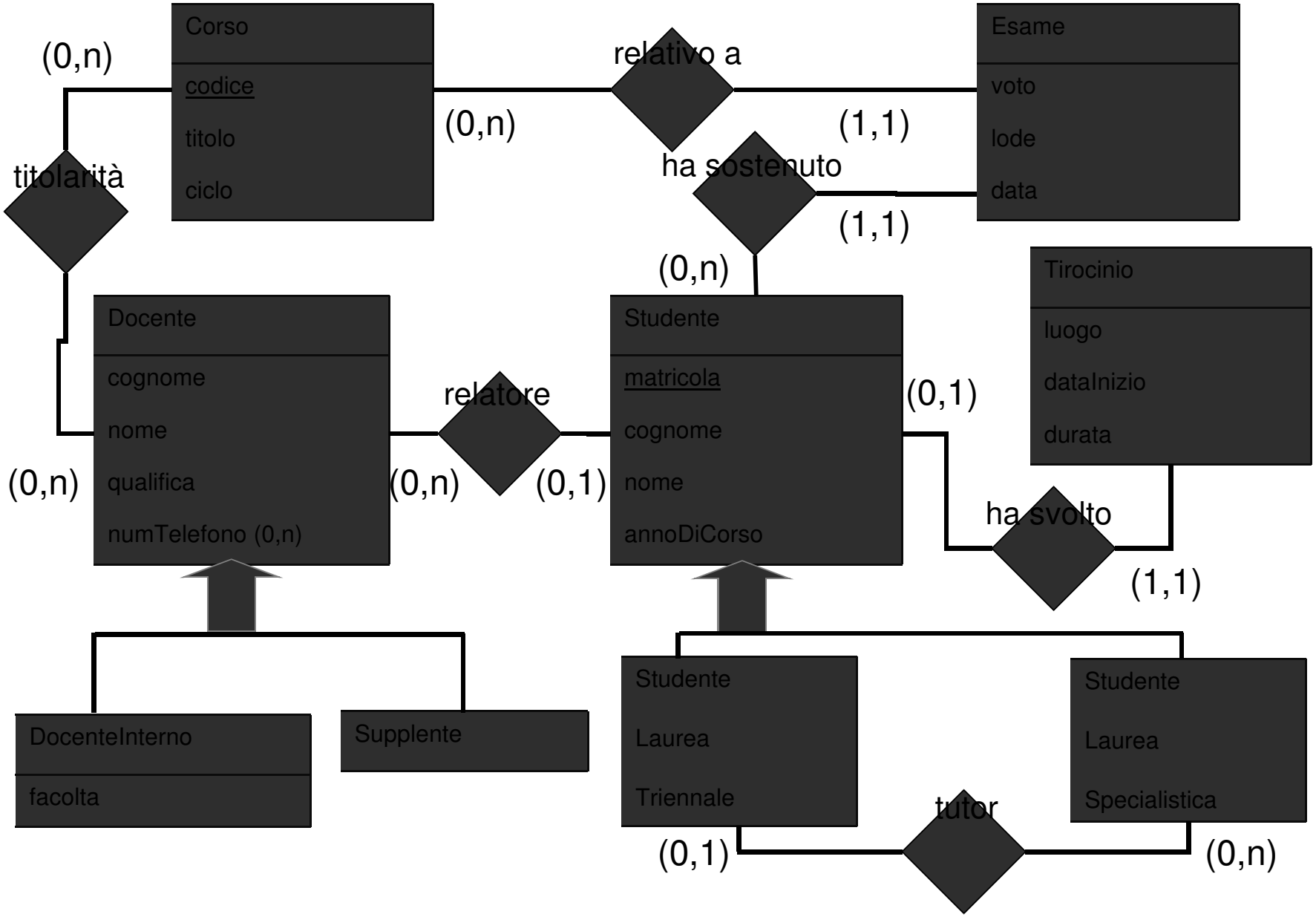


- **Generalizzazione: freccia**

Progettazione della BD >> Dettagli e Approfondimenti >> Modello ER



Progettazione della BD >> Dettagli e Approfondiementi >> Modello ER



Basi di Dati

La Progettazione Logica

Il Processo di Progetto della BD

- **Punto di partenza**
  - il modello concettuale dei dati
- **Progettazione Logica**
  - dallo schema concettuale viene derivato uno schema logico standard e i necessari schemi esterni
- **Progettazione Fisica**
  - lo schema logico viene sottoposto a verifica e viene ottimizzato

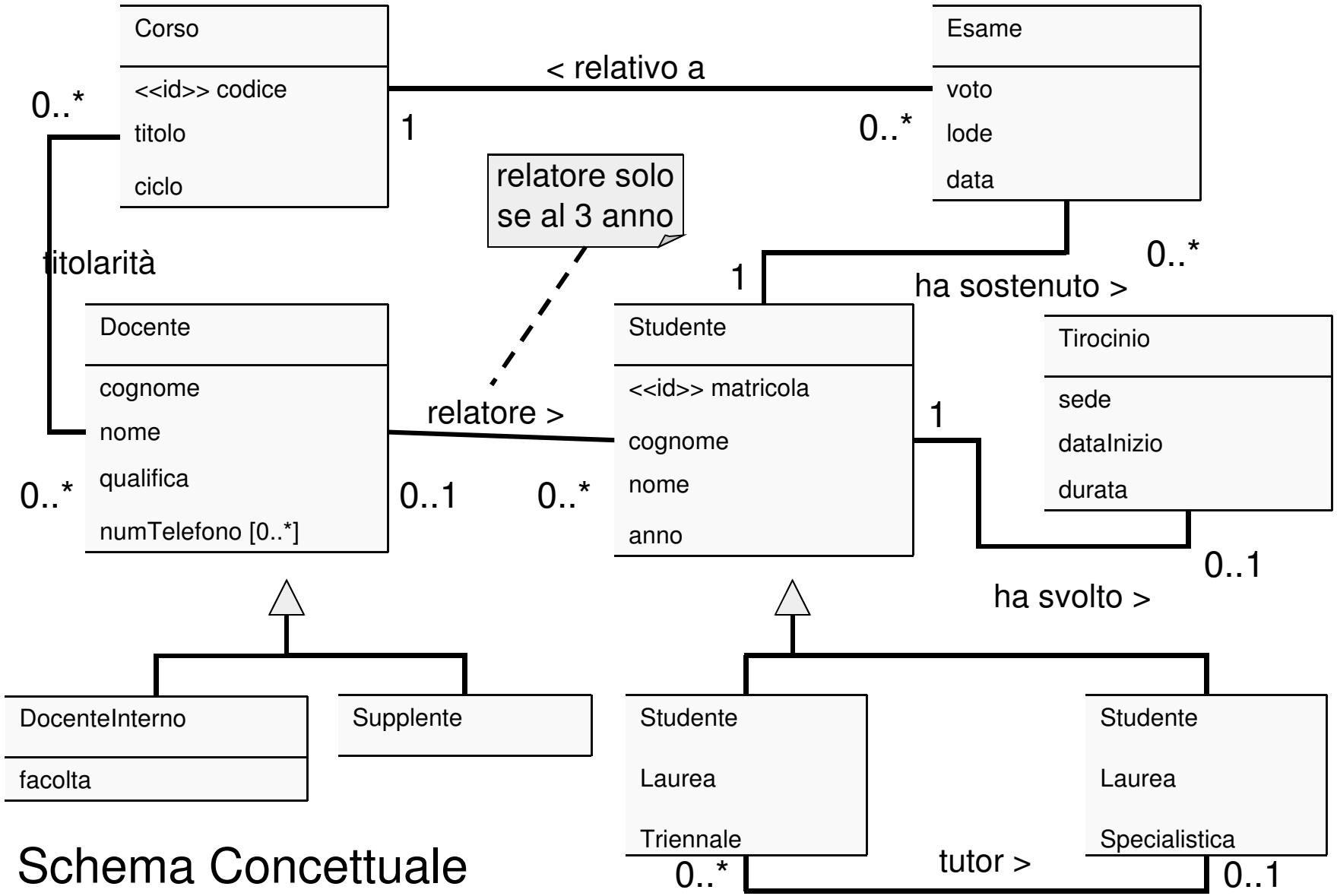
Progettazione logica

- Traduzione delle Classi
- Traduzione delle Gerarchie
- Traduzione delle Associazioni molti a molti
- Traduzione delle Associazioni 1-1 e 1-molti

Algoritmo di Progettazione Logica

- I passo: trad. delle classi non coinvolte in gerarchie
- II passo: trad. delle gerarchie
- III passo: trad. degli attributi multivalore
- IV passo: trad. delle assoc. molti a molti
- V passo: trad. delle assoc. uno a molti
- VI passo: trad. delle assoc. uno a uno

Progettazione Logica >> Algoritmo di Traduzione

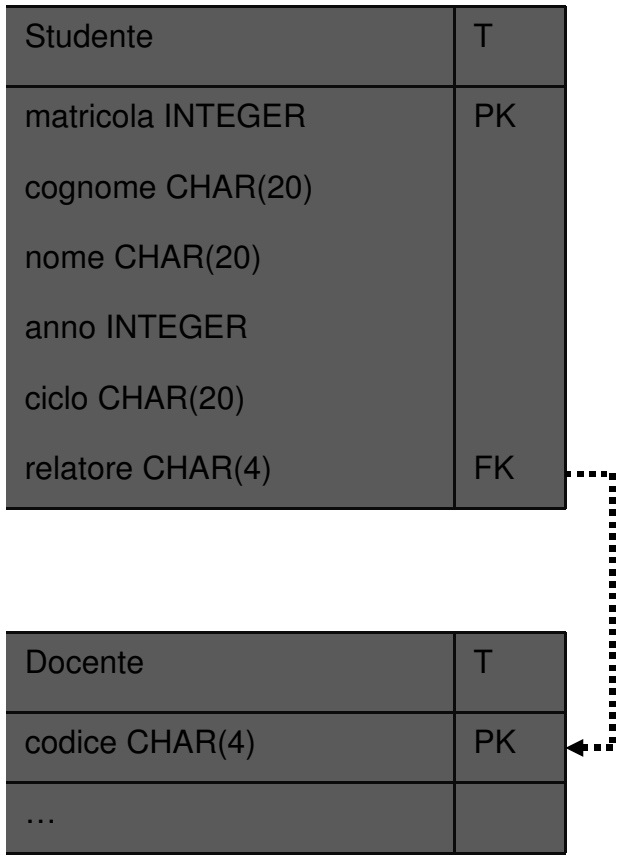


Schema Concettuale

Notazione Grafica per le Tabelle

- **Stereotipo di UML**
  - tabella e attributi
  - chiave primaria
  - chiave esterna

- **Esempio:**




```
CREATE TABLE Studente (
   matricola integer PRIMARY KEY,
   cognome char(20),
   nome char(20),
   anno integer,
   ciclo char(20),
   relatore char(4) REFERENCES Docente(codice));
```

I Passo: Traduzione delle Classi

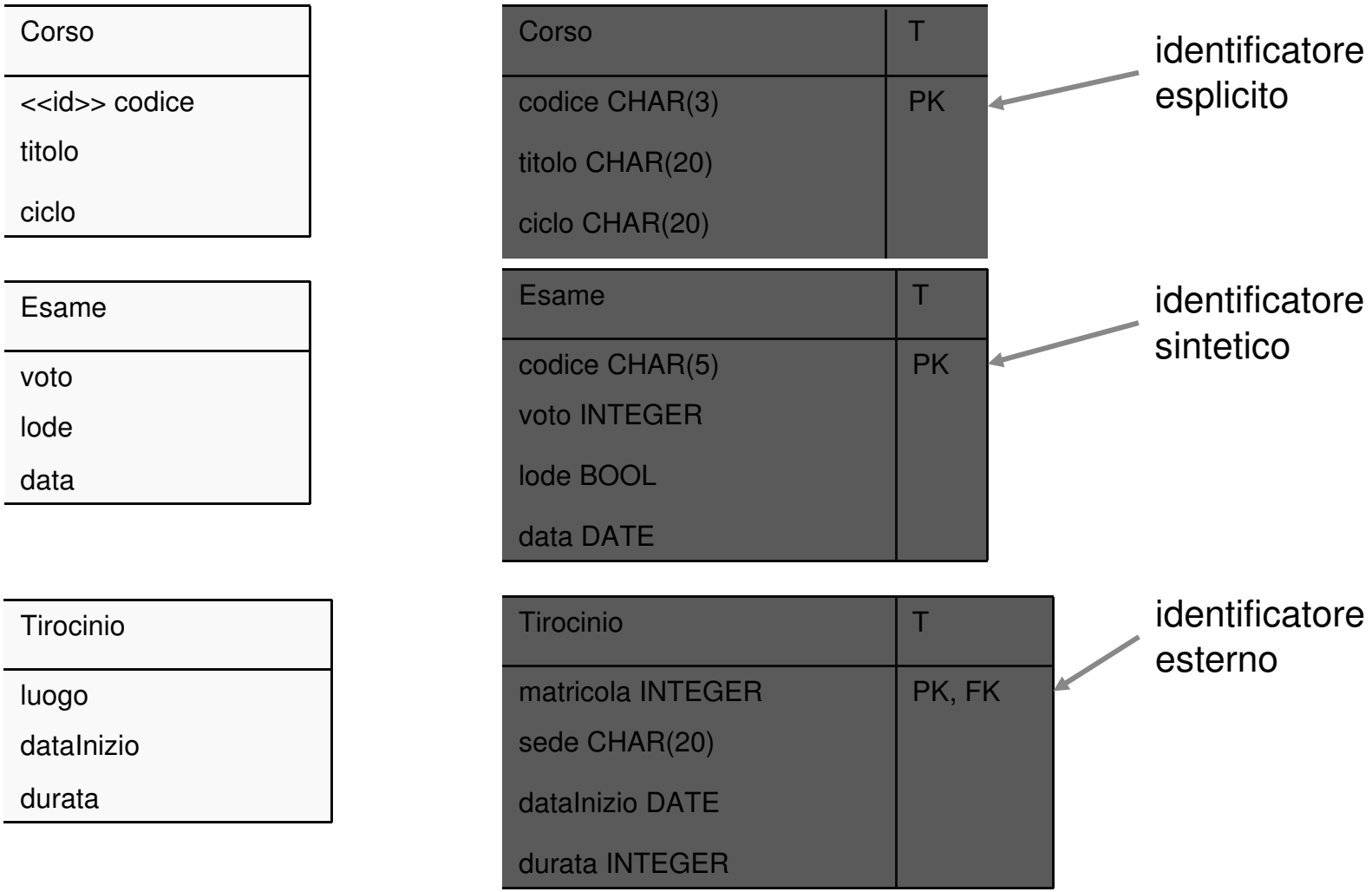
- **Idea**
  - ogni classe diventa una tabella
  - inizialmente gli stessi attributi monovalore
  - successivamente possono essere aggiunti altri attributi

- **E' necessario**
  - individuare il tipo degli attributi
  - individuare la chiave primaria
  - individuare eventuali chiavi esterne

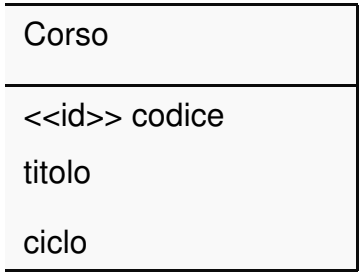
I Passo: Traduzione delle Classi

- **Chiave primaria**
  - deve essere semplice da usare e compatta
  - identificatore interno esplicito (es: matricola per Studente, codice per Corso)
  - un identificatore esterno può diventare una chiave primaria esterna (es: matricola dello studente per Tirocinio) purchè sia compatto
  - altrimenti si aggiunge un identificatore sintetico

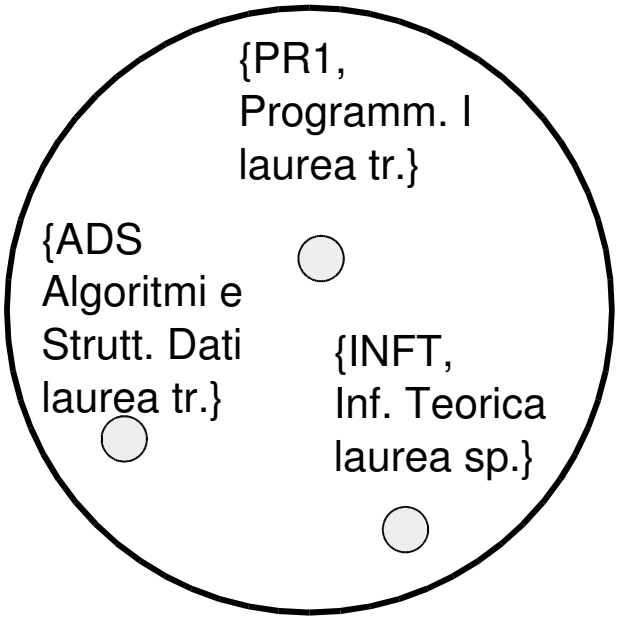
I Passo: Traduzione delle Classi



I Passo: Traduzione delle Classi



Corso	T
codice CHAR(3)	PK
titolo CHAR(20)	
ciclo CHAR(20)	

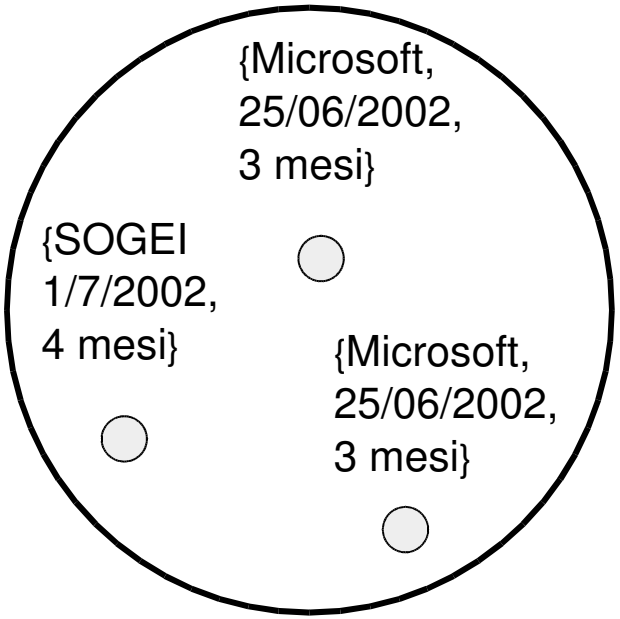


codice	titolo	ciclo	…
PR1	Programmazione I	laurea tr.	…
ASD	Algoritmi e Str. Dati	laurea tr.	…
INFT	Informatica Teorica	laurea sp.	…

I Passo: Traduzione delle Classi

Tirocinio
luogo dataInizio durata

Tirocinio	T
matricola INTEGER sede CHAR(20) dataInizio DATE durata INTEGER	PK, FK



studente	sede	dataInizio	durata	…
444	Microsoft	2002-05-15	3	…
77777	Microsoft	2002-05-15	3	…
88888	Basica	2002-09-01	3	…

II Passo: Traduzione delle Gerarchie

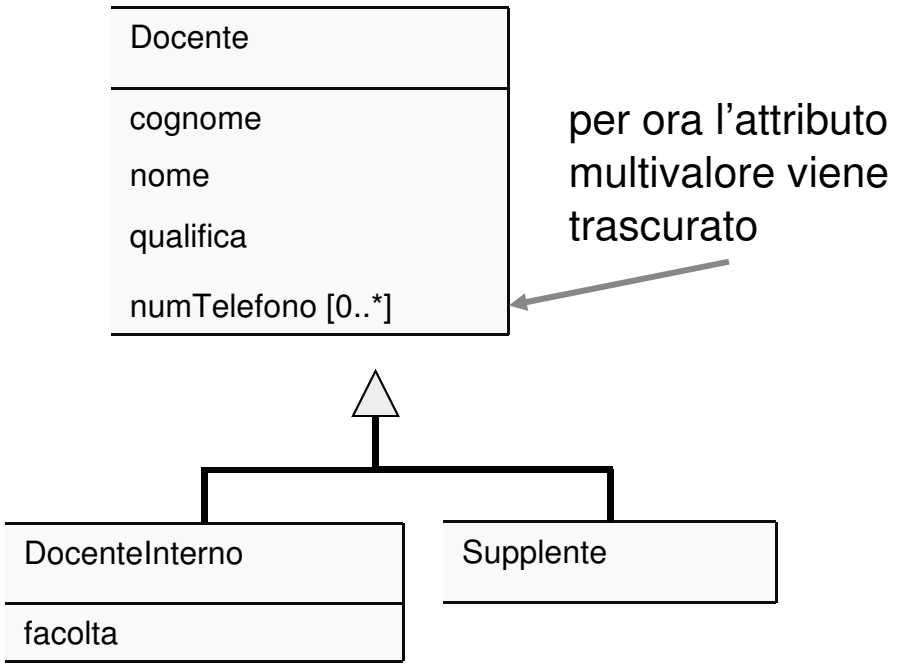
- **E' l'unico passo di una certa complessità**
  - non esiste la generalizzazione nel modello relazionale
- **Tre possibili strade**
  - tradurre solo il padre della gerarchia
  - tradurre solo i figli della gerarchia
  - tradurre il padre e i figli collegandoli con chiavi esterne

II Passo: Traduzione delle Gerarchie

- **I Soluzione: Solo il padre**
  - un'unica tabella con il nome del padre
  - la tabella deve avere tutti gli attributi di padre e figli
  - serve un ulteriore attributo (es: tipo) per distinguere le istanze dei figli
  - conveniente se le operazioni sui figli non sono particolarmente rilevanti nell'appl.
  - genera valori nulli

II Passo: Traduzione delle Gerarchie

- **I Soluzione: Solo il padre**



per ora l'attributo multivalore viene trascurato

Docente	T
codice CHAR(4) cognome CHAR(20) nome CHAR(20) facolta CHAR(10) qualifica CHAR(15) tipo CHAR(10)	PK

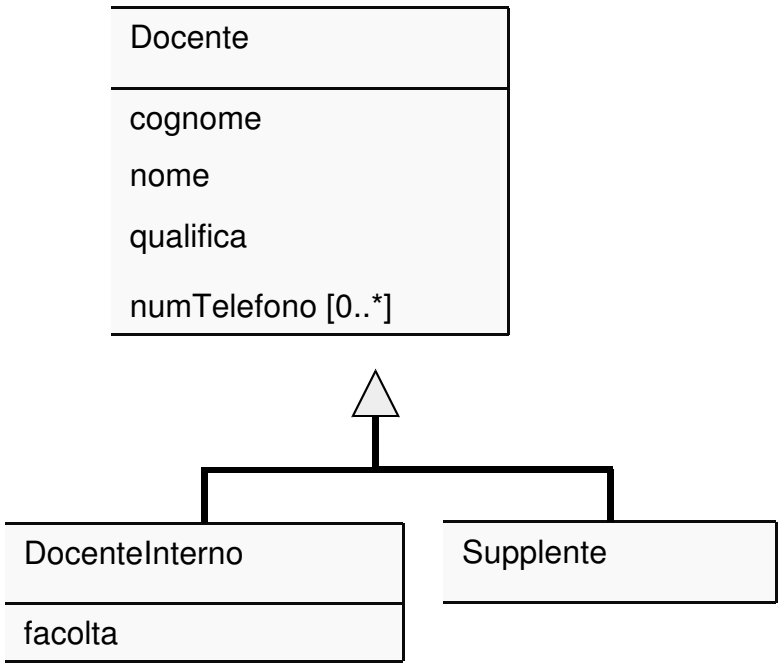
"tipo" può valere:
-interno oppure
-supplente

II Passo: Traduzione delle Gerarchie

- **II Soluzione: Solo i figli**
  - una tabella per ciascun figlio
  - ciascun figlio eredita le associazioni e gli attributi del padre
  - possibile solo se la gerarchia è completa
  - conveniente se l'applicazione richiede spesso di accedere singolarmente ai figli
  - costringe ad effettuare molte unioni

II Passo: Traduzione delle Gerarchie

- **II Soluzione: Solo i figli**



DocenteInterno	T
codice CHAR(4) cognome CHAR(20) nome CHAR(20) facolta CHAR(10) qualifica CHAR(15)	PK

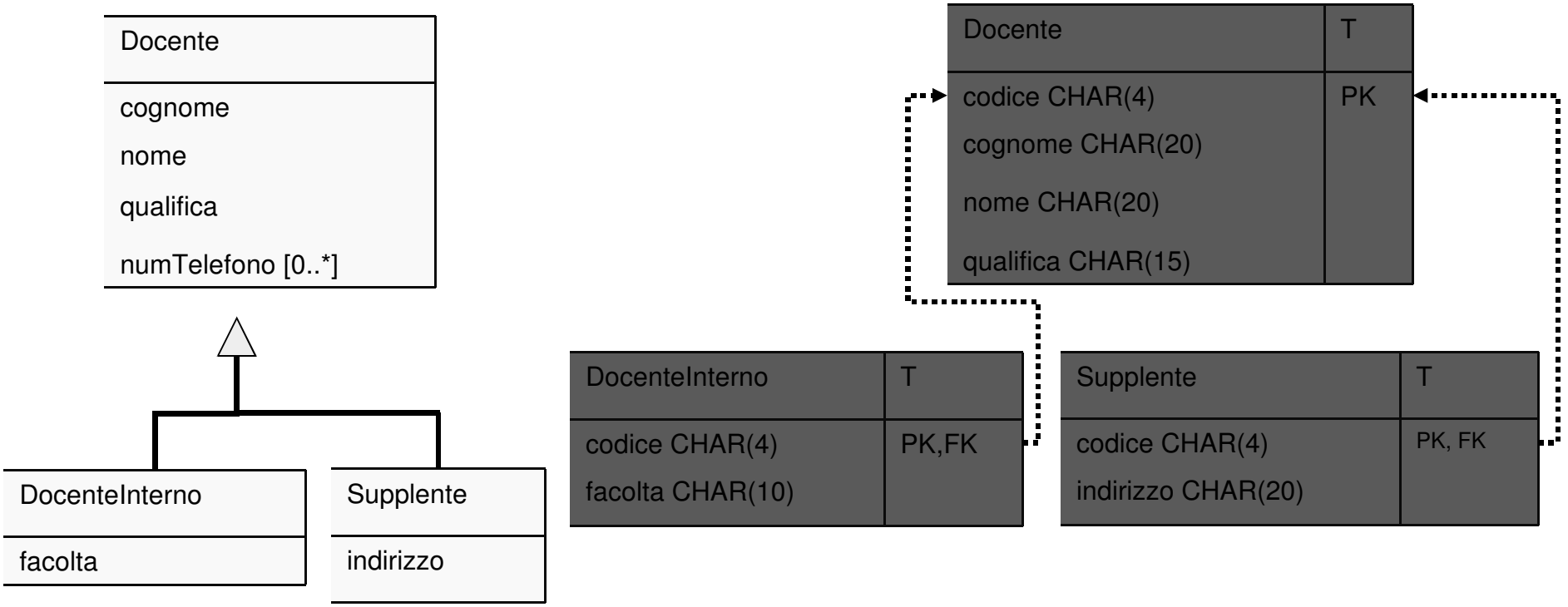
Supplente	T
codice CHAR(4) cognome CHAR(20) nome CHAR(20) qualifica CHAR(15)	PK

II Passo: Traduzione delle Gerarchie

- **III Soluzione: Sia il padre che i figli**
  - una tabella per il padre e una per ciascun figlio (per ogni istanza del figlio: parte degli attributi nella tabella specifica, parte nella tabella generale)
  - riferimento da ciascun figlio al padre
  - conveniente se bisogna spesso accedere tanto al padre che singolarmente ai figli
  - costringe ad effettuare molti join

II Passo: Traduzione delle Gerarchie

- **III Soluzione: Sia il padre che i figli**



II Passo: Traduzione delle Gerarchie

- **III Soluzione: Sia il padre che i figli**

Docente

codice	cognome	nome	qualifica
FT	Totti	Francesco	ordinario
CV	Vieri	Christian	associato
ADP	Del Piero	Alessandro	null

DocenteInterno

codice	facolta
FT	Ingegneria
CV	Scienze

Supplente

codice	Indirizzo
ADP	Stadio delle Alpi, Torino

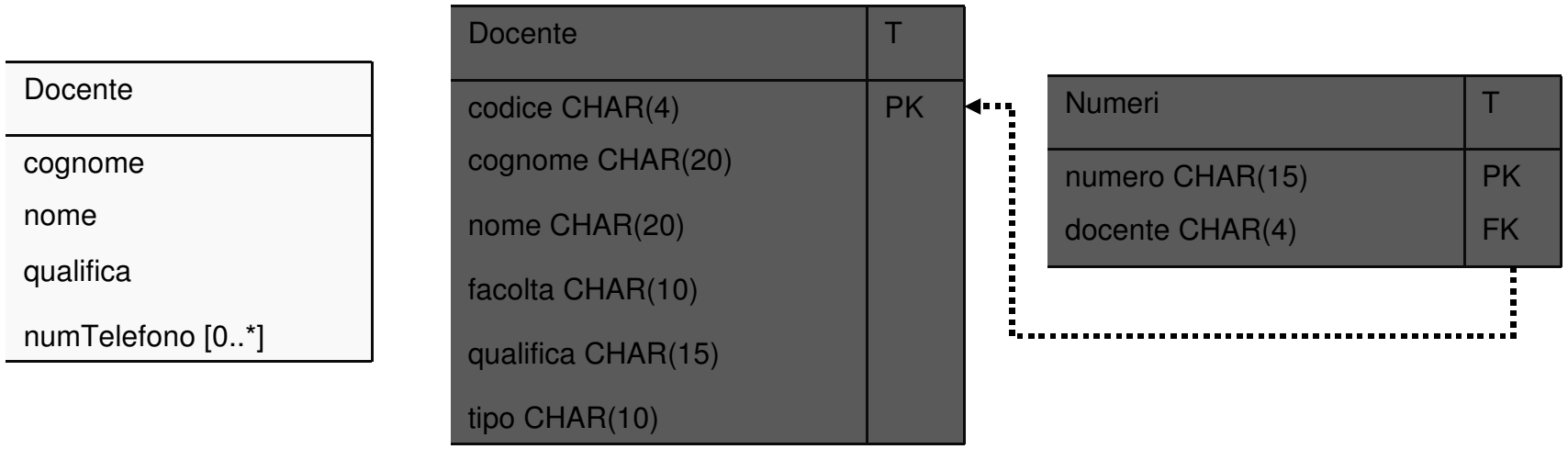
II Passo: Traduzione delle Gerarchie

- **Nel nostro esempio**
  - soluzione n.1 per i docenti
  - un'unica tabella "Docente"
  - soluzione n.1 per gli studenti
  - un'unica tabella "Studente"

Studente	T
matricola INTEGER cognome CHAR(20) nome CHAR(20) anno INTEGER ciclo CHAR(15)	PK

III Passo: Trad. degli Attributi Multiv.

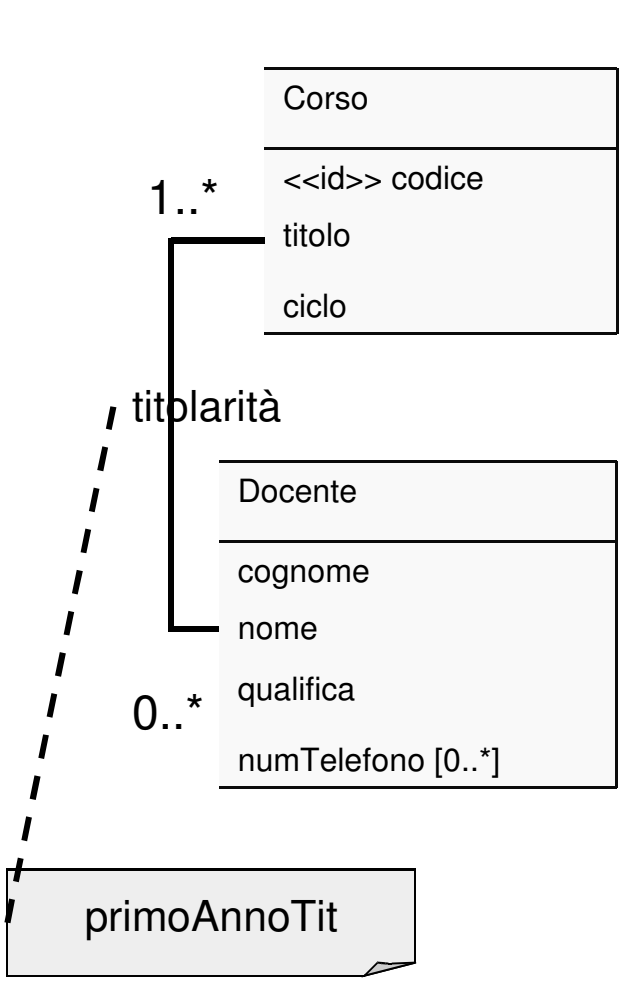
- **Ogni attributo multivalore genera una nuova tabella**
  - chiave esterna per fare riferimento alla tabella che traduce la classe originale



IV Passo: Trad. delle Associazioni m-m

- **Ogni associazione molti a molti genera una tabella**
  - riferimenti (chiavi esterne) alle tabelle che traducono le classi coinvolte
  - eventuali attributi dell'associazione
  - la chiave della tabella deve includere le chiavi esterne

IV Passo: Trad. delle Associazioni m-m



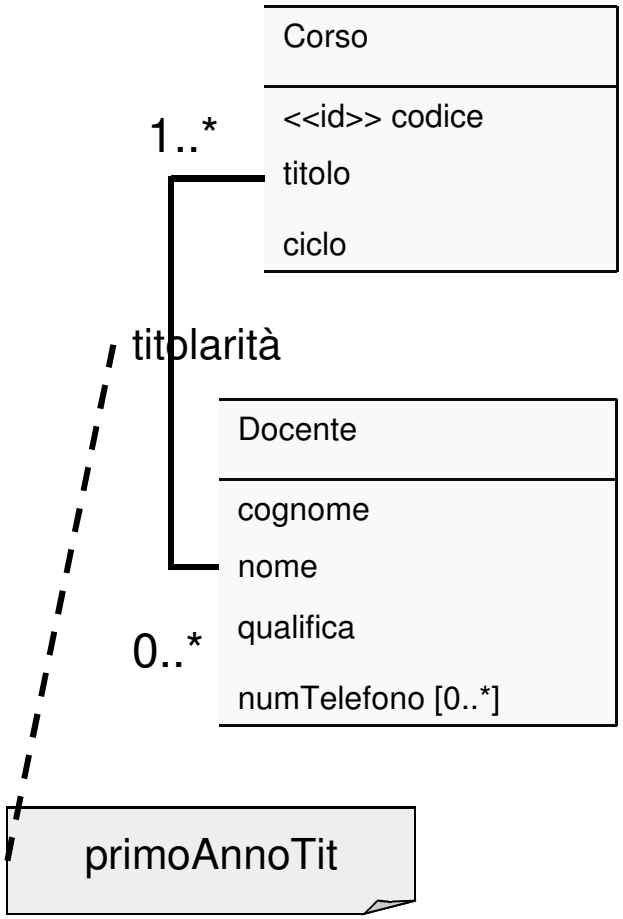
Corso	T
codice CHAR(3) titolo CHAR(20) ciclo CHAR(20)	PK

Titolarità	T
corso CHAR(3) docente CHAR(4) primoAnnoTit INTEGER	PK, FK PK, FK

Docente	T
codice CHAR(4) cognome CHAR(20) nome CHAR(20) …	PK …

attributo dell' associazione (nel seguito omesso)

IV Passo: Trad. delle Associazioni m-m



codice	titolo	ciclo
PR1	Programmazione I	laurea tr.
ASD	Algoritmi e Str. Dati	laurea tr.
INFT	Informatica Teorica	laurea sp.

docente	corso	primoAnnoTit
FT	PR1	2001
CV	ASD	2002
FT	ASD	1999
…	…	

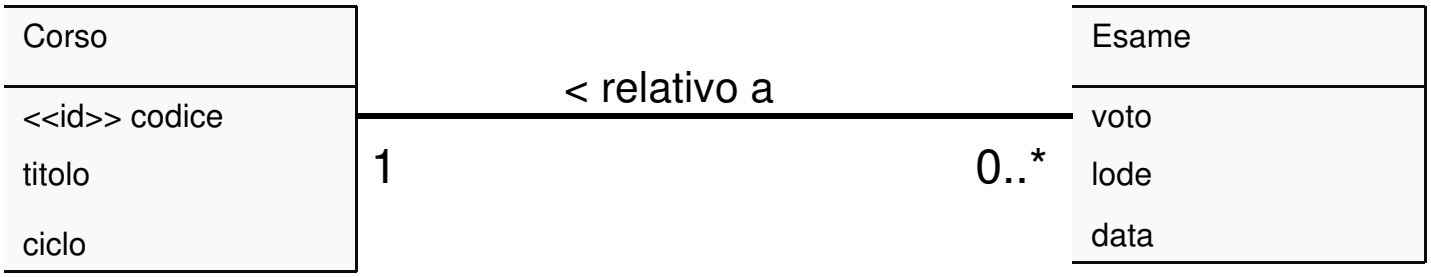
codice	cognome	nome	…
FT	Totti	Francesco	…
CV	Vieri	Christian	…
ADP	Del Piero	Alessandro	…

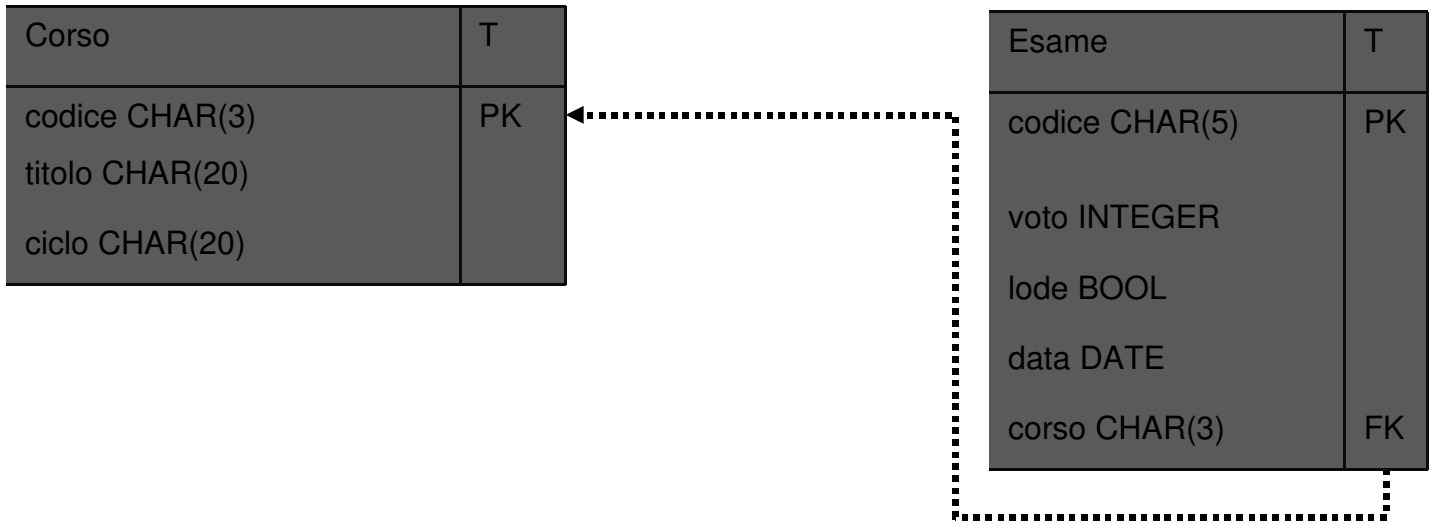
V Passo: Trad. delle Associazioni 1-m

- **Potrebbero essere tradotte con nuove tabelle**
  - sarebbe inefficiente
  - costringerebbe a più join del normale

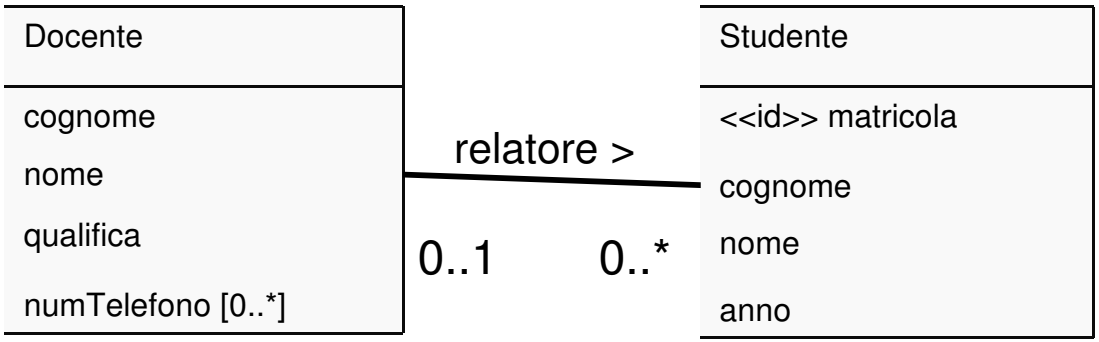
- **Generano chiavi esterne**
  - ciascuna istanza dell'associazione è identificata dall'oggetto dal lato 1
  - chiave esterna della tabella dal lato 1 nella tabella corrispondente alla classe dal lato m

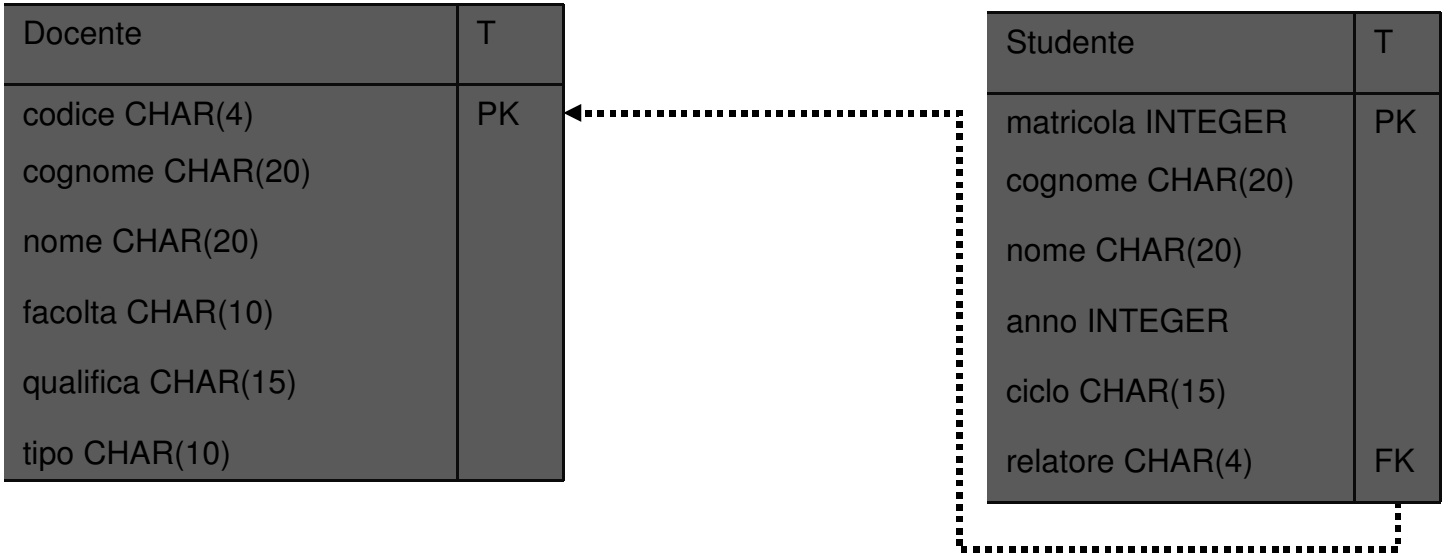
V Passo: Trad. delle Associazioni 1-m



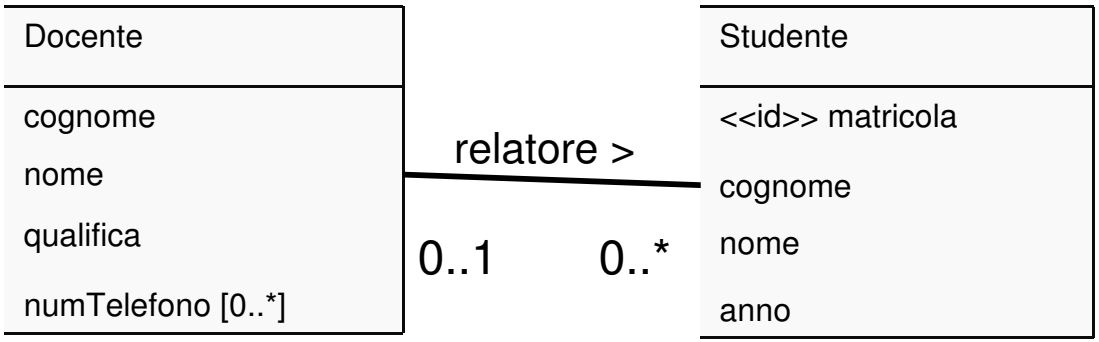


V Passo: Trad. delle Associazioni 1-m





V Passo: Trad. delle Associazioni 1-m

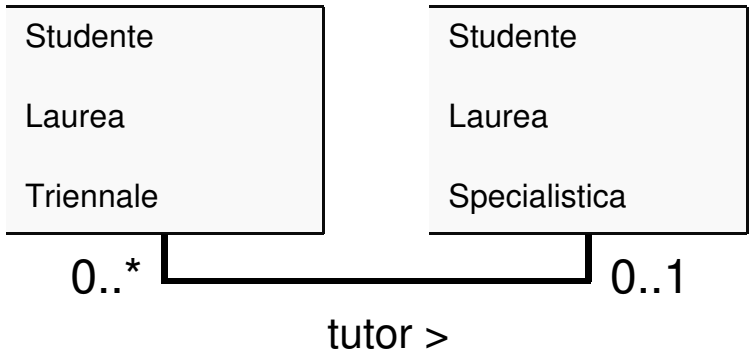


codice	cognome	nome	…
FT	Totti	Francesco	…
CV	Vieri	Christian	…
ADP	Del Piero	Alessandro	…

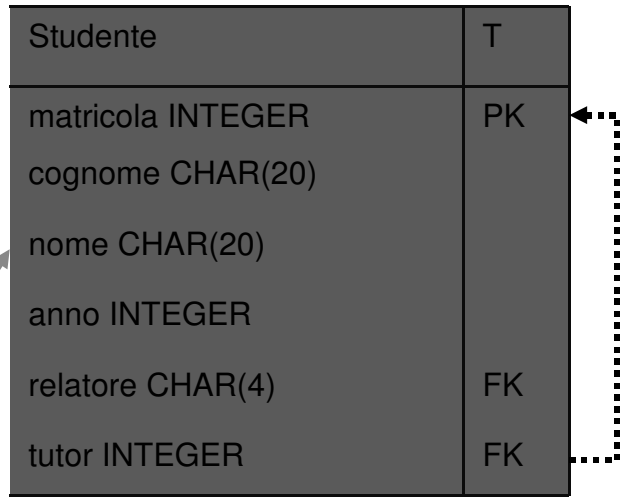
matricola	cognome	nome	…	relatore
111	Rossi	Mario	…	null
222	Neri	Paolo	…	null
333	Rossi	Maria	…	null
444	Pinco	Palla	…	FT
77777	Bruno	Pasquale	…	FT
88888	Pinco	Pietro	…	CV

V Passo: Trad. delle Associazioni 1-m

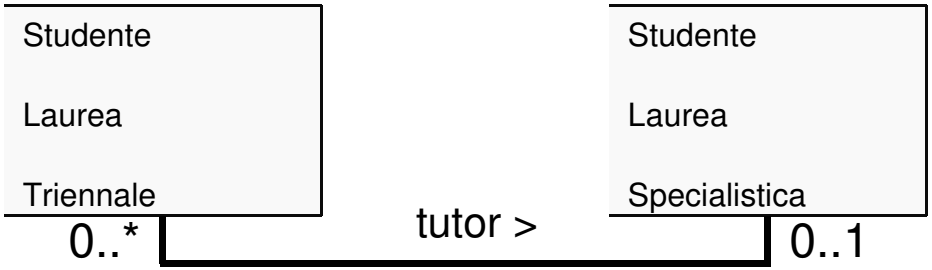
- **Attenzione:**
  - nel caso degli studenti, l'associazione del tutorato produrrebbe un vincolo di riferimento ricorsivo (scomodo)



nonostante non sia scorretta, non adotteremo questa soluzione



V Passo: Trad. delle Associazioni 1-m

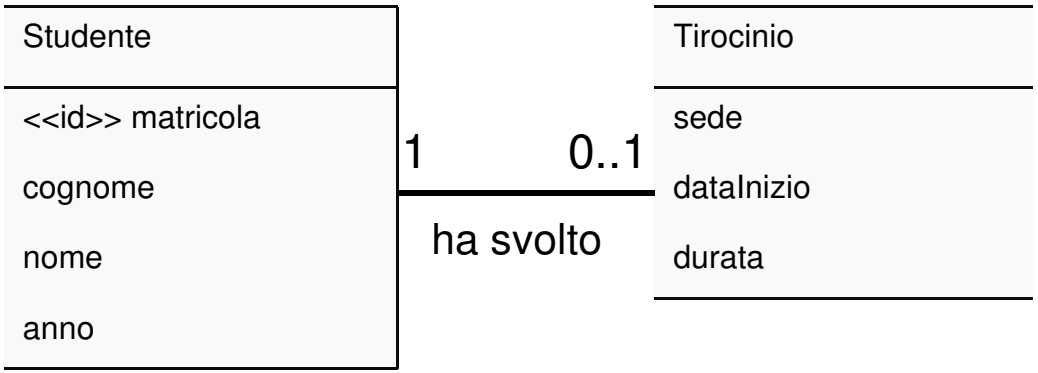


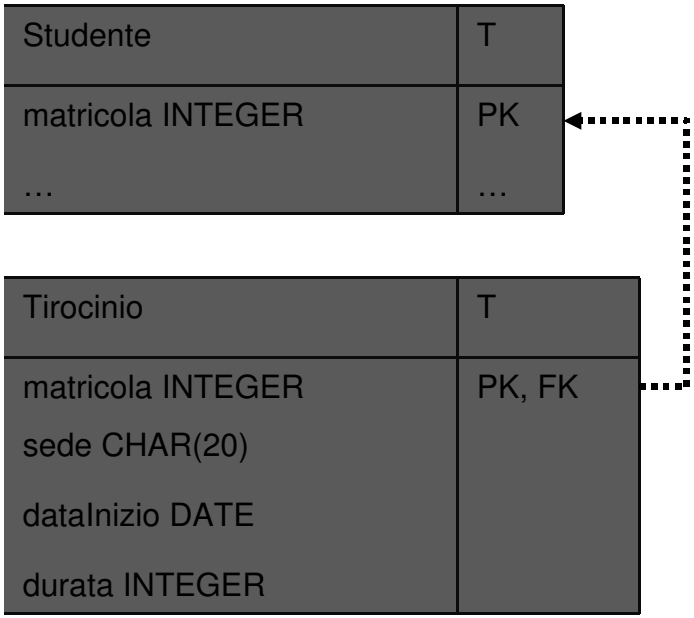
Tutorato	T
studente INTEGER	PK, FK
tutor INTEGER	PK, FK

Studente	T
matricola INTEGER	PK
cognome CHAR(20)	
nome CHAR(20)	
anno INTEGER	
relatore CHAR(4)	FK

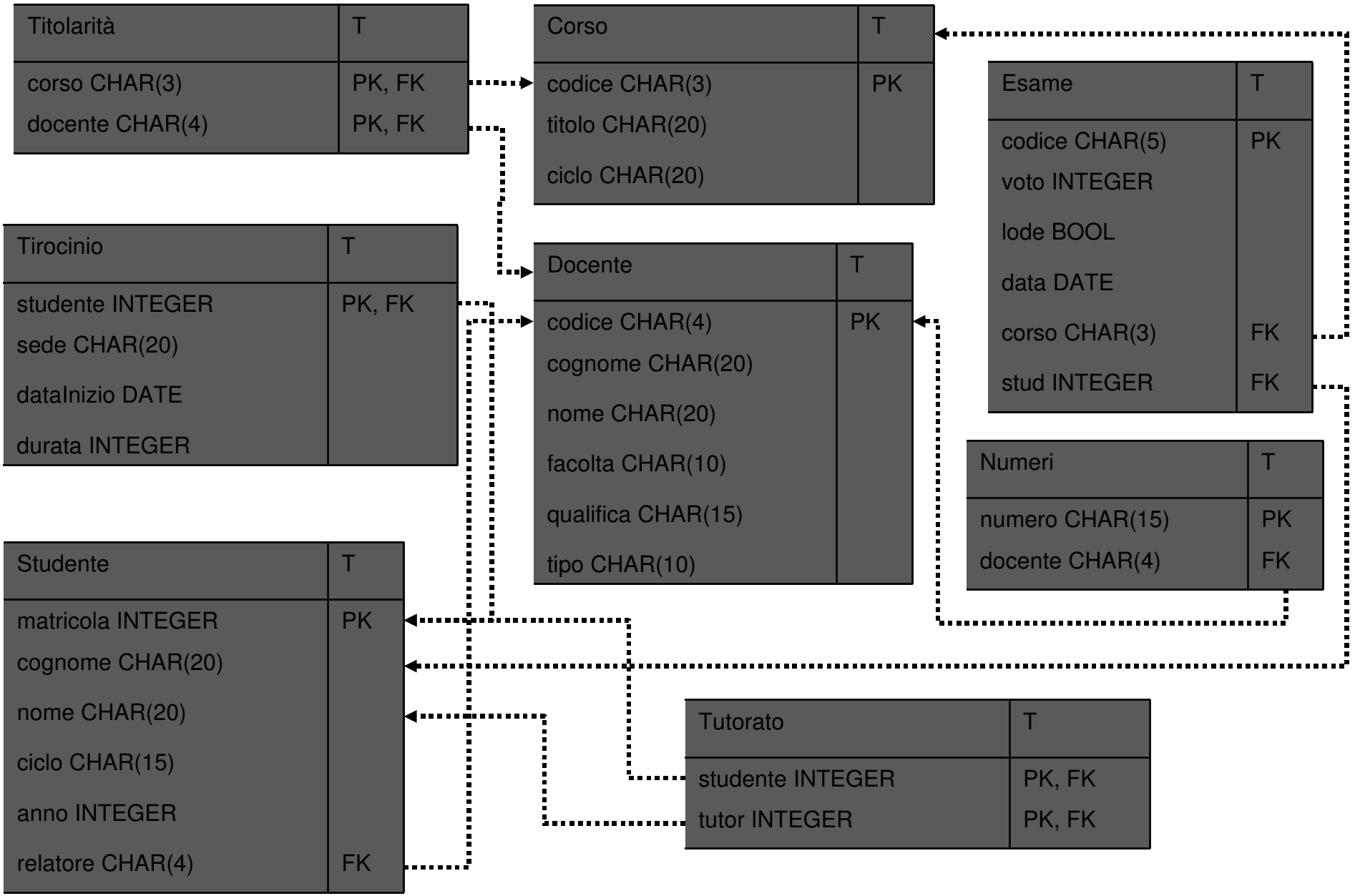
VI Passo: Trad. delle Associazioni 1-1

- **Discorso simile a quelle 1 a molti**
  - posso scegliere dove mettere la chiave est.
  - si preferisce un lato in cui la card. min. è 1





Progettazione Logica >> Algoritmo di Traduzione



Lo Schema Finale

```
CREATE TABLE Docente (
   codice char(4) PRIMARY KEY,
   cognome varchar(20) NOT NULL,
   nome varchar(20) NOT NULL,
   qualifica char(15),
   facolta char(10),
   tipo char(10) NOT NULL
);

CREATE TABLE Studente (
   matricola integer PRIMARY KEY,
   cognome varchar(20) NOT NULL,
   nome varchar(20) NOT NULL,
   ciclo char(20),
   anno integer,
   relatore char(4) REFERENCES Docente(codice),
   CHECK(relatore is NULL or anno=3 or ciclo='Laurea sp.')
);
```

Progettazione Logica >> Lo Schema Finale

```
CREATE TABLE Corso (
   codice char(3) PRIMARY KEY,
   titolo varchar(20) NOT NULL,
   ciclo char(20)
);

CREATE TABLE Esame (
    codice char(5) PRIMARY KEY,
    studente integer NOT NULL REFERENCES Studente(matricola)
      ON DELETE cascade ON UPDATE cascade,
    corso char(3) NOT NULL REFERENCES Corsi(codice),
    voto integer,
    lode bool,
    data date,
    CHECK (voto>=18 and voto<=30),
    CHECK (not lode or voto=30),
    UNIQUE (studente, corso)
);


CREATE TABLE Tutorato (
   studente integer REFERENCES Studente(matricola),
   tutor integer REFERENCES Studente(matricola),
   PRIMARY KEY (studente, tutor)
);
```

Progettazione Logica >> Lo Schema Finale

```
CREATE TABLE Numeri (
   numero char(9) PRIMARY KEY,
   docente char(4) REFERENCES Docente(codice)
);


CREATE TABLE Tirocinio (
   studente integer PRIMARY KEY REFERENCES Studente(matricola),
   sede char(20) NOT NULL,
   dataInizio date,
   durata integer
);


CREATE TABLE Titolarita (
   docente char(4) REFERENCES Docente(codice),
   corso char(3) REFERENCES Corso(codice),
   PRIMARY KEY (docente, corso)
);
```

Una Possibile Istanza

Docente

codice	cognome	nome	qualifica	facolta	tipo
FT	Totti	Francesco	ordinario	Ingegneria	interno
CV	Vieri	Christian	associato	Scienze	interno
ADP	Del Piero	Alessandro	null	null	supplente

Studente

matricola	cognome	nome	ciclo	anno	relatore
111	Rossi	Mario	laurea tr.	1	null
222	Neri	Paolo	laurea tr.	2	null
333	Rossi	Maria	laurea tr.	1	null
444	Pinco	Palla	laurea tr.	3	FT
77777	Bruno	Pasquale	laurea sp.	1	FT
88888	Pinco	Pietro	laurea sp.	1	CV

Progettazione Logica >> Lo Schema Finale

Corso

codice	titolo	ciclo
PR1	Programmazione I	laurea tr.
ASD	Algoritmi e Str. Dati	laurea tr.
INFT	Informatica Teorica	laurea sp.

Tutorato

studente	tutor
111	77777
222	77777
333	88888
444	88888

Esame

codice	studente	corso	voto	lode	data
pr101	111	PR1	27	false	2002-06-12
asd01	222	ASD	30	true	2001-12-03
inft1	111	INFT	24	false	2001-09-30
pr102	77777	PR1	21	false	2002-06-12
asd02	77777	ASD	20	false	2001-12-03
asd03	88888	ASD	28	false	2002-06-13
pr103	88888	PR1	30	false	2002-07-01
inft2	88888	INFT	30	true	2001-09-30

Progettazione Logica >> Lo Schema Finale

Tirocinio

studente	sede	dataInizio	durata
444	Microsoft	2002-05-15	3
77777	Microsoft	2002-05-15	3
88888	SOGEI	2002-09-01	3

Numeri

numero	docente
0971205145	FT
347123456	FT
0971205227	VC
0971205363	ADP
338123456	ADP

Titolarita

docente	corso
FT	PR1
CV	ASD
ADP	INFT
ADP	PR1
FT	ASD