



Informatica **U**manistica

# Basi di Dati

*La Progettazione Logica*



UNIVERSITÀ DI PISA

# Sommario

## ◆ Introduzione

- Il Processo di Progetto della BD

## ◆ Algoritmo di Progettazione Logica

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

# Introduzione

- ◆ **Siamo nella fase di progettazione**
  - si è conclusa (un'iterazione del)la fase di analisi
  
- ◆ **Attività da svolgere**
  - definire l'architettura dell'applicazione
  - definire la struttura e i metodi delle classi
  - definire la struttura della base di dati
  
- ◆ **Fase successiva: sviluppo**

## 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

# Il Processo di Progetto della BD

## ◆ Progettazione logica

- viene condotta sulla base di un semplice algoritmo sistematico

## ◆ Progettazione fisica

- attività mista: progettazione e “tuning”
- difficilmente sistematizzabile

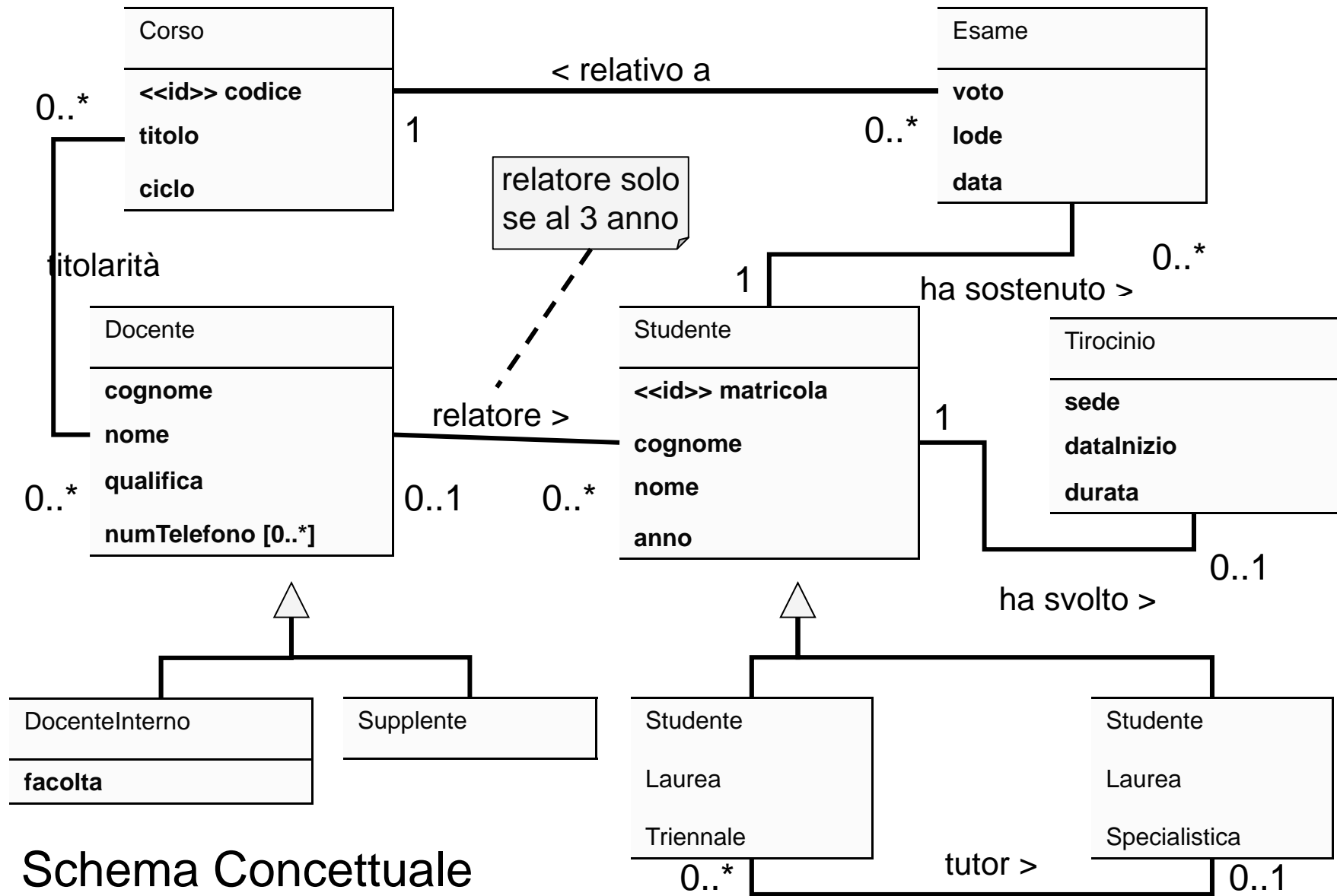
## ◆ In questa lezione

- ci concentriamo sulla progettazione logica

## Algoritmo di Progettazione Logica

- ◆ I passo: trad. iniziale delle classi non coinvolte in gerarchie
- ◆ II passo: trad. iniziale 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
- ◆ VII passo: introduzione di eventuali ulteriori vincoli
- ◆ VIII passo: progettazione degli schemi esterni

## 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));
```

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

Docente	T
codice CHAR(4)	PK
...	





## 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

Corso
<<id>> codice
titolo
ciclo

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

identificatore  
esplicito

Esame
voto
lode
data

Esame	T
codice CHAR(5)	PK
voto INTEGER	
lode BOOL	
data DATE	

identificatore  
sintetico

Tirocinio
luogo
dataInizio
durata

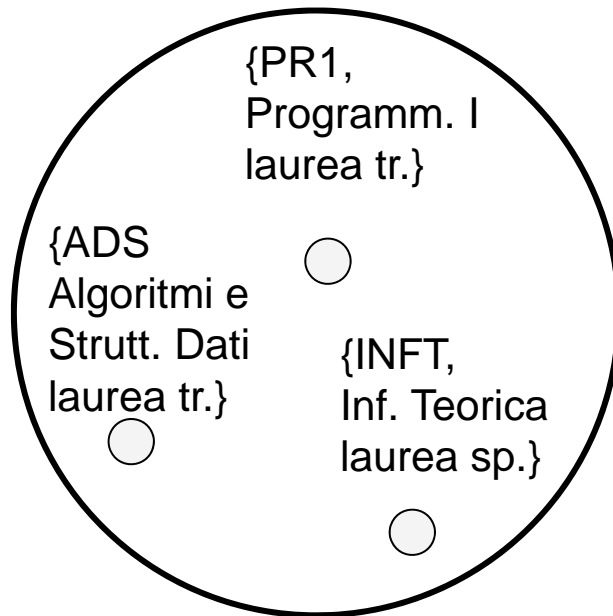
Tirocinio	T
matricola INTEGER	PK, FK
sede CHAR(20)	
dataInizio DATE	
durata INTEGER	

identificatore  
esterno

# I Passo: Traduzione delle Classi

Corso
<b>&lt;&lt;id&gt;&gt; codice</b>
<b>titolo</b>
<b>ciclo</b>

Corso	T
<b>codice CHAR(3)</b>	<b>PK</b>
<b>titolo CHAR(20)</b>	
<b>ciclo CHAR(20)</b>	

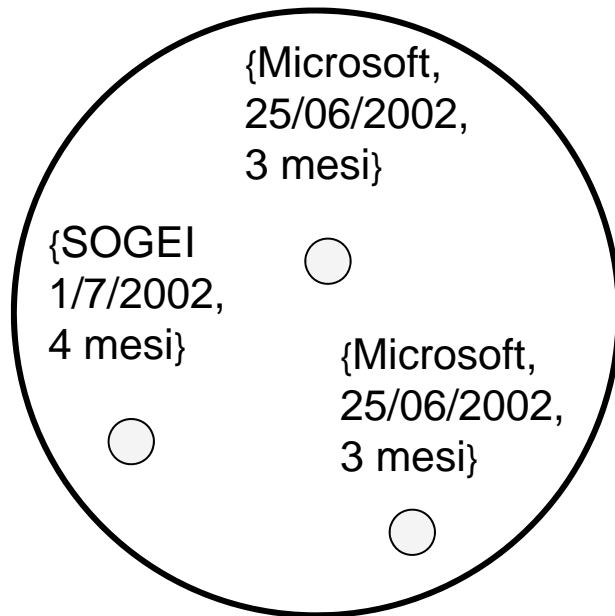


<u>codice</u>	titolo	ciclo	...
PR1	Programmazione I	laurea tr.	...
ASD	Algoritmi e Str. Dati	laurea tr.	...
INFT	Informatica Teorica	laurea sp.	...

# I Passo: Traduzione delle Classi

Tirocinio
<b>luogo</b>
<b>dataInizio</b>
<b>durata</b>

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



<u>studente</u>	sede	dataInizio	durata	...
444	Microsoft	2002-05-15	3	...
77777	Microsoft	2002-05-15	3	...
88888	Basica	2002-09-01	3	...

## Il 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

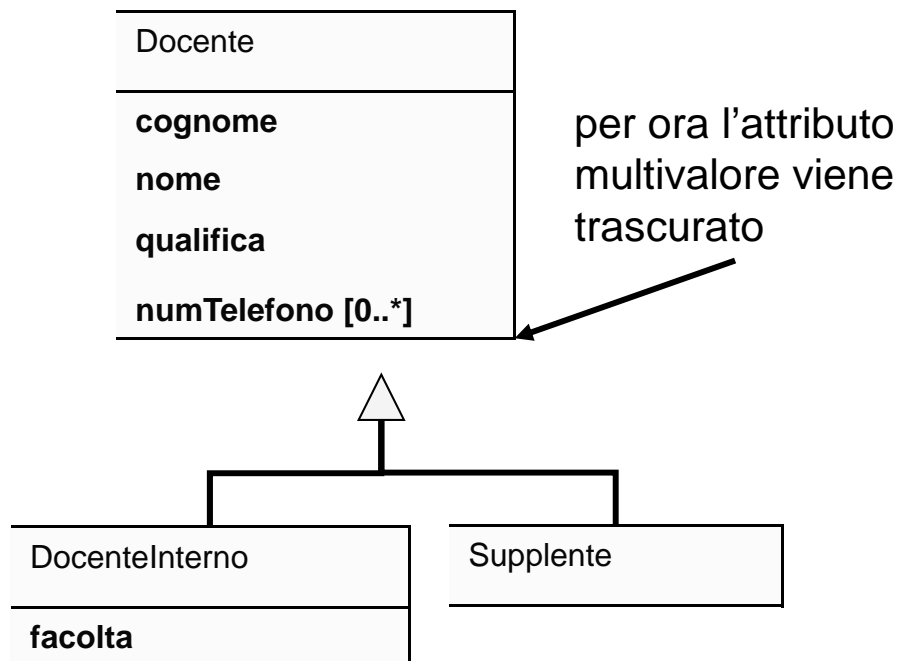
## Il Passo: Traduzione delle Gerarchie

### ◆ 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

## Il Passo: Traduzione delle Gerarchie

### ◆ I Soluzione: Solo il padre



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

“tipo” può valere:  
-interno oppure  
-supplente



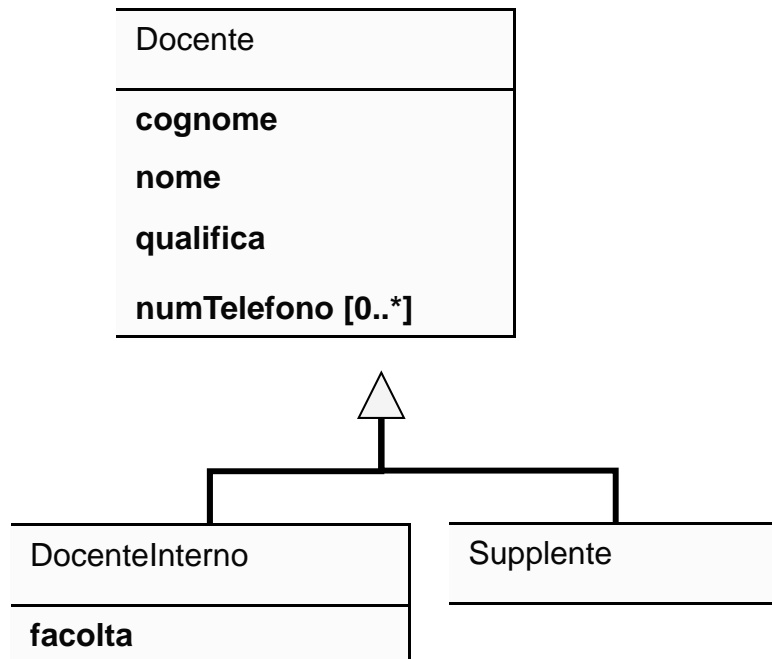
## Il Passo: Traduzione delle Gerarchie

### ◆ Il 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

## Il Passo: Traduzione delle Gerarchie

### ◆ Il Soluzione: Solo i figli



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

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

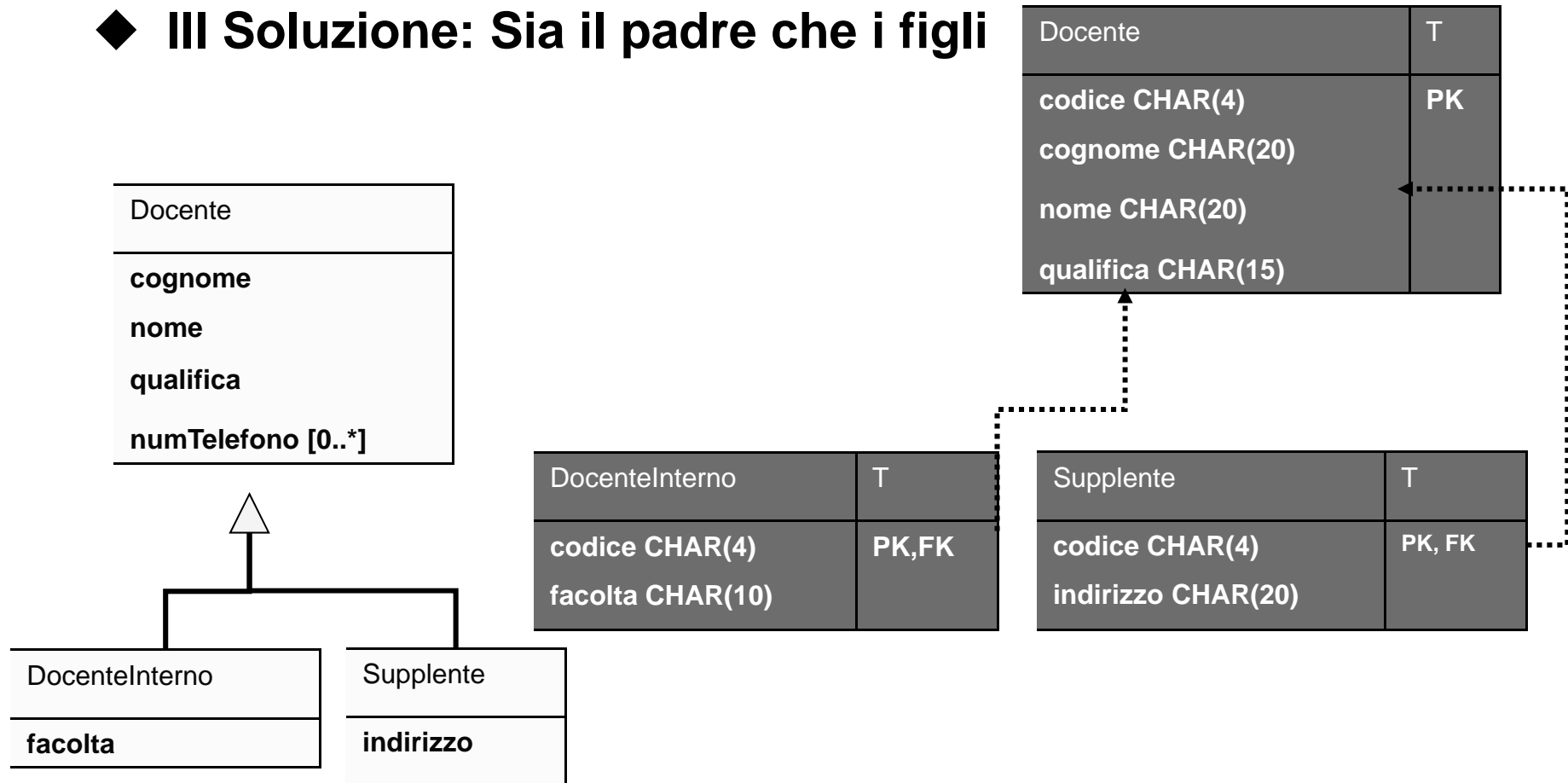
## Il Passo: Traduzione delle Gerarchie

### ◆ 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

## Il Passo: Traduzione delle Gerarchie

◆ III Soluzione: Sia il padre che i figli



## Il Passo: Traduzione delle Gerarchie

### ◆ III Soluzione: Sia il padre che i figli

Docente

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

DocenteInterno

<u>codice</u>	facolta
FT	Ingegneria
CV	Scienze

Supplente

<u>codice</u>	Indirizzo
ADP	Stadio delle Alpi, Torino

## Il Passo: Traduzione delle Gerarchie

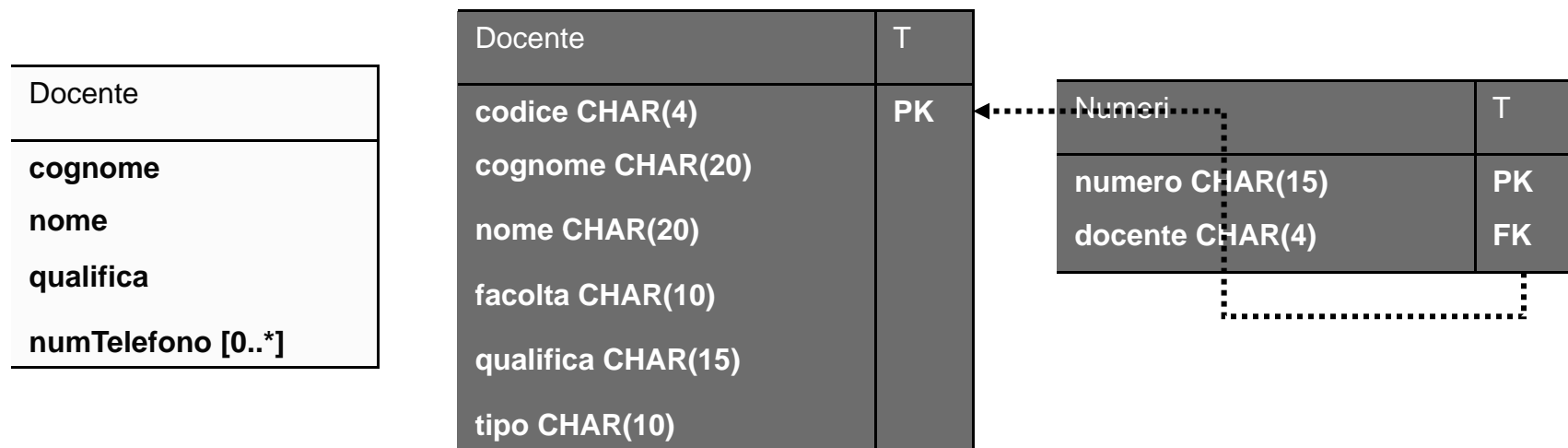
### ◆ Nel nostro esempio

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

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

## 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

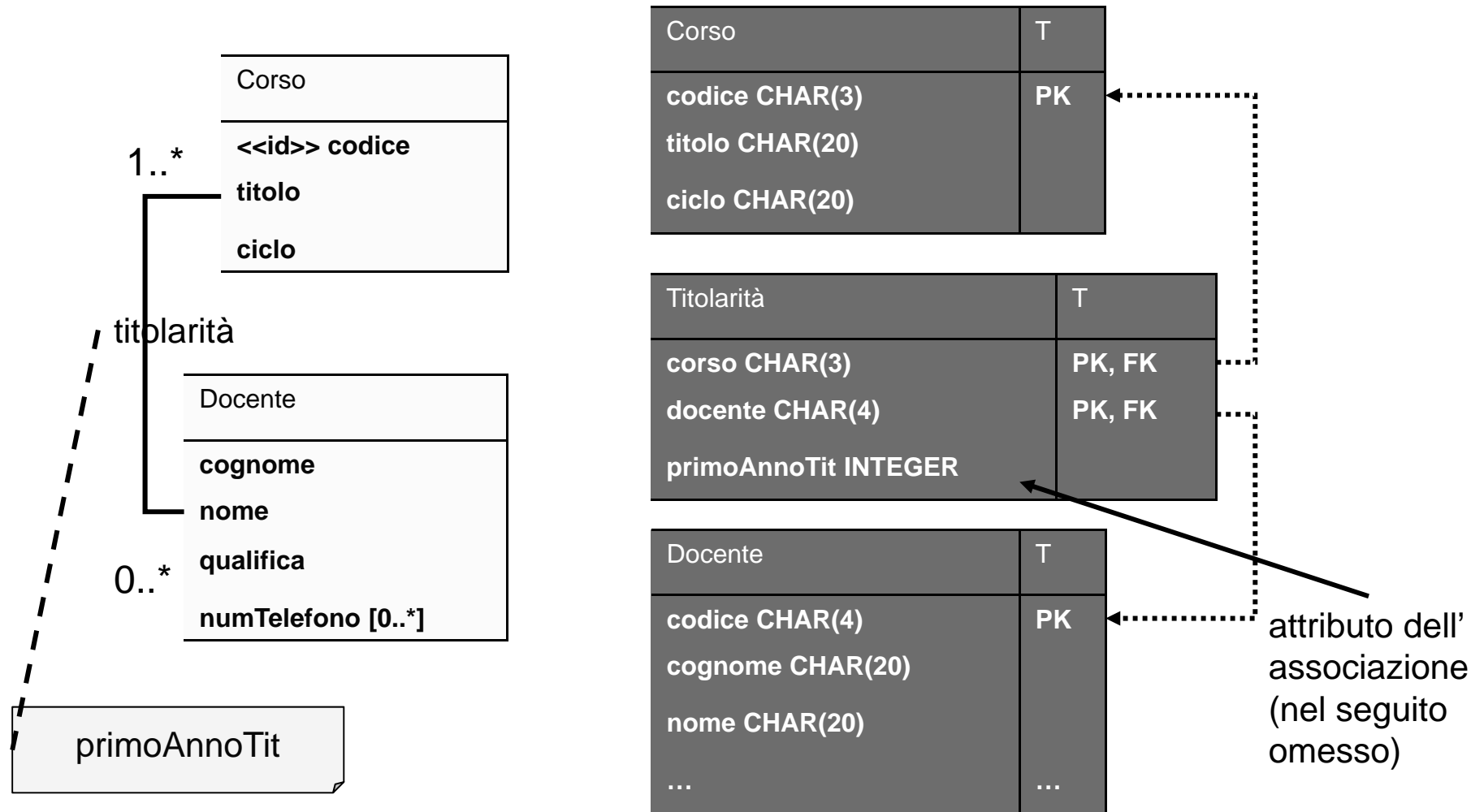


## IV Passo: Trad. delle Associazioni m-m

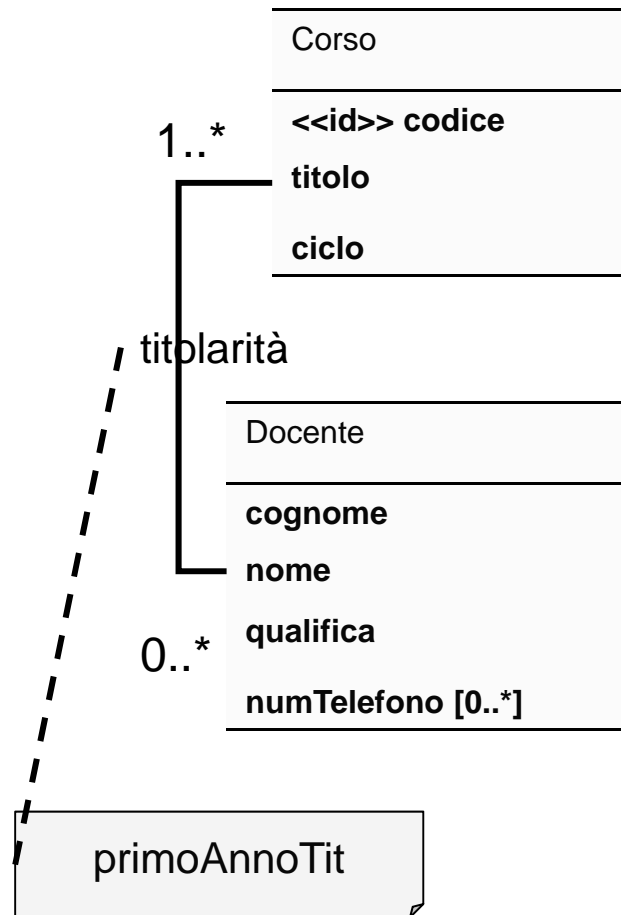
- ◆ **Ogni associazione molti a molti genera una tabella**
  - riferimenti (chiavi esterne) alle tabelle che traducono le classi coinvolte
  - eventuali attributi dell'associazione
  - la chiave della tabella deve includere le chiavi esterne



## IV Passo: Trad. delle Associazioni m-m



## IV Passo: Trad. delle Associazioni m-m



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

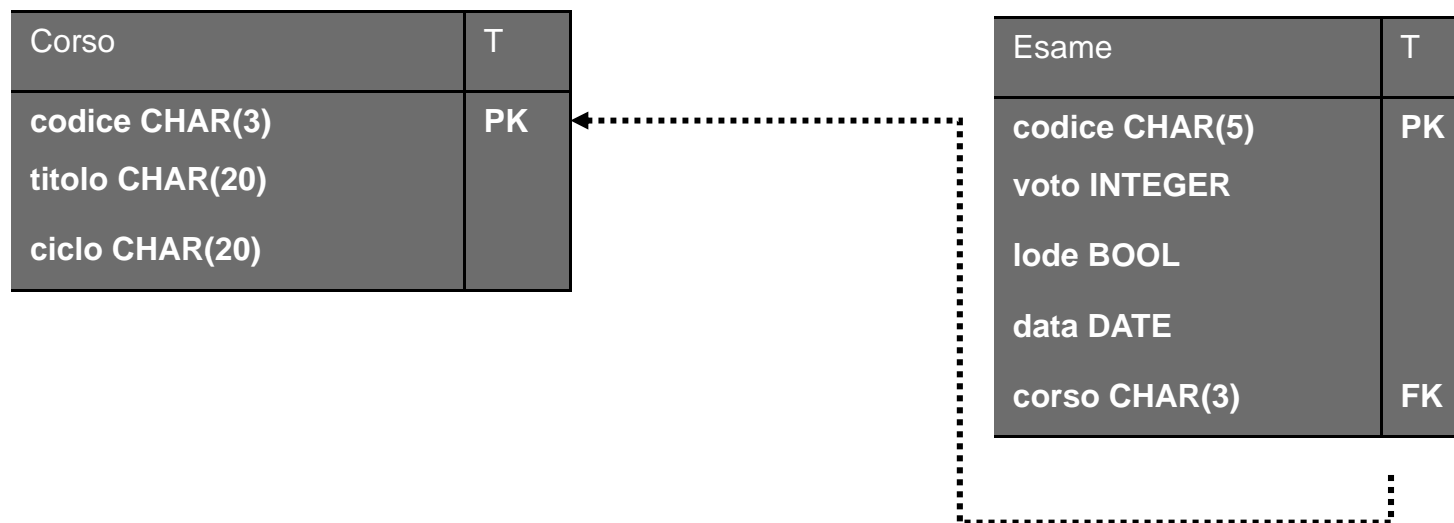
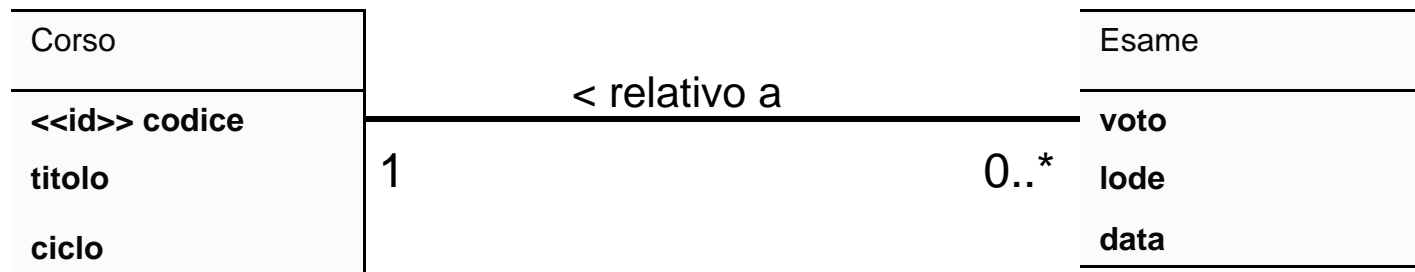
<u>docente</u>	<u>corso</u>	primoAnnoTit
FT	PR1	2001
CV	ASD	2002
FT	ASD	1999
...	...	

<u>codice</u>	cognome	nome	...
FT	Totti	Francesco	...
CV	Vieri	Christian	...
ADP	Del Piero	Alessandro	...

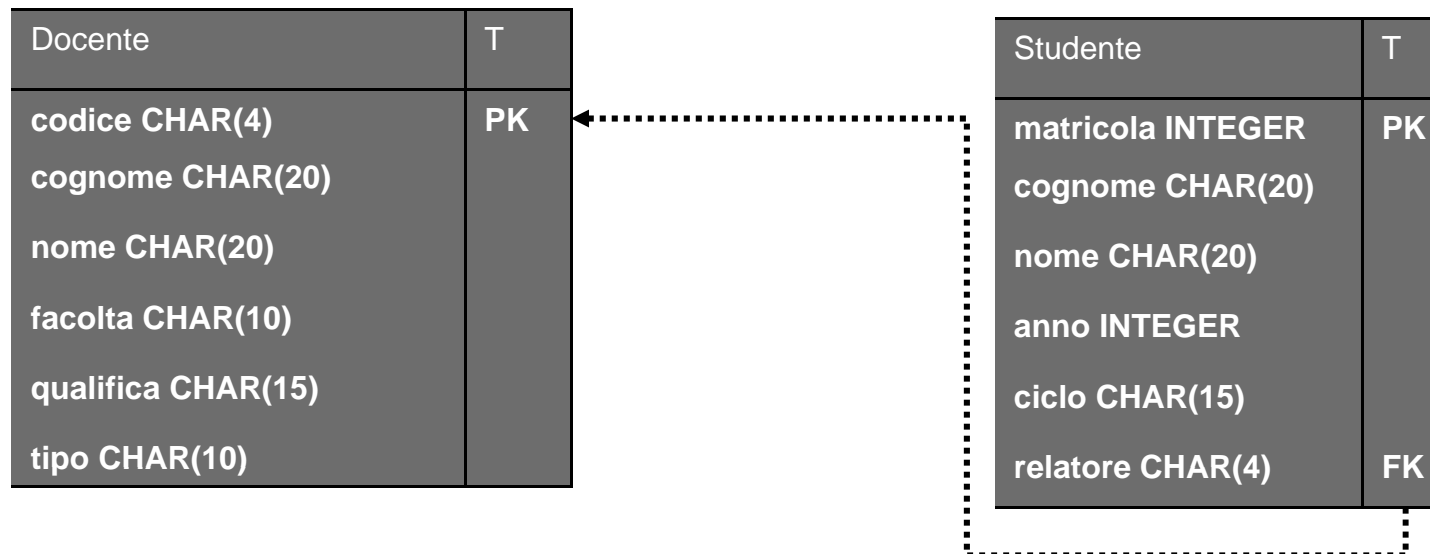
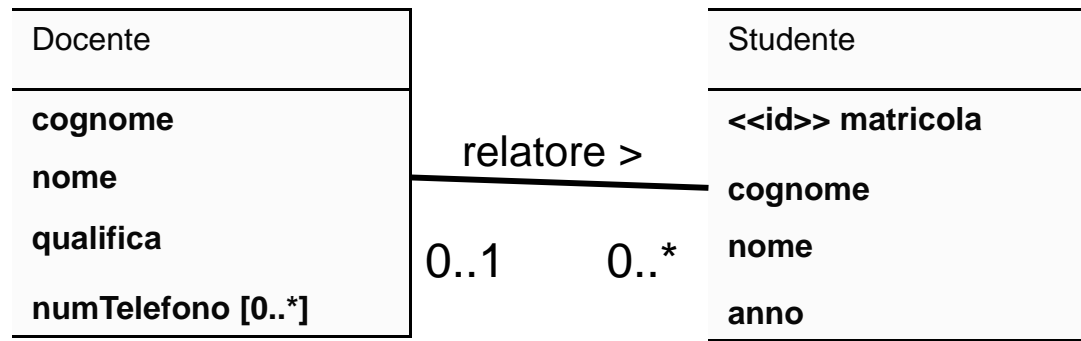
## V Passo: Trad. delle Associazioni 1-m

- ◆ **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

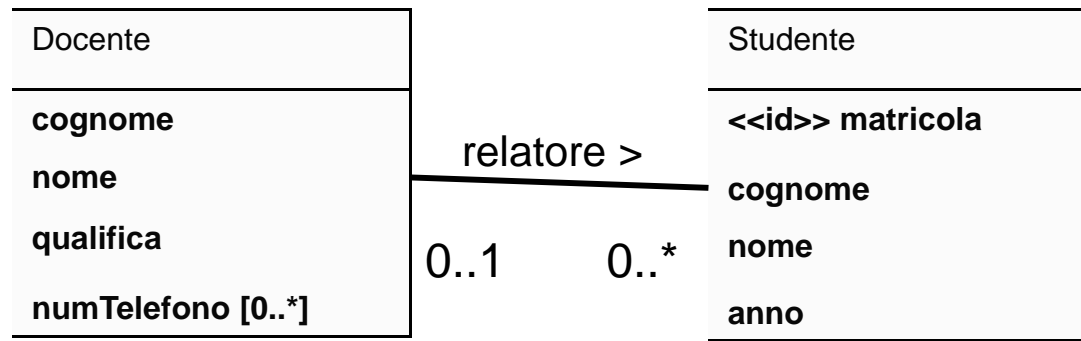
## V Passo: Trad. delle Associazioni 1-m



## V Passo: Trad. delle Associazioni 1-m



## V Passo: Trad. delle Associazioni 1-m



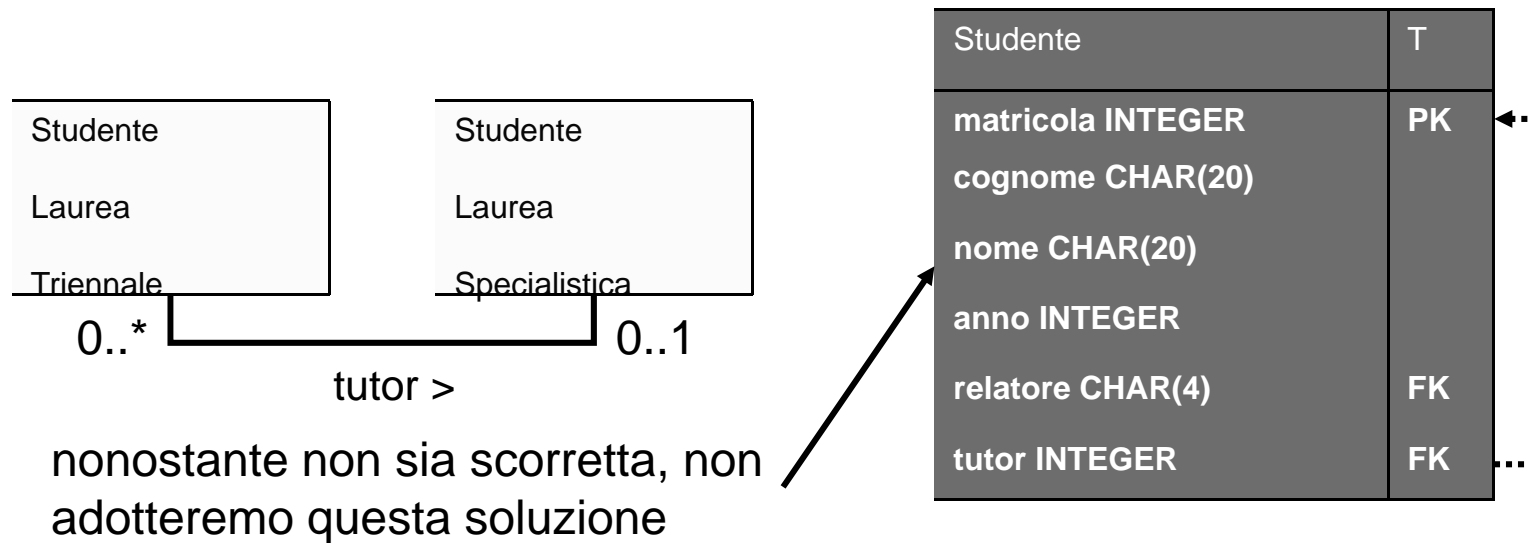
<u>codice</u>	cognome	nome	...
FT	Totti	Francesco	...
CV	Vieri	Christian	...
ADP	Del Piero	Alessandro	...

<u>matricola</u>	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

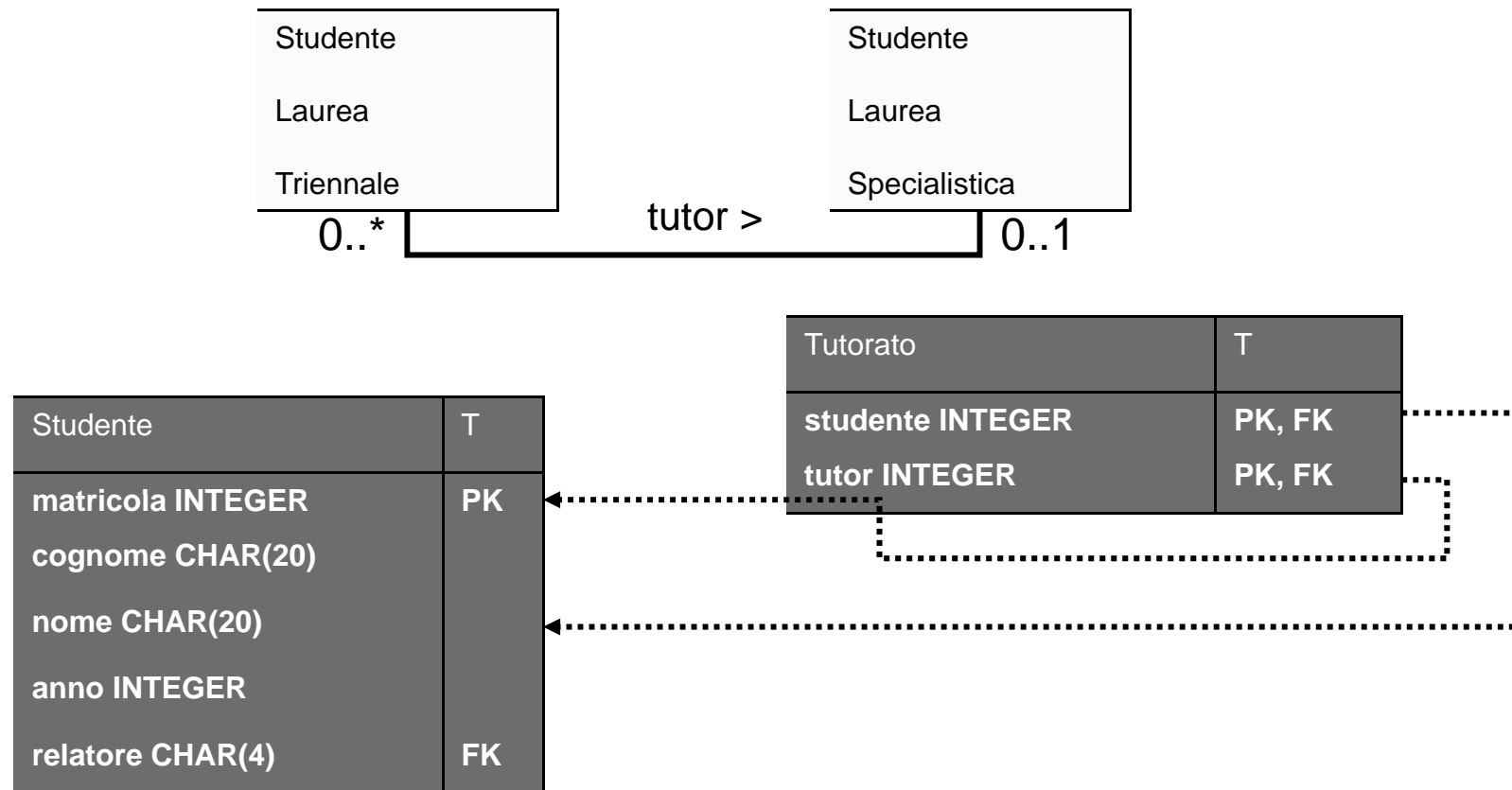
## V Passo: Trad. delle Associazioni 1-m

### ◆ Attenzione:

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



## V Passo: Trad. delle Associazioni 1-m

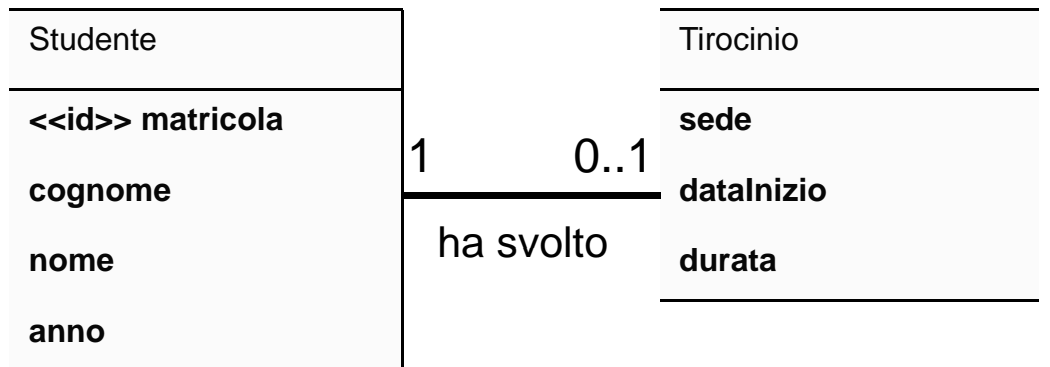




## VI Passo: Trad. delle Associazioni 1-1

### ◆ Discorso simile a quelle 1 a molti

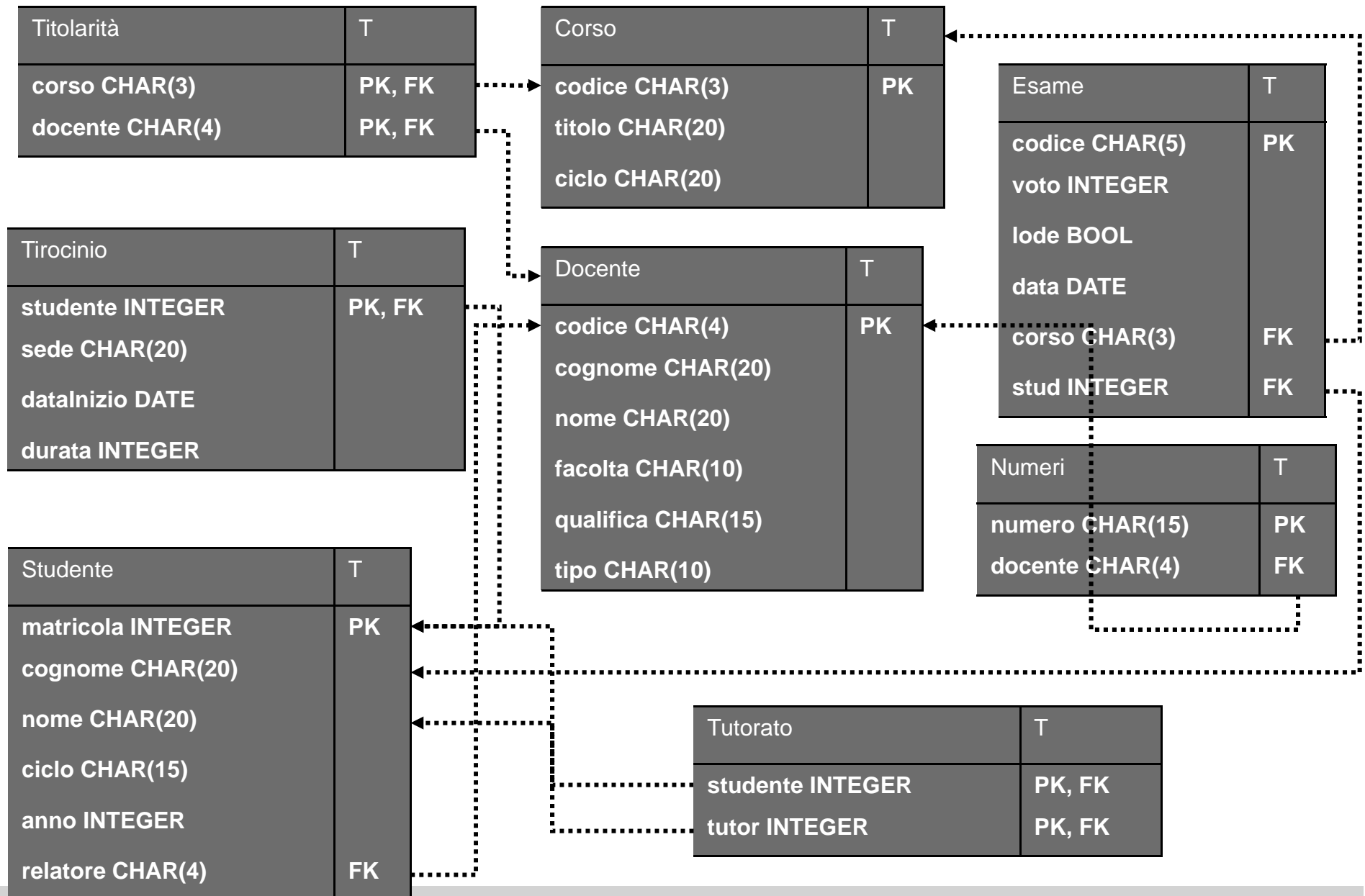
- posso scegliere dove mettere la chiave est.
- si preferisce un lato in cui la card. min. è 1



Studente	T
matricola INTEGER	PK
...	...

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

## Progettazione Logica >> Algoritmo di Traduzione



## VII Passo: Aggiunta di Vincoli Ulteriori

### ◆ A questo punto sono definite

- le tabelle
- gli attributi
- le chiavi primarie
- i vincoli di riferimento

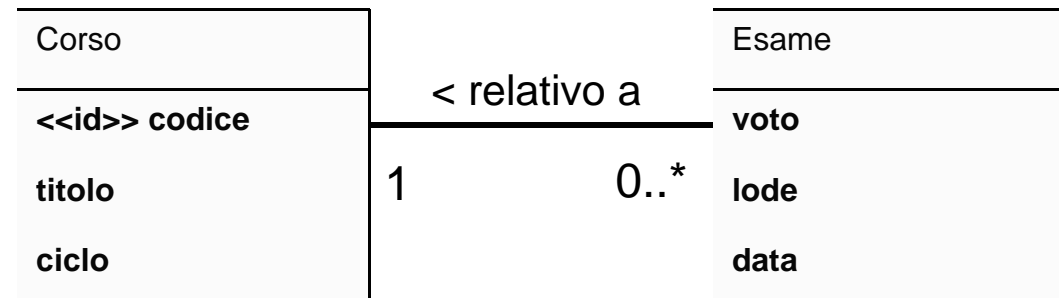
### ◆ Per ottenere lo schema conclusivo

- è possibile aggiungere altri vincoli (NOT NULL, DEFAULT, CASCADE, CHECK ecc.)

## VII Passo: Aggiunta di Vincoli Ulteriori

### ◆ In particolare

- le cardinalità minime danno origine a vincoli NOT NULL
- Esempio:



```
CREATE TABLE Esame (  
    codice char(5) PRIMARY KEY,  
    corso char(3) NOT NULL REFERENCES Corso(codice),  
    ...  
);
```

## 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

<u>codice</u>	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

<u>matricola</u>	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

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

### Tutorato

<u>studente</u>	<u>tutor</u>
111	77777
222	77777
333	88888
444	88888

### Esame

<u>codice</u>	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

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

### Numeri

<u>numero</u>	docente
0971205145	FT
347123456	FT
0971205227	VC
0971205363	ADP
338123456	ADP

### Titolarita

<u>docente</u>	<u>corso</u>
FT	PR1
CV	ASD
ADP	INFT
ADP	PR1
FT	ASD

## VIII Passo: Schemi Esterni

- ◆ **Dallo schema logico è necessario derivare gli schemi esterni**
  - eventuali viste
  - autorizzazioni agli utenti su tabelle e viste
  
- ◆ **Esempio: due categorie di utenti**
  - segreteria: accesso a tutti i dati
  - docenti: accesso a dati ristretti sugli esami (es: una vista “EsameSenzaVoto”)

# Sommario

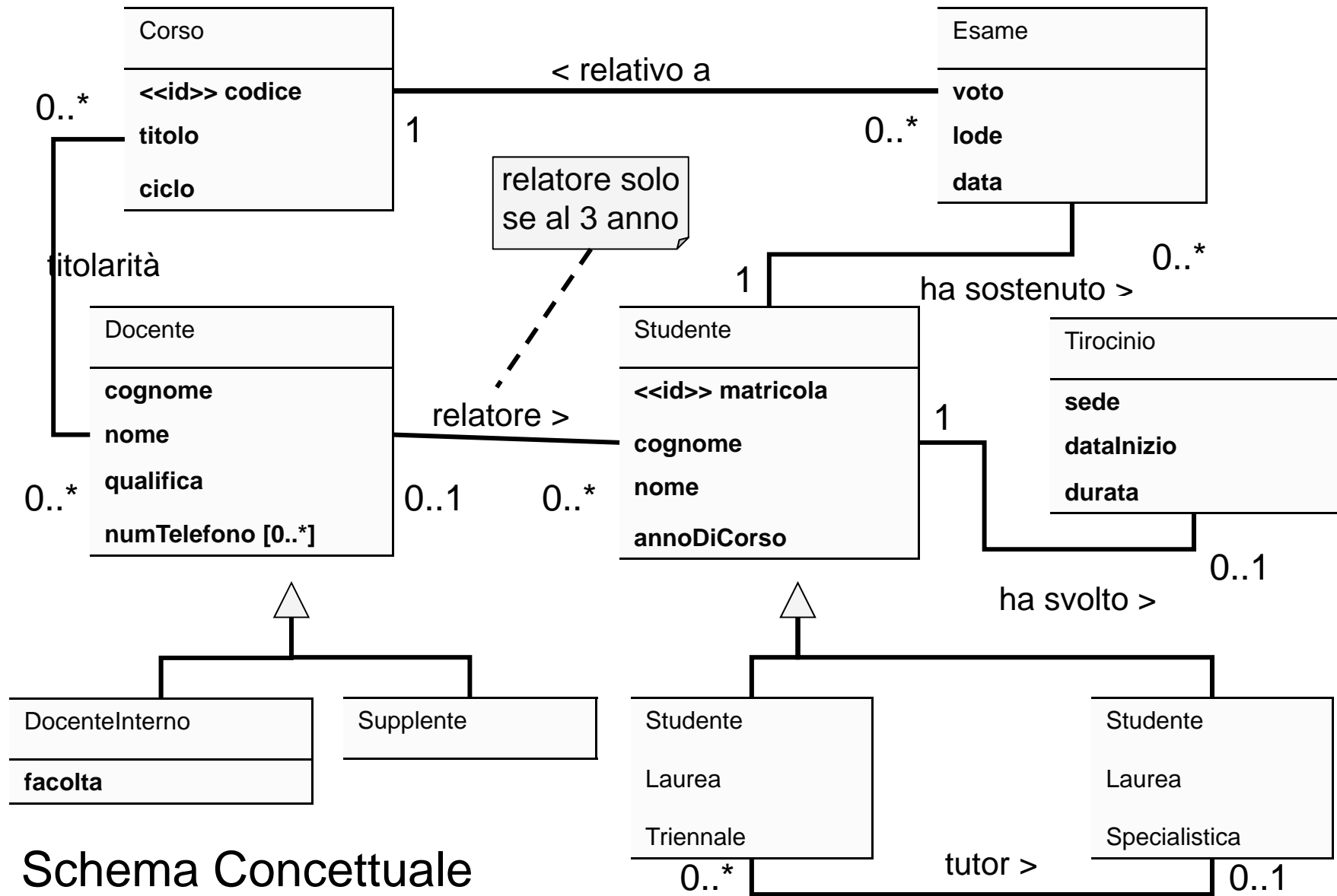
## ◆ Introduzione

- Il Processo di Progetto della BD

## ◆ Algoritmo di Progettazione Logica

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

## Progettazione Logica >> Modello Concettuale



Schema Concettuale