

Basi di Dati - IV

Corso di Laurea in Informatica
Anno Accademico 2024/2025

Alessandra Raffaetà
raffaeta@unive.it

Progettazione Logica

- Il **modello dei dati relazionale** (Edgar F. Codd, 1970)
- **Trasformazione** dal modello **concettuale ad oggetti** al modello **logico relazionale**
- **Algebra relazionale**

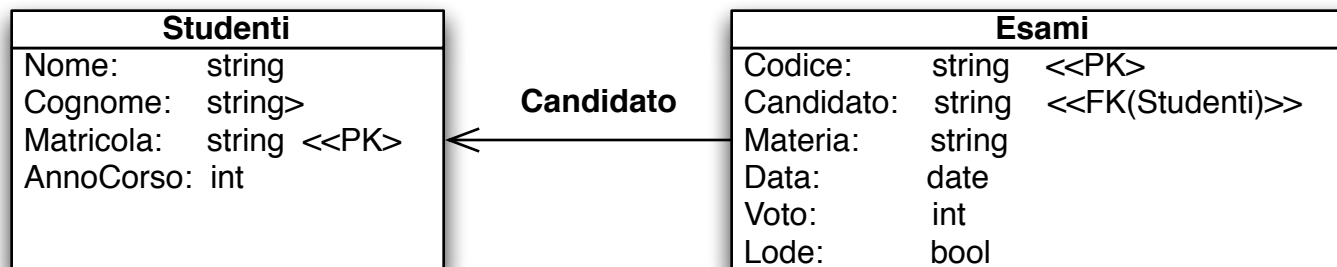
Il Modello Relazionale

- **Collezioni** come **relazioni** (**tabelle**)

Studenti	
Nome:	string
Cognome:	string
Matricola:	string
AnnoCorso:	int

Esami	
Codice:	string
Candidato:	string
Materia:	string
Data:	date
Voto:	int
Lode:	bool

- **Associazioni** tramite **chiavi**



- I meccanismi per definire una base di dati con il modello relazionale sono l'**ennupla** e la **relazione**.
- Dal punto di vista matematico
 - **relazione** $R \subseteq D1 \times D2 \times \dots \times Dn$
 - $D1, \dots, Dn$ domini
 - **ennupla** $\langle d1, \dots, dn \rangle \in R$
 - $d1 \in D1, \dots, dn \in Dn$
- in Informatica si associa un'etichetta distinta a ciascun dominio $D1, \dots, Dn$ (record!)

- **Tipo ennupla T**: insieme finito di coppie (Attributo, **Tipo primitivo**):
(A1: T1, ..., An: Tn)
- **Tipo relazione o tipo insieme di ennuple**: Se T è un tipo ennupla, allora {T} è un tipo relazione.
- **Schema di relazione**
 $R : \{ T \}$ (T tipo ennupla, {T} tipo relazione)
- Spesso scriveremo R(T) invece di R:{T}.
- **Istanza** di uno schema R:{T} o **relazione**: insieme finito di ennuple di tipo T.
 - **cardinalità**: numero delle sue ennuple.
- **Schema relazionale di una BD**:
 - insieme di schemi di relazione **Ri:{Ti}**;
 - vincoli di integrità

- Studenti (Nome: string, Cognome: string, Matricola: string, Anno:int)

Nome	Cognome	Matricola	Anno
Paolo	Verdi	71523	2005
Anna	Rossi	76366	2006
Giorgio	Zeri	71347	2005

Studenti

- se non interessa evidenziare il tipo degli attributi scriviamo
Studenti(Nome, Cognome, Matricola, Anno)

Schema relazionale:

Studenti (Nome: string, Cognome: string, Matricola: string, Anno: int)

Esami (Codice: string ,Materia: string, Candidato: string, Data: string, Voto: int, Lode:char)

Studenti

Nome	Cognome	Matricola	Anno
Paolo	Verdi	71523	2005
Anna	Rossi	76366	2006
Giorgio	Zeri	71347	2005

Esami

Codice	Materia	Candidato	Data	Voto	Lode
B112	BD	71523	08.07.06	27	N
F31	FIS	76366	08.07.07	26	N
B247	CN	71523	28.12.06	30	S

- Considereremo
 - **chiavi**
 - **chiavi esterne**
 - **valori non nulli**

r è un'**istanza valida** di uno schema di relazione R se rispetta tutti i



vincoli definiti su R .

- **Superchiave in R:** sottoinsieme X di attributi di uno schema di relazione R tale che il valore degli attributi in X determina univocamente una ennupla
 - Esempio: (Matricola) e (Cognome, Matricola) sono superchiavi in:
Studenti(Nome, Cognome, Matricola, Anno)
- **Chiave:** superchiave **minimale**; gli attributi che appartengono ad una chiave sono detti **primi**
 - Esempio: Matricola
- **Chiave primaria:** una delle chiavi, in genere di lunghezza minima
- Altre chiavi sono indicate con **<<UNIQUE>>** oppure **<<CK>>**

● Chiave esterna in R

- insieme di attributi $X = \{A_1, \dots, A_n\}$ di R che riferisce la chiave primaria $Y = \{B_1, \dots, B_n\}$ di S:
- per ogni ennupla r in R esiste una ennupla s in S t.c.
 $r.X = s.Y$ (r “riferisce” s). [integrità referenziale]

● Associazioni

- realizzate con il meccanismo di chiave

Schema:

Studenti(Nome: string, Cognome: string, Matricola: string, Anno: int)

Esami(Codice: string, Materia: string, Candidato*: string, Data: string, Voto: int, Lode:char)

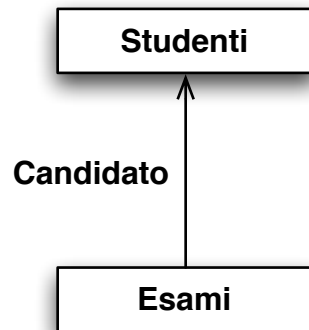
● Associazione:

Studenti

Nome	Cognome	<u>Matricola</u>	Anno
Paolo	Verdi	71523	2005
Anna	Rossi	76366	2006
Giorgio	Zeri	71347	2005

Esami

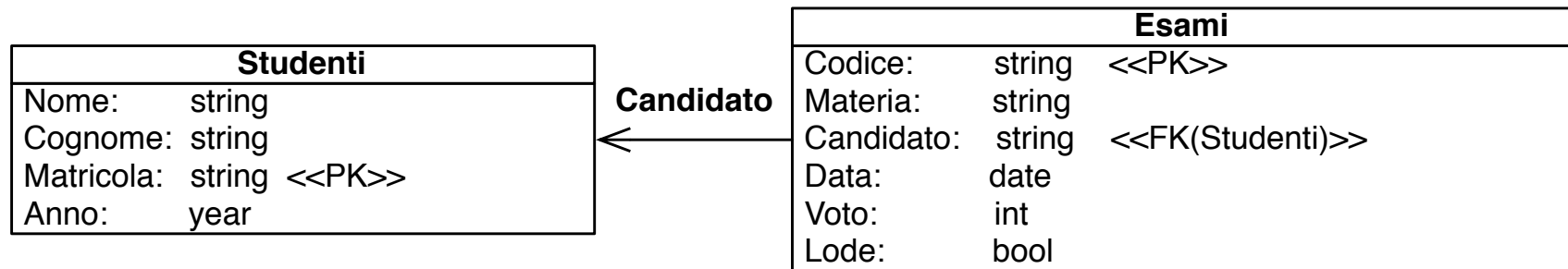
<u>Codice</u>	Materia	<u>Candidato*</u>	Data	Voto	Lode
B112	BD1	71523	08.07.06	27	N
F31	FIS	76366	08.07.07	26	N
B247	BD2	71523	28.12.06	30	S



- Studenti(Nome, Cognome, Matricola, Anno, Esame*)
Esami(Codice, Materia, Data, Voto, Lode)
 - Studenti(Nome, Cognome, Matricola, Anno, Esame*)
Esami(Codice, Materia, Data, Voto, Lode)
 - Studenti(Nome, Cognome, Matricola, Anno)
Esami(Codice, Materia, Data, Voto, Lode)
StudentiEsami(Esame*, Candidato*)
 - Studenti(Nome, Cognome, Matricola, Anno)
Esami(Materia, Crediti)
ProvaEsame(Codice, Esame*, Candidato*, Data, Voto, Lode)
- Quali sono sensate?

- Un attributo può avere valore non specificato (proprietà parziali), per varie ragioni:
 - non applicabile
 - sconosciuto
- si usa **NULL**
- Es.: Per lo schema di relazione nella biblioteca
Utenti(Nome, Cognome, CodiceFiscale, ...)
CodiceFiscale per un ospite potrebbe non aver valore perché nel paese di provenienza il CF non si usa o perché il CF non è noto nel momento della creazione dell'utente.

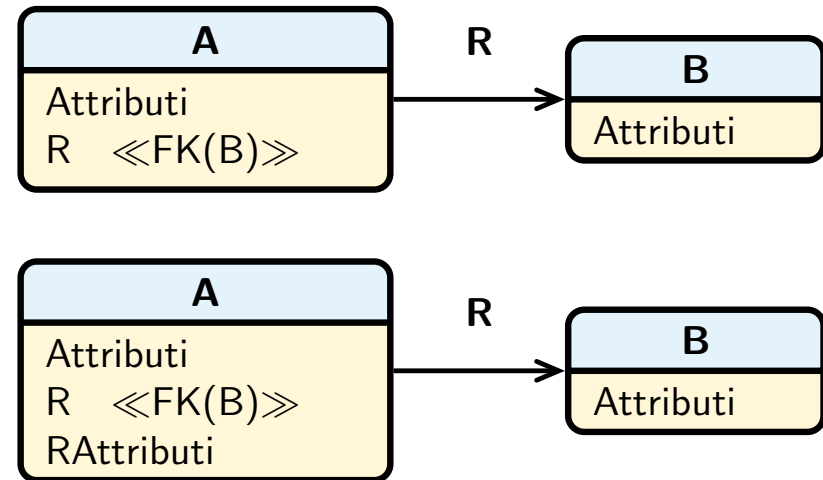
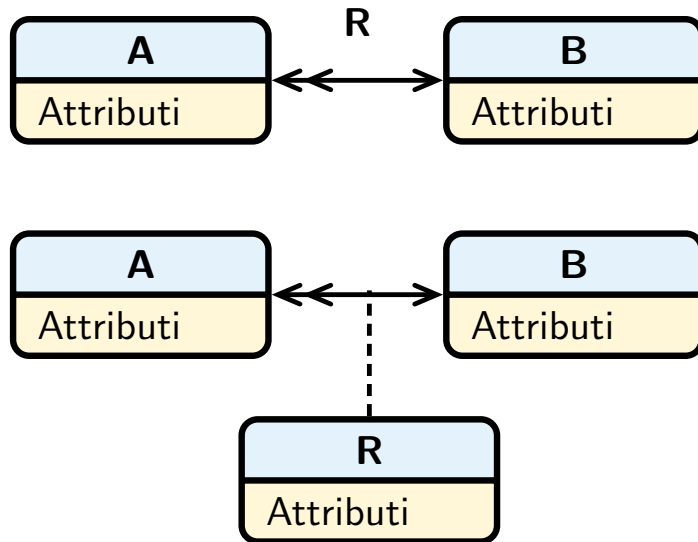
- Negli schemi relazionali si può imporre il vincolo **NOT NULL** per un attributo
- Gli attributi della **chiave primaria** (e delle chiavi in generale) devono assumere valori **non nulli**
- Una **chiave esterna** può avere valore nullo se rappresenta una associazione parziale.



Dal Modello a Oggetti al Modello Relazionale

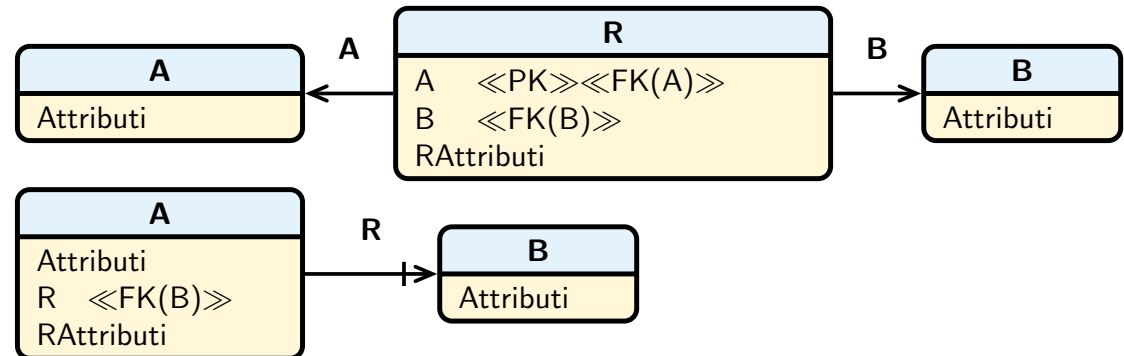
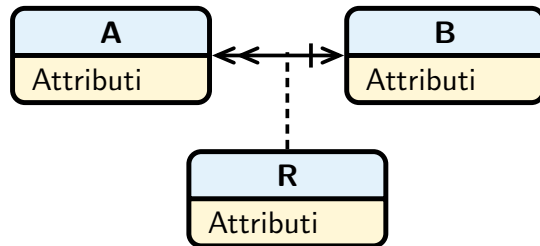
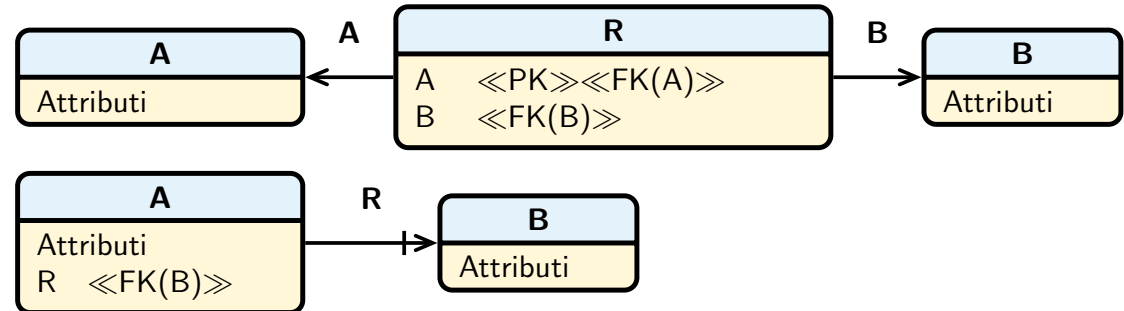
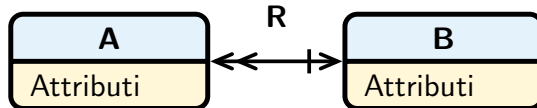
- Trasformazione per passi:
 1. associazioni molti a uno (e uno a uno)
 2. associazioni molti a molti
 3. gerarchie di inclusione
 4. identificazione chiavi primarie
 5. attributi multivalore
 6. attributi composti

- Associazioni N:1 (univoche e totali)



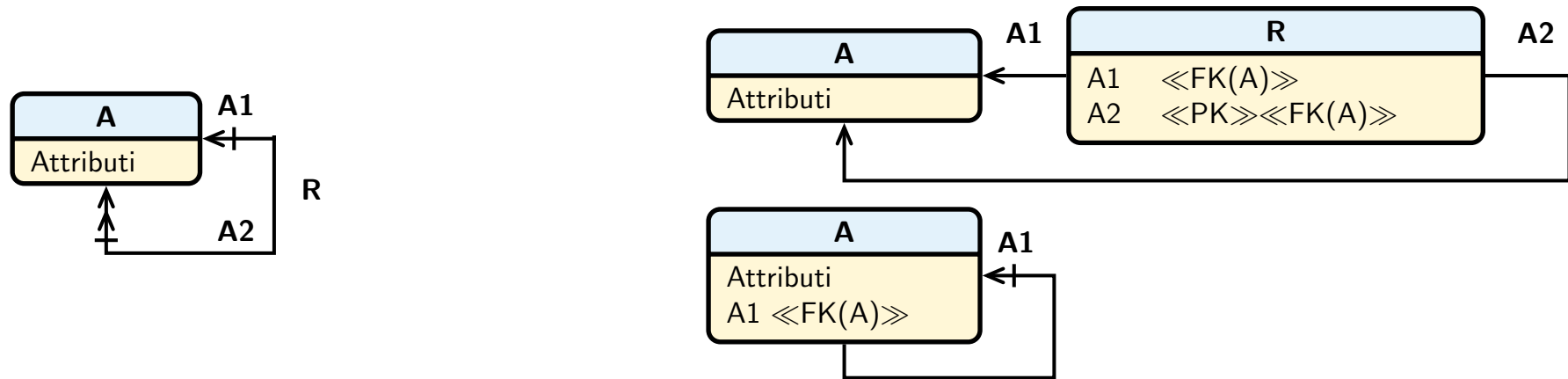
- Prestiti <<-|----> Utenti

- Associazioni N:1 (univoche e parziali)

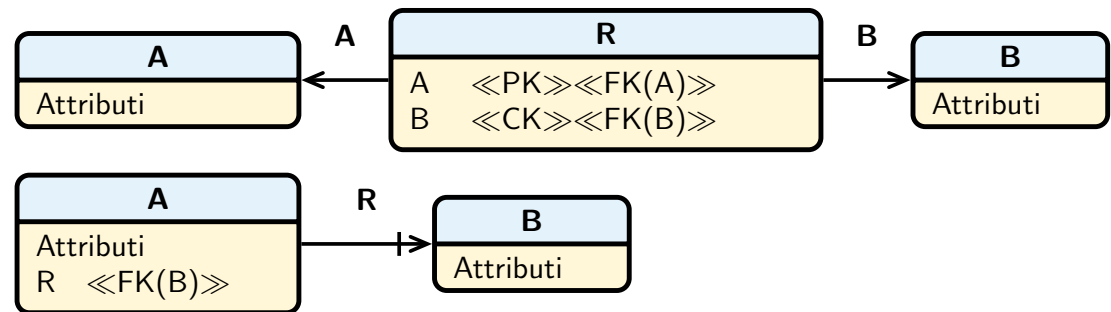
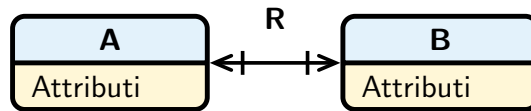
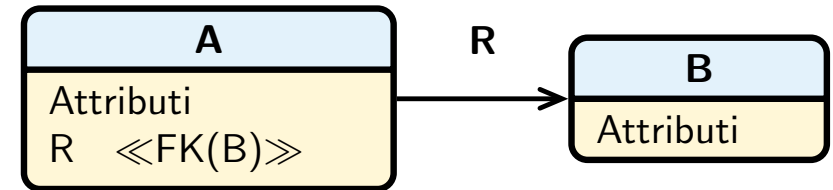
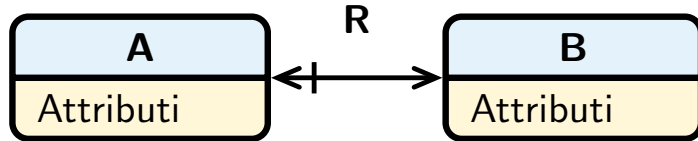


- EsamiEsterni <<-|---|-> EsamiInterni (attributo: Colloquio)

- Associazioni N:1 (ricorsive)

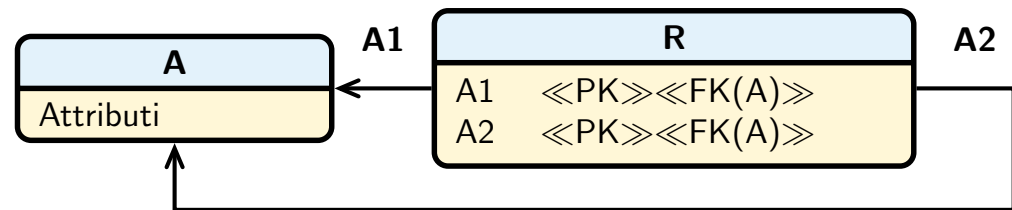
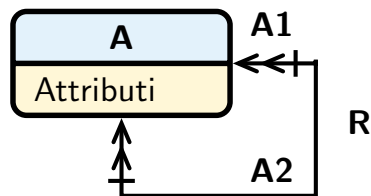
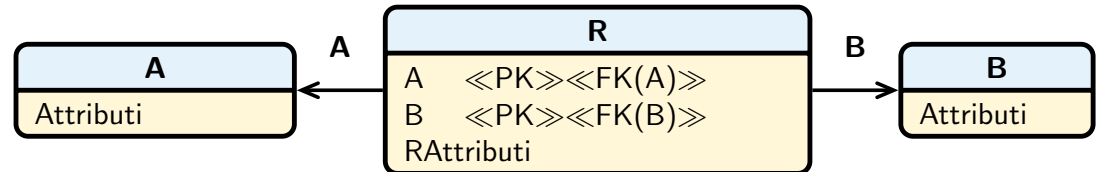
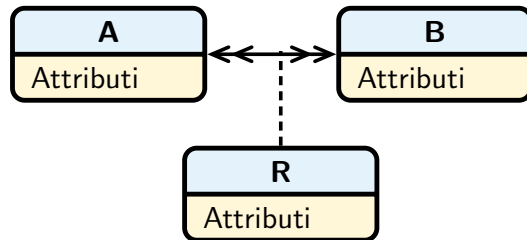
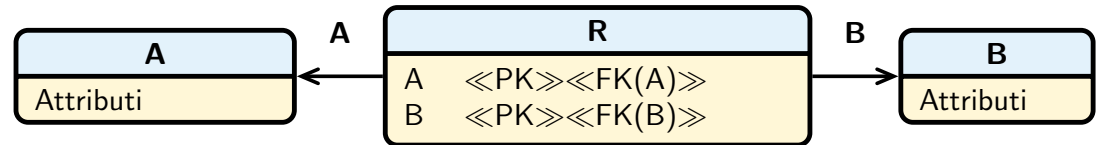
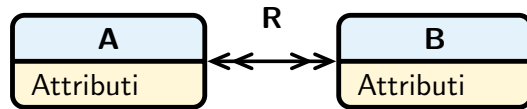


- Associazioni 1:1 (univoche con inversa univoca)

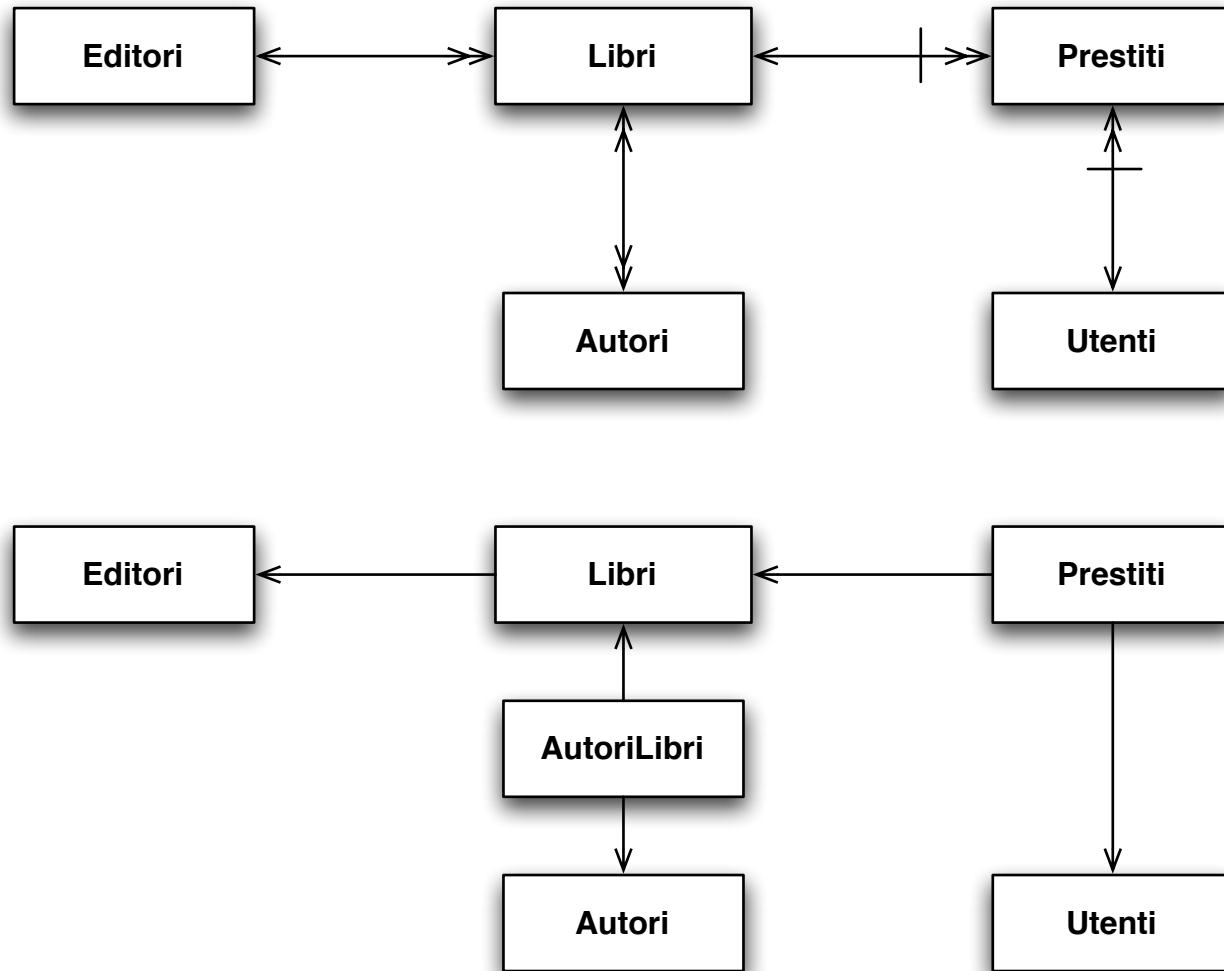


- Es.: Domande Trasferimento <---|> Pratiche Trasferimento

- Associazioni N:M (multivalore con inversa multivalore)



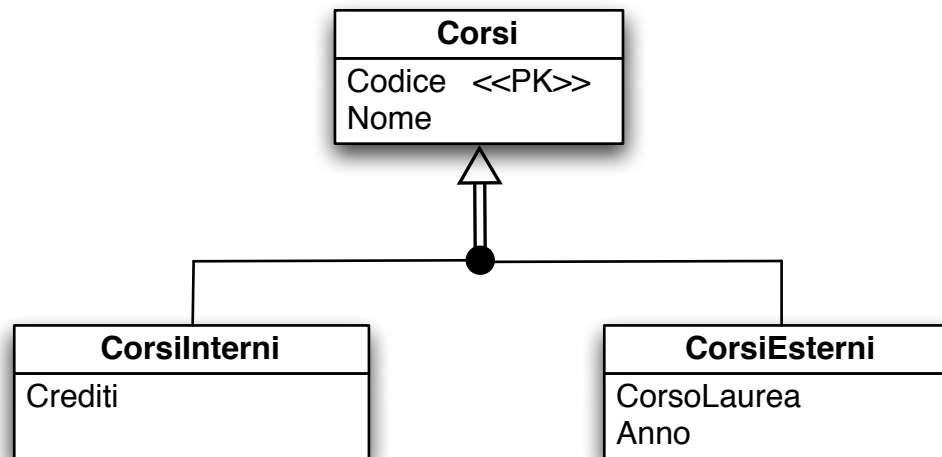
- totalità non rappresentabile



- Data la classe A (attr. X_A , chiave K_A) con sottoclassi B (attr. X_B) e C (attr. X_C)
- Tre possibili soluzioni
 - Relazione unica
 - $R(X_A, X_B, X_C, \text{Discr})$
 - Discr indica la classe alla quale appartiene l'elemento
 - X_B e X_C possono avere valore nullo
 - Partizionamento verticale
 - $R_A(X_A)$: tutti gli elementi di A,
 - $R_B(X_B, K_A)$: attributi propri per gli elementi di B
 - $R_C(X_C, K_A)$: attributi propri per gli elementi di C

- Partizionamento orizzontale
 - $R_A(X_A)$: solo gli elementi di A - (B \cup C)
 - $R_B(X_A, X_B)$: elementi di B (tutti gli attributi)
 - $R_C(X_A, X_C)$: elementi di C (tutti gli attributi)

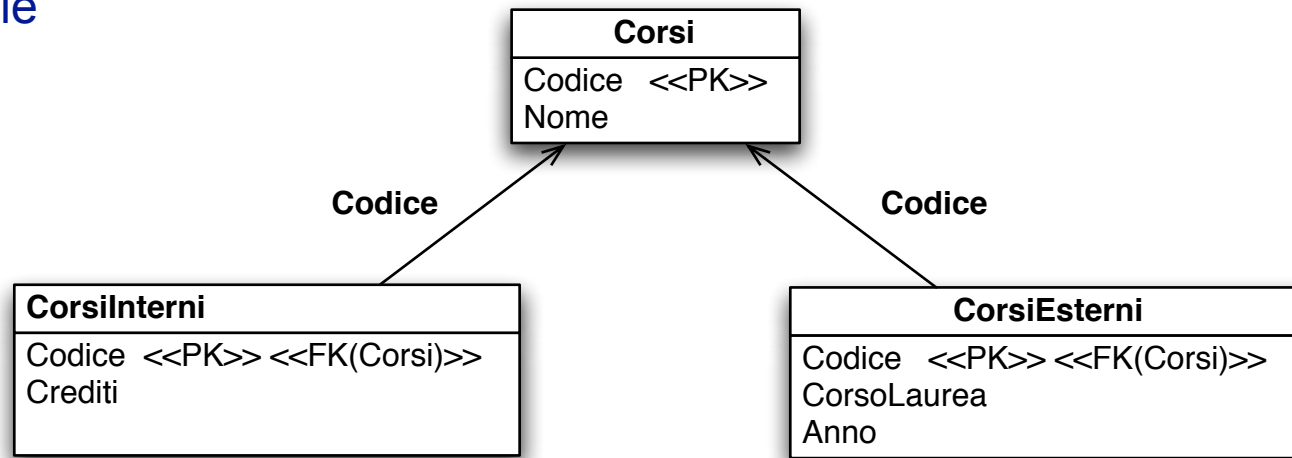
- Si consideri la gerarchia seguente:



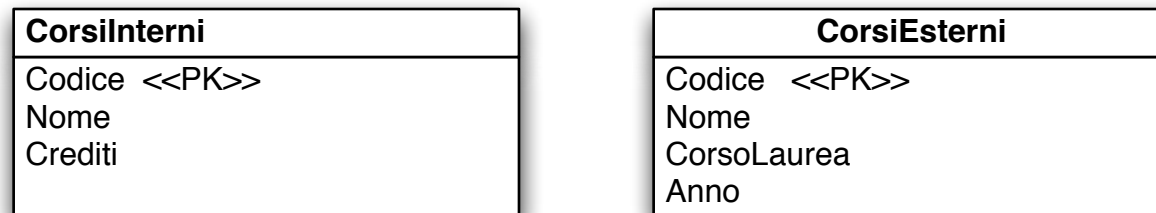
- L'attributo **InterniEsterni** svolge il ruolo di discriminatore

Corsi	
Codice	<<PK>>
Nome	
Crediti	
CorsoLaurea	
Anno	
InterniEsterni	

● Verticale



● Orizzontale



- **Relazione unica**

- conveniente se le sottoclassi differiscono per pochi attributi

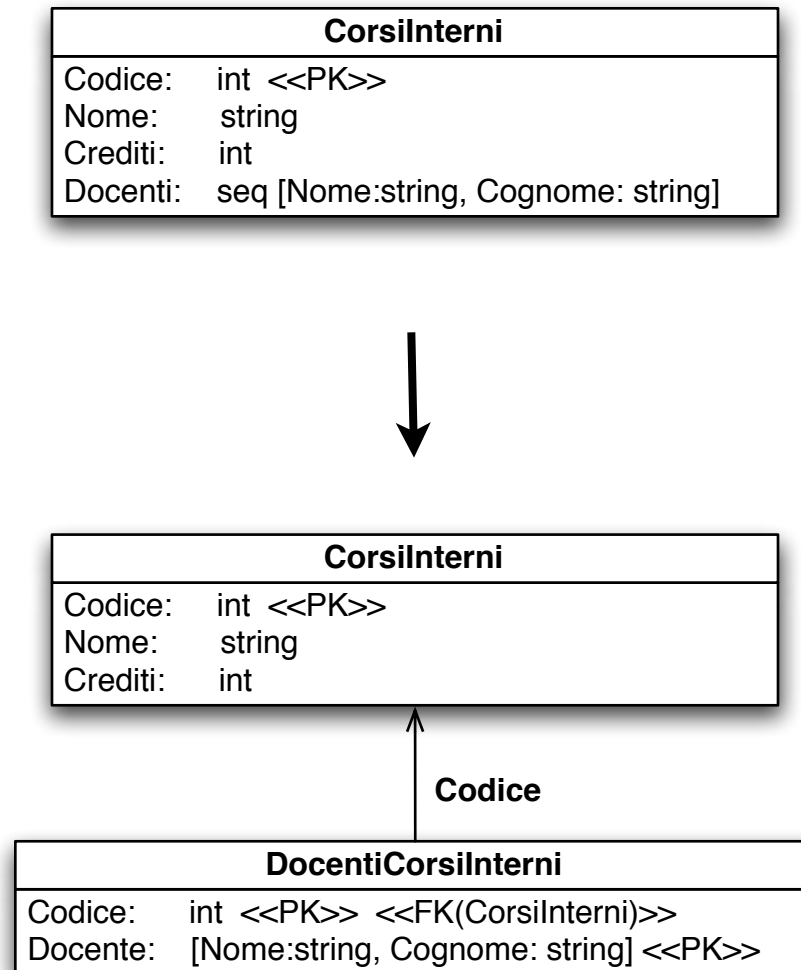
- **Partizionamento orizzontale**

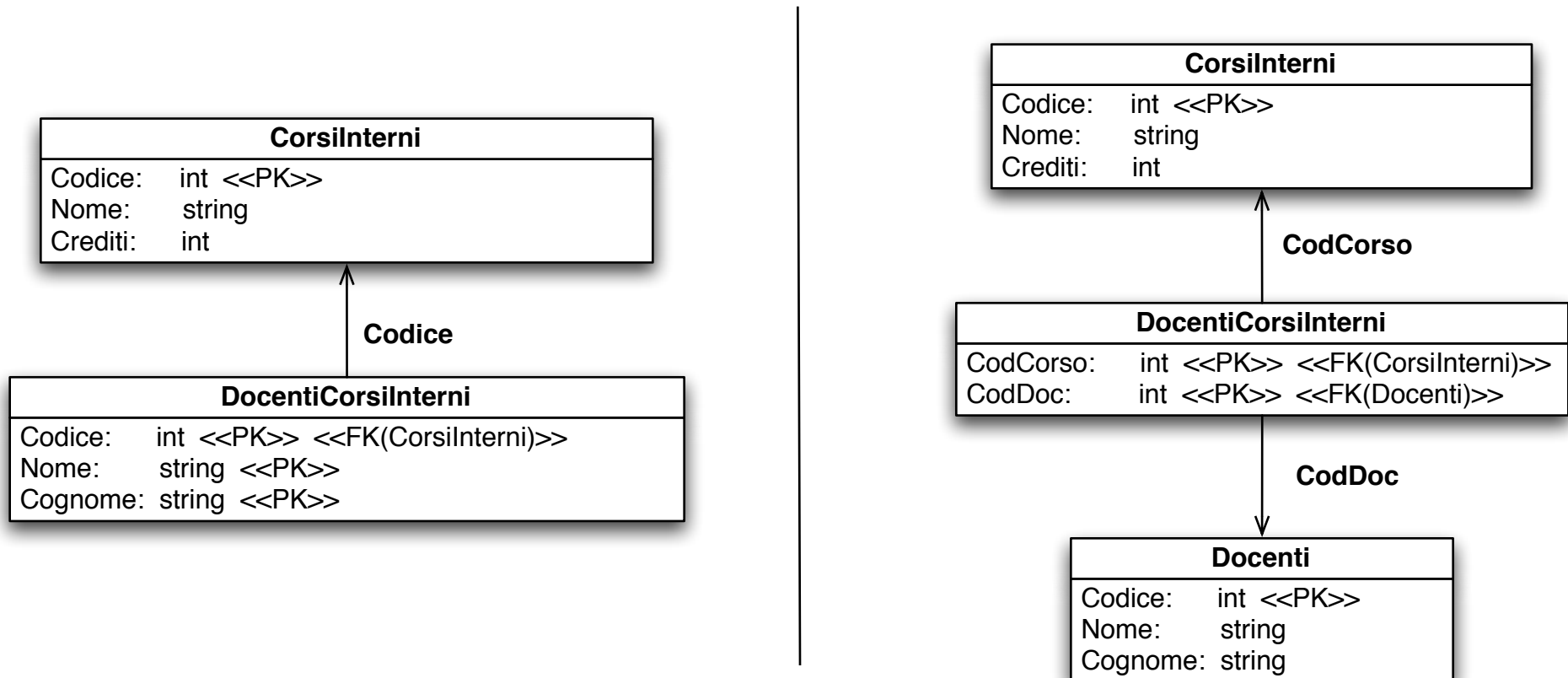
- complica la visita di tutti gli elementi della superclasse
- divide la superclasse in più relazioni: sconsigliato se vi è una associazione entrante nella superclasse
- problematico senza vincolo di disgiunzione

- **Partizionamento verticale**

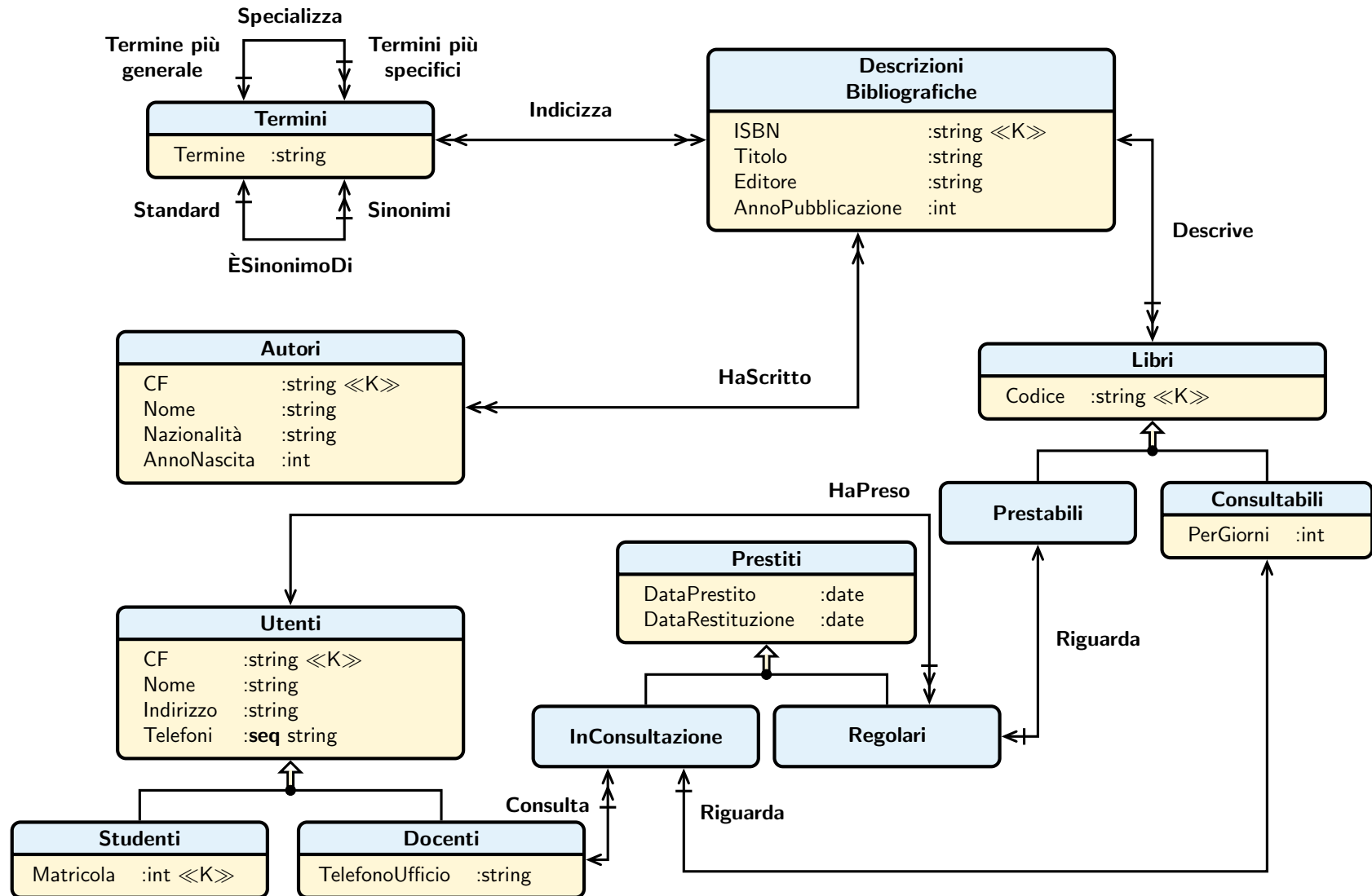
- complica il recupero di tutte le informazioni relative ad un'entità (distribuite in varie relazioni)

- Relazioni corrispondenti a **classi radice** (prive di superclasse)
 - attributo univoco, totale, costante
 - attributo artificiale (chiave sintetica)
- Relazioni che corrispondono a **sottoclassi**
 - chiave della superclasse
- Relazioni per **associazioni N:M**
 - concatenazione delle chiavi esterne

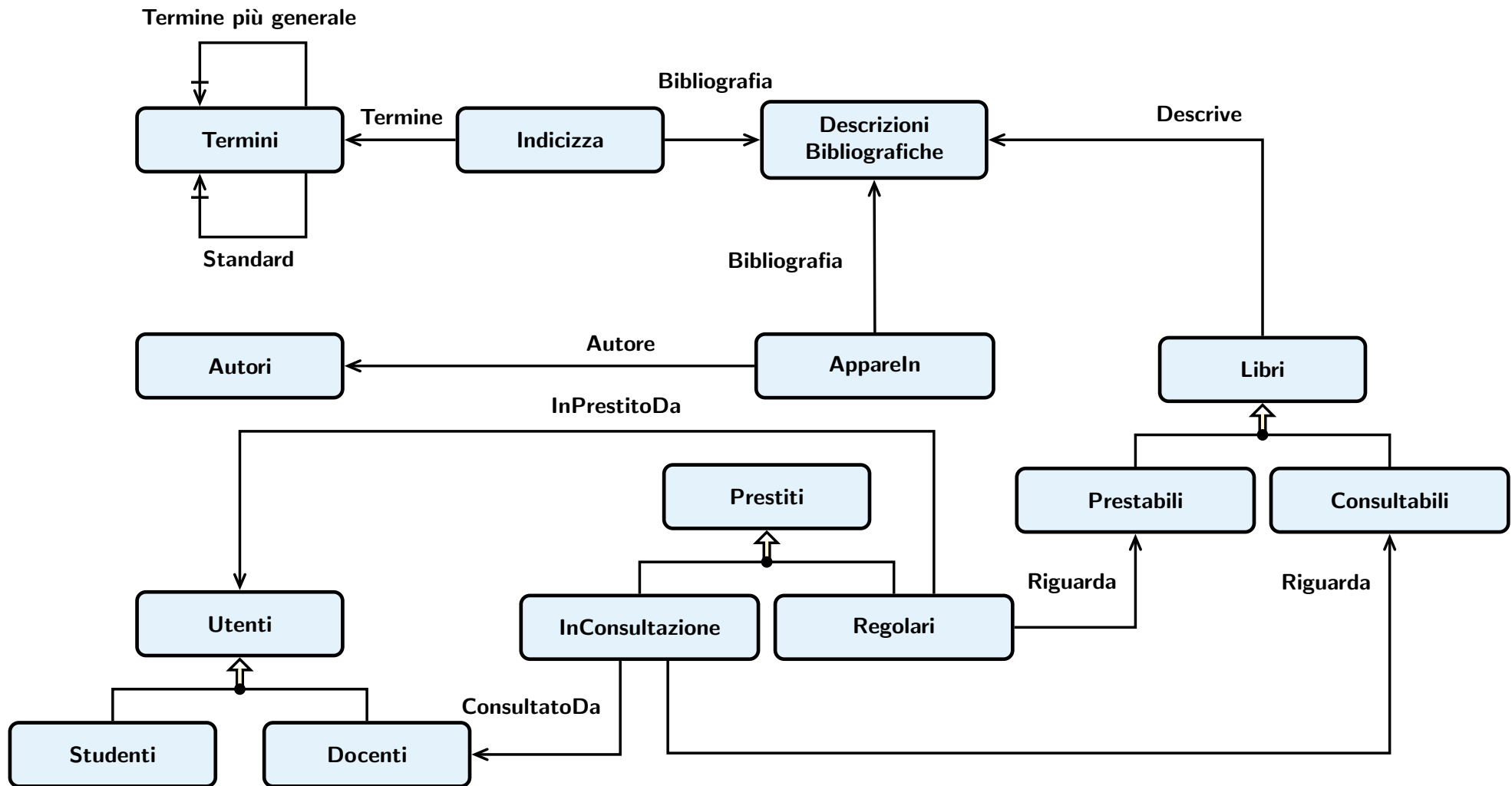


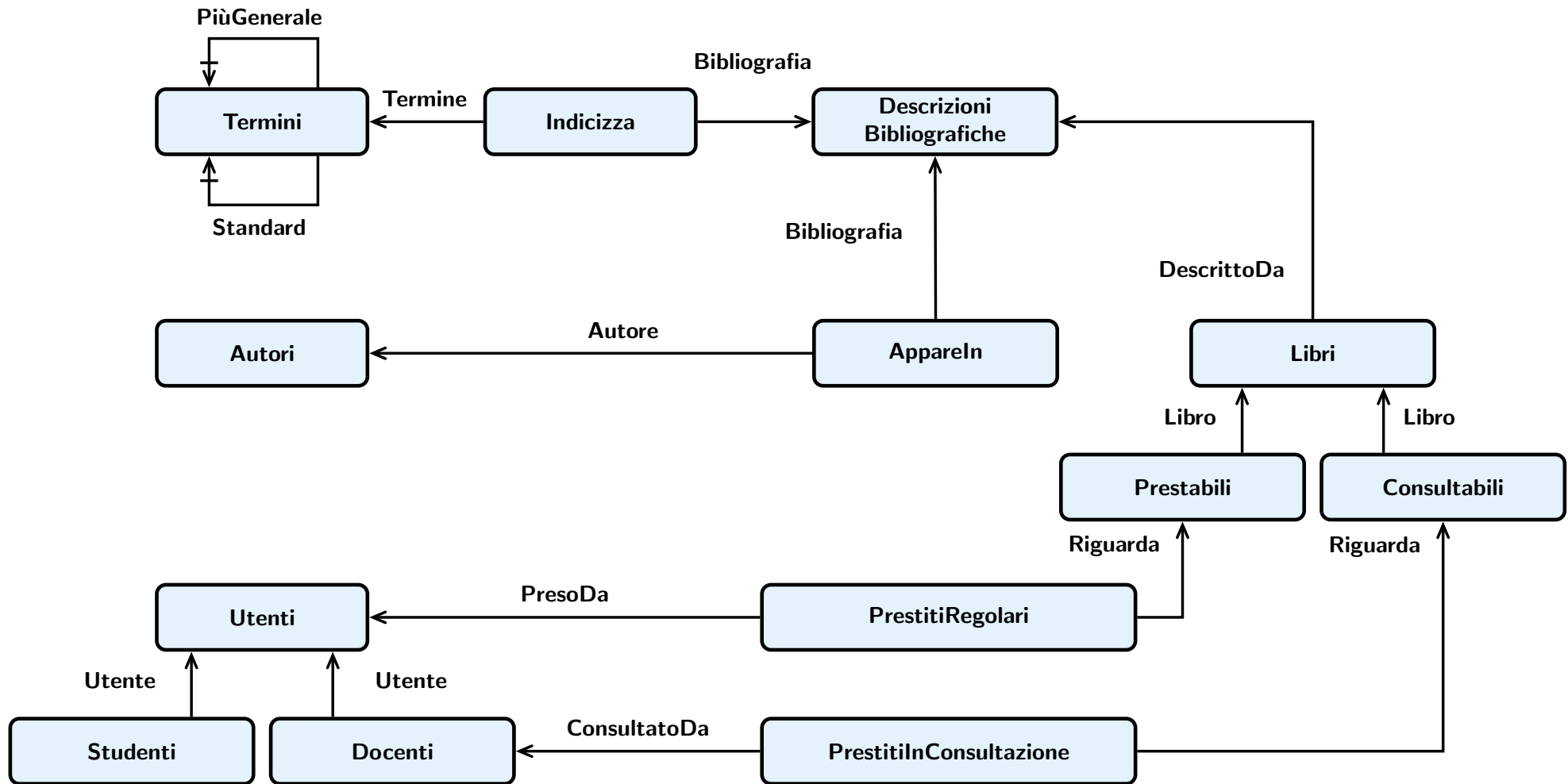


Un esempio: BD per una Biblioteca









- Termini(Termine: string, PiùGenerale*: string, Standard*: string)
 - PK(Termine)
 - PiùGenerale FK(Termini), Standard FK(Termini)
- DescrizioneBib(ISBN: string, Titolo: string, Editore: string, Anno: int)
 - PK(ISBN)
- Indicizza (Termine*: string, Bibliografia*: string)
 - PK(Termine, Bibliografia)
 - Termine FK(Termini), Bibliografia FK(DescrizioniBib)
- Autori (CF: string, Nome: string, Nazionalita: string, DataNascita: date)
 - PK(CF)
- AppareIn(Autore*: int, Bibliografia*: string)
 - PK(Autore, Bibliografia)
 - Autore FK(Autori), Bibliografia FK(DescrizioniBib)

- Libri(Codice: string, DescrittoDa*: string)
 - PK(Codice)
 - DescrittoDa FK(DescrizioniBib)
- Consultabili(Libro*: string, PerGiorni: int)
 - PK(Libro)
 - Libro FK(Libri)
- Prestabili(Libro*: string)
 - PK(Libro)
 - Libro FK(Libri)
- Utenti (CF: int, Nome: string, Indirizzo: string)
 - PK(CF)
- Telefoni(Numero: string, Utente*: int)
 - PK(Numero, Utente)
 - Utente FK(Utenti)

- Studenti (Utente*: int, Matricola: string)
 - PK(Utente)
 - Utente FK(Utenti)
 - CK(Matricola)
- Docenti (Utente*: int, TelefonoUfficio: string)
 - PK(Utente)
 - Utente FK(Utenti)
- PrestitiRegolari(DataPrestito: date, DataRestituzione: date, PresoDa*: int, Riguarda*: string)
 - PK(Riguarda)
 - PresoDa FK(Utenti), Riguarda FK(Prestabili)
- PrestitiInConsultazione(DataPrestito: date, DataRestituzione: date, ConsultatoDa*: string, Riguarda*: string)
 - PK(Riguarda)
 - ConsultatoDa FK(Docenti), Riguarda FK(Consultabili)

