

# Basi di Dati - IV

Corso di Laurea in Informatica

Anno Accademico 2013/2014

---

Alessandra Raffaetà

[raffaeta@dsi.unive.it](mailto:raffaeta@dsi.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**

Studenti	
Nome:	string
Cognome:	string>
Matricola:	string <<PK>
AnnoCorso:	int

**Candidato**

Esami	
Codice:	string <<PK>
Candidato:	string <<FK(Studenti)>>
Materia:	string
Data:	date
Voto:	int
Lode:	bool

- 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)
- **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:

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**

Codic	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 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

## ● 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

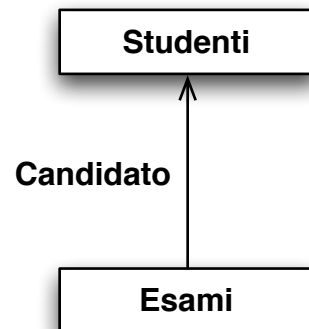
- realizzate con il meccanismo di chiave

## 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**

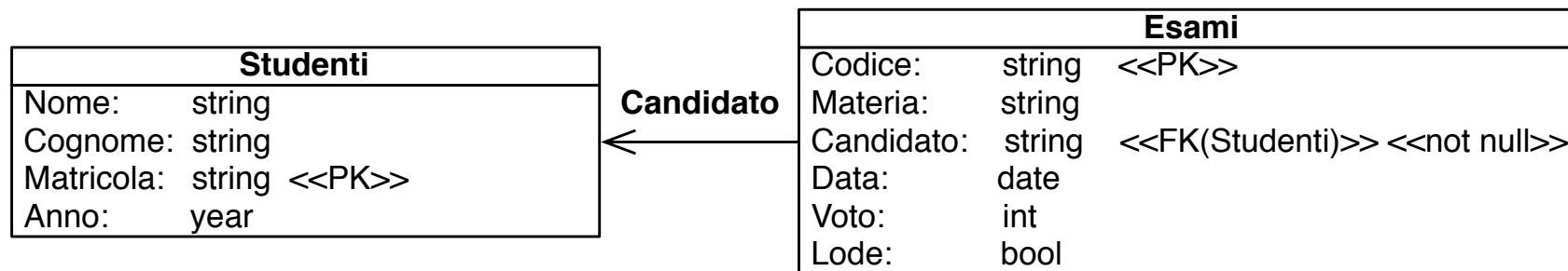
<u>Codic</u>	Materia	<u>Candidato*</u>	Data	Voto	Lode
B112	BD1	71523	08.07.06	27	N
F31	FIS	76366	08.07.07	26	N
B247	BD2	71523	28.12.06	30	S

- 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  
`Utente(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 cioè 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.

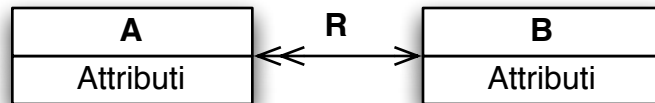




# **Dal Modello a Oggetti al Modello Relazionale**

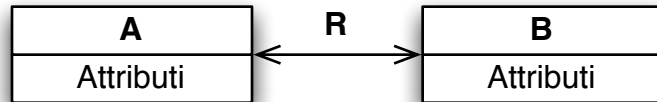
- Trasformazione per passi:
  - associazioni molti a uno (e uno a uno)
  - associazioni molti a molti
  - gerarchie di inclusione
  - identificazione chiavi primarie
  - attributi multivalore
  - attributi composti

## ● Associazioni N:1 (univoche)



- totalità di R con vincolo **not-null** sulla chiave esterna
  - totalità dell'inversa non rappresentabile
  - eventuali **attributi dell'associazione** si possono inserire in A
- 
- Prestiti <<-|----> Utenti
  - EsamiEsterni <<-|---|-> EsamiInterni (attributo: Colloquio)

- Associazioni 1:1 (univoche con inversa univoca)

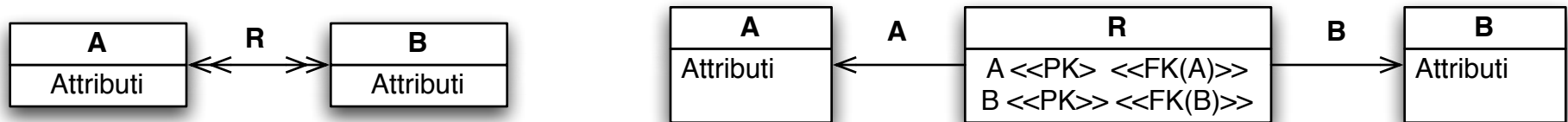


- univocità di  $R^{-1}$  (inversa di R) con vincolo di chiave (<<unique>>, <<key>>) sulla chiave esterna

- possibile solo se R è totale
- la direzione di R scelta in modo che sia totale, se possibile

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

- Associazioni N:M (multivalore con inversa multivalore)



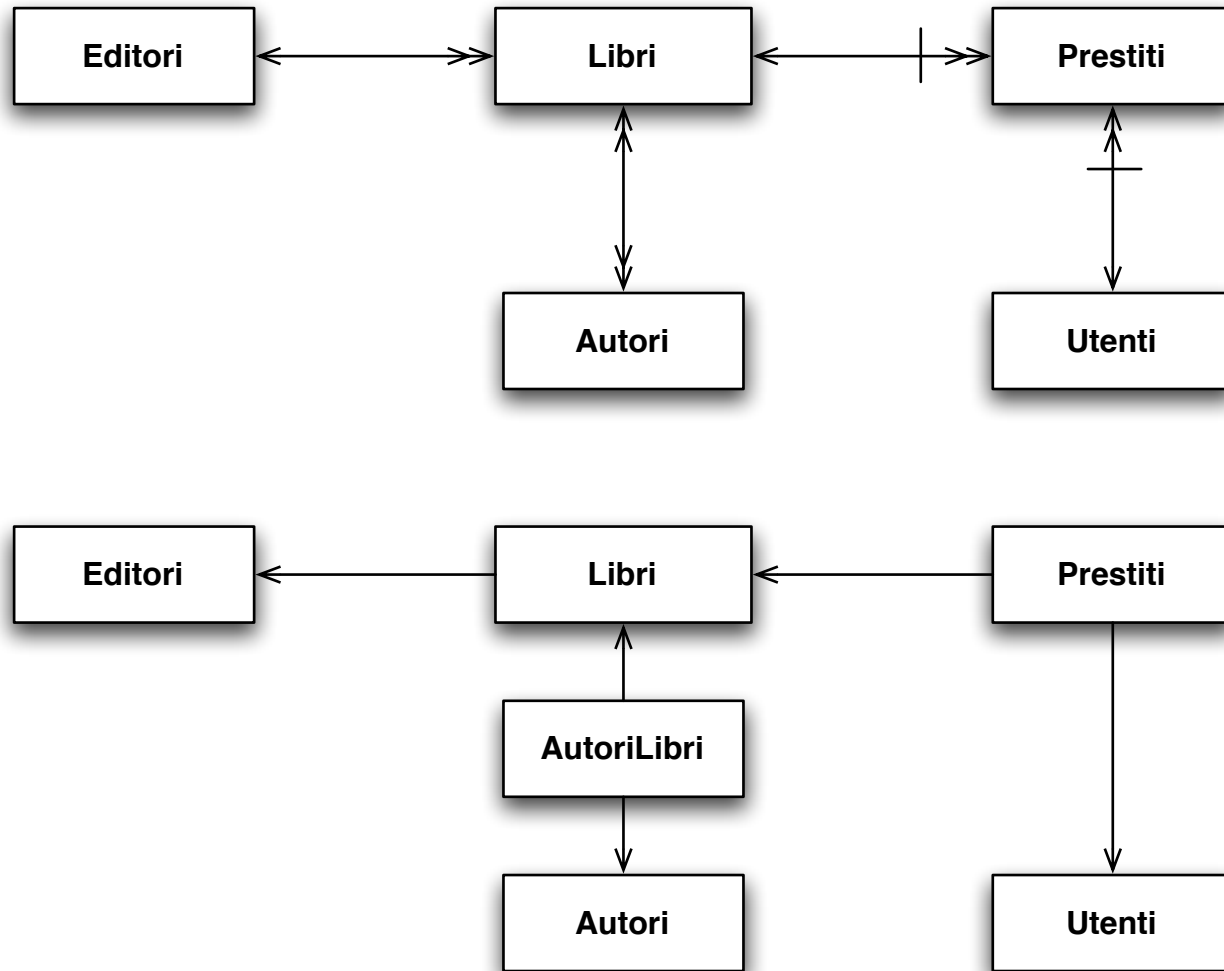
- eventuali attributi dell'associazione si inseriscono in R  
(e possono far parte della chiave primaria)

- totalità non rappresentabile

- Es.:

Relazione HaSostenuto, con attributo Data

TipoEsami <<---->> Studenti

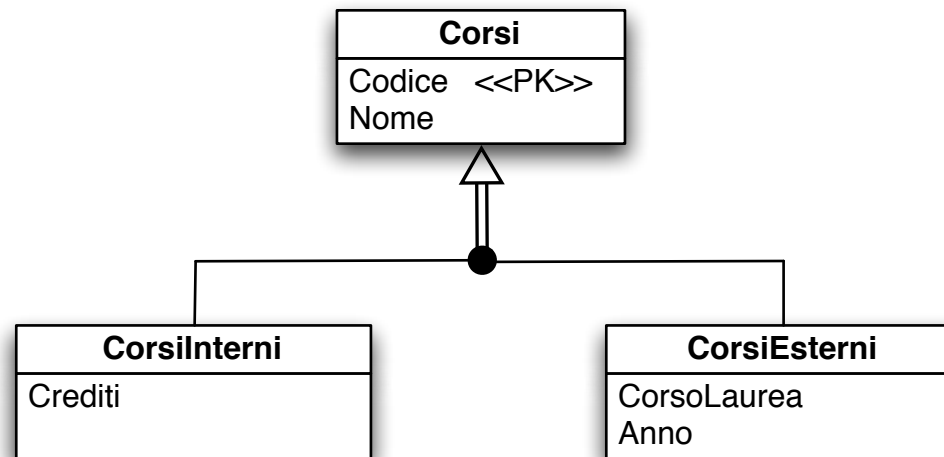


- Data la classe  $A$  (attr.  $X_A$ , chiave  $K_A$ ) con sottoclassi  $B$  (attr.  $X_B$ ) e  $C$  (attr.  $X_C$ )
- Tre possibili soluzioni
  - Relazione unica
    - $R(X_A, X_B, X_C, \text{Discr})$
    - Discr indica la classe alla quale appartiene l'elemento
    - $X_B$  e  $X_C$  possono avere 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)

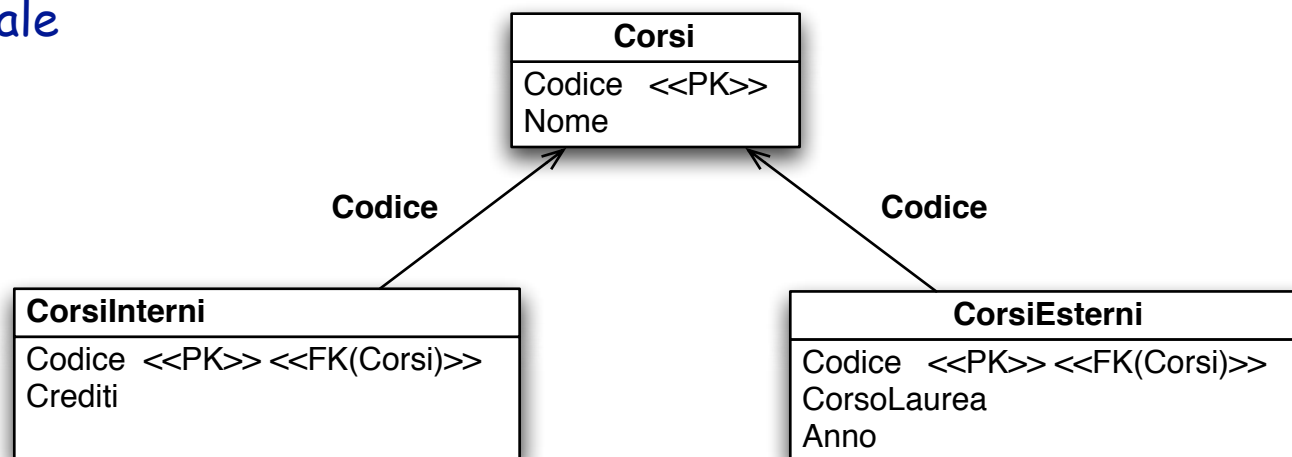
- Si consideri la gerarchia seguente:



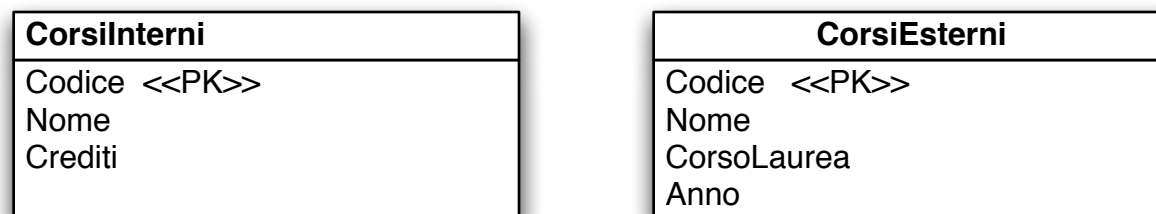
- L'attributo **InterniEsterni** svolge il ruolo di discriminatore

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

## ● Verticale



## ● Orizzontale



## ● Tabella unica

- conveniente se le sottoclassi differiscono per pochi attributi

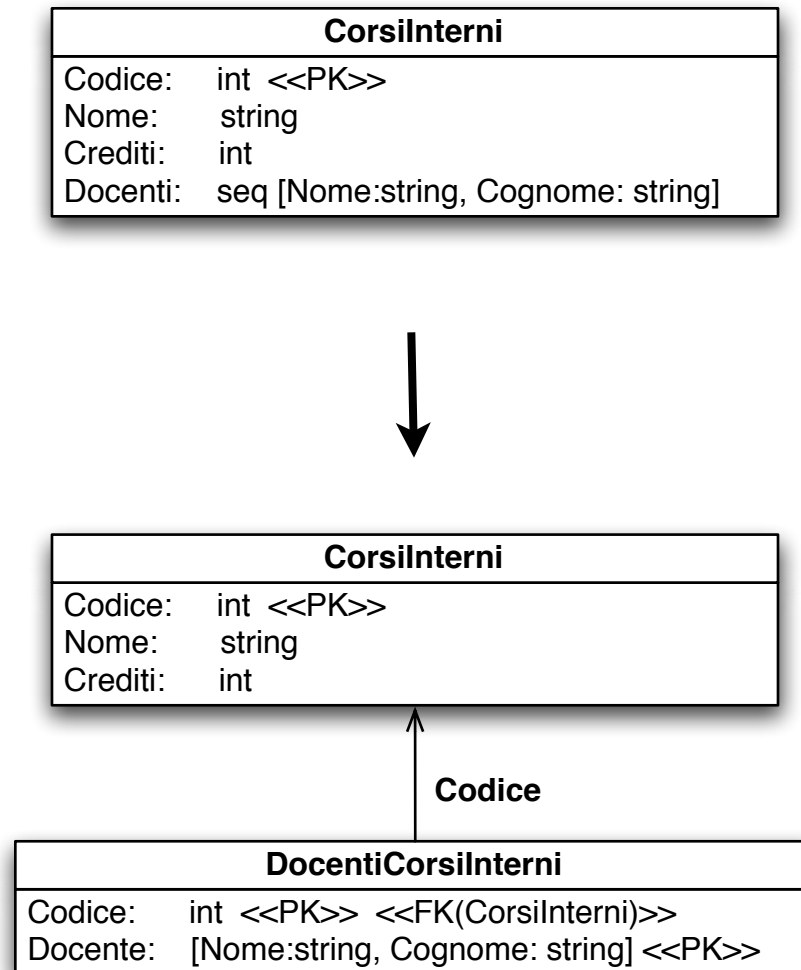
## ● Partizionamento orizzontale

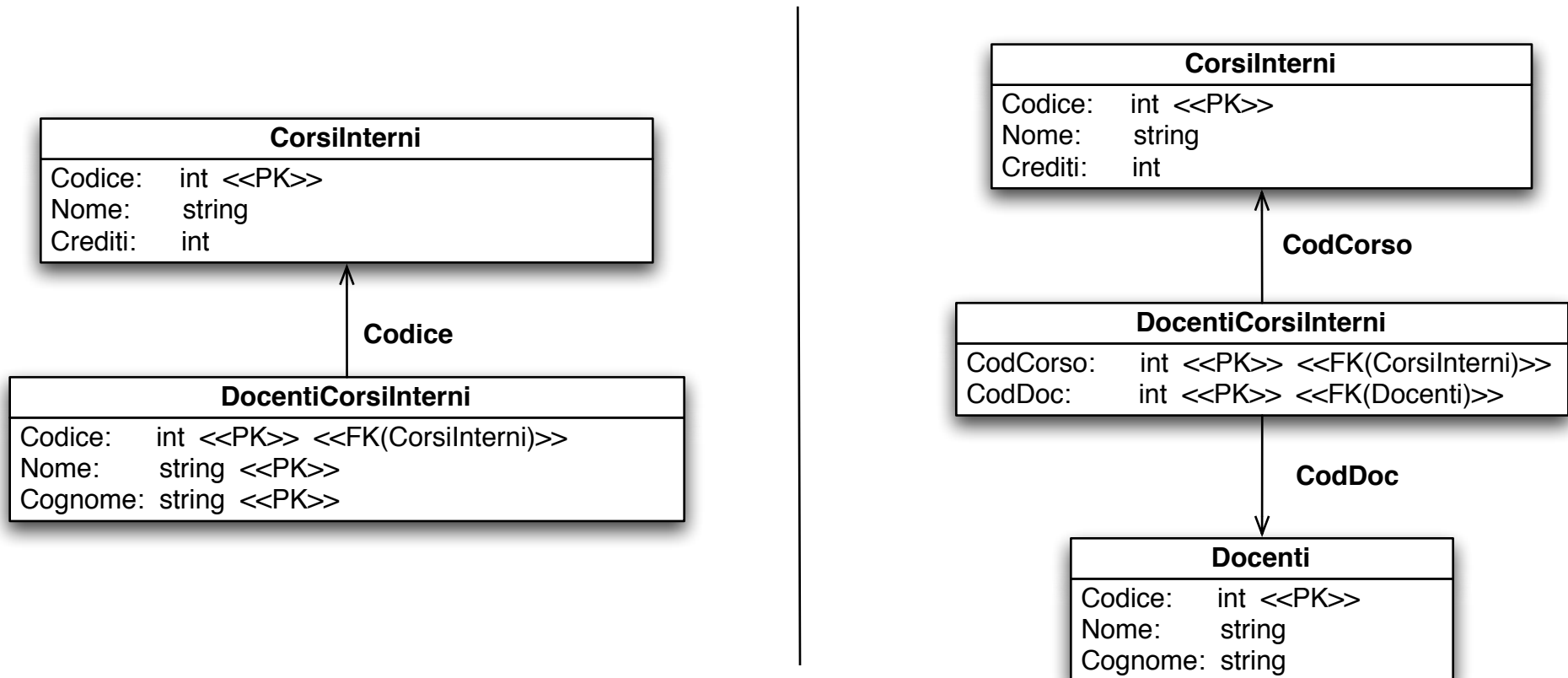
- complica la visita di tutti gli elementi della superclasse
- divide la superclasse in più relazioni: sconsigliato se vi è una associazione entrante nella superclasse
- problematico senza vincolo di disgiunzione

## ● 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

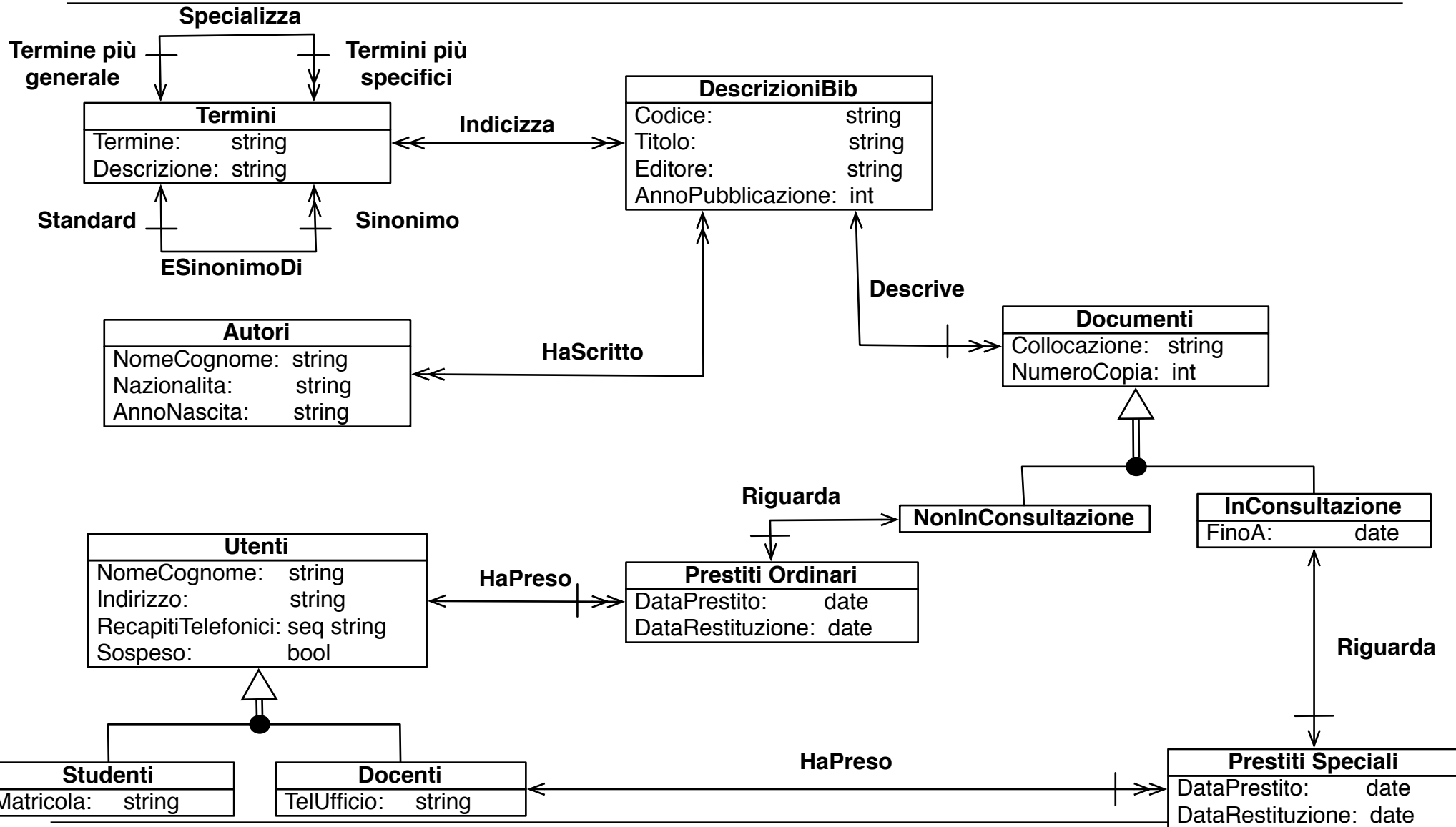


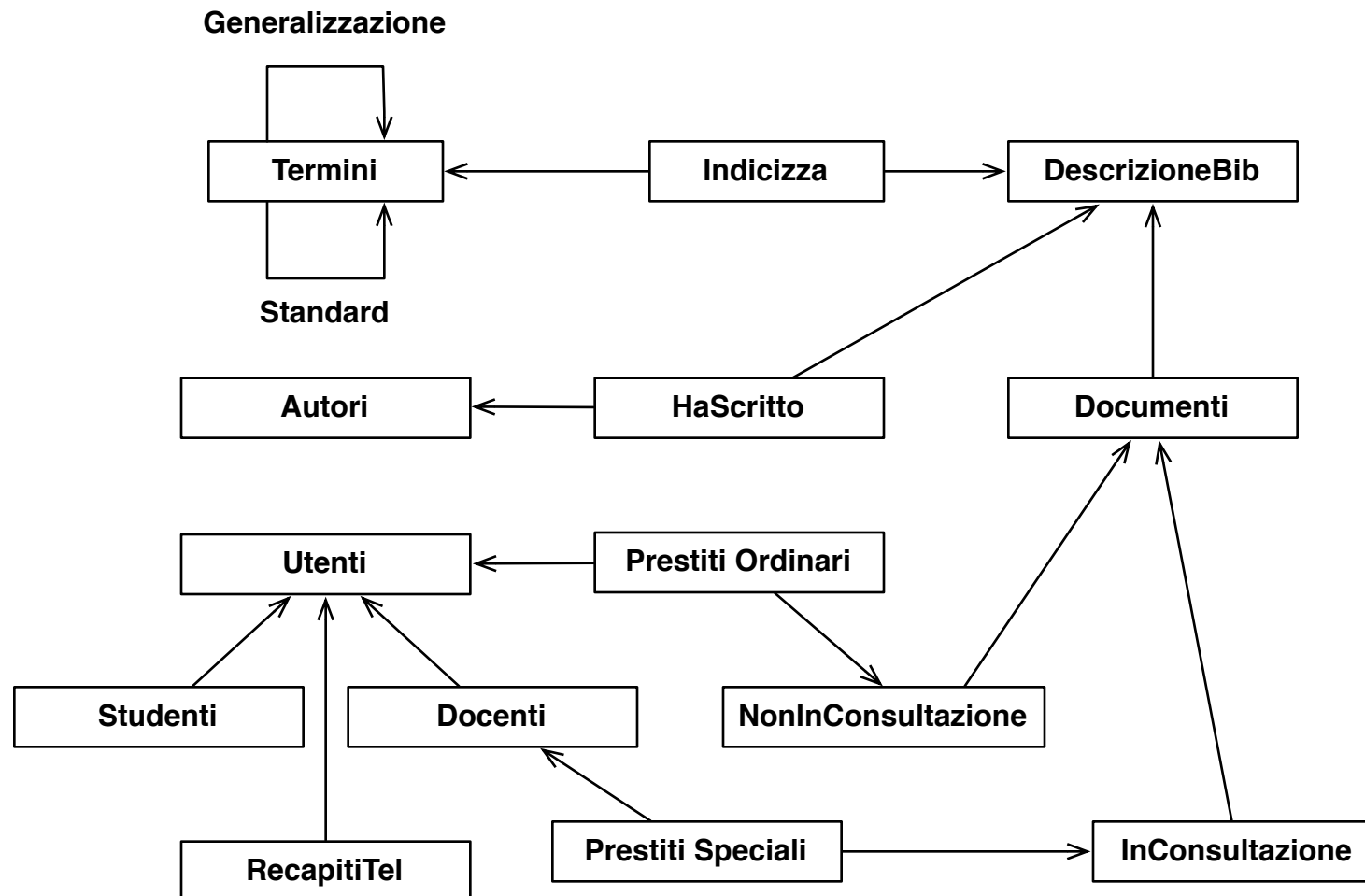




- Modello relazionale
  - relazione + { chiave primaria + chiave esterna + not null }
- Dal modello concettuale (a oggetti) al modello logico relazionale
  - associazioni  $A \leftrightarrow B$  (N:1 oppure 1:1) con chiave esterna in A
  - associazioni  $A \leftrightarrow B$  (M:N) con una nuova relazione R che riferisce con chiave esterna sia A che B
  - sottoclassi: relazione unica, partizionamento (verticale, orizzontale)
  - attributi multivalore e strutturati

**Un esempio:  
BD per una Biblioteca**

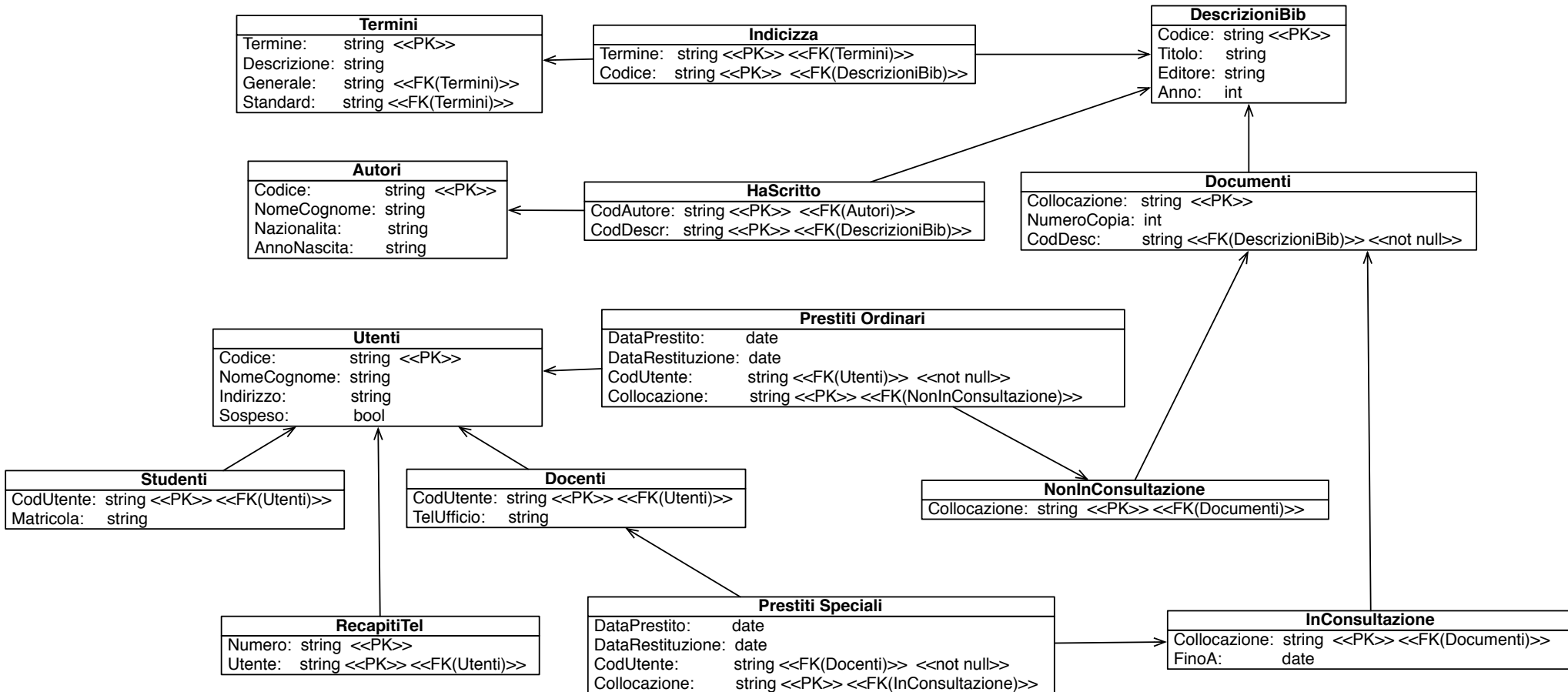




- Termini(Termine: string, Descrizione: string, Generalizzazione\*: string, Standard\*: string)
  - PK(Termine)
  - Generalizzazione FK(Termini), Standard FK(Termini)
- DescrizioneBib(Codice: string, Titolo: string, Editore: string, Anno: int)
  - PK(Codice)
- Indicizza (Termine\*: string, Codice\*: string)
  - PK(Termine, Codice)
  - Termine FK(Termini), Codice FK(DescrizioniBib)
- Autori (Codice: string, NomeCognome: string, Nazionalita: string, AnnoNascita: string)
  - PK(Codice)
- HaScritto(CodAutore\*: string, CodDescr\*: string)
  - PK(CodAutore, CodDescr)
  - CodAutore FK(Autori), CodDescr FK(DescrizioniBib)

- Documenti(Collocazione: string, NumeroCopia: int, CodDesc\*: string)
  - PK(Collocazione)
  - CodDesc FK(DescrizioniBib) NOT NULL
- InConsultazione(Collocazione\*: string, FinoA: string)
  - PK(Collocazione)
  - Collocazione FK(Documenti)
- NonInConsultazione(Collocazione\*: string)
  - PK(Collocazione)
  - Collocazione FK(Documenti)
- Utenti (Codice: string, NomeCognome: string, Indirizzo: string, Sospeso: bool)
  - PK(Codice)
- RecapitiTel(Numero: string, Utente\*: string)
  - PK(Numero, Utente)
  - Utente FK(Utenti)

- Studenti (CodUtente\*: string, Matricola: string)
  - PK(CodUtente)
  - CodUtente FK(Utenti)
- Docenti (CodUtente\*: string, TelUfficio: string)
  - PK(CodUtente)
  - CodUtente FK(Utenti)
- PrestitiOrdinari(DataPrestito: data, DataRestituzione: data, CodUtente\*: string, Collocazione\*: string)
  - PK(Collocazione)
  - CodUtente FK(Utenti) NOT NULL, Collocazione FK(NonInConsultazione)
- PrestitiSpeciali(DataPrestito: data, DataRestituzione: data, CodUtente\*: string, Collocazione\*: string)
  - PK(Collocazione)
  - CodUtente FK(Docenti) NOT NULL, Collocazione FK(InConsultazione)





- Si vogliono rappresentare informazioni relative alle visite di uno studio dentistico. Dottori, infermieri e pazienti hanno cognome e nome, codice fiscale, indirizzo, telefono. Un paziente può prenotare una visita in una certa data e ora con un certo dottore. Per ogni visita effettuata c'è un costo, e il tipo di visita (una stringa). Inoltre, se è la prima visita, ci sarà una diagnosi (una stringa) e un preventivo. Ogni paziente ha un totale da saldare.
- Si dia uno schema grafico a oggetti (secondo la notazione del libro di testo) della base di dati e si trasformi nello schema relazionale mostrandone la rappresentazione grafica (anche questa secondo la notazione del libro di testo, indicando la chiave primaria e le chiavi esterne).

- Si vogliono memorizzare informazioni per un sito che vende brani musicali da scaricare. Gli utenti hanno nome, cognome, login (unica), password, e possono acquistare sia brani singoli che album, composti da più brani. Di un album interessa il nome, l'autore, l'interprete, il genere, la casa discografica, la data e il prezzo. Di un brano interessa il titolo, la durata, il prezzo. Un acquisto può riguardare un numero qualunque di prodotti, e ha una data e un numero di carta di credito usato per il pagamento. I clienti possono giudicare sia brani che album, dando un numero di stelle (da 1 a 5) e un commento.