

Unità **C2**



ESERCIZI COMMENTATI
Segui la risoluzione passo passo

La progettazione concettuale: il modello ER

In questa unità imparerai...

- A progettare una base di dati
- A riconoscere le informazioni significative per realizzare una base di dati
- A comprendere le relazioni che esistono tra i dati
- A inquadrare i vari tipi di associazioni che legano i dati

Rifletti

Il termine **concettuale** si riferisce al fatto che si evita il più possibile di descrivere dettagli realizzativi. In particolare, si astrae dal tipo di sistema e dal linguaggio che verranno utilizzati nella realizzazione e si cerca, invece, di attribuire una certa struttura ai dati.

1 La progettazione concettuale

La **progettazione concettuale** consiste nel riorganizzare tutti gli elementi presenti nella documentazione sulle specifiche per rappresentare la realtà di interesse in termini di una descrizione formale e completa, indipendentemente dai criteri di rappresentazione usati nei DBMS. Il prodotto di tale attività è lo **schema concettuale**, che rappresenta il documento ufficiale di riferimento e di comunicazione per i progettisti della successiva fase di progettazione logica.

2 Il modello concettuale ER

Tra i possibili modelli concettuali è molto diffuso il **modello Entità/Associazioni** (in inglese **ER**, da *Entity/Relationship*), introdotto nel 1976 da Peter P. Chen. Si tratta di un modello grafico per la descrizione dei dati e delle loro relazioni all'interno di una realtà di interesse. È uno strumento per svolgere l'analisi delle caratteristiche di una realtà indipendentemente dagli eventi che in essa accadono. Il risultato di questo lavoro è rappresentato dallo **schema ER** o **diagramma ER**.

Per la costruzione di uno schema ER, si parte dal semplice concetto che la realtà da rappresentare è composta da **entità**, ognuna delle quali è caratterizzata da specifiche **proprietà** (o **attributi**). Le varie entità che caratterizzano la realtà oggetto di studio sono connesse tra loro attraverso specifiche associazioni che evolvono nel tempo.

3 Le entità

L'**entità** permette di rappresentare una classe di oggetti (cioè fatti, persone, cose e così via) che appartengono a una realtà e che:

- hanno proprietà comuni;
 - hanno esistenza autonoma;
 - sono di interesse per l'applicazione.
-

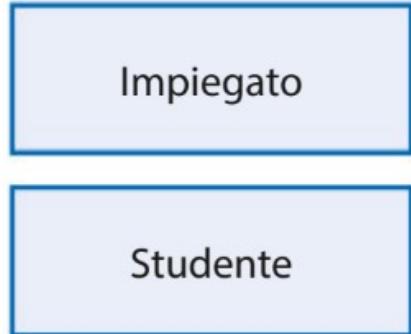
Ad esempio, si può scegliere di descrivere:

- un'**università** attraverso le entità *Studente*, *Corso*, *Docente*, *Dipartimento*, ecc.;
- un'**azienda** attraverso le entità *Impiegato*, *Progetto*, *Dipartimento*, *Città*, ecc.

Ogni singolo oggetto che appartiene a un'entità è detto **istanza di entità** (o anche **occorrenza di entità**).

Un'istanza dell'entità *Impiegato* corrisponde a un singolo impiegato (ad esempio, l'impiegato Gianfranco Bianchi). Un'istanza dell'entità *Studente* corrisponde a un singolo studente (ad esempio, lo studente Marco Rossi), mentre un'istanza dell'entità *Dipartimento* corrisponde a uno dei dipartimenti di un'università (ad esempio, Informatica).

La **rappresentazione grafica** di un'entità, in uno schema concettuale, è un rettangolo con al centro il nome che identifica univocamente l'entità in questione. Seguiremo la convenzione di assegnare all'entità un nome significativo al singolare, con l'iniziale maiuscola.



Impiegato

Studente

4 Gli attributi

Gli **attributi** sono le proprietà ossia i fatti che si intendono rappresentare e che interessano poiché descrivono caratteristiche di un'entità.

Ad esempio, per l'entità *Impiegato* sono utili attributi: *Cognome*, *Nome*, *DataNascita*, *Stipendio*, *Età*, *Sesso*. Per l'entità *Studente* si possono considerare gli stessi attributi dell'entità *Impiegato*, escludendo *Stipendio* e aggiungendo *Matricola*.

Ogni attributo è specificato da:

- un **nome**, come abbiamo visto precedentemente;
- un **formato**, che indica il tipo di valori che può assumere (ad esempio Stringa, Reale, Intero, Booleano, Data);
- una **dimensione** (non vale per i tipi Data o Booleano), che indica la quantità massima di caratteri o cifre inseribili;
- un **valore**; i diversi valori assunti dagli attributi determinano le diverse istanze dell'entità. L'insieme dei possibili valori assunti da un attributo si chiama **dominio** dell'attributo;
- Considerando l'entità **Impiegato** possiamo dire che:
 - il dominio dell'attributo *Età* è rappresentato dall'insieme di valori interi {1..120};
 - il dominio dell'attributo *Cognome* è rappresentato da {tutte le possibili stringhe di caratteri} o, più semplicemente, da stringhe di caratteri;
 - il dominio dell'attributo *Sesso* è {uomo, donna};

Classificazione e rappresentazione degli attributi

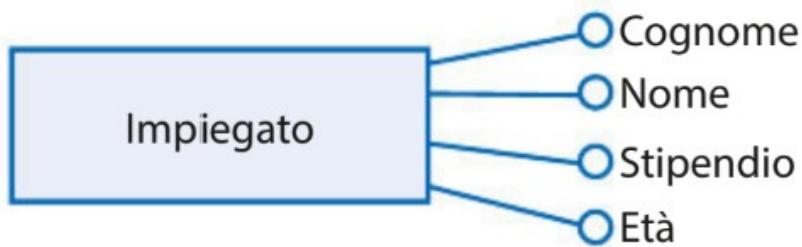
Gli attributi di un'entità possono essere classificati come:

- attributi semplici;
- attributi composti;
- attributi multipli;
- attributi opzionali;
- attributi derivati.

Un attributo si definisce **semplice** (o **elementare**) quando non è ulteriormente scomponibile, cioè rappresenta un'unità informativa di base che caratterizza un'entità e alla quale è associato un singolo valore.

Ad esempio, nell'entità *Impiegato* gli attributi *Cognome*, *Nome*, *Stipendio*, *Età* sono tutti semplici, in quanto non ulteriormente scomponibili. In altri termini, ogni impiegato ha un solo cognome, un solo nome, un solo stipendio e una sola età.

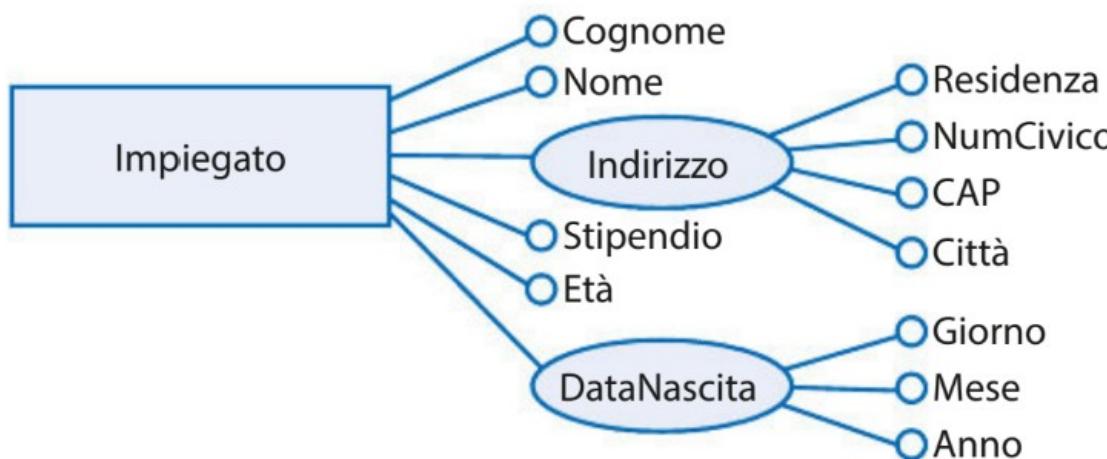
Nello schema ER, l'attributo semplice si rappresenta graficamente con una linea che origina dall'entità e termina con un pallino vuoto cui segue il nome dell'attributo.



Un **attributo** si definisce **composto** o **aggregato** quando è costituito da un'aggregazione di altri attributi, che a loro volta possono essere semplici o frutto di altre aggregazioni di attributi. L'attributo composto è, pertanto, scomponibile.

Nell'entità *Impiegato*, l'attributo *DataNascita* è composto, infatti contiene tre attributi semplici (*Giorno*, *Mese*, *Anno*). Anche l'attributo *Indirizzo* è di tipo composto, in quanto è formato dai seguenti attributi semplici: *Residenza*, *NumCivico*, *CAP*, *Città*.

Nello schema ER, l'attributo composto si rappresenta graficamente con una linea che origina dall'entità e termina con un'ellisse (che riporta il nome dell'attributo) alla quale vengono agganciati gli attributi semplici di cui l'attributo è composto.



Un **attributo** si definisce **multiplo** quando possono essergli associati contemporaneamente anche più valori dello stesso tipo.

Rifletti

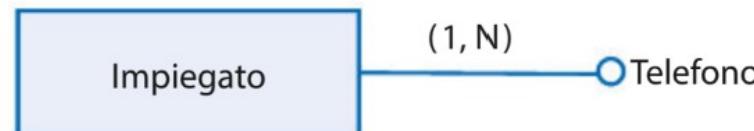
Quando la cardinalità massima di un attributo non è 1, l'**attributo non rappresenta più una funzione ma una relazione**.

Nello schema ER, per rappresentare l'attributo multiplo si ricorre al concetto di **cardinalità**.

La **cardinalità di attributo** è descritta da una coppia di valori (x, y) che specificano, rispettivamente, il numero minimo e massimo di valori dell'attributo associati a ogni istanza dell'entità.

I parametri x (cardinalità minima) e y (cardinalità massima) possono assumere i valori 0, 1 e N . Quando la cardinalità massima è pari a N siamo di fronte al caso di attributi multipli.

Ad esempio, nell'entità *Impiegato*, l'attributo *Telefono* è di tipo multiplo in quanto a esso si può associare l'elenco dei numeri di telefono dell'impiegato. La sua rappresentazione nello schema ER è la seguente:

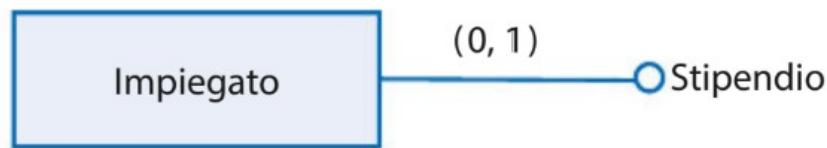


Il valore nullo, in inglese **NULL** (da non confondere con la stringa di caratteri vuota o con un numero di valore zero), rappresenta un'informazione mancante, inapplicabile o sconosciuta. Chiariamo meglio con un esempio.

L'attributo *NumerоАppartamento* di un indirizzo ha senso solo per gli indirizzi relativi a edifici con appartamenti e non per quelli di altri tipi di abitazioni come, ad esempio, le case monofamiliari. Il significato di NULL, in questo caso, è **inapplicabile**.

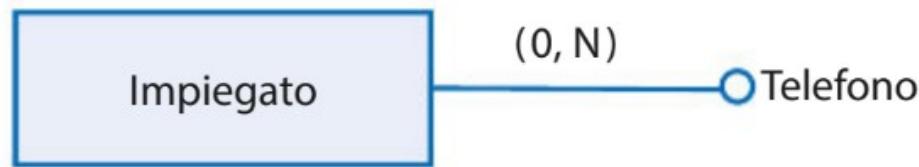
L'attributo *NumerоТелефоноСа* di uno studente potrebbe contenere un valore NULL se non si conosce questo dato. Il significato di NULL, in questo caso, è **sconosciuto**.

L'opzionalità viene rappresentata sempre attraverso la cardinalità, operando sulla cardinalità minima (x). Ad esempio, nell'entità *Impiegato*, l'attributo *Stipendio* potrebbe essere opzionale. La sua rappresentazione nello schema ER sarà, pertanto, la seguente:



Ciò significa che per ogni istanza di *Impiegato*, l'attributo *Stipendio* potrà non assumere un valore, o accettarne al più uno solo.

Anche un attributo multiplo può essere opzionale. Riprendiamo l'attributo *Telefono*; se la sua rappresentazione è la seguente:



significa che per ogni istanza di *Impiegato* potrà essere associata una lista di N numeri telefonici oppure una lista vuota.

In sintesi:

Cardinalità	Valore	Significato
Minima	0	L'attributo è opzionale (l'informazione potrebbe essere non disponibile)
Minima	1	L'attributo è obbligatorio
Massima	N	L'attributo è multivaleore

Rifletti

Non è utile creare campi che assumono un unico valore. Ad esempio, se inseriamo l'attributo *Nazione* nell'entità *Impiegato* e sappiamo che gli impiegati sono tutti residenti nella provincia o regione dove si tiene il corso, tale campo assumerà sempre il valore *Italia*. Così il campo non è utile all'economia del database, occupa spazio senza offrire un'informazione utile all'utilizzatore.

Consideriamo l'attributo *Età* dell'entità *Impiegato*.



L'età può essere calcolata partendo dal valore dell'attributo *DataNascita*. Pertanto, memorizzare questo dato significa incorrere in un serio problema di ridondanza, in quanto questo attributo necessiterebbe del suo aggiornamento ogni anno, pena la non corrispondenza tra l'età e la data di nascita delle varie istanze.

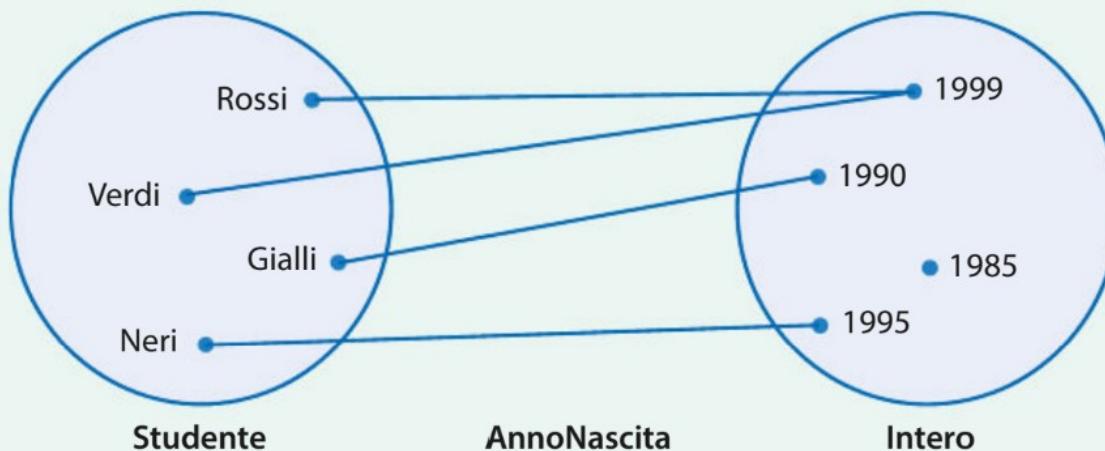
È importante, quindi, *non prevedere la presenza di attributi derivati* durante la fase di formalizzazione dello schema ER, a meno che il calcolo del valore corrispondente non sia eccessivamente complesso e richieda molto tempo, nel qual caso si opta per la presenza di questo attributo, facendo però attenzione alla sua gestione al fine di evitare incongruenze e inconsistenze della base di dati.

ORA TOCCA A TE

- Osserva il seguente schema (livello intensionale):



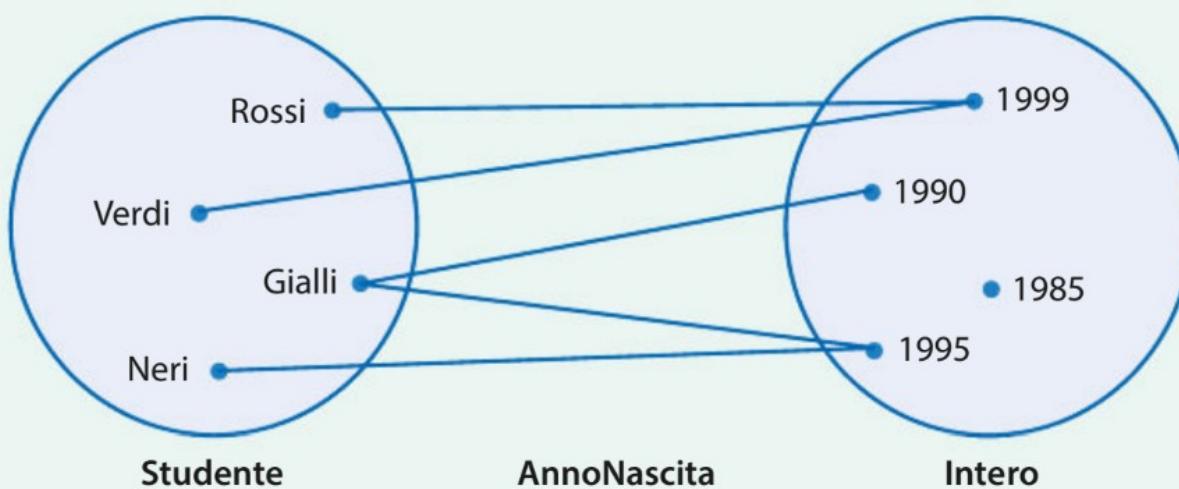
dove il dominio dell'attributo *AnnoNascita* è rappresentato dall'insieme dei numeri interi compresi nell'intervallo [1960,2015]. Nel seguente livello estensionale è rappresentata un'istanza corretta?



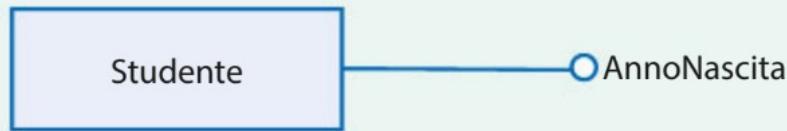
2. Osserva il seguente schema (livello intensionale):



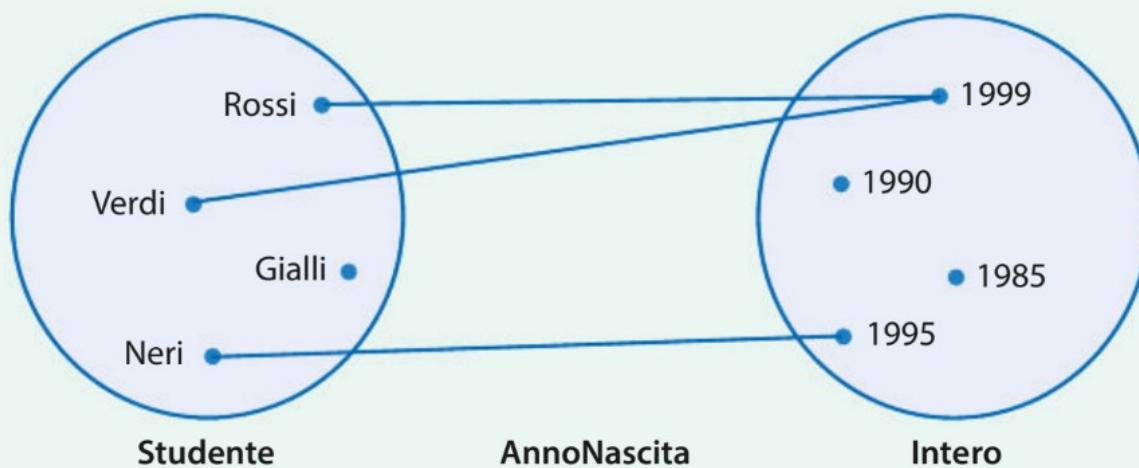
dove il dominio dell'attributo *AnnoNascita* è rappresentato dall'insieme dei numeri interi compresi nell'intervallo [1960,2015]. Nel seguente livello estensionale è rappresentata un'istanza corretta?



3. Osserva il seguente schema (livello intensionale):



dove il dominio dell'attributo *AnnoNascita* è rappresentato dall'insieme dei numeri interi compresi nell'intervallo [1960,2015]. Nel seguente livello estensionale è rappresentata un'istanza corretta?



5 Gli attributi chiave

Il concetto di **chiave** è particolarmente importante nell'ambito dei database, poiché le entità devono essere costruite in modo tale che ogni istanza possa essere distinta per mezzo di uno o più attributi. Scopo della chiave è, quindi, quello di identificare univocamente le istanze, ma per ottenere ciò si vuole usare “la minima informazione possibile”; si vuole, cioè, che la chiave sia la più piccola possibile.

Si definisce **chiave candidata** o **superchiave** un attributo o un insieme di attributi in grado di distinguere un'istanza di entità dall'altra in modo **univoco**.

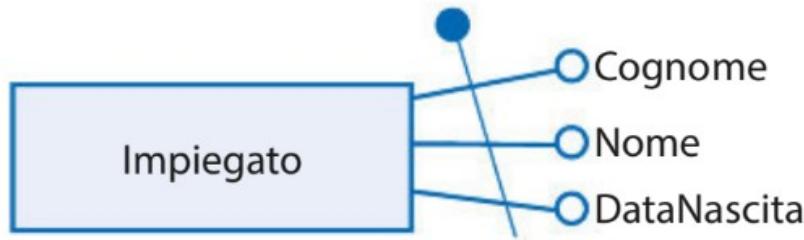
Ad esempio, se consideriamo l'entità *Impiegato*, per distinguere in modo univoco due istanze, possiamo utilizzare:

- un attributo *CodiceFiscale*, poiché sappiamo che i valori di tale attributo sono diversi per ogni persona;
- un insieme di attributi (ad esempio *Cognome*, *Nome* e *DataNascita*), facendo attenzione a scegliere attributi che, presi nel loro insieme, riescano a identificare univocamente un'istanza.

Una chiave composta da un solo attributo si rappresenta come un attributo semplice, con la sola variante che il pallino è pieno.

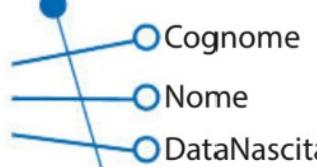


Nel caso di chiave composta da più attributi (ad esempio *Cognome*, *Nome* e *DataNascita*), la rappresentazione avviene utilizzando una linea terminante con un pallino pieno che unisce gli attributi che la compongono.



Tra le chiavi candidate, quella con il minor numero di attributi prende il nome di **chiave**, o meglio **chiave primaria** (*primary key*).

Ricapitolando:

Tipo di attributo	Rappresentazione
Semplice	
Composto	
Multiplo (obbligatorio)	
Opzionale	
Multiplo opzionale	
Chiave (composta da un solo attributo)	
Chiave (composta da più attributi)	

6 Le associazioni

Finora abbiamo mappato sullo schema concettuale:

- gli oggetti della nostra realtà di interesse (*entità*);
- i dati (le *proprietà*) che li caratterizzano.

Dobbiamo ora esprimere il modo in cui questi oggetti sono in relazione tra loro.

L'**associazione** (in inglese *relationship*) è un legame logico tra due o più entità rilevanti nella realtà che si sta considerando.

Così come le entità sono classi di oggetti del mondo reale, le associazioni sono **classi di fatti**.

Per rappresentare graficamente in un diagramma ER un'associazione tra due tipi di entità, si utilizza la seguente notazione:



dove il rombo posto a metà della linea che unisce le due entità rappresenta la loro associazione. Per convenzione utilizzeremo iniziali minuscole per la descrizione dell'associazione.

Ad esempio, tra l'entità *Persona* e l'entità *Automobile* esiste un'associazione che descrive, nel linguaggio naturale, il fatto per cui “una persona possiede una o più automobili” e viceversa “un'automobile è posseduta da una persona”.

Rifletti

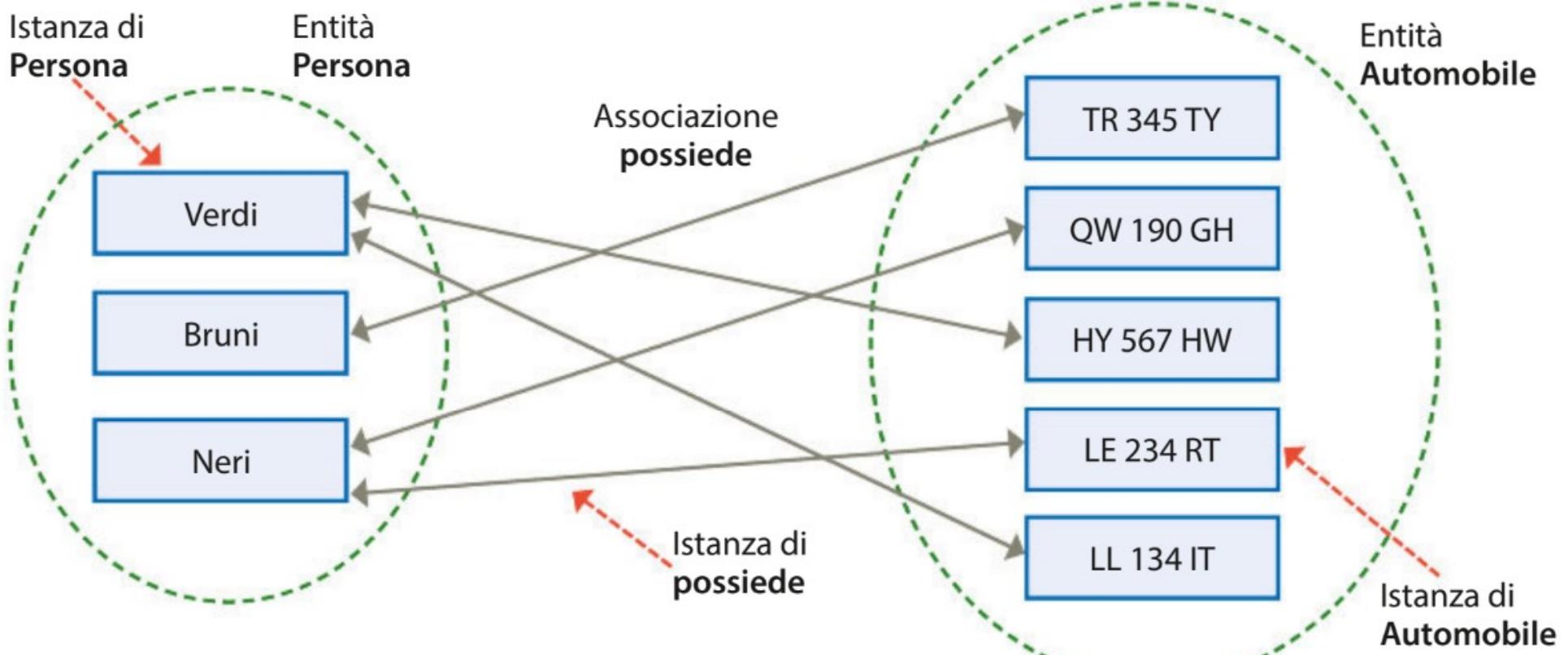
Soltamente, i sostantivi del linguaggio naturale corrispondono alle entità, mentre i verbi corrispondono alle associazioni.

L'associazione *possiede* tra le due entità *Persona* e *Automobile* può essere rappresentata nel seguente modo:



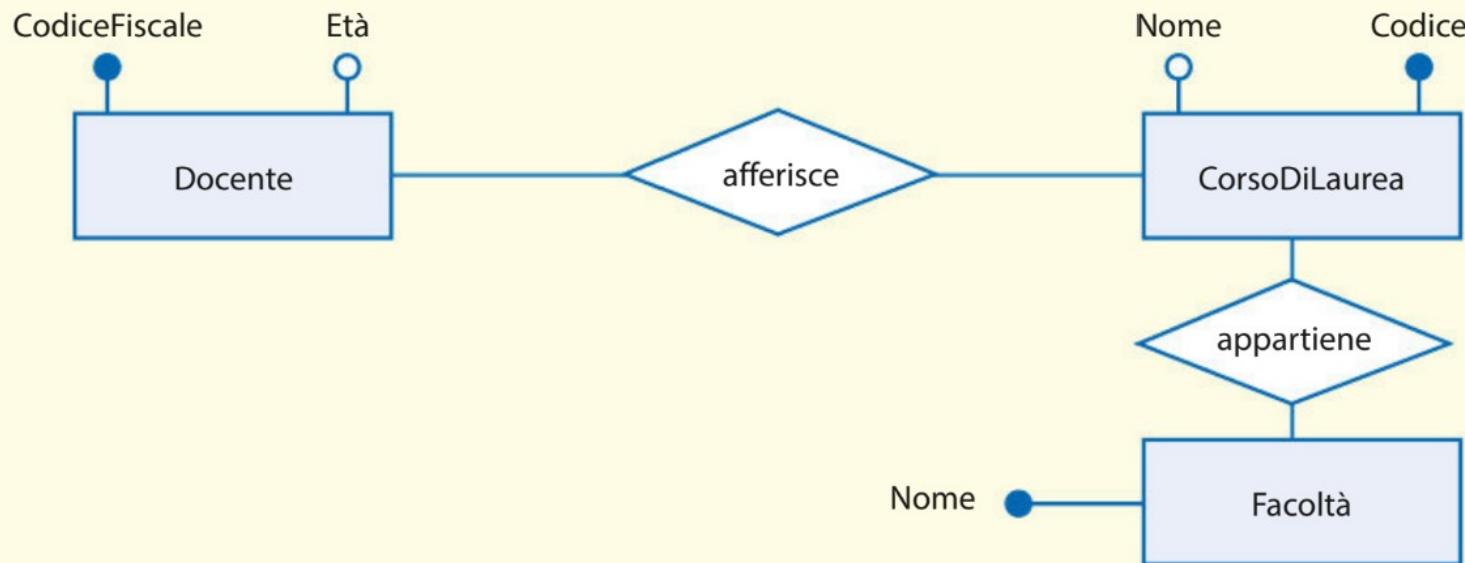
A livello estensionale, un'associazione *A* tra le entità *E* e *F* è costituita da un insieme di coppie (k,h) , tali che *k* è un'istanza di *E*, e *h* è un'istanza di *F*. Ogni coppia è detta **istanza dell'associazione *A***.

Un possibile livello estensionale dell'associazione *possiede* potrebbe essere il seguente:



OSSERVA COME SI FA

1. Descriviamo lo schema concettuale della seguente realtà:



I docenti operano nei corsi di laurea (si dice che *afferiscono* ai corsi di laurea). I docenti hanno un codice fiscale e una età. I corsi di laurea hanno un codice e un nome, e appartengono alle facoltà. Ogni facoltà ha un nome.

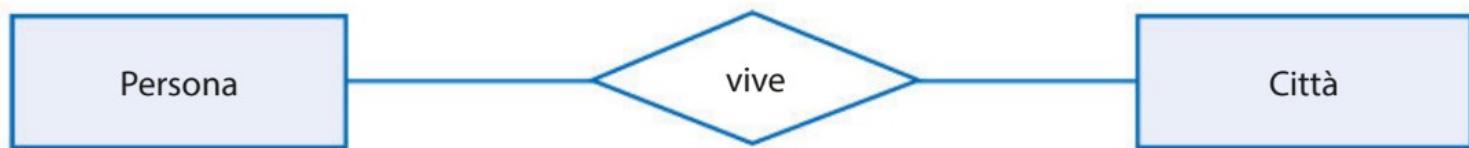
Il grado dell'associazione

Si definisce **grado** il numero di entità che partecipano all'associazione.

Rifletti

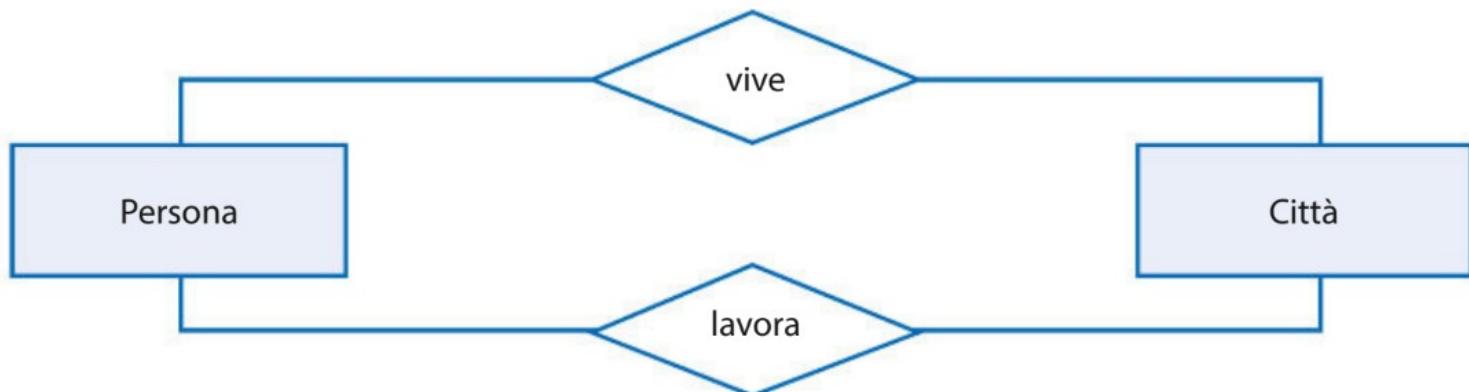
Nelle associazioni non binarie (ad esempio, nelle associazioni ternarie) un'occorrenza di un'associazione R non è una coppia, ma un'ennupla.

Ad esempio, la seguente associazione *vive*:

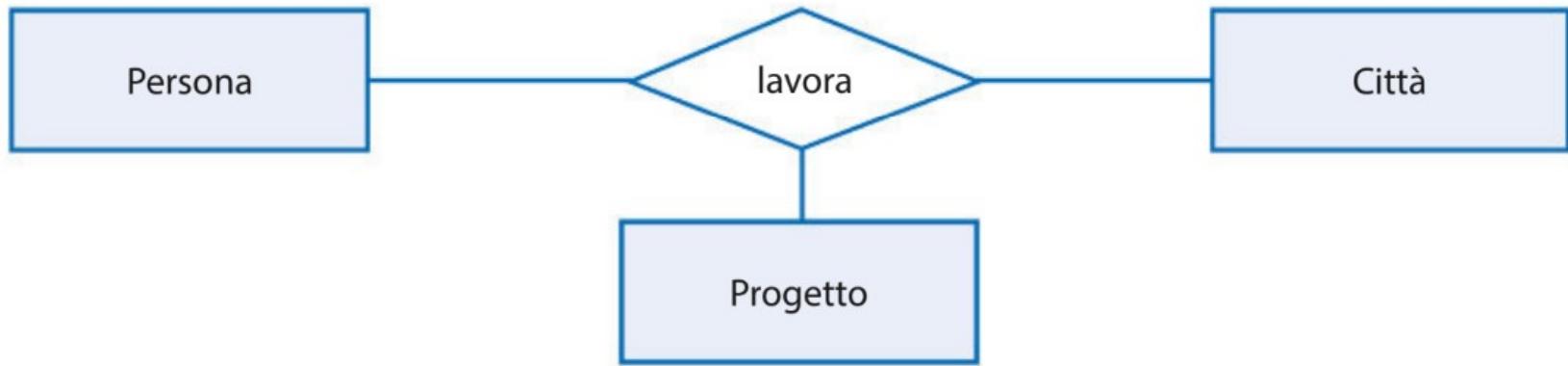


è di grado 2 e si definisce **binaria**.

Tra due entità è possibile definire più associazioni binarie, come nel seguente esempio:



La seguente, invece, è un'associazione di grado 3, definita **ternaria** (o, genericamente, **n-aria**):



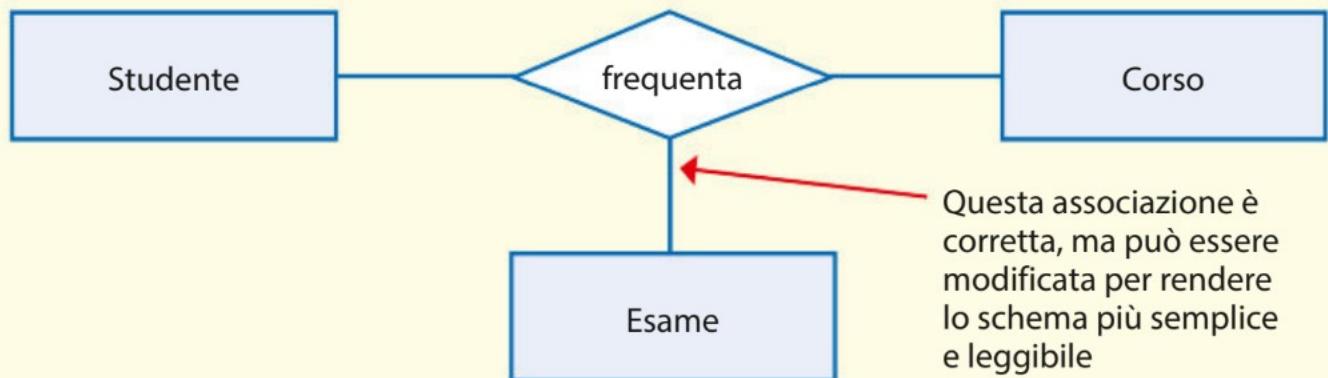
Rifletti

Nelle associazioni non binarie (ad esempio, nelle associazioni ternarie) un'occorrenza di un'associazione R non è una coppia, ma un'ennupla.

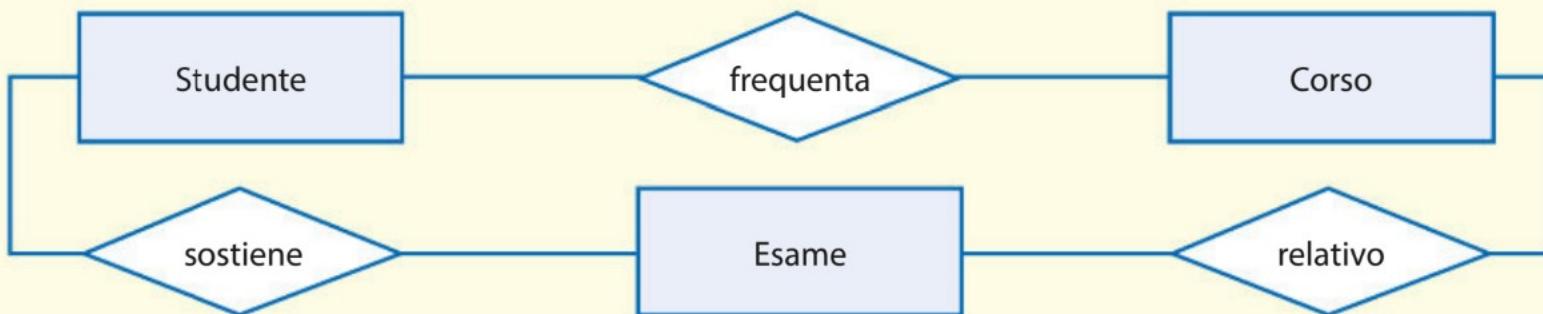
OSSERVA COME SI FA

- Le associazioni ternarie possono essere “esplose” al fine di ottenere diagrammi più leggibili.

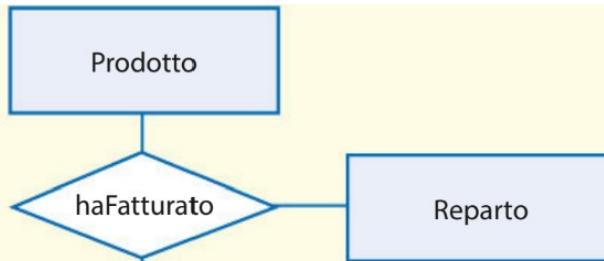
Osserva la seguente associazione ternaria:



Esplodendo l'associazione ternaria otteniamo:

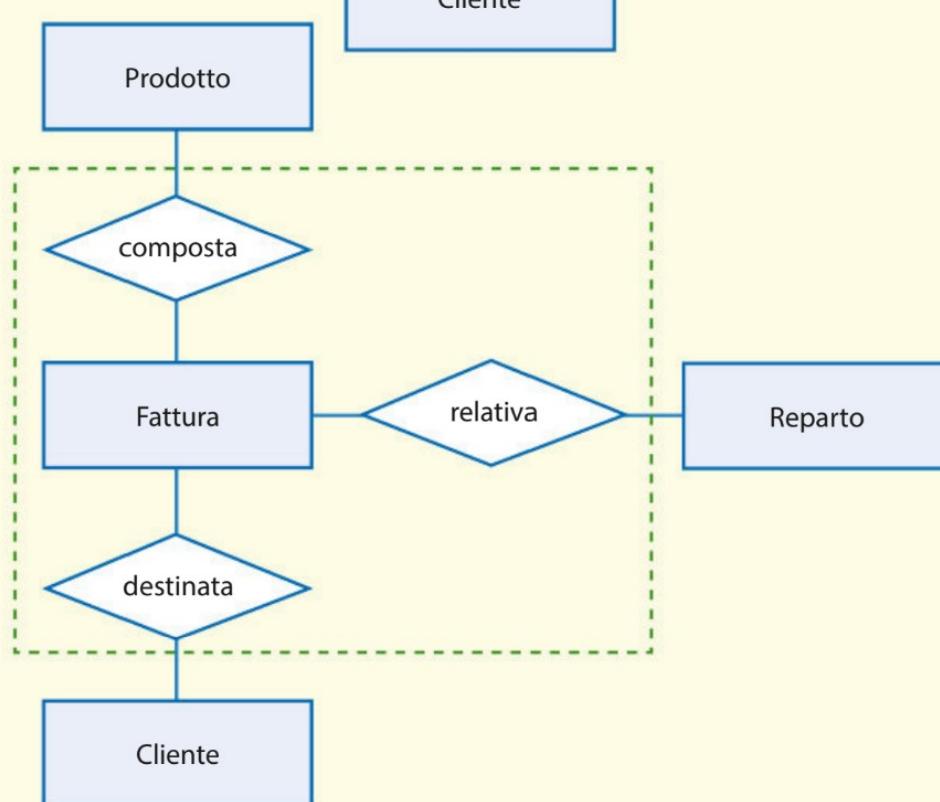


Vediamo ora un altro esempio. Consideriamo la realtà di un supermercato, in cui si vuole rappresentare il fatto che ogni reparto possa emettere fattura per i clienti. Di ogni reparto si vogliono conoscere alcuni dati riassuntivi e identificativi. Possiamo utilizzare l'associazione ternaria *haFatturato*.



Esplodiamo l'associazione *haFatturato* nel modo illustrato qui a lato.

L'associazione *haFatturato* è stata trasformata in entità (*Fattura*) e sono state aggiunte tre associazioni necessarie per garantire il collegamento con le tre entità presenti. Non è quindi riduttivo considerare solo le associazioni binarie tra entità, poiché qualsiasi associazione multipla potrà sempre essere trasformata in più associazioni binarie.



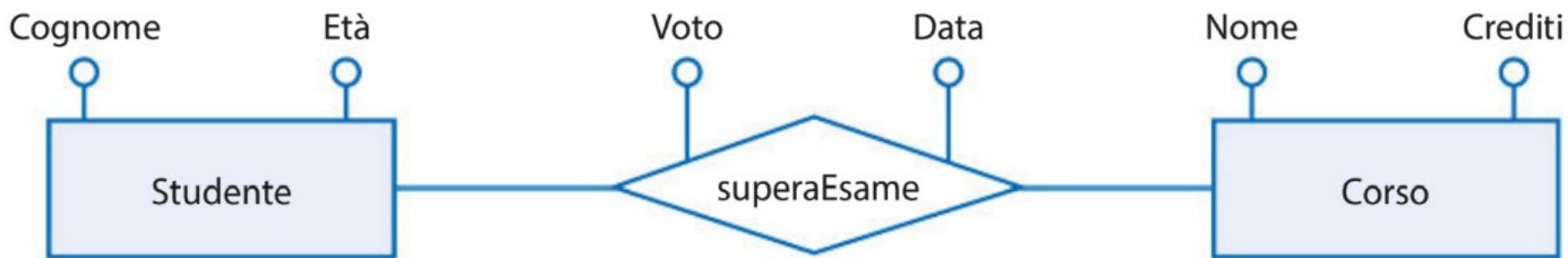
Gli attributi dell'associazione

Anche le associazioni, esattamente come le entità, possono avere degli attributi.

Un **attributo di associazione** è una proprietà locale di un'associazione, di interesse ai fini dell'applicazione.

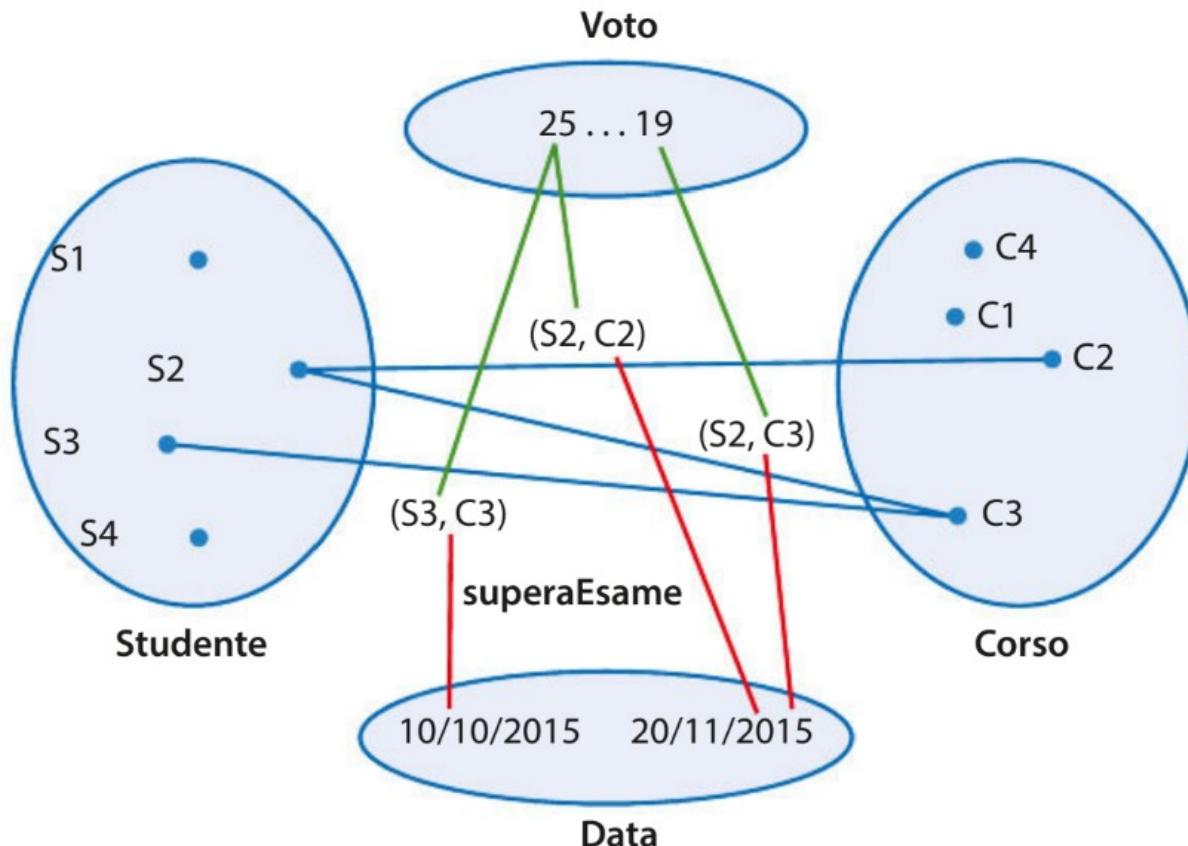
Un attributo associa a ogni istanza di relazione R un valore appartenente a un insieme D detto **dominio** dell'attributo. Quindi, a livello estensionale, l'attributo A è una funzione totale, cioè è una funzione che associa a ogni istanza di R un valore appartenente al dominio D dell'attributo.

Diamo uno sguardo al seguente schema (livello intensionale):



Gli attributi *Voto* e *Data* non sono proprietà né di uno studente né di un corso, **ma del legame Studente-Corso** che si crea in occasione dello svolgimento di un esame.

Un possibile livello estensionale del precedente schema potrebbe essere il seguente:



7 Quando modellare con un attributo e quando con un'entità

Quando modelliamo una realtà di interesse, dobbiamo decidere se modellare un particolare concetto come entità o come attributo. Occorre scegliere:

- a seconda del contesto;
- a seconda dell'uso che ne vogliamo fare.

Vediamo un esempio chiarificatore, considerando il concetto di “Comune” nei seguenti contesti.

Primo contesto

Per creare un'applicazione di gestione anagrafica siamo interessati a descrivere diverse caratteristiche anagrafiche dell'entità *Persona*. In questo caso possiamo avere un'entità *Persona* che ha un attributo *Comune* (inteso come “comune di nascita”):

entità *Persona*:
attributi: *Nome*, *Cognome*, *Comune*.

L'attributo *Comune* è una caratteristica dell'entità *Persona* e sarà di tipo stringa. Assumerà valori del tipo:

“Milano”, “Roma”, “Pescara”, “Palermo” e così via.

Secondo contesto

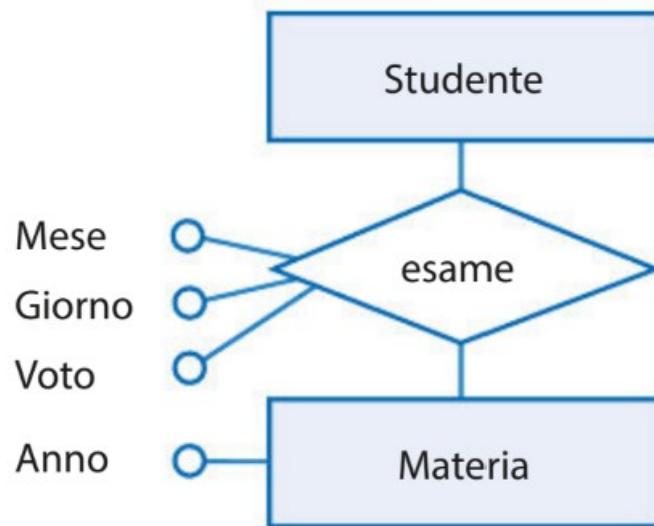
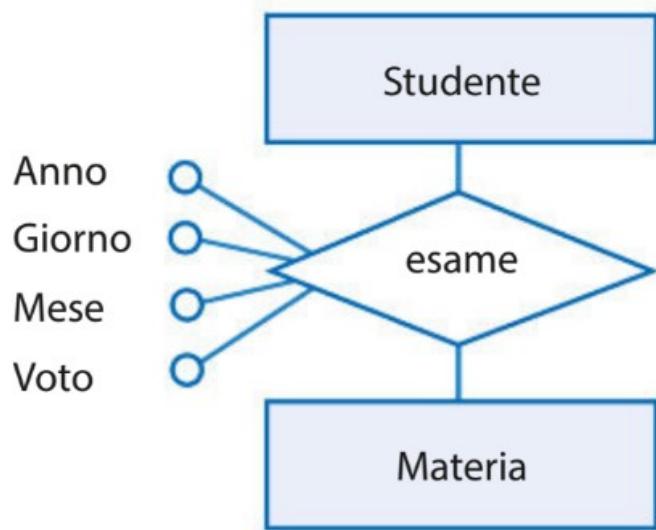
Per creare un'applicazione di gestione territoriale siamo interessati a descrivere diverse caratteristiche dei comuni. In questo caso possiamo avere un'entità *Comune* con i seguenti attributi:

entità *Comune*:
attributi: *Nome*, *Provincia*, *Regione*.

La domanda da porsi è: dobbiamo descrivere un insieme di proprietà che caratterizzano questo concetto?

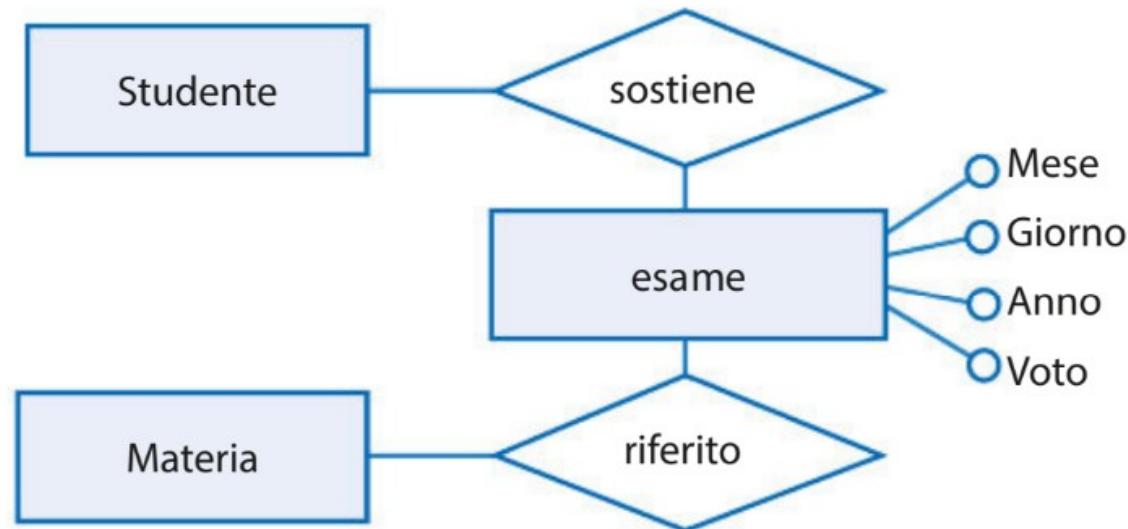
Se la risposta è affermativa creiamo un'entità, altrimenti creiamo un attributo.

Ora consideriamo il diagramma ER rappresentato nella figura qui sotto a sinistra, in cui abbiamo quattro attributi per specificare la data dell'interrogazione. Se ci stiamo riferendo alle interrogazioni relative a uno stesso anno, ovvero se l'attributo *Anno* ha un valore fisso, il diagramma diventa quello della seguente figura a destra.



L'attributo *Anno* è diventato un attributo dell'entità *Materia*.

Per eliminare tutti gli attributi di un'associazione, introdurremo una nuova entità che li contiene. Variando gli schemi appena visti otterremo:

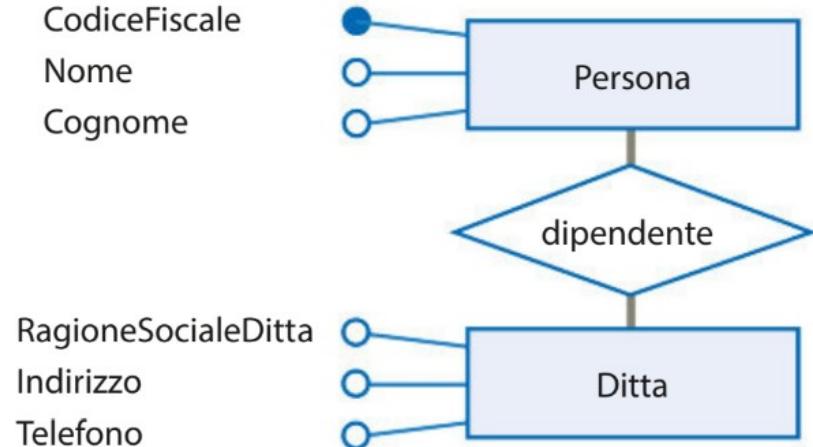
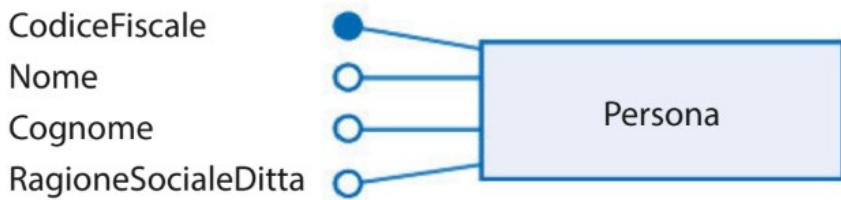


Ora analizziamo un diagramma ER in cui è presente l'entità *Persona*, caratterizzata tra l'altro dall'attributo *RagioneSocialeDitta* che indica il nome della ditta presso la quale una persona lavora.

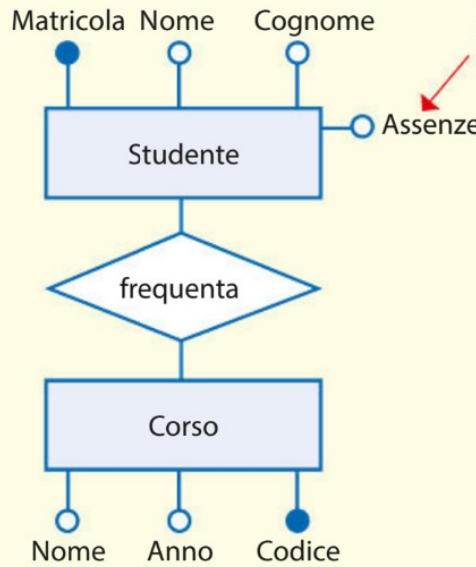
Se siamo interessati a conoscere solo la ragione sociale della ditta presso cui è impiegata quella persona, allora l'attributo *RagioneSocialeDitta* è sufficiente, ma se occorre conoscere anche altre informazioni relative a quella ditta, ad esempio indirizzo, numero di dipendenti, numero di telefono, allora occorrerà aggiungere altri attributi all'entità *Persona*.

Questa soluzione, però, non è conveniente poiché i valori di tali attributi verrebbero duplicati in ogni altra istanza di *Persona* che sia dipendente della stessa ditta. Inoltre, una modifica al numero di telefono di una particolare ditta dovrebbe essere riportata in tutte le istanze dove tale numero è duplicato.

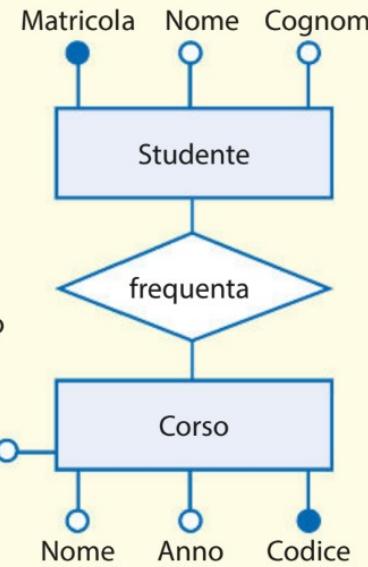
Per evitare questi inconvenienti aggiungiamo un'altra entità che chiamiamo *Ditta* e la mettiamo in relazione con *Persona*, ottenendo il diagramma ER visibile qui a destra.



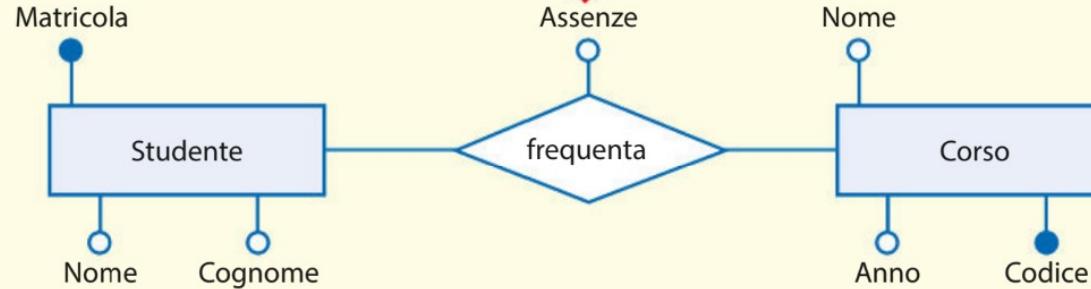
OSSERVA COME SI FA



Se inseriamo le assenze sullo studente, queste varranno per tutti i suoi corsi!

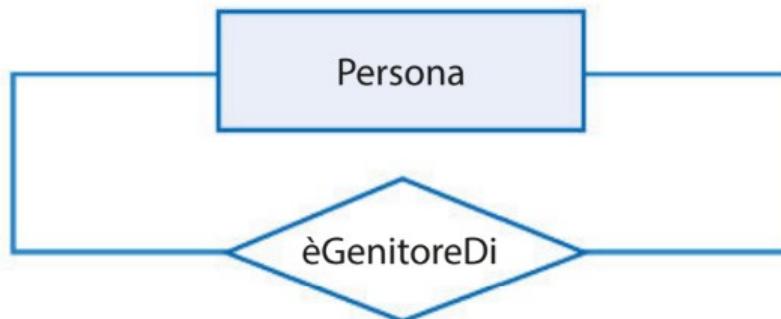


Se inseriamo le assenze sullo corso, queste varranno per tutti gli studenti!



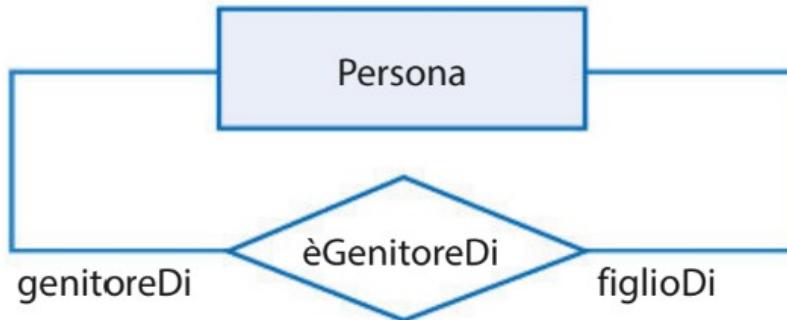
8 Le associazioni ricorsive e i ruoli

Quando un'associazione crea un legame tra una stessa entità, si ha il caso particolare di un'**associazione ricorsiva (o unaria)**. Ad esempio, consideriamo l'entità *Persona* e supponiamo di voler rappresentare l'associazione *èGenitoreDi*:



Dobbiamo distinguere, in ogni fatto che costituisce un'istanza dell'entità *Persona*, chi è il genitore e chi è il figlio. In termini grafici dobbiamo distinguere qual è il ramo che rappresenta il genitore e qual è il ramo che rappresenta il figlio.

Possiamo allora scrivere:



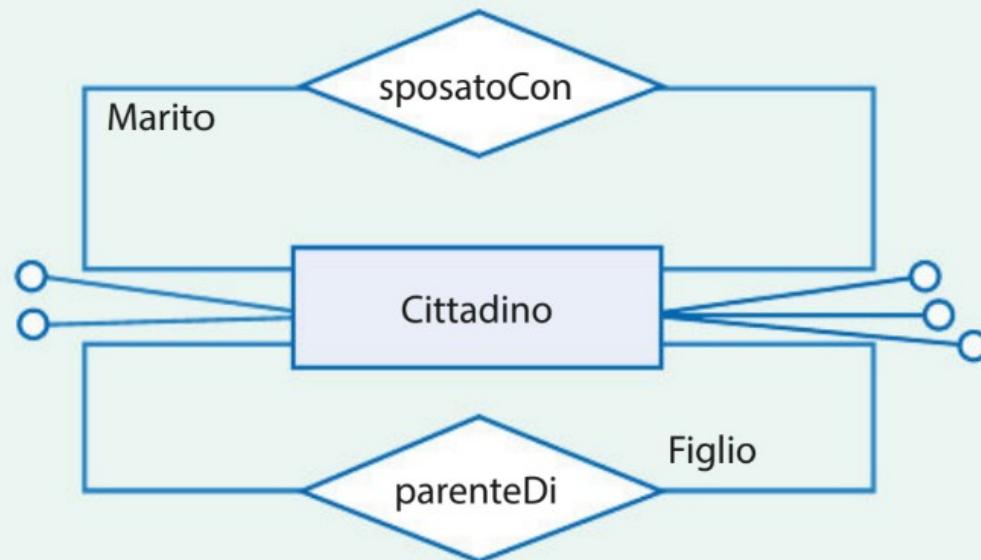
dove *genitoreDi* e *figlioDi* sono anche chiamati **ruoli** dell'entità nell'ambito dell'associazione. In questo diagramma l'entità *Persona* ha due diversi ruoli nell'ambito dell'associazione *genitoreDi*: il ruolo di genitore e il ruolo di figlio.

Il concetto di ruolo, che ha una funzione chiarificatrice, se necessario può essere esteso a qualsiasi associazione binaria, come nella seguente figura:

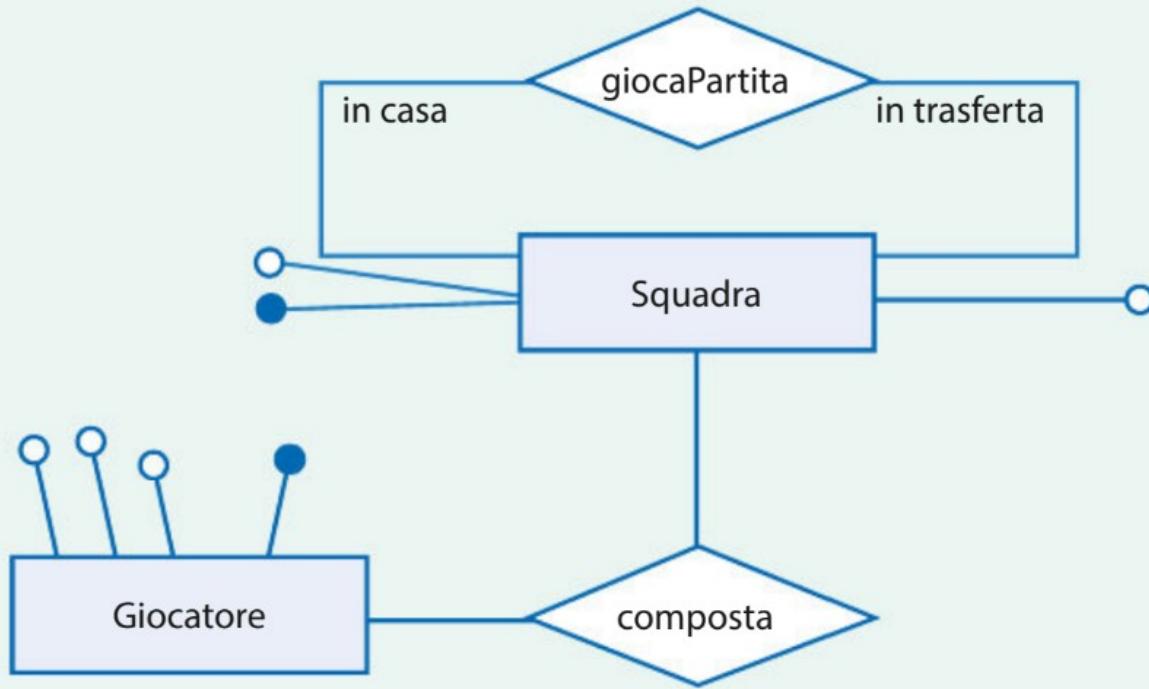


ORA TOCCA A TE

1. Completa il seguente schema ER inserendo i ruoli mancanti e alcuni attributi che ritieni utili:

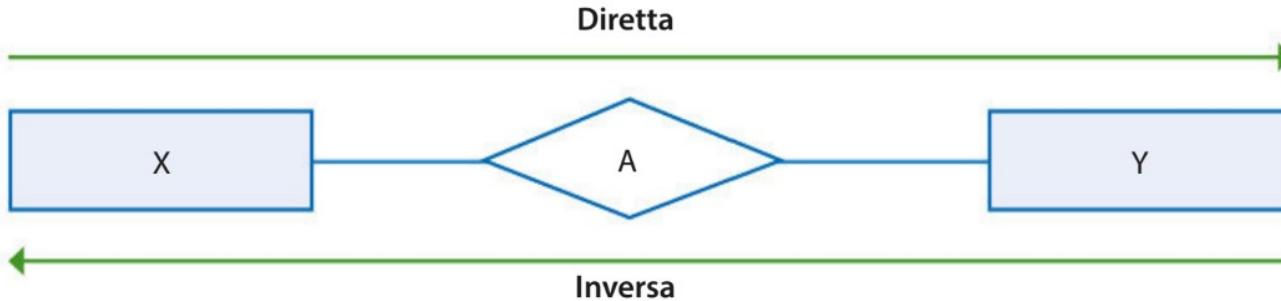


2. Completa il seguente schema ER inserendo i ruoli mancanti e gli attributi che ritieni utili. Inserisci anche le chiavi primarie:



9 Le proprietà delle associazioni

Data un'associazione A tra le entità X e Y , si attribuisce un verso da X a Y che definisce l'**associazione diretta** da X a Y . Definiamo **inversa** l'associazione da Y a X , che indichiamo con A^{-1} . Possiamo allora dire che la coppia (y, x) con $y \in Y$ e $x \in X$ appartiene ad A^{-1} se, e solo se, la coppia (x, y) appartiene ad A .



L'associazione *guida* ha quindi due versi: uno da *Persona* ad *Automobile* (relativo all'associazione diretta) e l'altro da *Automobile* a *Persona* (relativo all'associazione inversa).

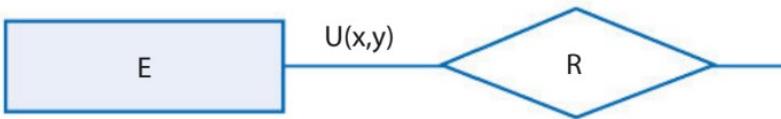


Un'associazione è caratterizzata da un vincolo di cardinalità.

Rifletti

Sottolineiamo che A e A^{-1} sono una l'inversa dell'altra. Sono quindi la stessa associazione, salvo che per l'ordine delle coppie. La scelta del verso relativo all'associazione diretta e di quello relativo all'inversa, quindi, è puramente arbitraria. Possiamo indicare l'associazione diretta e l'inversa con i nomi dei ruoli.

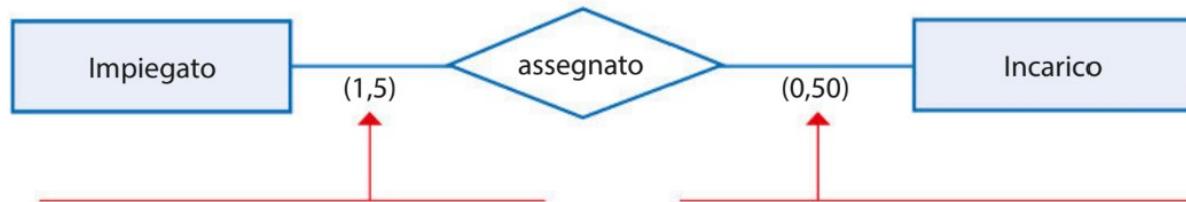
Un **vincolo di cardinalità** si associa a un ruolo U in una relazione R e impone un limite minimo x e un limite massimo y di istanze della relazione a cui può partecipare ogni istanza dell'entità E nel ruolo U .



Un vincolo di cardinalità si esprime mediante una coppia (x,y) associata al ruolo U della relazione R , dove:

- x è la **cardinalità minima**, cioè un intero ≥ 0 ;
- y è la **cardinalità massima**, che è n , oppure un intero positivo $\geq x$.

Analizziamo l'associazione *assegnato* tra le entità *Impiegato* e *Incarico* presente nel seguente schema ER:



Il primo ruolo (ossia l'associazione diretta), ha cardinalità minima 1 e cardinalità massima 5. Ciò significa che un impiegato può partecipare a un minimo di un'occorrenza e a un massimo di cinque occorrenze dell'associazione *assegnato*. In altri termini, ogni impiegato può avere da 1 a 5 incarichi.

Il secondo ruolo (ossia l'associazione inversa), ha cardinalità minima 0 e cardinalità massima 50. Ciò significa che un certo incarico può partecipare a nessuna occorrenza oppure al massimo a 50 occorrenze dell'associazione *assegnato*. In altri termini, un incarico può non essere assegnato ad alcun impiegato oppure può essere assegnato al massimo a 50 impiegati.

Nella maggiore parte dei casi, per esprimere la cardinalità è sufficiente utilizzare solo tre valori:

- 0;
 - 1;
 - N (che indica genericamente un intero maggiore di uno).
-

Se la cardinalità minima è

- 0: la partecipazione dell'entità E in R è **opzionale** (o **parziale**);
 - 1: la partecipazione dell'entità E in R è **obbligatoria** (o **totale**).
-

Cardinalità minima

Osserva il seguente schema ER:



La cardinalità minima tra *Persona* e *possiede* è 0. Ciò significa che la partecipazione dell'entità *Persona* è **facoltativa (parziale)**: possono esistere, infatti, persone che non possiedono auto.

La cardinalità minima tra *Automobile* e *possiede* è 0. Ciò significa che anche la partecipazione dell'entità *Automobile* è **facoltativa (parziale)**: possono esistere, infatti, auto che non hanno ancora proprietari (non sono state ancora immatricolate).

Ora osserva attentamente questo nuovo schema ER:



La cardinalità minima tra *Comune* e *ubicato* è 1. Ciò significa che la partecipazione dell'entità *Comune* è **obbligatoria (totale)**: ogni comune dove essere ubicato in una provincia.

La cardinalità minima tra *Provincia* e *ubicato* è 1. Ciò significa che anche la partecipazione dell'entità *Provincia* è **obbligatoria (totale)**: ogni provincia dove prevedere dei comuni al suo interno, ossia non possono esserci province prive di comuni.

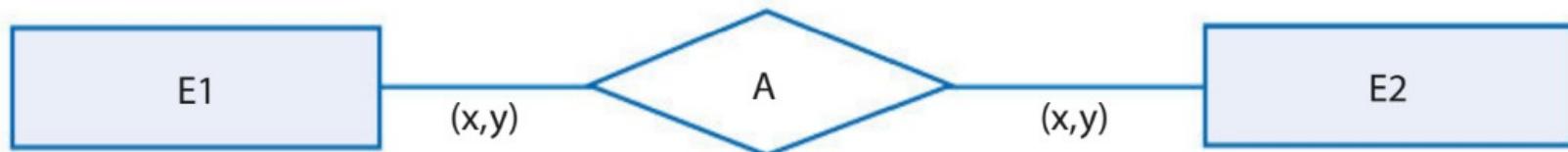
Cardinalità massima

Esaminiamo ora la cardinalità massima; quando è:

- **1**, le istanze dell'entità partecipano al più 1 volta alle istanze dell'associazione;
- **N**, le istanze dell'entità partecipano un numero qualsiasi di volte alle istanze dell'associazione.

10 Tipi di associazioni

In base al valore assunto dal parametro della cardinalità massima, un'associazione binaria A tra due entità $E1$ e $E2$:



è definita come segue.

- **Uno a Uno (1, 1) o biunivoca**, se le cardinalità massime di entrambe le entità rispetto ad A sono 1.
- **Uno a Molti (1, N) o semplice**, se la cardinalità massima di $E1$ rispetto ad A è 1 e la cardinalità massima di $E2$ rispetto ad A è N .
- **Molti a Molti (N, N) o complessa**, se le cardinalità massime di entrambe le entità rispetto ad A sono N .

ASSOCIAZIONE 1:1

Osserviamo il seguente schema ER:

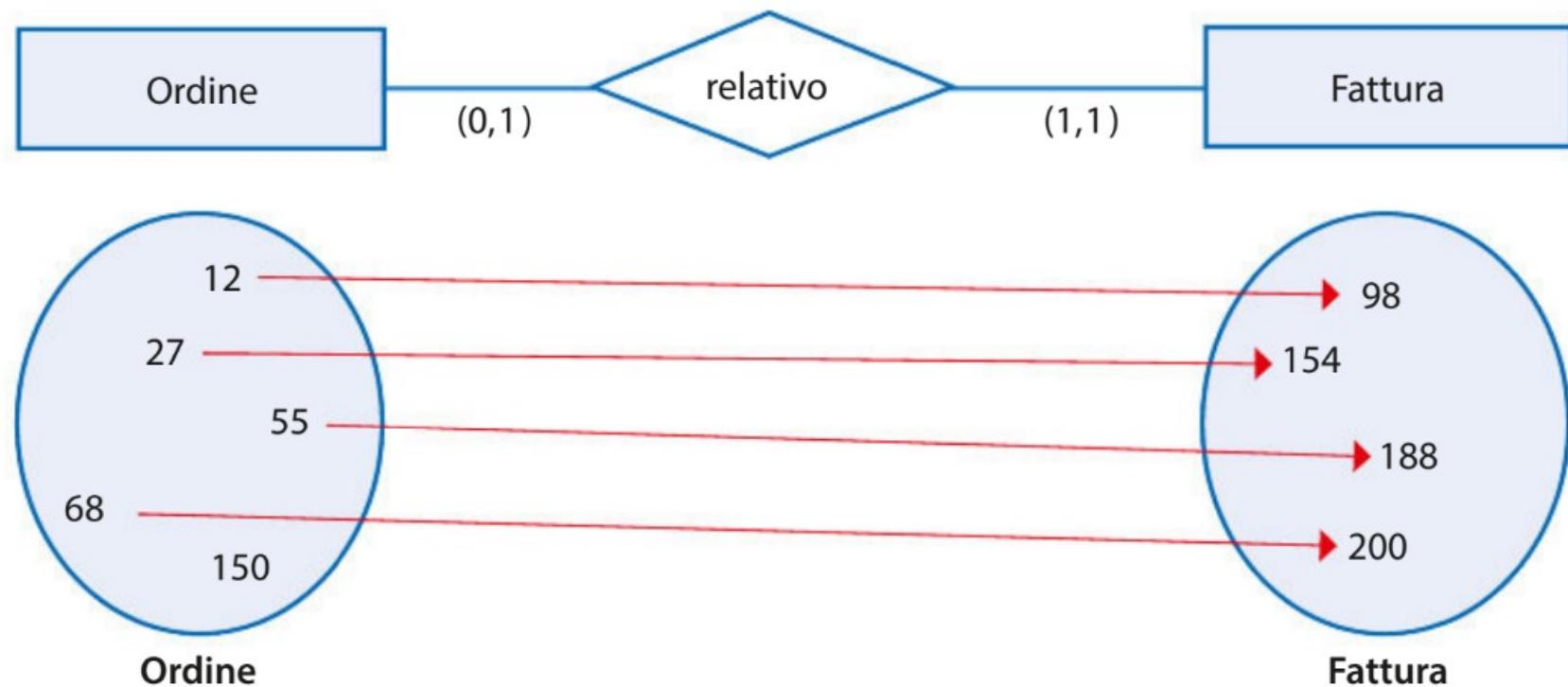


Analizziamolo cominciando con la diretta. Un ordine può fare riferimento a una sola fattura. Una volta effettuato, l'ordine potrebbe non avere ancora una fattura a cui fare riferimento (ordini ancora da evadere). La diretta è, pertanto, parziale, e la cardinalità è (0,1).

Passiamo all'inversa. Una fattura, per poter essere emessa, deve obbligatoriamente fare riferimento a un solo ordine. L'inversa è, pertanto, totale e la cardinalità è (1,1).

La cardinalità dell'associazione *relativo* è, quindi, 1:1.

Lo schema ER completo è, quindi, il seguente:



ORA TOCCA A TE

- Completa il seguente schema ER riportando gli attributi delle due entità che ritieni utili, la cardinalità e la partecipazione dei due ruoli.



Associazione diretta: Totale Parziale

Associazione inversa: Totale Parziale

Associazione *appartiene*: (,)

2. Completa il seguente schema ER riportando gli attributi delle due entità che ritieni utili, la cardinalità e la partecipazione dei due ruoli.



Associazione diretta: Totale Parziale

Associazione inversa: Totale Parziale

Associazione *coordina*: (_,_)

3. Completa il seguente schema ER riportando gli attributi delle due entità che ritieni utili, la cardinalità e la partecipazione dei due ruoli.



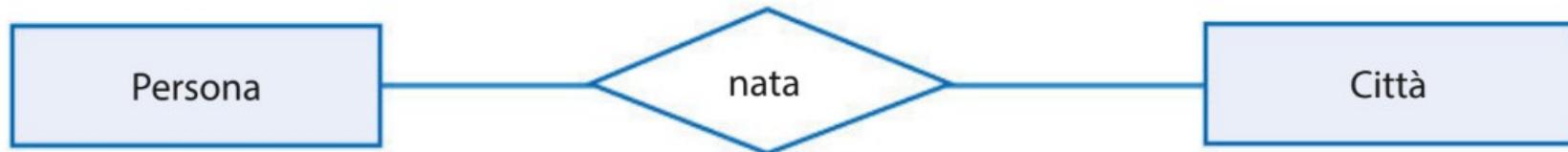
Associazione diretta: Totale Parziale

Associazione inversa: Totale Parziale

Associazione *diretto*: (,)

ASSOCIAZIONE 1:N

Osserviamo il seguente schema ER:

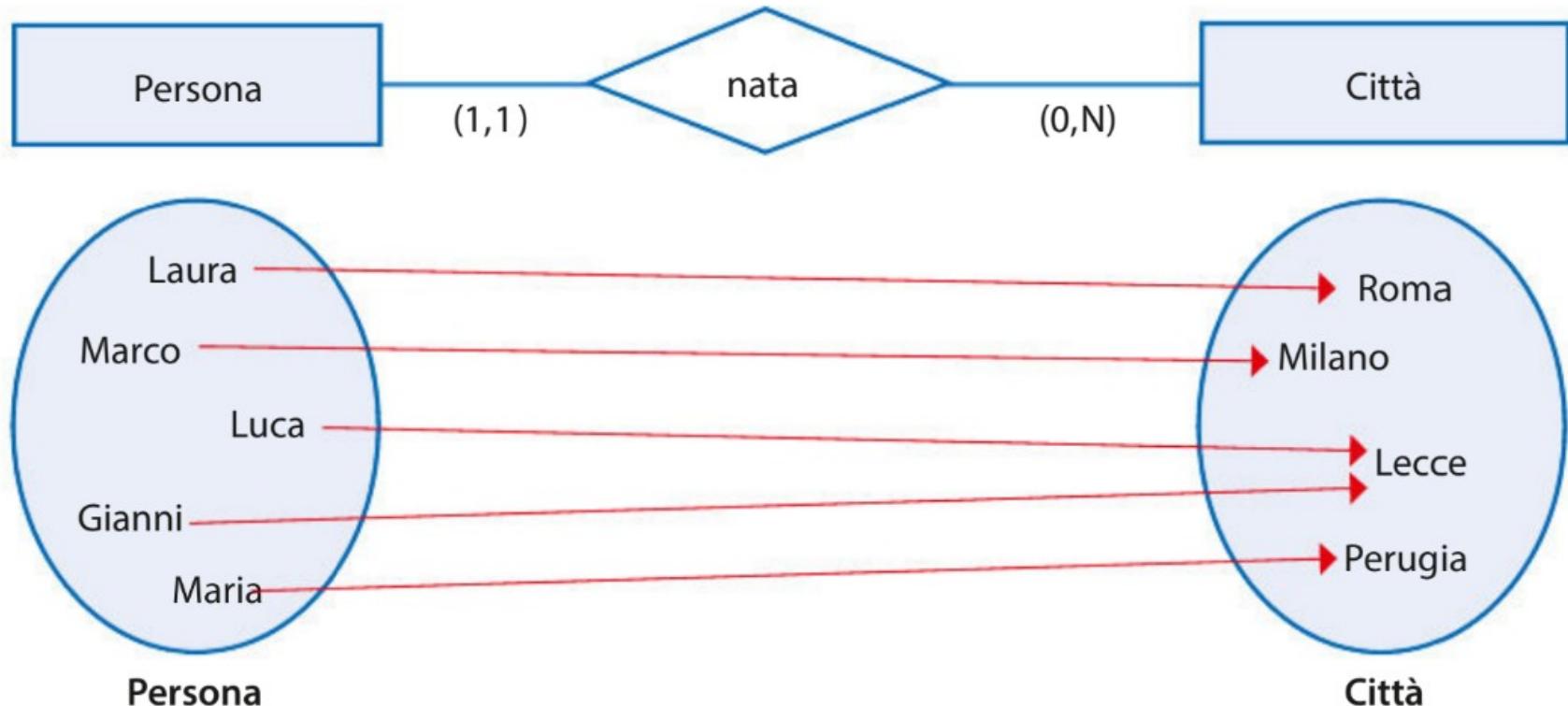


Analizziamolo partendo dalla diretta. Una persona nasce in una e una sola città. La diretta è, pertanto, totale e la cardinalità è (1,1)

Passiamo all'inversa. In una città può anche non nascere nessuna persona, ma ne potrebbero nascere molte. L'inversa è, pertanto, parziale e la cardinalità è (0,N).

La cardinalità dell'associazione *nata* è, quindi, 1:N.

Lo schema ER completo è il seguente:



ORA TOCCA A TE

1. Completa il seguente schema ER riportando gli attributi delle due entità che ritieni utili, la cardinalità e la partecipazione dei due ruoli.



Associazione diretta: Totale Parziale

Associazione inversa: Totale Parziale

Associazione *lavora*: (_, _)

2. Completa il seguente schema ER riportando gli attributi delle due entità che ritieni utili, la cardinalità e la partecipazione dei due ruoli.



Associazione diretta: Totale Parziale

Associazione inversa: Totale Parziale

Associazione *dispone*: (_, _)

- 3. Completa il seguente schema ER riportando gli attributi delle due entità che ritieni utili, la cardinalità e la partecipazione dei due ruoli.**



Associazione diretta: Totale Parziale

Associazione inversa: Totale Parziale

Associazione *ubicato*: (__ , __)

ASSOCIAZIONE N:N

Osserviamo il seguente schema ER:

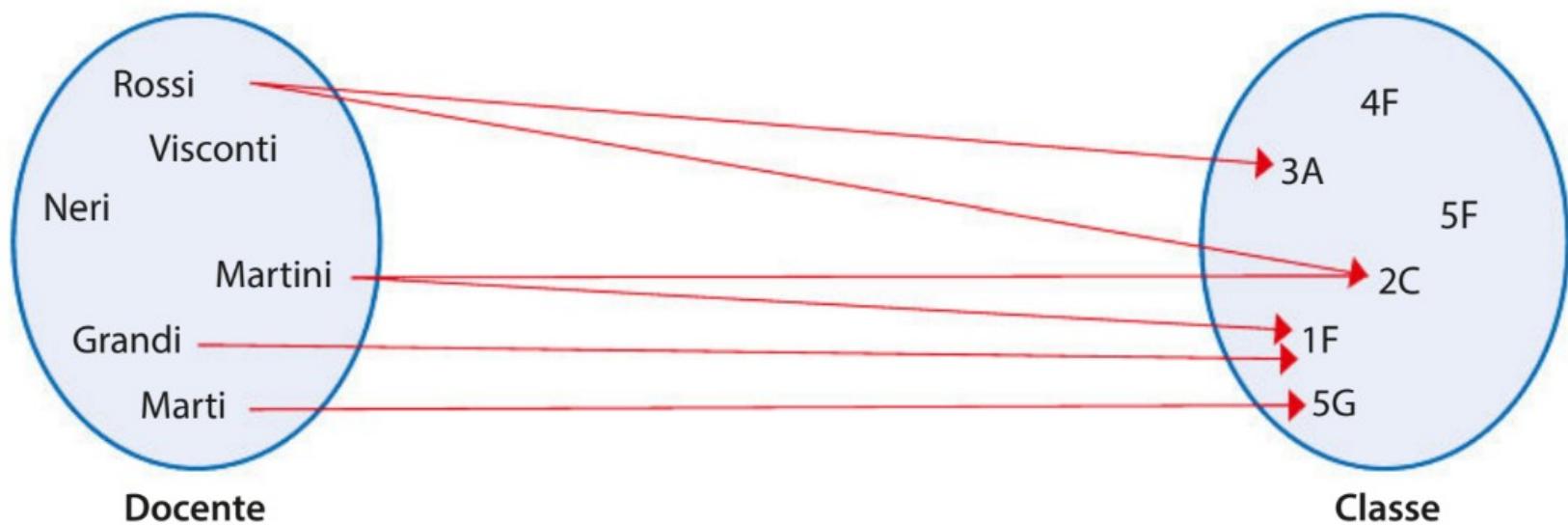


Analizziamolo partendo dalla diretta. Un docente può insegnare in più di una classe, ma potrebbe anche non essere stato ancora assegnato a nessuna. La diretta è, pertanto, parziale e la cardinalità è (0,N).

Passiamo all'inversa. In una classe insegnano più docenti, ma può anche accadere che in un determinato momento (ad esempio prima dell'inizio dell'anno scolastico) la classe non abbia ancora assegnati i docenti che insegheranno. L'inversa è, pertanto, parziale e la cardinalità è (0,N).

La cardinalità dell'associazione *insegna* è, quindi, N:N.

Lo schema ER completo è il seguente:



ORA TOCCA A TE

1. Completa il seguente schema ER riportando gli attributi delle due entità che ritieni utili, la cardinalità e la partecipazione dei due ruoli.



Associazione diretta: Totale Parziale

Associazione inversa: Totale Parziale

Associazione *sostiene*: (_,_)

- 2.** Completa il seguente schema ER riportando gli attributi delle due entità che ritieni utili, la cardinalità e la partecipazione dei due ruoli.



Associazione diretta: Totale Parziale

Associazione inversa: Totale Parziale

Associazione scalata: (_, _)

- 3.** Completa il seguente schema ER riportando gli attributi delle due entità che ritieni utili, la cardinalità e la partecipazione dei due ruoli.



Associazione diretta: Totale Parziale

Associazione inversa: Totale Parziale

Associazione effettua: (_, _)

Ricapitolando:

Cardinalità	Partecipazione
(1,1)	Obbligatoria, una sola volta
(1,N)	Obbligatoria, almeno una volta
(0,1)	Opzionale, una sola volta
(0,N)	Opzionale, N volte

11 Un primo esempio svolto

Per un corretto sviluppo di uno schema ER è necessario svolgere l'analisi del problema, l'analisi dei dati e l'analisi delle associazioni, prima di pervenire allo schema concettuale. Ricordiamo inoltre che è doveroso elencare i vincoli di integrità (che tratteremo più avanti e che, pertanto, non compaiono in questo esempio) e che, nel caso in cui la traccia dell'esercizio sia molto analitica, si può evitare di svolgere l'analisi del problema (sarebbe ridondante) e iniziare direttamente con quella dei dati.

Iniziamo con una semplice realtà di interesse: quella relativa a una videoteca. Diciamo "semplice" poiché la soluzione che forniremo è semplice e riduttiva, considerato che questo è il primo esercizio completo che presentiamo. La soluzione fornita potrà essere ulteriormente dettagliata e perfezionata.

OSSERVA COME SI FA

1. Automatizziamo la gestione del prestito di DVD in un negozio specializzato.

Analisi del problema

La videoteca gestisce il noleggio di DVD; ogni DVD è caratterizzato da alcune informazioni generali e dai dati relativi agli attori (con i rispettivi ruoli) e ai registi, inoltre appartiene a una o più categorie (film d'azione, romantico, dell'orrore e così via). I DVD vengono noleggiati dai clienti, i quali devono aver richiesto e ottenuto precedentemente una tessera valida fino al termine dell'anno solare di emissione e sono identificati dal numero di tale tessera. Quando un cliente richiede la tessera, non è obbligato a noleggiare subito un film, ma nel momento in cui lo fa si apre un contratto di noleggio al quale sono associati il cliente e il DVD (almeno uno) noleggiato. Vogliamo avere disponibile sia la situazione dello stato di noleggio dei DVD, sia la descrizione completa di tutti i noleggi effettuati. Un cliente può restituire in date diverse i DVD noleggiati con un unico contratto.

Analisi dei dati (entità)

Dall'analisi del problema si possono facilmente individuare le seguenti entità:

- *Cliente*, per rappresentare le informazioni riguardanti i clienti della videoteca;
- *DVD*, per memorizzare le informazioni inerenti le caratteristiche del DVD;
- *Attore* per le informazioni degli attori che interpretano il film registrato su DVD;
- *Regista* per le informazioni dei registi che dirigono il film;
- *Categoria* per memorizzare i tipi di film (azione, giallo, storico e così via).

Gli attributi di *Cliente* sono numero della tessera, data di emissione della tessera, cognome, nome, indirizzo, città di residenza, telefono. Il numero della tessera è di tipo numerico, mentre i restanti attributi sono di tipo stringa.

Gli attributi di *DVD* sono codice, titolo e durata; tutti sono di tipo stringa.

Gli attributi di *Attore* sono codice, cognome e nome; tutti sono di tipo stringa.

Gli attributi di *Regista* sono codice, cognome e nome; tutti sono di tipo stringa.

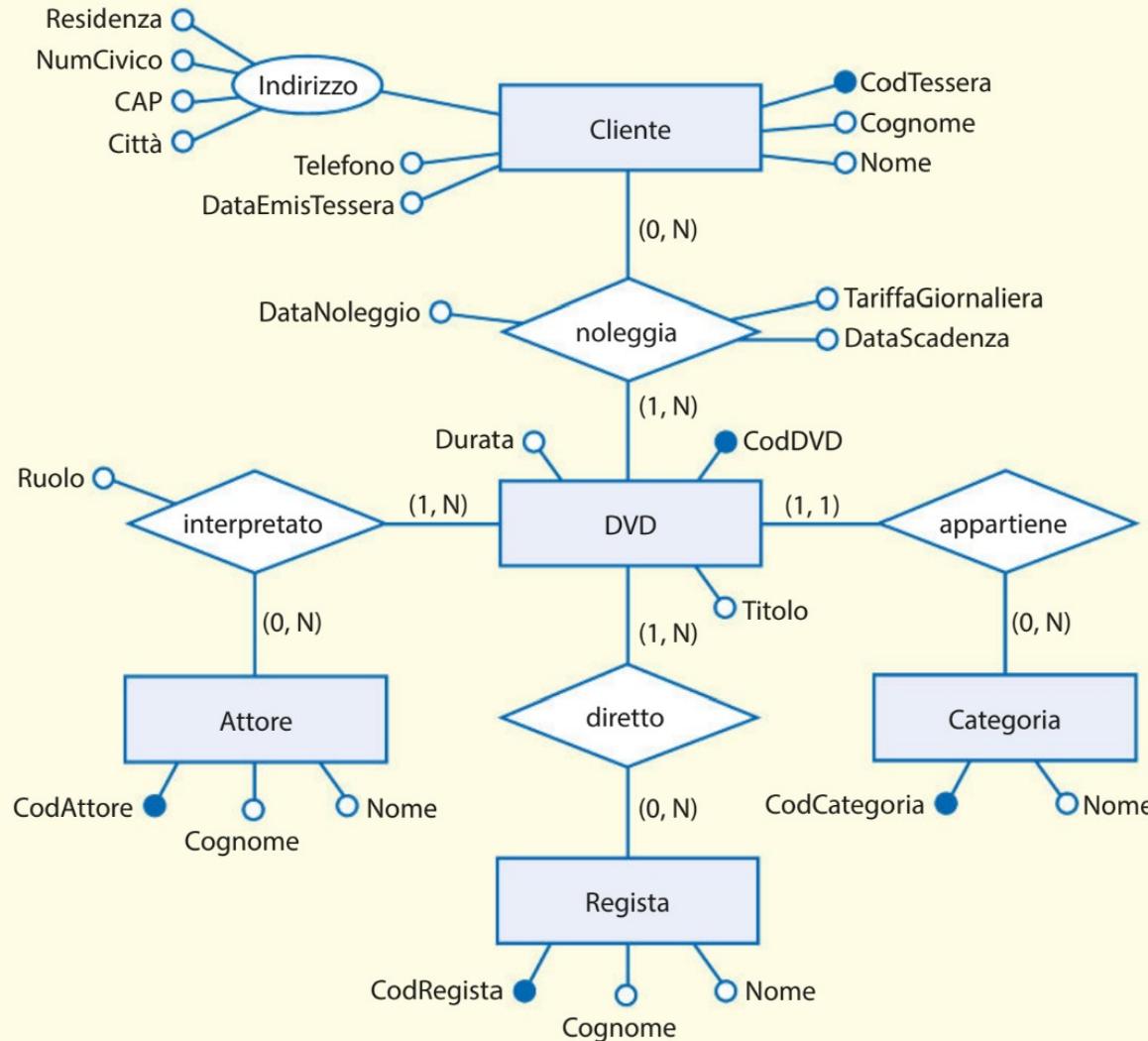
Gli attributi di *Categoria* sono codice e nome, entrambi di tipo stringa.

Analisi delle associazioni

- Tra l'entità *Cliente* e l'entità *DVD* esiste un'associazione *noleggia* di tipo molti a molti (N,N). Infatti, la cardinalità massima di entrambe le entità rispetto all'associazione è pari a N. Ciò significa che un cliente può noleggiare più DVD, e un DVD può essere noleggiato da più clienti. La partecipazione dell'entità *Cliente* in *noleggia* è opzionale (parziale) in quanto un cliente può non aver ancora noleggiato alcun DVD, mentre la partecipazione dell'entità *DVD* in *noleggia* è obbligatoria (totale) poiché il noleggio di un DVD deve essere sempre riferito a un cliente. L'associazione *noleggia* ha tre attributi, rispettivamente *DataNoleggio*, *DataScadenza* e *TariffaGiornaliera*, utili per la gestione del contratto di noleggio.
- Tra l'entità *DVD* e l'entità *Attore* esiste un'associazione *interpretato* di tipo (cardinalità) molti a molti (N,N). Infatti, la cardinalità massima di entrambe le entità rispetto all'associazione è pari a N. Ciò significa che un DVD ha memorizzato al suo interno un film interpretato da più attori e un attore può interpretare più film. La partecipazione dell'entità *DVD* in *interpretato* è obbligatoria (totale) poiché il film registrato su un DVD deve essere interpretato da almeno un attore, mentre la partecipazione dell'entità *Attore* in *interpretato* è facoltativa (parziale) poiché un attore può non essere il protagonista di alcun film. L'associazione *interpretato* ha l'attributo *Ruolo* che identifica il ruolo dell'attore nell'ambito del film.
- Tra l'entità *DVD* e l'entità *Categoria* esiste un'associazione *appartiene* di tipo uno a molti (1,N). Infatti, la cardinalità massima dell'entità *DVD* rispetto ad *appartiene* (ossia la diretta) è 1, mentre la cardinalità massima di *Categoria* rispetto ad *appartiene* (ossia l'inversa) è N. Ciò significa che un DVD appartiene a una sola categoria, e a una categoria possono appartenere più DVD. La partecipazione dell'entità *DVD* in *appartiene* è obbligatoria (totale) poiché un DVD deve appartenere a una sola categoria, mentre la partecipazione dell'entità *Categoria* in *appartiene* è facoltativa (parziale) poiché a una specifica categoria può non appartenere alcun DVD.
- Tra l'entità *DVD* e l'entità *Regista* esiste un'associazione *diretto* di tipo molti a molti (N,N). Infatti, la cardinalità massima di entrambe le entità rispetto all'associazione è pari a N. Ciò significa che il film registrato su un DVD può essere diretto da più registi, e un regista può dirigere più film. La partecipazione dell'entità *DVD* in *diretto* è obbligatoria (totale) poiché un film memorizzato su DVD deve essere diretto da almeno un regista, mentre la partecipazione dell'entità *Regista* in *diretto* è facoltativa (parziale) poiché un regista può anche non dirigere alcun film tra quelli presenti nella nostra videoteca.

Schema concettuale

Siamo ora in grado di formalizzare lo schema concettuale secondo le regole del modello ER:



12 I vincoli di integrità

Finora abbiamo visto come modellare una realtà di interesse usando le entità, con le loro proprietà, e le associazioni. Gran parte delle strutture che si vogliono modellare possono essere espresse in una di queste due notazioni, ma esistono alcuni importanti aspetti del mondo reale che non possono essere modellati con gli strumenti visti finora: i vincoli di integrità.

Possiamo definire **vincolo di integrità** una proprietà che deve essere soddisfatta dalle istanze che rappresentano informazioni corrette per l'applicazione che utilizza la base di dati.

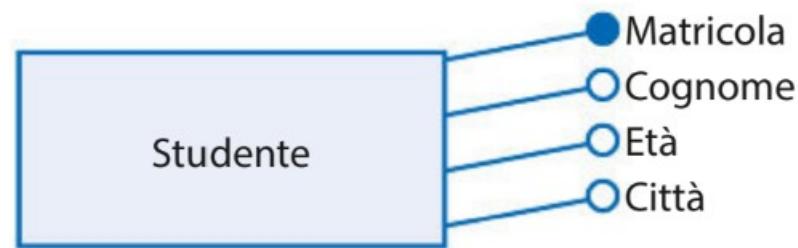
I vincoli di integrità sono costituiti da predicati che devono essere soddisfatti dalle istanze delle varie entità che compongono la base di dati.

Consideriamo l'entità *Studente*, che contiene informazioni anagrafiche relative a uno studente. Supponiamo che l'attributo *Matricola* sia un numero progressivo che identifica esattamente uno studente.

L'attributo *Età*, di tipo intero, potrebbe assumere un valore molto alto, ad esempio pari a 250. Dal punto di vista concettuale questa assegnazione è corretta (infatti 250 è un valore del dominio intero), ma non ha molto senso in questa realtà: è del tutto irreale (anche se desiderabile!) vivere fino a 250 anni.

È necessario, quindi, specificare un vincolo di integrità che indichi che l'età deve essere inferiore a 120.

Un altro vincolo imporrà l'impossibilità di avere due studenti con il medesimo numero di matricola.



Riprendiamo in esame il problema del noleggio dei DVD appena esaminato. In base alle nostre conoscenze abbiamo potuto modellare la realtà di interesse individuando entità, associazioni e caratteristiche dei singoli attributi. Per condurre una progettazione concettuale ampia e completa, però, occorre precisare alcune informazioni che non possono essere rappresentate con i simboli che abbiamo studiato. Riferendoci al problema del noleggio dei DVD, informazioni aggiuntive utili possono essere le seguenti:

1. la data di emissione della tessera deve essere antecedente a quella odierna;
2. la data del noleggio deve essere successiva a quella di emissione della tessera;
3. la data di scadenza del contratto di noleggio deve essere successiva a quella di stipula del contratto stesso.

Questi sono vincoli che la base di dati deve soddisfare per non compromettere l'integrità dei dati.

Vincoli impliciti ed esplicativi

Nell'ambito del modello concettuale, si è soliti classificare i vincoli di integrità in impliciti ed esplicativi.

I **vincoli impliciti** sono quelli imposti dalla semantica e struttura dei dati e si dividono in:

- **vincoli di chiave primaria**; impongono che le istanze di una categoria debbano essere tutte diverse tra loro;
- **vincoli referenziali**; date due entità A e B e un'associazione tra loro, questi vincoli impongono che non si possa inserire un elemento in A (o non esista un elemento in A) che non sia associato a un elemento in B. Ciò può anche essere visto come un vincolo sulla partecipazione delle associazioni.

I **vincoli esplicativi** sono quelli che impongono delle restrizioni sul modo in cui i dati possono cambiare. Un esempio di vincolo esplicito è: “il valore dell'attributo *Età* non può essere negativo né maggiore di 120”.

I vincoli impliciti si rappresentano direttamente con simboli del diagramma ER e precisamente:

- i vincoli di chiave primaria si rappresentano sottolineando i relativi attributi;
- i vincoli referenziali si rappresentano con le linee continue delle associazioni tra entità (rappresentazione della partecipazione di un'associazione).

Per rappresentare i vincoli esplicativi ricorriamo a una nostra sintassi, che possiamo definire ricorrendo a uno pseudolinguaggio:

V<NumProgressivo>: (<Espressione>)

dove:

- <NumProgressivo> è un numero progressivo del vincolo;
- <Espressione> è una qualsiasi espressione in pseudolinguaggio o in linguaggio naturale che serve per specificare il vincolo.

Per esprimere il vincolo sull'*Età* scriveremo:

V1: (0 < Studente.Età < 120)

Per esprimere il vincolo (poco realistico, ma inserito solo per comprendere l'argomento) che si possono immatricolare solo studenti che vivono a Roma, scriveremo:

V2: (Studente.Città = "Roma")

Come si può notare, per fare riferimento a un attributo di una particolare entità o associazione, utilizziamo la seguente sintassi:

```
<NomeEntità>.<NomeAttributo>
```

oppure:

```
<NomeAssociazione>.<NomeAttributo>
```

Un altro tipico esempio di vincolo esplicito è il **vincolo di cardinalità delle associazioni**.

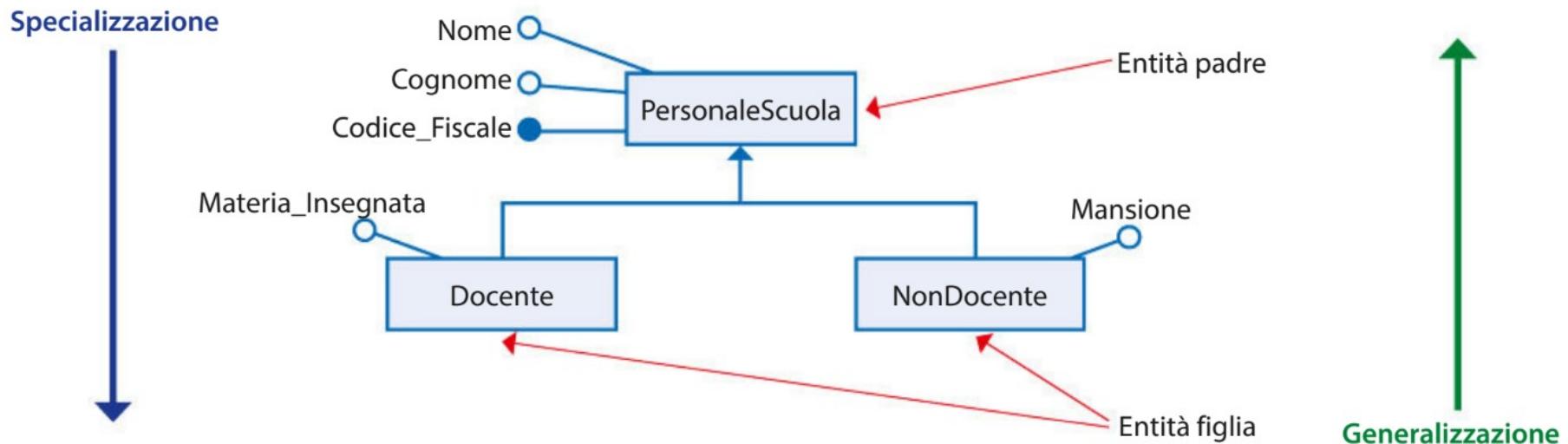
13 Collezione di entità e gerarchie

Quando alcune entità possono avere delle istanze in comune, oppure quando nasce l'esigenza di voler particolareggiare o generalizzare qualche entità, è **utile procedere con un'organizzazione per gerarchie**.

Una gerarchia ha una struttura ad albero: la radice rappresenta l'entità che contiene le caratteristiche comuni (**entità padre**), mentre i vari nodi (**entità figlie**) rappresentano le entità che contengono caratteristiche più specifiche.

Per creare una gerarchia si attua un **processo di specializzazione**, ossia si “specializza” l'entità padre creando entità figlie sempre più specifiche (aggiungendo agli attributi del padre altri attributi che le specializzano). Percorrendo la gerarchia dal basso verso l'alto si attua automaticamente un **processo di generalizzazione** in quanto si risale l'albero passando dalle entità più specializzate a quelle più generiche (si risale verso gli attributi comuni).

Ad esempio, l'entità *PersonaleScuola* può essere pensata come una generalizzazione delle entità *Docente* e *NonDocente*. In questo modo è possibile rappresentare il fatto che istanze di *Docente* e *NonDocente* possono anche essere classificate come istanze di *PersonaleScuola*. Questo approccio consente di prescindere dalle proprietà che le rendono semanticamente diverse (ad esempio *Materia_Insegnata* per i docenti e *Mansione* per i non docenti) ed evidenziare, invece, le proprietà comuni, ad esempio *Codice_Fiscale*, *Cognome*, *Nome*.



Rifletti

Nel modello ER un'associazione di generalizzazione è anche conosciuta con il termine **associazione ISA** (dall'inglese *is a*, cioè "è un").

Naturalmente il processo di specializzazione può proseguire, nel senso che una entità figlia può essere a sua volta una entità padre per altre entità modellando, così, gerarchie a più livelli. Ad esempio, l'entità figlia *NonDocente* potrebbe essere specializzata ulteriormente in *PersonaleSegreteria* e *CollaboratoreScolastico*.

Ogni istanza di un'entità figlia è anche istanza dell'entità padre. Per questo motivo, ogni entità figlia **eredita** tutte le proprietà (attributi) dell'entità padre. Ad esempio, nel nostro caso l'entità *NonDocente*, oltre all'attributo *Mansione*, possiede anche gli attributi *Cognome*, *Nome* e *Codice_Fiscale*.

Le generalizzazioni possono essere classificate in base a due proprietà, **copertura** e **disgiunzione**, che sono ortogonali tra loro; quindi è possibile avere quattro configurazioni di generalizzazione:

- la generalizzazione ha copertura **totale** se ogni istanza della classe padre è una istanza di almeno una delle figlie, altrimenti è **parziale**;
- la generalizzazione ha disgiunzione **esclusiva** se ogni istanza della classe padre è una istanza di al più una delle classi figlie, altrimenti si dice **sovraposta**.

		DISGIUNZIONE	
		Esclusiva	Sovrapposta
COPERTURA	Total	<pre> graph TD Mansione((Mansione)) --- Docente[Docente] Mansione --- NonDocente[NonDocente] Docente --- PersonaleScuola[PersonaleScuola] NonDocente --- PersonaleScuola PersonaleScuola --- Nome1((Nome)) PersonaleScuola --- Cognome1((Cognome)) PersonaleScuola --- CodFiscale1((Cod_fiscale)) </pre> <p>Totale (i Docenti e i NonDocenti costituiscono tutto il personale della scuola). Esclusiva (una persona della scuola è un Docente o un NonDocente).</p>	<pre> graph TD Mansione((Mansione)) --- DocentiCentrale[Docenti Centrale] Mansione --- DocentiSuccursale[Docenti Succursale] Mansione --- NonDocenti[NonDocenti] DocentiCentrale --- PersonaleScuola[PersonaleScuola] DocentiSuccursale --- PersonaleScuola NonDocenti --- PersonaleScuola PersonaleScuola --- Nome2((Nome)) PersonaleScuola --- Cognome2((Cognome)) PersonaleScuola --- CodFiscale2((Cod_fiscale)) </pre> <p>Totale (l'unione delle tre figlie concorrono a formare tutte le istanze di PersonaleScuola). Sovrapposta (alcuni docenti possono insegnare sia nella sede Centrale sia in Succursale).</p>
	Parziale	<pre> graph TD Mansione((Mansione)) --- Docenti[Docenti] Mansione --- Impiegati[Impiegati] Docenti --- PersonaleScuola[PersonaleScuola] Impiegati --- PersonaleScuola PersonaleScuola --- Nome3((Nome)) PersonaleScuola --- Cognome3((Cognome)) PersonaleScuola --- CodFiscale3((Cod_fiscale)) </pre> <p>Parziale (i docenti e gli impiegati non concorrono a formare tutto il personale della scuola). Esclusiva (il docente non può ricoprire il ruolo di impiegato e viceversa).</p>	<pre> graph TD Mansione((Mansione)) --- DocentiCentrale[Docenti Centrale] Mansione --- DocentiSuccursale[Docenti Succursale] DocentiCentrale --- PersonaleScuola[PersonaleScuola] DocentiSuccursale --- PersonaleScuola PersonaleScuola --- Nome4((Nome)) PersonaleScuola --- Cognome4((Cognome)) PersonaleScuola --- CodFiscale4((Cod_fiscale)) </pre> <p>Parziale (i soli docenti non concorrono a formare tutto il personale della scuola). Sovrapposta (alcuni docenti possono insegnare sia nella sede Centrale sia in Succursale).</p>