

La progettazione concettuale: il modello ER

- Terminologia e concetti introduttivi sulle basi di dati.
- Elementi di metodologia per la progettazione di basi di dati non a oggetti.

Prerequisiti

- Modellare una realtà utilizzando il diagramma ER.

Obiettivi

- Conoscere il concetto di entità e attributo.
- Conoscere il concetto di attributo chiave.
- Conoscere le rappresentazioni grafiche di entità e attributo.
- Conoscere le rappresentazioni grafiche delle associazioni tra entità.
- Conoscere le associazioni totali e parziali.
- Conoscere la rappresentazione delle associazioni di aggregazione e generalizzazione.
- Conoscere il concetto di vincolo.

Conoscenze da apprendere

- Saper rappresentare una realtà di interesse con un diagramma ER.

Competenze da acquisire

1 La progettazione concettuale

La documentazione sulle specifiche, prodotta dopo la fase di raccolta delle richieste, va in ingresso alla successiva fase di **progettazione concettuale**.

La **progettazione concettuale** consiste nel riorganizzare tutti gli elementi che si hanno a disposizione per definire un modello astratto della base di dati. Tale modello è destinato a diventare un documento ufficiale di riferimento e di comunicazione per i progettisti della successiva fase di *progettazione logica*.

In molti casi, il documento concettuale costituisce il tramite di verifica tra committenti e progettisti. I termini **astratto** e **concettuale** si riferiscono 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. Si cerca invece di attribuire una certa *struttura* ai dati (il significato verrà chiarito più avanti).

Nella progettazione concettuale assumono un ruolo di primaria importanza le **astrazioni**.

1.1 Le astrazioni

Abbiamo già introdotto il *concetto di astrazione* nel secondo volume, in particola-

re nelle unità dedicate agli oggetti. È importante a questo punto riprenderlo e considerare gli aspetti più importanti per i nostri scopi.

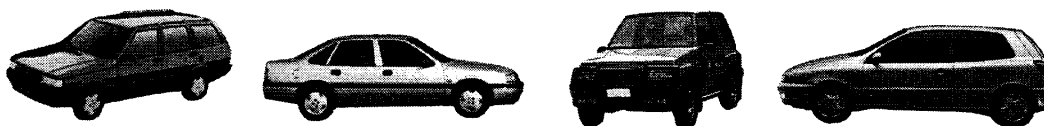
L'astrazione è un procedimento mentale che permette di evidenziare alcune proprietà, ritenute significative, degli oggetti osservati, escludendone altre giudicate non rilevanti.

Tramite l'astrazione si arriva a definire nuove classi di oggetti.

Esempio

Osserviamo i seguenti oggetti.

♦ Fig. A2.1
Il processo di astrazione che porta al concetto di automobile



È possibile selezionare gli aspetti in comune (colore, numero ruote ecc.) ed escludere gli aspetti che li differenziano, realizzando un processo di astrazione che porta al concetto di automobile. È questo il modo più adatto di descrivere questa classe di oggetti.

L'automobile è un'astrazione di una vasta classe di oggetti con caratteristiche simili. Se fossimo costruttori di automobili e partecipassimo a un congresso in cui si discute di tecnologia delle auto, usare il termine automobile sarebbe troppo astratto e ci converrebbe utilizzare termini più concreti come *station wagon*, *monovolume*, *berlina*, *fuoristrada*; utilizzeremmo cioè un diverso livello di astrazione nel descrivere gli oggetti di interesse.

Procedimenti di
astrazione

Nella progettazione di basi di dati vengono utilizzati principalmente tre procedimenti di astrazione:

- *classificazione*;
- *aggregazione*;
- *generalizzazione*.

1.2 Astrazione per classificazione

L'astrazione per classificazione permette di definire una classe a partire da un insieme di oggetti di cui si individuano proprietà comuni.

È importante notare che gli elementi di partenza di un processo di *astrazione per classificazione* sono oggetti della realtà e l'esito dell'utilizzo di tale processo è una classe di oggetti. Tali oggetti vengono detti *istanze* o *esemplari*.

Esempio

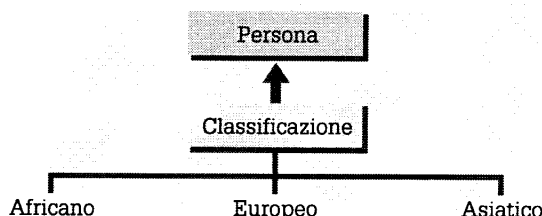
Analizziamo le seguenti immagini.

♦ Fig. A2.2
Un esempio di astrazione per classificazione



Se individuiamo le caratteristiche comuni ai soggetti presi in esame (respirano; hanno una testa, due braccia e due gambe; parlano; hanno un nome e cognome ecc.) e scartiamo quelle che differenziano gli oggetti della precedente Fig. A2.1, arriviamo a definire una classe *Persona*.

Possiamo rappresentare tale processo con il seguente schema:



♦ Fig. A2.3
La rappresentazione grafica del processo di astrazione per classificazione

1.3 Astrazione per aggregazione

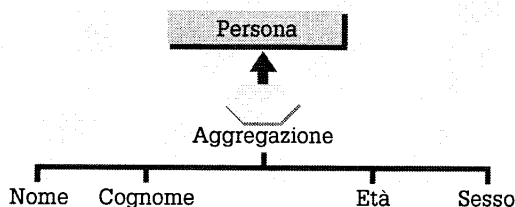
L'**astrazione per aggregazione** è il procedimento mentale che usiamo quando, a partire da una o più classi, generiamo una nuova classe. Le classi di partenza vengono dette **classi componenti o proprietà**, in quanto la loro unione contribuisce a realizzare la classe di arrivo.

Proseguiamo con il nostro esempio.

Esempio

Consideriamo le seguenti classi: *Nome*, *Cognome*, *Età*, *Sesso* delle persone e cerchiamo di individuare la classe di cui nome, cognome, età, sesso sono caratteristiche o parti componenti: è naturale arrivare all'individuazione della classe *Persona*, che rappresentiamo con il seguente schema:

♦ Fig. A2.4
Rappresentazione grafica del processo di astrazione per aggregazione



Come possiamo notare, l'aggregazione è un procedimento mentale diverso rispetto a quello che abbiamo usato nell'astrazione per classificazione. Nell'aggregazione si parte da un insieme di classi per ottenere un'altra classe. Nella classificazione si parte da un insieme di oggetti per ottenere una classe.

Differenze tra aggregazione e classificazione

1.4 Astrazione per generalizzazione

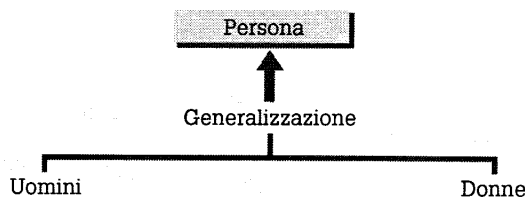
L'**astrazione per generalizzazione** è il procedimento mentale mediante il quale si può giungere alla definizione di una classe come unione di un insieme di classi, ognuna delle quali è contenuta nella classe da definire.

Poiché ogni classe è formata da un insieme di istanze, il legame che unisce le classi di partenza e la classe di generalizzazione è un legame di **sottoinsieme** di contenimento, e quindi un legame diverso rispetto al legame di aggregazione.

Serviamoci anche questa volta di un esempio.

Esempio

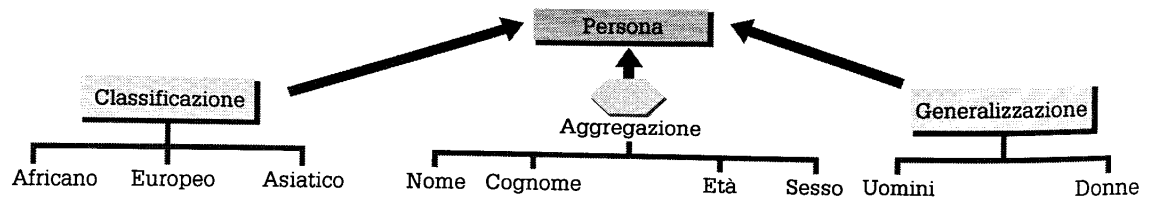
Consideriamo le seguenti classi di partenza: *Uomini* e *Donne*. Cerchiamo ora una classe che è l'unione delle due classi di partenza. Naturalmente ci viene in mente la classe *Persona* che è unione di queste due classi. Possiamo rappresentare la generalizzazione nel seguente modo:



♦ Fig. A2.5
Rappresentazione grafica del processo di astrazione per generalizzazione

Da quanto abbiamo visto nei nostri esempi possiamo arrivare alla definizione della stessa classe *Persona* tramite tre processi mentali completamente differenti e cioè:

♦ Fig. A2.6
La rappresentazione grafica dei vari processi di astrazione



I tre processi mentali (le tre astrazioni) sono indipendenti, ovvero non è possibile, ad esempio, dar luogo a una generalizzazione utilizzando un'aggregazione o una classificazione e viceversa.

Riassumendo:

- nel caso della **classificazione** partiamo da un insieme di oggetti e generiamo una classe;
- nel caso dell'**aggregazione** e della **generalizzazione** partiamo da un insieme di classi e generiamo una nuova classe ma:
 - nel caso dell'**aggregazione** le classi di partenza sono **classi componenti**, in quanto la loro unione contribuisce a realizzare la classe di arrivo;
 - nel caso della **generalizzazione** le classi di partenza sono un **sottoinsieme** della classe di arrivo.

Le *astrazioni* sono una modalità di *descrizione della realtà* comune a tutti i modelli, che possiamo utilizzare per la progettazione di una base di dati.

2 Il diagramma ER

Chen

Il diagramma *Entità/Associazioni* (in inglese **E/R**, da *Entity/Relationship*), introdotto nel 1976 da **Peter P. Chen**, è un modello grafico per la descrizione dei dati e delle loro relazioni in una realtà di interesse. È uno *strumento per l'analisi* delle caratteristiche di una realtà indipendentemente dagli eventi che in essa accadono.

Il diagramma ER, quindi, costruisce un modello concettuale dei dati *indipendente dalle applicazioni*.

Gli elementi di base di un diagramma E/R sono:

- le *entità*;
- gli *attributi*;
- le *associazioni*;
- la *generalizzazione*;
- l'*aggregazione*;
- i *vincoli di integrità*;
- gli *identificatori*.

Da ora in poi, come già fatto in precedenza, utilizzeremo la scrittura ER al posto di E/R per questi diagrammi.

3 Entità, istanze e loro attributi

Le **entità** corrispondono a **classi di oggetti** del mondo reale. Sono definite specificando le proprietà o attributi dei loro elementi.

Gli elementi di un'entità vengono chiamati **istanze dell'entità**, così come chiamavamo *istanze* gli oggetti di una classe.

Esempi di *entità* sono:

- una *persona*;
- un'*automobile*;
- un *movimento* contabile;
- una *prova* sostenuta da uno studente.

Esempio

Gli studenti sono classificabili nell'entità *Studente* (ogni studente rappresenta un'istanza dell'entità *Studente*); i diversi modelli di automobile sono classificabili nell'entità *Automobile*.

La scelta delle entità e dei suoi attributi dipende dal contesto della realtà di interesse.

Esempio

Se stiamo considerando come realtà di interesse una profumeria che vende esclusivamente profumi, è ovvio considerare *Profumo* come entità. Essere o meno un'entità dipende dal contesto nel quale la si considera. Se infatti consideriamo come realtà di interesse un supermercato, l'entità da considerare sarà *Articolo* e non più *Profumo* (pur avendo i supermercati l'area profumeria).

In alcuni contesti, utilizzeremo *entità* e *classe* come sinonimi, pur tenendo ben presente la differenza (nell'accezione più comune le classi infatti comprendono anche i metodi).

3.1 Attributi semplici

Le proprietà elementari (cioè non strutturabili in proprietà più atomiche) delle entità e delle associazioni vengono descritte attraverso gli **attributi semplici**.

Esempi di attributi per l'entità *Persona* nel contesto di un ufficio anagrafico sono:

Nome, Cognome, Età, Sesso

Attributi per l'entità *BottigliaDiVino* nel contesto di una enoteca, possono essere, ad esempio:

Nome, Produttore, Capacità, Prezzo, Vitigno

Quest'ultimo esempio è interessante perché se di una certa bottiglia di vino sono disponibili diverse istanze allora con *BottigliaDiVino* possiamo intendere due concetti diversi: la singola bottiglia fisica o l'insieme delle bottiglie che hanno tutte la stessa caratteristica.

Chiaramente, nel caso intendessimo la singola bottiglia di vino, dovremmo introdurre un altro attributo (ad esempio un numero progressivo), che possa distinguere quella particolare bottiglia dalle altre, altrimenti va bene quanto abbiamo specificato sopra.

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);
- una **dimensione** (non vale per i tipi data o booleani), 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;

Caratteristiche
di un attributo

Esempio

Considerando l'entità *Persona* possiamo dire che:

- il *dominio* dell'attributo *età*: 1, 2, 3, ... può arrivare a un valore massimo che possiamo assumere di 120;
- il *dominio* dell'attributo *Sesso* è {uomo, donna};
- il *dominio* di *Cognome* è rappresentato da {tutte le possibili stringhe di caratteri}.

Valore nullo (Null)

- una **opzionalità**, che indica la possibilità di non essere sempre valorizzato. Diremo che un attributo è **obbligatorio** se deve avere valore non nullo (ad esempio il nome di una persona in un'anagrafica); è **facoltativo** quando sono accettabili valori nulli (ad esempio il titolo di studio di una persona). Il **valore nullo**, in inglese *Null* (da non confondere con la stringa di caratteri blank o con un numero di valore zero) rappresenta un'informazione mancante, inapplicabile o sconosciuta.

3.2 Attributi composti o aggregati

Gli **attributi composti o aggregati** sono costituiti mediante aggregazione di altri attributi, che a loro volta possono essere semplici o frutto di altre aggregazioni di attributi.

Con **aggregato** facciamo riferimento alla definizione di aggregazione precedentemente utilizzata, cioè *attributi* che possono a loro volta considerarsi *astrazioni di aggregazione di attributi semplici*. La definizione è ricorsiva.

Abbiamo già visto che per l'entità *Persona* sono attributi semplici: *Nome*, *Cognome*, *Età*, *Sesso*; se consideriamo l'attributo *DataDiNascita*, quest'ultimo è sicuramente un attributo aggregato, in quanto è composto da altri attributi che sono: *Giorno*, *Mese*, *Anno*. In questo caso, quindi, *DataDiNascita* è un attributo aggregato dei tre attributi semplici: *Giorno*, *Mese*, *Anno*. Un altro esempio è quello dell'attributo *Indirizzo*, composto dagli attributi: *Via*, *NumCivico*, *Città*, *Cap*.

3.3 Attributi multipli

Gli **attributi multipli** sono un elenco di lunghezza variabile di attributi dello stesso tipo.

Se consideriamo l'entità *Persona* un esempio di attributo multiplo è *Hobby*.

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

Quando modelliamo una realtà di interesse per mezzo di entità e di attributi, abbiamo varie scelte possibili per decidere se uno stesso concetto possa essere modellato come entità o come attributo. Occorre scegliere:

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

Vediamo subito un esempio chiarificatore.

Esempio

Consideriamo 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* sarà una caratteristica dell'entità *Persona* e sarà di tipo stringa. Prenderà valori del tipo: "Milano", "Roma", "Pescara", "Palermo" ecc.

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 propri attributi:

entità *Comune*:

attributi *Nome, Provincia, Regione*.

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

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

3.5 Attributi chiave

Si indica con il termine **chiave** o **chiave candidata** l'insieme di uno o più attributi che consentono di distinguere un'istanza dall'altra in **modo univoco**.

Utilizziamo ancora come esempio l'entità *Persona*.

Esempio

Se consideriamo l'entità *Persona* per distinguere in *modo univoco* due persone, possiamo utilizzare:

- un attributo *CodiceFiscale*, poiché sappiamo che i valori di tale attributo sono diversi per ogni persona;
- un *CodiceProgressivo* che creiamo ogni volta che viene aggiunta una nuova persona, cioè una nuova istanza all'entità;
- gli attributi *Nome, Cognome* e *DataDiNascita*.

Tra le *chiavi candidate*, quella con il minor numero di attributi prende il nome di **chiave primaria** (*primary key*). Nell'esempio precedente può essere chiave primaria: *CodiceFiscale* o *CodiceProgressivo*.

Gli attributi chiave giocano un ruolo fondamentale in un diagramma ER, in particolare nel modello logico relazionale che vedremo di seguito.

3.6 Rappresentazione grafica di entità e attributi

Per rappresentare graficamente in un diagramma ER le entità e gli attributi si utilizza la seguente notazione:



◆ Fig. A2.7
La rappresentazione grafica di un'entità

dove:

- per le *entità* si utilizza un rettangolo, contenente all'interno il nome dell'entità;
- per gli *attributi non chiave* si utilizza una linea, che parte dal rettangolo dell'entità e termina con un piccolo cerchio e il nome dell'attributo;
- gli *attributi chiave* si rappresentano come gli attributi non chiave e in più si sottolinea il nome oppure si cambia il colore del cerchio.

Regole per la rappresentazione di entità e attributi

È possibile collocare gli attributi in qualsiasi posizione attorno al rettangolo dell'entità.

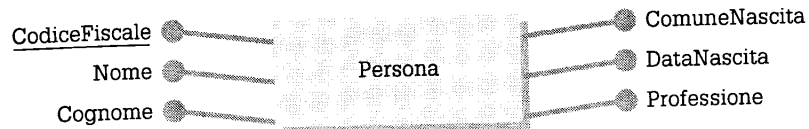
Convenzioni per la scrittura di entità e attributi

Per convenzione utilizziamo:

- *nomi al singolare* per indicare le entità;
- *iniziali maiuscole* per i nomi delle entità e degli attributi;
- *iniziali maiuscole* per i nomi di attributi composti da più parole (ad esempio *DataNascita*).

Una possibile rappresentazione dell'entità *Persona* è la seguente:

Fig. A2.8
La rappresentazione grafica dell'entità *Persona*



4 Le associazioni

Classi di oggetti e classi di fatti

L'**associazione** (in inglese *relationship*) è un legame esistente tra due o più entità. Così come le *entità* sono **classi di oggetti** del mondo reale, le *associazioni* sono **classi di fatti**.

Le **associazioni** sono classi di fatti del mondo reale che hanno proprietà omogenee ai fini della loro applicazione; tali fatti mettono in corrispondenza istanze di due o più entità.

Esempio

Tra l'entità *Persona* e l'entità *Automobile* esiste un'associazione che descrive nel linguaggio naturale il *fatto* secondo il quale: "una persona possiede una o più automobili" e viceversa "un'automobile è posseduta da una persona".

Versi dell'entità

Ogni associazione tra due entità ha due **versi**. Quindi si può dire che tra l'entità *Persona* e l'entità *Automobile* esiste l'associazione **possiede**; tra l'entità *Automobile* e l'entità *Persona* esiste l'associazione **possedutaDa**.

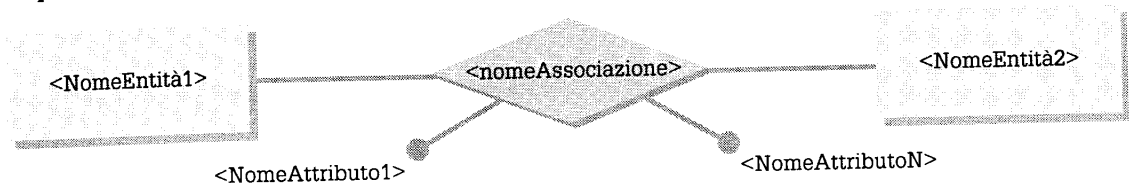
L'esempio mostra che solitamente i **sostantivi** del linguaggio naturale corrispondono alle *entità*, mentre i **verbi** corrispondono alle *associazioni*.

4.1 Rappresentazione grafica delle associazioni binarie

Associazione binaria

Per rappresentare graficamente in un diagramma ER un'associazione tra due entità, o **associazione binaria**, si utilizza la seguente notazione, dove per semplicità di esposizione non compaiono gli attributi delle due entità:

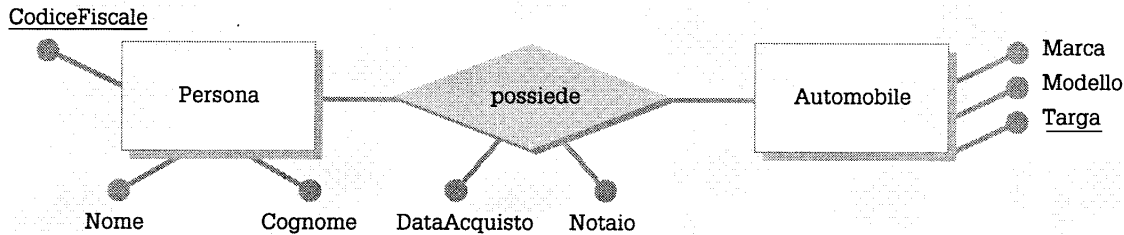
Fig. A2.9
La rappresentazione grafica delle associazioni



dove il *rombo* a metà della linea che unisce le due entità rappresenta la loro associazione. Per gli eventuali attributi dell'associazione si seguono le stesse regole degli attributi di un'entità. Per convenzione utilizzeremo *iniziali minuscole* per la descrizione dell'associazione.

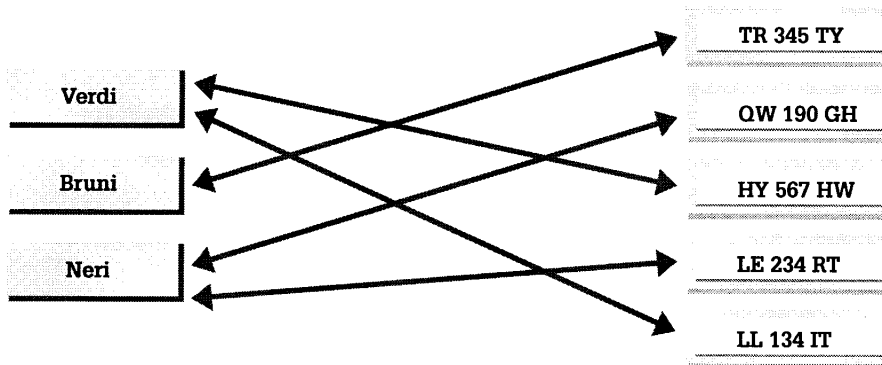
Esempio

Ad esempio, l'associazione *possiede* tra le due entità *Persona* e *Automobile* può essere rappresentata nel seguente modo:



♦ Fig. A2.10
Un esempio di associazione

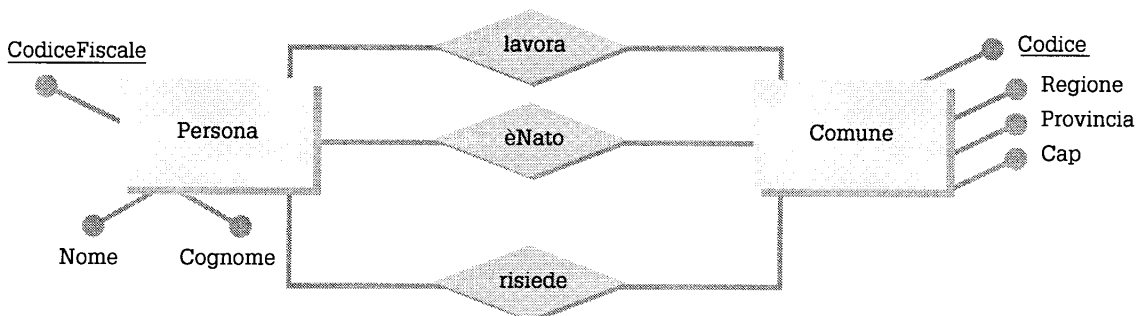
Come possiamo notare, gli attributi *DataAcquisto* e *Notaio* sono attributi che caratterizzano l'associazione e non le entità in relazione. Questo simbolismo rappresenta i possibili collegamenti tra istanze dell'entità *Persona* e istanze dell'entità *Automobile*, come nella figura seguente:



♦ Fig. A2.11
Associazione binaria tra due entità

4.2 Rappresentazione di più associazioni tra due entità

Tra due entità possiamo definire più associazioni, come nella figura seguente:



♦ Fig. A2.12
Due entità con più associazioni tra di loro

Le tre associazioni rappresentano rispettivamente i seguenti fatti:

- "il comune dove le persone *lavorano*";
- "il comune dove le persone sono *nate*";
- "il comune dove le persone *risiedono*".

4.3 Rappresentazione di associazioni multiple

Finora abbiamo considerato le associazioni binarie, che collegano cioè due entità.

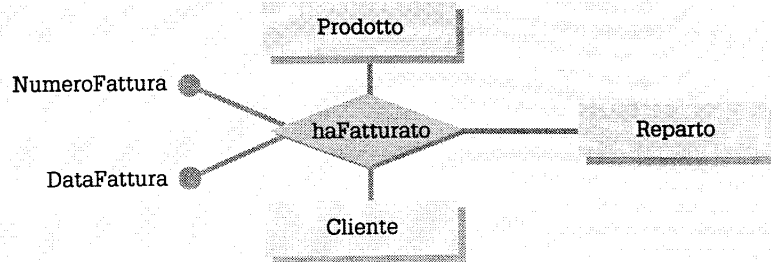
Vediamo come si possono rappresentare le **associazioni multiple**, ovvero quelle associazioni che collegano più di due entità.

Esempio

Consideriamo la *realtà* di un supermercato, in cui si vuole rappresentare il *fatto* che ogni reparto possa emettere fattura per i clienti.

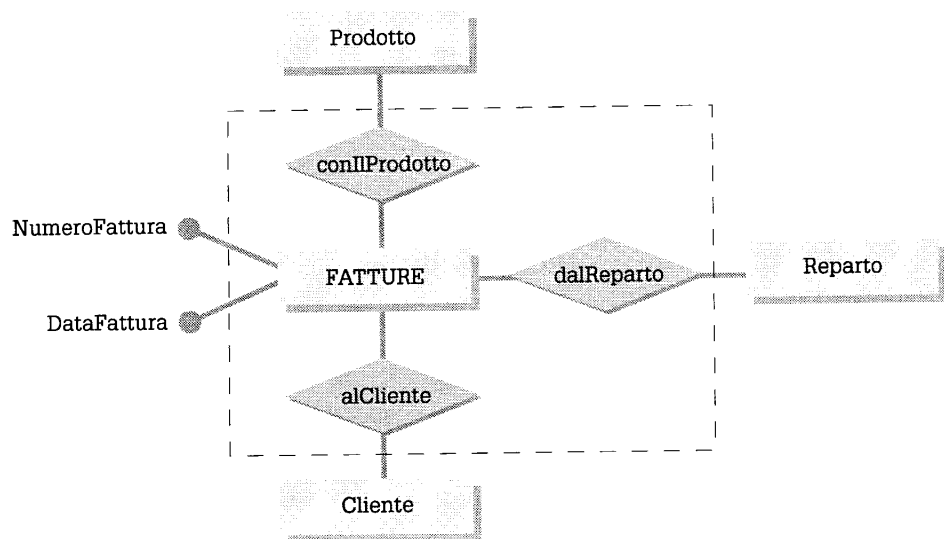
Di ogni reparto si vogliono conoscere alcuni dati riassuntivi e identificativi. Possiamo utilizzare l'associazione ternaria *haFatturato*.

◆ Fig. A2.13
Un'associazione ternaria



Le *associazioni multiple* possono essere trasformate in *associazioni binarie*. La precedente associazione *haFatturato* può essere trasformata nel seguente modo:

◆ Fig. A2.14
Come trasformare un'associazione ternaria in due binarie



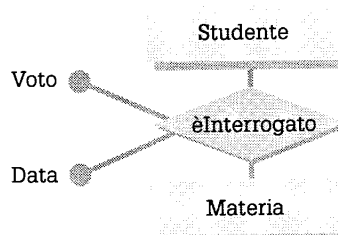
dove l'associazione *haFatturato* è stata trasformata in entità (FATTURE) e sono state aggiunte tre associazioni necessarie per garantire il collegamento con le tre entità presenti.

Non è quindi riduttivo considerare solo le associazioni binarie tra entità, poiché qualsiasi associazione multipla potrà sempre essere trasformata in *più associazioni binarie*.

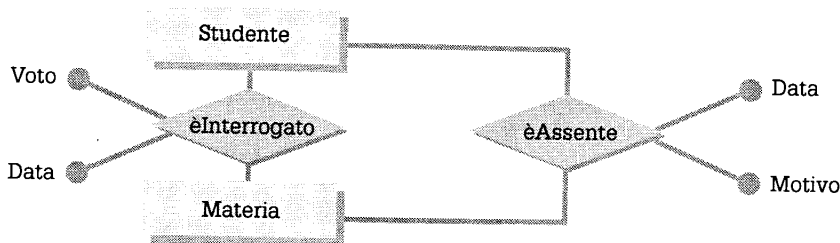
4.4 Contenuto informativo di un diagramma ER

Consideriamo la *realtà* di una *classe di alunni* in cui si vogliono rappresentare le *interrogazioni*. Possiamo utilizzare il seguente diagramma ER:

◆ Fig. A2.15
Un diagramma ER che fornisce informazioni sulle interrogazioni degli studenti

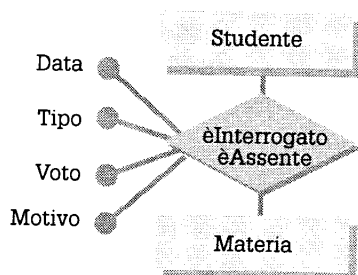


Se vogliamo avere informazioni sulle *assenze* possiamo aggiungere al precedente diagramma un'altra associazione ottenendo:



◆ Fig. A2.16
Diagramma ER
che fornisce
informazioni
su assenze e
interrogazioni con due
associazioni distinte

Un'altra possibile soluzione consiste nell'avere un'unica associazione così composta:



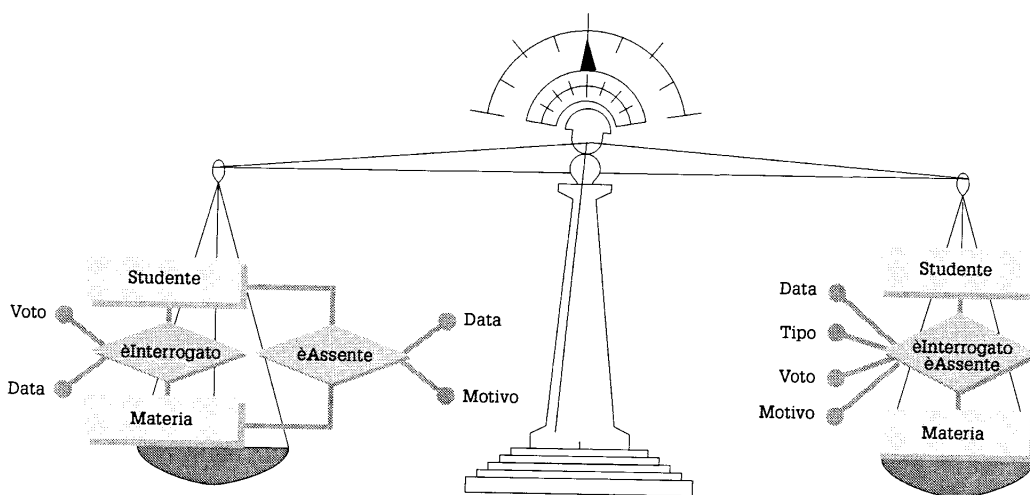
◆ Fig. A2.17
Diagramma ER
analogo a quello
precedente, ma
che opera con una
sola associazione

L'attributo *Tipo* indicherà se si tratta di assenza o di interrogazione. Se si trattasse di assenza, gli attributi *Voto* e *Motivo* rimarrebbero non valorizzati.

Come possiamo notare, data una realtà d'interesse, è possibile descriverla con diversi schemi del modello ER; l'importante è che abbiano lo stesso contenuto informativo.

Diremo che:

due diagrammi D1 e D2 hanno lo stesso **contenuto informativo** quando per ogni interrogazione che possiamo fare sul diagramma D1 esiste un'interrogazione che possiamo fare sul diagramma D2 che fornisce la stessa risposta. In questo caso i diagrammi si dicono **equivalenti**.



◆ Fig. A2.18
Equivalenza del
contenuto informativo
di due diagrammi ER

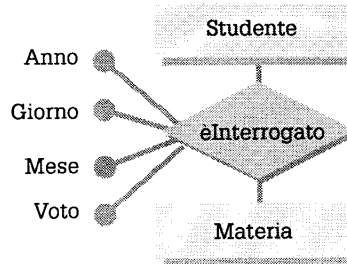
Analizziamo ora i due diagrammi posti sui piatti della bilancia. Dal punto di vista della leggibilità, della comprensibilità e della capacità di trasferirvi il contenuto informativo della realtà che modella, si può affermare che lo schema di sinistra sia più chiaro perché separa concetti che sono intrinsecamente diversi. Quello di destra è più sintetico e più compatto, ma con i concetti meno facilmente distinguibili e meno facilmente ritrovabili. Seguiremo da ora in poi il criterio generale che privilegia la **chiarezza**.

Comprensibilità e
chiarezza di un
diagramma

4.5 Attributo che passa dall'associazione all'entità

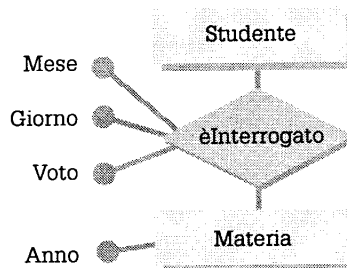
Consideriamo il seguente diagramma ER, in cui abbiamo tre attributi per specificare la data dell'interrogazione:

♦ Fig. A2.19
Diagramma ER con
l'attributo *Anno*
sull'associazione



Se ci stiamo riferendo alle interrogazioni relative a uno stesso anno, ovvero se *Anno* avesse un valore fisso, il diagramma precedente diventerebbe come il seguente:

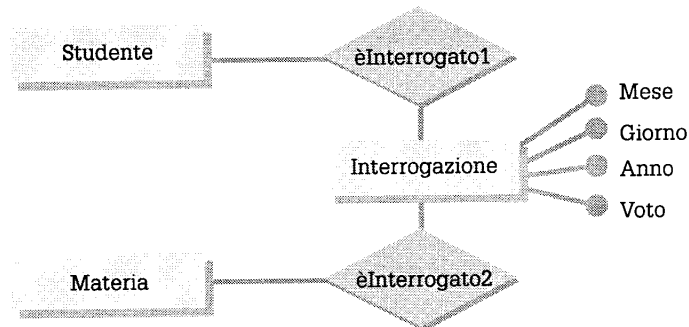
♦ Fig. A2.20
Diagramma ER con
l'attributo *Anno*
sull'entità



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

Per eliminare tutti gli attributi di un'associazione, introdurremo una nuova entità, che contiene tali attributi. Variando le *Figg. A2.19 e A2.20*, otterremo:

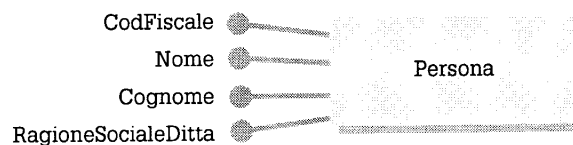
♦ Fig. A2.21
Creazione di un'entità
per eliminare attributi
dall'associazione



4.6 Attributo che diventa un'entità

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 lavora una persona.

♦ Fig. A2.22
Attributi dell'entità
Persona



Se siamo interessati a conoscere solo la ragione sociale della ditta presso cui è impiegata quella persona, allora l'attributo *RagioneSocialeDitta* è sufficiente.

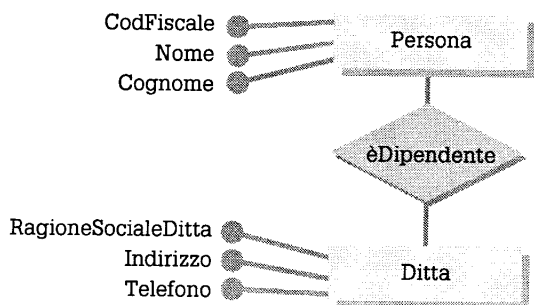
Ma se occorre conoscere anche altre informazioni relative a quella ditta, come, ad esempio, indirizzo, numero di dipendenti, numero di telefono, allora occorrerà aggiungere altri attributi all'entità *Persona*.

Questa soluzione, però, non è conveniente poiché i valori di tali attributi verreb-

bero duplicati in ogni altra istanza di *Persona*, che è dipendente della stessa ditta.

Inoltre una modifica al numero di telefono di una particolare ditta dovrebbe essere riportata in tutte le istanze dove è duplicato tale numero di telefono.

Per evitare questi inconvenienti si aggiunge un'altra entità che chiamiamo *Ditta* e la si mette in relazione con *Persona*, ottenendo il seguente diagramma ER:

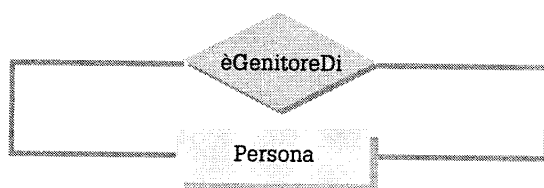


◆ Fig. A2.23
L'attributo
RagioneSocialeDitta
diviene entità

4.7 Rappresentazione di un'associazione sulla stessa entità: i ruoli

Quando un'associazione crea un legame tra due entità identiche, si ha il caso particolare di un'associazione sulla stessa entità.

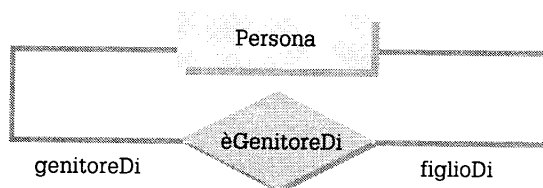
Ad esempio, consideriamo l'entità *Persona* e supponiamo di voler rappresentare l'associazione *èGenitoreDi*. È questo un "classico" esempio di associazione sulla stessa entità, che si rappresenta nel seguente modo:



◆ Fig. A2.24
Un esempio di
associazione su una
stessa entità

Poiché però l'associazione è una *classe di fatti* che collegano due o più entità, 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 quale è il ramo che rappresenta il genitore e quale è il ramo che rappresenta il figlio.

Possiamo allora scrivere:



◆ Fig. A2.25
Associazione con ruoli

Dove *genitoreDi* e *figlioDi* sono anche chiamati **ruolo** 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.

Nell'ambito dei fatti che rappresentano le istanze dell'associazione sappiamo sempre qual è il genitore e qual è il figlio, ovvero qual è l'istanza di entità che rappresenta il genitore e quale è l'istanza di entità che rappresenta il figlio. Il concetto di ruolo, se necessario, poiché ha una funzione chiarificatrice, può essere esteso a qualsiasi associazione binaria, come nella seguente figura:



◆ Fig. A2.26
Ancora
un'associazione
con ruoli

4.8 Associazioni dirette e inverse

L'associazione *guida* (Fig. A2.26) ha quindi due **versi**: uno da *Persona* ad *Automobile* (relativo all'associazione che possiamo chiamare *diretta*) e l'altro da *Automobile* a *Persona* (relativo all'associazione che chiameremo *inversa*).

In modo più generale, possiamo dire che:

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 .

Da sottolineare 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* quindi del *verso* relativo all'associazione diretta e di quello relativo all'associazione inversa è puramente arbitrario. Possiamo indicare l'associazione diretta e inversa con i nomi dei ruoli.

4.9 Associazioni totali e parziali

Un'associazione A tra le entità X e Y può essere **totale** quando il legame tra le entità *deve* essere sempre presente (cioè a ogni elemento di X deve corrispondere almeno un elemento di Y), altrimenti si dice **parziale**.

È quindi *parziale* quando il legame tra le entità *può* non essere presente, cioè può esistere un elemento di X a cui non corrisponda alcun elemento di Y o, viceversa, un elemento di Y a cui non corrisponda alcun elemento di X . Per rappresentare un'associazione parziale utilizzeremo una **linea tratteggiata**.

Può succedere che l'associazione *diretta* sia *totale* mentre l'*inversa* è *parziale* o viceversa (la diretta è parziale mentre l'inversa è totale). Quest'ultimo caso è rappresentato nel seguente esempio:

Rappresentazione di un'associazione parziale

♦ Fig. A2.27 Associazioni totale e parziale



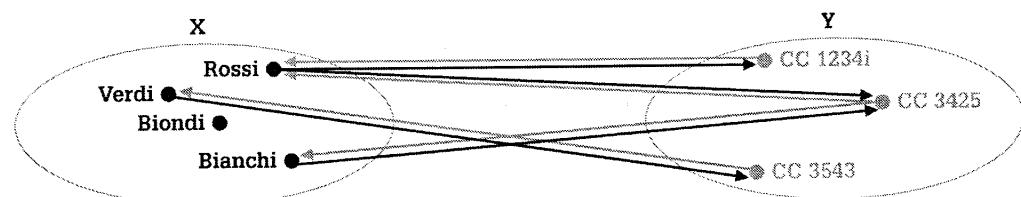
In tale associazione notiamo:

- l'associazione diretta *titolareDi*, rappresentata con *linea tratteggiata*, indica la *parzialità*, nel senso che non tutte le persone possiedono un conto corrente;
- l'associazione inversa *intestatoA*, rappresentata con *linea continua*, indica la *totalità*, nel senso che ogni conto corrente deve essere intestato a una persona.

Esempio

Il signor Biondi non possiede un *ContoCorrente* come è mostrato dalla seguente figura, in cui l'associazione diretta è rappresentata con le frecce di colore azzurro, mentre l'inversa con frecce di colore grigio. Come si può notare, il signor Biondi di X non ha alcun legame con un elemento di Y ; pertanto l'associazione diretta è *parziale*. Invece, tutti gli elementi di Y hanno un legame con qualche elemento di X . Pertanto l'associazione inversa è *totale*. Si noti che il conto CC 3425 è un conto cointestato.

♦ Fig. A2.28 Un esempio di associazione diretta parziale e associazione inversa totale



4.10 Associazioni univoche e biunivoche

Un'associazione *A* tra le entità *X* e *Y* può essere:

- **univoca**, se a ogni elemento di *X* corrisponde al più un elemento di *Y*, altrimenti si dice **multipla**;
- **biunivoca** o **surgettiva** se è univoca e la sua associazione inversa è totale, altrimenti si dice **non-biunivoca** o **non-surgettiva**.

4.11 Tipi di associazioni

Un'associazione *binaria* tra due entità *A* e *B* può essere di tipo:

- 1) **uno a uno (1:1)**, quando a un'istanza di *A* corrisponde una e una sola istanza di *B* e viceversa, o anche quando la *diretta* e l'*inversa* sono *univoche*.

Rappresentiamo l'associazione nel seguente modo:



Associazione binaria "uno a uno"

♦ Fig. A2.29 Rappresentazione dell'associazione 1:1

Esempio

Un'associazione *uno a uno* è quella che esiste tra l'entità *DirigenteScolastico* e l'entità *Scuola*. Un dirigente scolastico infatti può dirigere una sola scuola. Viceversa una scuola può essere diretta da un solo dirigente scolastico.



♦ Fig. A2.30 Un esempio di associazione 1:1

Se si utilizza un verbo come descrizione dell'associazione, come nel diagramma precedente, la **regola di lettura** è:

Regola di lettura

per l'associazione *diretta*: "un dirigente scolastico dirige una scuola";



per l'associazione *inversa*: "una scuola è diretta da un dirigente scolastico".

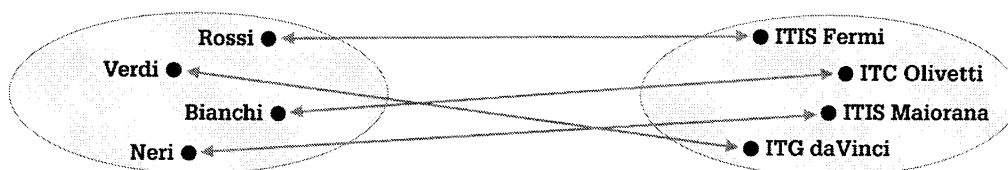


Se si esplicitano i ruoli, le regole di lettura diventano più evidenti, come mostra il seguente diagramma.



♦ Fig. A2.31 Esempio di un'associazione 1:1 con ruoli sulla diretta e sull'inversa

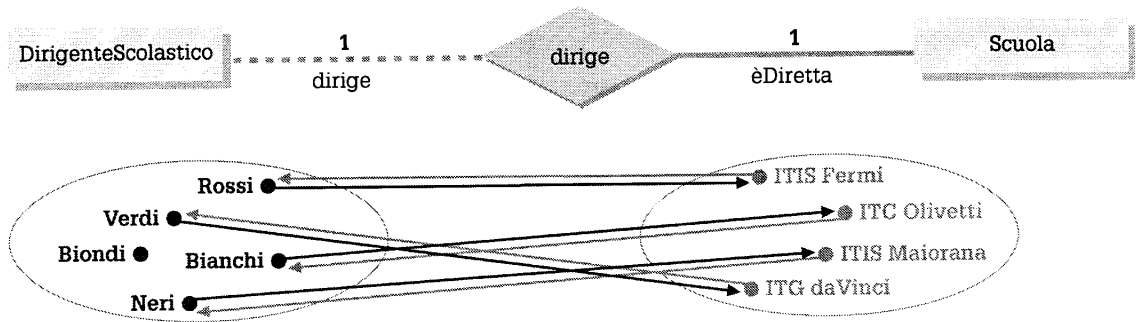
Nella figura seguente si evidenzia il legame esistente, in un'associazione uno a uno, tra le istanze delle due entità. Le associazioni *diretta* e *inversa* sono *totali* (e anche *univoche*), come si evince sia dalla linea continua del diagramma precedente sia dalla seguente figura.



♦ Fig. A2.32 Un esempio di associazione diretta e associazione inversa univoche (e totali)

L'associazione precedente potrebbe avere la *diretta* (per semplicità, da ora in poi utilizzeremo a volte questo termine non preceduto da *associazione*, come anche nel caso di *inversa*) parziale, nel caso vengano considerati i presidi in pensione. Avremmo così:

◆ Fig. A2.33
Un esempio di associazione con diretta e inversa univoche e diretta parziale



2) **uno a molti (1:N) o semplice**, quando a un'istanza di *A* possono corrispondere una o più istanze di *B* e a ogni istanza di *B* deve corrispondere una sola istanza di *A* (la scelta di *A* o di *B* come entità di partenza è casuale). Possiamo anche dire che un'associazione binaria è *uno a molti* se è multipla la diretta o l'inversa ma non entrambe. Si rappresenta nel seguente modo:

◆ Fig. A2.34
Rappresentazione di un'associazione 1:N



Esempio

Un'associazione *uno a molti* è quella che esiste tra una scuola e il suo personale di segreteria. Una scuola, infatti, ha più persone che lavorano in segreteria, mentre il personale di segreteria lavora in una sola scuola. Mostriamo il relativo diagramma in cui sono stati evidenziati i ruoli.

◆ Fig. A2.35
Un esempio di associazione 1:N



Regola di lettura

Se si utilizza un *verbo* come descrizione dell'associazione, come nella Fig. A2.35 precedente, la **regola di lettura** è:

per l'associazione *diretta*: **“una scuola ha in organico più personale di segreteria”**;



per l'associazione *inversa*: **“una persona di segreteria lavora in una/una sola scuola”**.



Se fossero stati indicati i ruoli, le regole di lettura sarebbero più evidenti. Nella figura seguente si evidenzia il legame esistente, in un'associazione uno a uno, tra le istanze delle due entità. In questo esempio, come possiamo notare sia dal diagramma precedente che dalla figura seguente, sia la diretta sia l'inversa sono totali.

◆ Fig. A2.36
Un esempio di associazione in cui la diretta è multipla (diretta e inversa sono totali)



3) **molti a molti (N:N) o complessa**, quando a ogni istanza di *A* possono corrispondere una o più istanze di *B* e viceversa a ogni istanza di *B* possono corri-

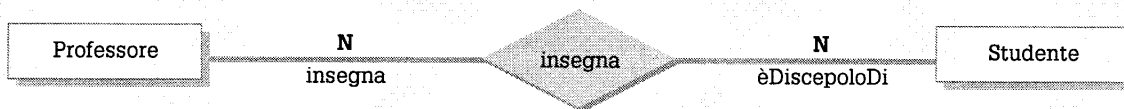
spondere una o più istanze di A. Possiamo anche dire che un'associazione è *molti a molti* se sono multiple sia la diretta che l'inversa. Si rappresenta nel seguente modo:



◆ Fig. A2.37
Rappresentazione dell'associazione N:N

Esempio

Un'associazione *molti a molti* è quella che esiste tra professori e studenti. Un professore infatti può insegnare a più studenti e viceversa uno studente ha più professori.



◆ Fig. A2.38
Un esempio di associazione N:N

Se si utilizza un verbo come descrizione dell'associazione, come nel diagramma precedente, la **regola di lettura** è:

Regola di lettura

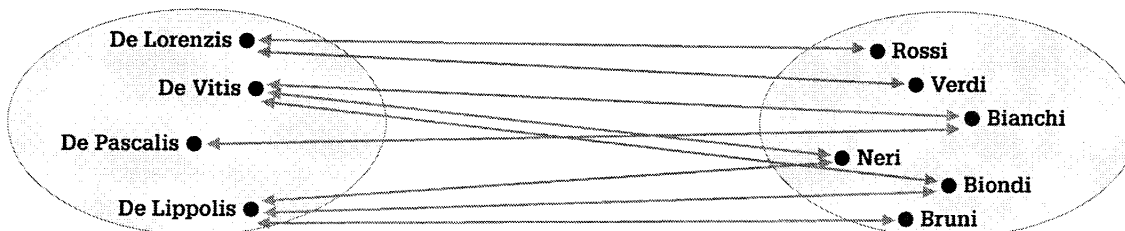
per l'associazione *diretta*: “**un** professore insegna a **più/molti** studenti”;



per l'associazione *inversa*: “**uno** studente è seguito da **più/molti** professori”.



Nella figura seguente si evidenzia il legame esistente, in un'associazione uno a uno, tra le istanze delle due entità.



◆ Fig. A2.39
Rappresentazione insiemistica di un'associazione N:N

4.12 L'importanza delle regole di lettura

Le regole di lettura del diagramma servono per effettuare un **controllo sulla bontà della modellazione effettuata**.

Se infatti una terza persona riesce a comprendere ciò che si sta modellando con il diagramma (traducendolo in italiano attraverso le regole di lettura delle associazioni), vorrà dire che è stato eseguito un buon lavoro di analisi della realtà di interesse.

4.13 Rappresentazione dell'astrazione per generalizzazione: gerarchie ISA

Un'associazione per generalizzazione nel modello ER è esattamente un'astrazione per generalizzazione così come l'abbiamo definita in precedenza.

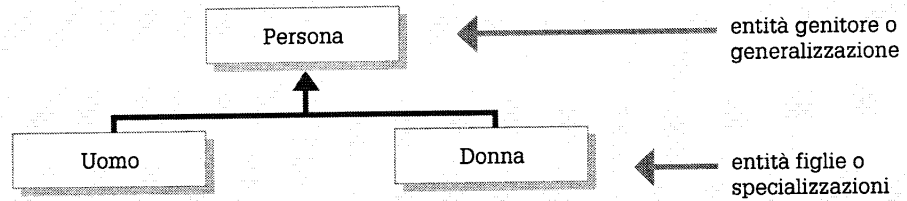
L'entità astrazione è detta anche entità *padre* (o *genitore*), le entità “inferiori” sono dette entità *figlie* o *specializzazioni*.

L'associazione per generalizzazione è anche chiamata associazione **ISA** (dall'inglese “is a” cioè “è un”).

Associazione ISA

Rappresentiamo la generalizzazione *Persona* tramite la seguente associazione:

◆ Fig. A2.40
Un esempio di generalizzazione



Diremo che *Persona* è generalizzazione di *Uomo* e *Donna*.

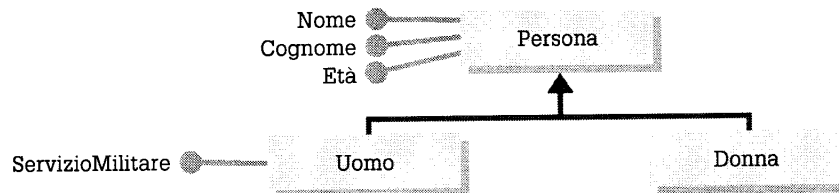
Ricordiamo che nell'astrazione per generalizzazione le classi *Uomo* e *Donna* sono sottoinsiemi della classe *Persona*. Come conseguenza della scelta della generalizzazione, dobbiamo associare alle entità figlie attributi che non compaiono tra gli attributi della classe genitore. Introduciamo allora una definizione molto importante per le generalizzazioni: quella di **ereditarietà**. In una generalizzazione, ogni proprietà dell'entità padre è anche proprietà delle entità figlie. Per proprietà intendiamo:

Ereditarietà

- gli attributi;
- le associazioni;
- le generalizzazioni cui partecipa l'entità.

Tale proprietà è chiamata **ereditarietà**. Vediamo alcuni esempi di tale proprietà:

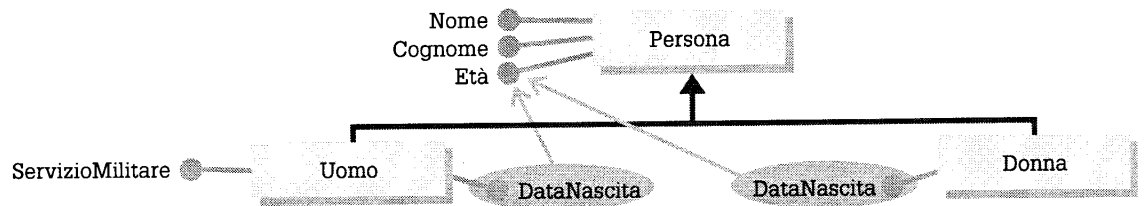
◆ Fig. A2.41
Attributi ereditati



L'attributo *Età* è definito nell'entità genitore *Persona* e viene ereditato dalle entità figlie *Uomo* e *Donna* (infatti un uomo e una donna hanno un'età).

Tale proprietà vale anche in senso opposto: se un attributo *DataNascita* è definito sia nell'entità *Uomo* sia nell'entità *Donna*, deve essere considerato un attributo dell'entità "superiore" (*Persona* nel nostro esempio).

◆ Fig. A2.42
Attributi delle entità figlie che migrano nell'entità padre



L'ereditarietà è un strumento molto importante perché ci consente di non duplicare l'informazione. Inoltre:

si parla di **generalizzazione totale** quando ogni occorrenza dell'entità padre è un'occorrenza di almeno una delle entità figlie (altrimenti è **parziale**).

Si parla di **generalizzazione esclusiva** se ogni occorrenza dell'entità padre è al massimo un'occorrenza di una delle entità figlie (altrimenti è **sovrapposta**).

Ad esempio, la generalizzazione tra *Persona*, *Uomo* e *Donna* è:

- *totale* poiché l'uomo e la donna costituiscono tutte le persone possibili;
- *esclusiva* poiché una persona o è uomo o è donna.

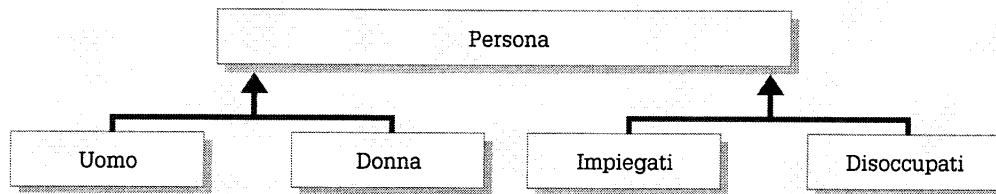
4.13.1 Rappresentazione della generalizzazione multipla

Una stessa entità può essere generalizzazione di diversi insiemi di entità tra loro

disgiunti. Si parla in questo caso di **generalizzazione multipla**.

Esempio

L'entità *Persona* è generalizzazione multipla da una parte delle entità figlie *Uomo* e *Donna* e dall'altra parte delle entità figlie *Impiegati*, *Disoccupati*.



◆ Fig. A2.43
Un esempio di generalizzazione multipla

4.14 Rappresentazione dell'astrazione per aggregazione

Possiamo dire che nel modello ER abbiamo a disposizione due distinti strumenti concettuali per rappresentare **aggregazioni implicite** tra concetti:

- l'entità che è un'aggregazione dei suoi *attributi*;
- l'associazione che è un'aggregazione di entità tra cui è definita.

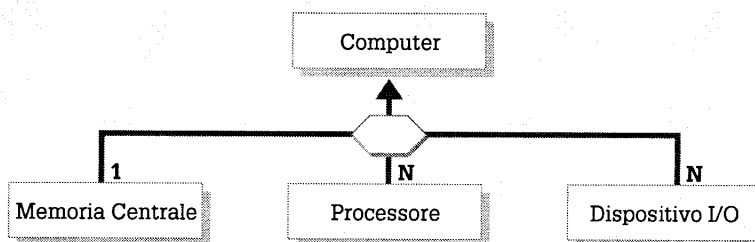
Un'associazione per aggregazione esplicita, nel modello ER, è esattamente un'astrazione per aggregazione così come l'abbiamo definita in precedenza.

L'associazione per aggregazione è anche chiamata associazione **HASA** o **ha un**. Le entità componenti contribuiscono a formare l'entità **composizione** o **contenitore**.

Associazione
"has a" o "ha un"

Esempio

Rappresentiamo l'aggregazione *Computer* tramite la seguente associazione:



◆ Fig. A2.44
Un esempio di aggregazione

Notiamo che è possibile indicare il **numero di elementi** delle entità componenti che concorrono a formare un elemento dell'entità composta.

Un'aggregazione generalmente viene classificata in **lasca** e **stretta**.

Un'aggregazione **lasca** indica l'indipendenza del ciclo di vita dell'elemento dell'entità componente dal rispettivo elemento dell'entità contenitore. Quindi, l'elemento "contenuto" potrà esistere indipendentemente dall'elemento "contenitore": ha cioè una vita propria. Possiamo dire che a un elemento di un'entità contenuta può non corrispondere alcun elemento dell'entità contenitore.

Aggregazione lasca

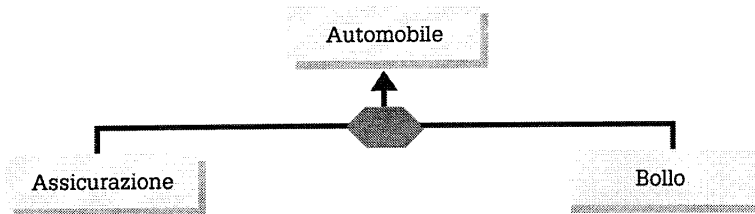
Graficamente un'aggregazione lasca si rappresenta con un esagono chiaro non vuoto come rappresentato nell'esempio precedente.

Un'aggregazione **stretta** indica che l'oggetto "contenuto" non ha una vita propria, ma esiste unicamente perché è stato creato l'oggetto "contenitore".

Aggregazione stretta

Graficamente si rappresenta con un esagono pieno, come mostrato nell'esempio di seguito, in cui si mette in relazione un'automobile al relativo bollo e assicurazione. Il bollo per un'auto non ha più senso di esistere senza l'auto su cui agisce. Quindi possiamo dire che a ogni elemento dell'entità contenuta deve corrispondere almeno un elemento dell'entità contenitore.

♦ Fig. A2.45
Un esempio di
aggregazione stretta



Aggregazione
totale

Un'aggregazione è **totale** se per ogni occorrenza dell'entità contenitore è possibile trovare occorrenze delle entità contenute.

Possiamo riassumere i simboli utilizzati nei diagrammi ER nel seguente schema:

♦ Fig. A2.46
Simboli utilizzati
nel diagramma ER

Descrizione	Simbolo
Entità	<NomeEntità>
Attributo semplice	<NomeAttributo>
Attributo chiave	<NomeAttributo>
Associazioni	<Ruolo> <nomeRelazione> <Ruolo> <Grado> <Grado>
Parzialità di un'associazione
Associazione per generalizzazione o ISA	
Associazione per aggregazione:	
lasca	
stretta	

5 I vincoli di integrità di un diagramma ER

Nell'unità precedente abbiamo detto che una base di dati definisce un insieme di regole che devono essere soddisfatte da tutte le istanze delle categorie.

Le *regole* a livello concettuale possono essere definite in un diagramma ER, precisando i vincoli di integrità.

Un **vincolo di integrità** è un'asserzione, ovvero un predicato che deve essere soddisfatto dalle istanze delle categorie.

I vincoli di integrità possono essere di due tipi:

- **vincoli impliciti**, che sono quelli imposti dalla stessa struttura dei dati e si dividono in:

- **vincoli di chiave primaria**. Questi vincoli 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*. Può anche essere visto come un **vincolo sulla totalità** delle associazioni.

- **vincoli espliciti**, che sono quelli che occorre esplicitare con apposite dichiarazioni. Un esempio di vincolo esplicito è: "il valore dell'attributo *Età* non può essere negativo né maggiore di 120".

Si chiamano vincoli di integrità perché devono essere rispettati, pena l'integrità stessa dei dati. Se il valore dell'attributo *Età* assume valori superiori a 120 può causare problemi a programmi che lo utilizzano e che suppongono che sia un valore "legittimo".

5.1 Rappresentazione dei vincoli di integrità

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 totalità di un'associazione).

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

V<NumProg>: (<Espressione>)

dove:

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

Esempio

Per esprimere il vincolo sull'*Età* scriveremo:

V1: (0<Età<120)

Per esprimere il vincolo che "non si può assumere un dipendente di età inferiore a 16 anni", scriveremo:

V2: (anno(Dipendente.DataAssunzione) – anno(Dipendente.DataNascita)>16)

Come possiamo notare, per riferirci a un attributo di una particolare entità o associazione, facciamo riferimento alla seguente sintassi:

<NomeEntità>. <NomeAttributo>

oppure

<NomeAssociazione>. <NomeAttributo>

6 Un esempio: una concessionaria di automobili

Problema n° 1

Si vuole automatizzare la gestione delle attività di una concessionaria di automobili multimarca. Nel database vanno memorizzate le informazioni che consentono di:

- 1) registrare le immatricolazioni di nuove automobili;
- 2) registrare le informazioni che riguardano le riparazioni;
- 3) elencare le automobili usate caratterizzate da un prezzo inferiore a un valore fornito in input;
- 4) elencare le riparazioni da effettuare per ogni auto usata;
- 5) elencare le auto nuove o usate di ogni marca presente in concessionaria;
- 6) elencare gli optional presenti su ogni automobile.

Analisi del
problema

Dalla lettura del testo si evince che la realtà di interesse da analizzare fa riferimento solo ad alcune delle funzioni svolte da una concessionaria. Sarà necessario raccogliere tutte le informazioni attinenti alle automobili: ad esempio quelle relative ai principali componenti quali il motore (cilindrata, numero di cilindri, potenza in kW ecc.), il tipo e il numero di telaio, le caratteristiche delle ruote (dimensione pneumatici, marca ecc.) nonché quelle relative agli optional che possono essere montati. Memorizzeremo inoltre anche le informazioni anagrafiche dei proprietari, supponendo che ogni automobile possa averne uno solo. Per la riparazione delle automobili creeremo una lista di riparazioni, individuando per ognuna di esse il livello di gravità. La lista sarà accompagnata anche da un preventivo di spesa. Dovendo trattare auto sia nuove che usate, per ottenere una memorizzazione efficiente delle informazioni relative alle automobili, scegliamo di utilizzare un'associazione per generalizzazione, attraverso la quale memorizzare nelle entità padre le informazioni comuni a tutte le automobili e nelle entità figlie solo quelle specifiche alle auto nuove e a quelle usate.

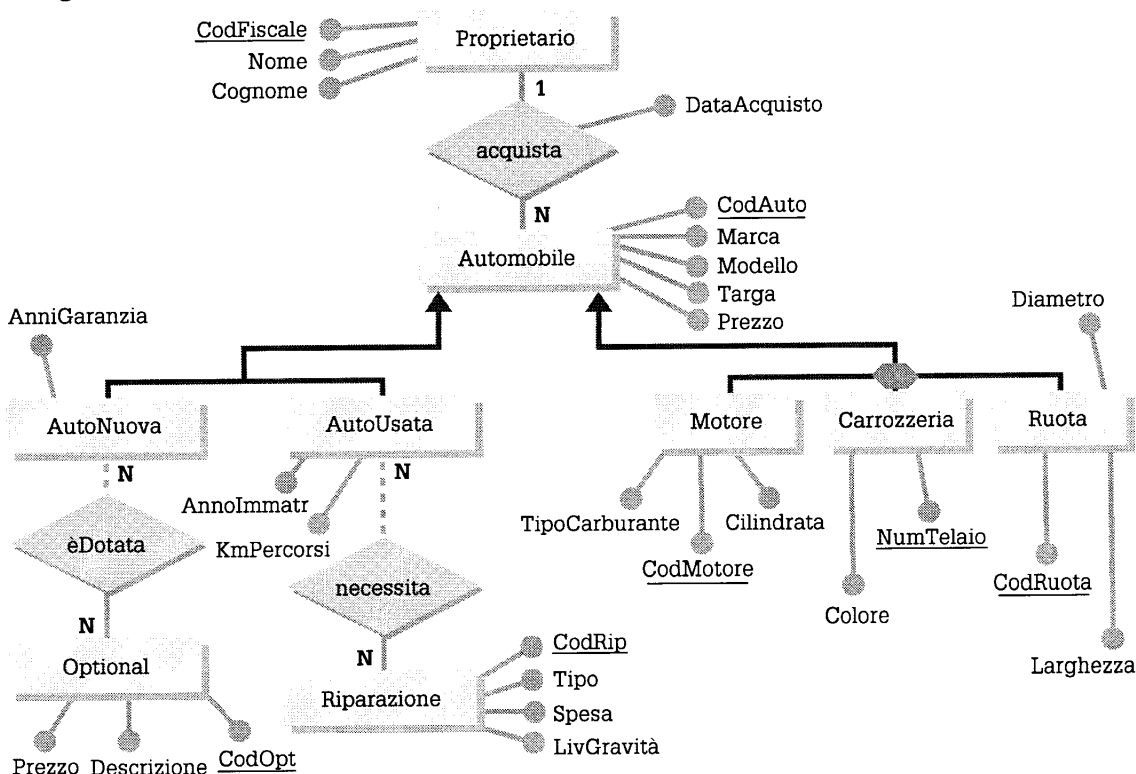
Nella realizzazione della base di dati ipotizzeremo i seguenti vincoli:

- non devono esistere riparazioni che abbiano una spesa inferiore a 1000 euro per una gravità uguale o superiore a livello 7;
- non saranno vendute auto usate immatricolate prima del 1990 o che abbiano percorso più di 300000 chilometri.

Progettazione
concettuale

◆ Fig. A2.47
Diagramma ER della
concessionaria di
automobili

Diagramma ER



Vincoli di integrità

V1:(Automobile.Prezzo > 0)

V2:(AutoUsata.AnnoImmatr > 01/01/1990)

V3:(AutoUsata.KmPercorsi < 300000)

Per esprimere il fatto che: “non devono esistere riparazioni che abbiano una spesa inferiore a 1000 euro per una gravità uguale o superiore a livello 7”, così come risulta dall’analisi delle specifiche della realtà d’interesse, scriveremo il seguente vincolo.

V4:(SE(Riparazione.LivGravità ≥ 7) ALLORA Riparazione.Spesa > 1000)

Verifica modello con Regole di lettura:

Vediamo qualche esempio di regole di lettura per la verifica della bontà del diagramma:

Associazione *necessita*:

diretta, “**un**’auto usata necessita di **nessuna** o **più** riparazioni”

inversa, “**una** riparazione è eseguita su **una** o **più** auto usate”.

Associazione *èDotata*:

diretta, “**un**’auto nuova è dotata di **nessuno** o **più** optional”

inversa, “**un** optional è installato su **una** o **più** auto nuove”.

Notiamo inoltre che l’associazione *necessita* ha la diretta **parziale**: infatti un’auto potrebbe non necessitare di riparazioni. L’inversa è totale in quanto una riparazione se presente vuol dire che è associata a un’auto usata.

7 La rappresentazione di informazioni non alfanumeriche: immagini e suoni

Spesso accade che nella realtà di interesse si debba considerare l’esigenza di rappresentare informazioni che non siano solo alfanumeriche. È frequentemente richiesta infatti la rappresentazione di informazioni multimediali quali suoni e immagini, come immagini relative ad articoli in vendita, immagini di quadri, foto di persone, suoni relativi a registrazioni di suonerie di telefoni cellulari ecc.

Con il modello ER non si hanno meccanismi in grado di rappresentare direttamente negli attributi informazioni non alfanumeriche. I tipi dei valori dei domini nel modello ER che stiamo considerando sono primitivi: *Interi*, *Stringhe*, *Reali*.

Per rappresentare quindi informazioni diverse dai tipi primitivi occorre far riferimento ai **file** che contengono tali informazioni.

Il problema allora si sposta a come **individuare il file** contenente informazioni non alfanumeriche.

La **visualizzazione** di tali file sarà un problema da risolvere a livello di linguaggio di programmazione e non a livello concettuale.

Si suppone allora che:

- i file siano **memorizzati** in directory di cui occorre individuare il percorso completo;
- i file abbiano un’**estensione** che ne specifica il formato;
- i **nomi** dei file siano autoesplicativi del loro contenuto, cioè leggendo il nome del file deve essere facile capire cosa esso contiene.

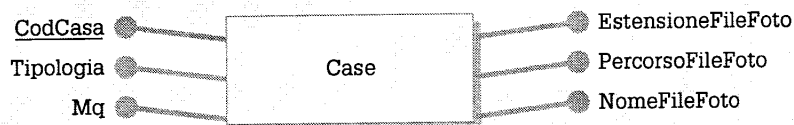
Occorre allora introdurre nuovi attributi per implementare le precedenti ipotesi:

- un attributo che specifichi il **percorso** completo del file;
- un attributo che specifichi l’**estensione** ovvero il formato del file;
- un attributo che specifichi il **nome** del file.

Per il nome del file generalmente si utilizza la **chiave** dell'entità, ove possibile e applicabile; altrimenti occorre creare un **codice** che sia in relazione **uno a uno** con la chiave dell'entità.

Esempio

Supponiamo di dover rappresentare tramite diagramma ER un'entità che contenga le abitazioni in vendita presso un'agenzia immobiliare. Oltre alle classiche informazioni quali: metri quadrati, prezzo, indirizzo, si vuole conservare anche una foto per ogni abitazione. Tra le varie soluzioni, consideriamo la seguente:



In particolare il *NomeFileFoto* avrà lo stesso valore di *CodCasa*, supponendo che *CodCasa* sia un numero progressivo.

Ad esempio:

<i>PercorsoFileFoto</i>	c:\database\immagini
<i>EstensioneFileFoto</i>	TIFF
<i>NomeFileFoto</i>	12347



Conoscenze

■ VERO o FALSO?

- | | V | F |
|--------------------------------------------------------------------------------------------------|--------------------------|--------------------------|
| 1. Un'entità può avere più attributi. | <input type="checkbox"/> | <input type="checkbox"/> |
| 2. Una chiave di interpretazione consente l'accesso a un dato. | <input type="checkbox"/> | <input type="checkbox"/> |
| 3. L'intensione o schema è la chiave di interpretazione di un dato. | <input type="checkbox"/> | <input type="checkbox"/> |
| 4. Un'istanza è uno dei possibili valori di uno schema. | <input type="checkbox"/> | <input type="checkbox"/> |
| 5. Un'associazione 1:1 contiene solo una coppia di valori. | <input type="checkbox"/> | <input type="checkbox"/> |
| 6. Un'associazione 1:N può avere propri attributi. | <input type="checkbox"/> | <input type="checkbox"/> |
| 7. L'associazione ISA avviene tra un'entità detta specializzazione e una detta generalizzazione. | <input type="checkbox"/> | <input type="checkbox"/> |
| 8. In un'associazione ISA la specializzazione non ha tutti gli attributi della generalizzazione. | <input type="checkbox"/> | <input type="checkbox"/> |
| 9. In un'associazione ISA la generalizzazione ha tutti gli attributi della specializzazione. | <input type="checkbox"/> | <input type="checkbox"/> |
| 10. È possibile avere la diretta totale e l'inversa parziale. | <input type="checkbox"/> | <input type="checkbox"/> |
| 11. È possibile avere la diretta parziale e l'inversa parziale. | <input type="checkbox"/> | <input type="checkbox"/> |

14. Quali delle seguenti associazioni possono essere rappresentate in un diagramma ER?
 - Associazioni per generalizzazione;
 - associazioni ISA;
 - associazioni per accrescimento;
 - associazioni per aggregazione;
 - associazioni HASA.
15. Quando una generalizzazione è parziale e quando è totale?
16. Cosa è una generalizzazione multipla?
17. Che differenza c'è tra aggregazione lasca e stretta?
18. Come si rappresentano i vincoli impliciti e come quelli espliciti?
19. Come si rappresentano i vincoli di integrità referenziale?

■ ESERCIZI

1. Fare un esempio di:
 - attributo;
 - attributo chiave;
 - istanza;
 per le seguenti entità:
 - Bicicletta;
 - Motocicletta;
 - Impiegato;
 - Medico;
 - Animale;
 - Fiore;
 - Televisore.

2. Quali dei seguenti attributi sono semplici e quali aggregati?
 - DataNascita;
 - LuogoNascita;
 - CodiceFiscale;
 - NumeroCivico;
 - Nome;
 - RagioneSociale;
 - Proprietario;
 - Studente.
3. Fai un esempio di associazione multipla tra più entità.
4. Fai un esempio di più associazioni tra due entità.
5. Fai un esempio di associazione tra una stessa entità.
6. Fai un esempio in cui è necessario l'utilizzo dei ruoli.
7. Fai un esempio di associazione 1:1 con diretta parziale.
8. Fai un esempio di associazione 1:1 totale.
9. Fai un esempio di associazione 1:N:
 - con diretta parziale e inversa totale;
 - con diretta e inversa parziale;
 - con diretta e inversa totale.
10. Fai un esempio di associazione N:N:

■ QUESITI

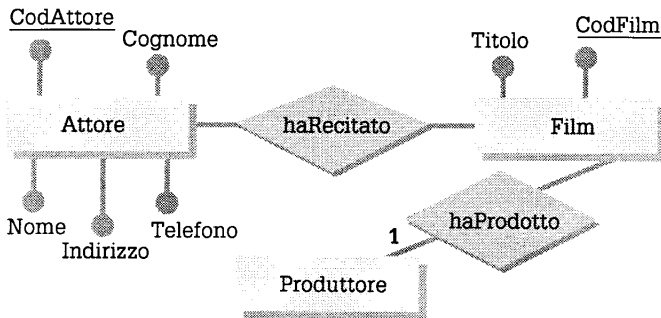
1. Che differenza c'è tra chiavi candidate e chiavi primarie?
2. Come si rappresenta un attributo chiave primaria in un diagramma ER?
3. Che differenza c'è tra associazioni multiple e associazioni binarie?
4. È possibile trasformare un'associazione multipla in associazioni binarie?
5. Perché si parla di equivalenza del contenuto informativo di due diagrammi ER?
6. Quando un attributo viene trasformato in un'entità?
7. Cosa indicano la diretta e l'inversa di un'associazione?
8. Quando un'associazione è totale e quando è parziale?
9. Quando un'associazione si dice univoca e quando biunivoca?
10. Come si definisce un'associazione di tipo 1:N?
11. Come si definisce un'associazione di tipo N:N?
12. Che differenza c'è tra aggregazioni e generalizzazioni?
13. Cosa indicano le associazioni ISA e cosa le associazioni HASA?

- con diretta parziale e inversa totale;
con diretta e inversa parziale;
con diretta e inversa totale.
11. Trasforma un'associazione N:N in più associazioni 1:N.
 12. Fai un esempio di generalizzazione.
 13. Fai un esempio di aggregazione lasca.
 14. Fai un esempio di aggregazione stretta.
 15. Fai un esempio di vincolo implicito.
 16. Fai un esempio di vincolo esplicito.
 17. Per i seguenti esercizi analizza la realtà di interesse proposta e descrivila sia in forma discorsiva sia utilizzando un diagramma ER che riporti:
 - le entità e le associazioni;
 - gli attributi di ogni entità e di ogni associazione;
 - gli attributi chiave di ogni entità;
 - le associazioni totali e quelle parziali;
 - il grado di ogni associazione;
 - i ruoli per le associazioni (se necessari);
 - i vincoli di integrità;
 - un glossario contenente:
 - per ogni entità:
 - descrizione del suo significato;
 - descrizione e dominio di ogni attributo (quando necessario); se l'attributo è derivato, la relativa regola di computazione;
 - per ogni associazione:
 - descrizione del suo significato;
 - descrizione e dominio di ogni attributo;
 - ruolo della partecipazione di ogni entità coinvolta nelle associazioni, mettendo in evidenza il significato del ruolo (motivando, se necessario, la scelta del grado).

Il diagramma deve essere infine verificato con le regole di lettura.
 18. Rappresenta i dati e le relazioni tra dati necessari alla stesura di una fattura commerciale che comprenda gli articoli venduti, il cliente e la data di vendita.
 19. Rappresenta i dati e le relazioni tra dati necessari a gestire un sistema di prenotazione di un teatro di 1000 posti suddivisi in 25 file di 40 poltrone l'una. Ogni poltrona è quindi individuata da una lettera e da un numero.
 20. Rappresenta i dati e le relazioni tra dati relativi alle partite dell'ultimo campionato di calcio. In particolare memorizza, per poi poterle rintracciare, le seguenti informazioni:
 - il numero di goal segnati dai giocatori;
 - il numero di goal segnati "in trasferta";
 - il numero di goal segnati "in casa";
 - le date relative alle partite;
 - i numeri di telefono dei giocatori;
 - il nome del presidente della squadra di calcio alla quale appartiene un giocatore.
 21. Modifica il diagramma ER relativo alla concessionaria di automobili visto nel paragrafo 6 di questa unità, per poter gestire anche l'officina meccanica. Tale officina deve poter:

- ricevere le richieste relative a una riparazione di un'automobile, classificate in base al tipo di guasto;
 - comunicare il preventivo di spesa, la data di consegna e la data di ritiro della vettura.
- Organizza il tutto in modo che la base di dati possa effettuare le seguenti query:
- Q1: visualizza le auto che devono essere consegnate in una determinata data;
 - Q2: visualizza le auto che hanno avuto un determinato tipo di guasto;
 - Q3: visualizza il meccanico responsabile della riparazione di quell'auto.
22. Rappresenta i dati e le relazioni tra dati relativi alla gestione di un autonoleggio.
 23. Rappresenta i dati e le relazioni tra dati relativi alla gestione dell'orario ferroviario.
 24. Rappresenta i dati e le relazioni tra dati relativi alla gestione dei reparti di un'ospedale.
Suggerimento: un ospedale è composto da diversi reparti. Ogni reparto ha un nome, una specialità, un primario, un numero di posti letto numerati. Ogni paziente ha un nome, una data di arrivo, un insieme di diagnosi, una tariffa giornaliera e occupa un certo posto letto in un reparto. I dottori dell'ospedale sono caratterizzati da un nome, una o più specialità; ciascun dottore ha sotto cura diversi pazienti, situati anche in reparti diversi. A ogni paziente un dottore può prescrivere, di tanto in tanto, una serie di esami clinici i cui risultati devono essere registrati (cartella clinica del paziente). Prevedi le procedure per l'arrivo e la partenza di un paziente, elenca tutti gli esami prescritti da un dottore per un certo paziente, elenca tutti i medici che visitano un certo reparto. Individua il numero di giorni di permanenza di un paziente in un certo reparto.
 25. Rappresenta i dati e le relazioni tra dati relativi a un sistema di rilevazione automatica delle presenze installato in una scuola.
 26. Rappresenta i dati e le relazioni tra dati relativi a un sistema di noleggio di videocassette e DVD, con le seguenti specifiche:
 - per ogni film sono rilevanti un codice identificativo, il titolo, il regista (supposto come unico), l'anno di produzione, gli interpreti (zero o più) e il costo del noleggio; possono esistere film senza attori e film il cui regista sia anche attore;
 - vengono anche memorizzate informazioni riguardo ad attori e registi: data e luogo di nascita, cittadinanza attuale ed eventuali cittadinanze passate, con data di variazione;
 - per ogni film possono esistere più copie;
 - per ogni cassetta sono rilevanti la data di fabbricazione e lo stato di conservazione;
 - per ogni DVD sono registrati i GB memorizzati;
 - per ogni noleggio sono rilevanti il cliente, la data di noleggio e la data di restituzione; non viene tenuta traccia dei noleggi conclusi;
 - per ogni cliente vengono memorizzati il nome, l'indirizzo e il telefono.

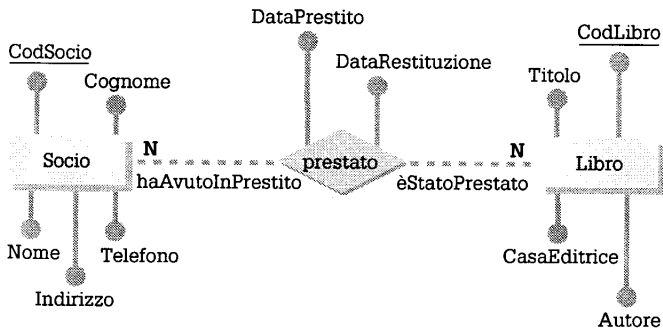
27. Modifica il diagramma dell'esercizio precedente affinché si tenga traccia anche dei noleggi conclusi e si consideri che ogni film può essere registrato su una o più videocassette.
28. Completa il seguente diagramma, che consente di memorizzare i dati e le relazioni tra dati relativi agli attori e produttori di film cinematografici. Scegli almeno due contesti diversi per stabilire gli altri attributi e associazioni.



29. Completa il diagramma precedente aggiungendo attributi e associazioni in modo da poter conoscere:
- il genere di un film;
 - il tempo di durata di un film;
 - il budget del produttore di un film;
 - il numero di comparse di un film;
 - i dati anagrafici delle comparse di un film.

ESEMPI SVOLTI

1. Rappresenta tramite diagramma ER la realtà di interesse: prestiti di libri ai soci di una biblioteca. Diagramma ER:



Verifica del diagramma

Regole di lettura:

- "un socio ha avuto in prestito nessuno o più libri";
- "un libro è stato prestato a nessuno o a più soci".

GLOSSARIO

Entità:	Socio, Libro
Attributi di Socio:	CodSocio, Cognome, Nome, Indirizzo, Telefono
Chiave:	CodSocio
Attributi di Libro:	CodLibro, Titolo, Autore, CasaEditrice
Chiave:	CodLibro
Associazione:	prestatato

Tipo:	N:N
Attributi:	DataPrestito, DataRestituzione
Diretta:	da Socio a Libro, parziale poiché un socio può non aver avuto in prestito alcun libro
Ruolo:	haAvutoInPrestito
Inversa :	da Libro a Socio, parziale poiché un libro può non essere stato mai prestato ad alcun socio
Ruolo:	èStatoPrestato

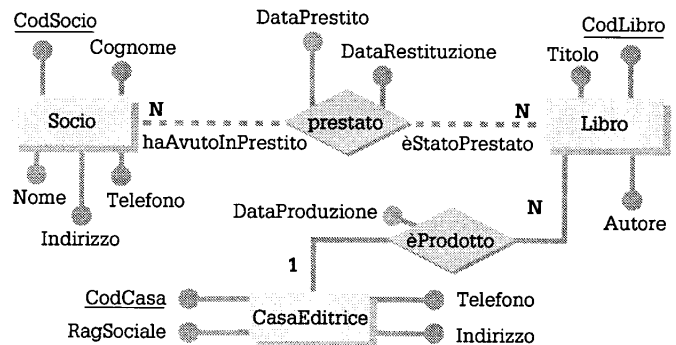
Vincoli di integrità:

- V1:(Prestato.DataPrestito < Prestato.DataRestituzione)
 V2:(Prestato.DataPrestito ≠ 0)

2. Modifica l'esempio precedente, aggiungendo le informazioni anagrafiche relative alle case editrici dei libri.

Realtà di interesse: prestiti di libri ai soci di una biblioteca, comprese le informazioni anagrafiche relative alla case editrici dei libri.

Diagramma ER:



Verifica del diagramma

Regole di lettura:

- "un socio ha avuto in prestito nessuno o più libri";
- "un libro è stato prestato a nessuno o a più soci".

Al glossario precedente occorre aggiungere:

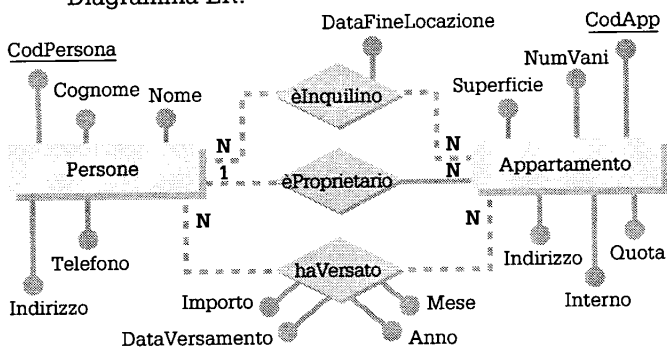
GLOSSARIO

Entità:	CasaEditrice
Attributi di Casa Editrice:	CodCasa, RagSociale, Indirizzo, Telefono
Chiave:	CodCasa
Associazione:	produce
Tipo:	1:N
Attributi:	DataProduzione
Diretta:	da CasaEditrice a Libro, totale poiché una casa editrice ha sicuramente prodotto qualche libro
Ruolo:	haProdotto (non compare sul diagramma)
Inversa:	da Libro a CasaEditrice, totale poiché ogni libro ha una casa editrice che lo ha prodotto
Ruolo:	èStatoProdotto (non compare sul diagramma)

Esempio di Vincoli di integrità:

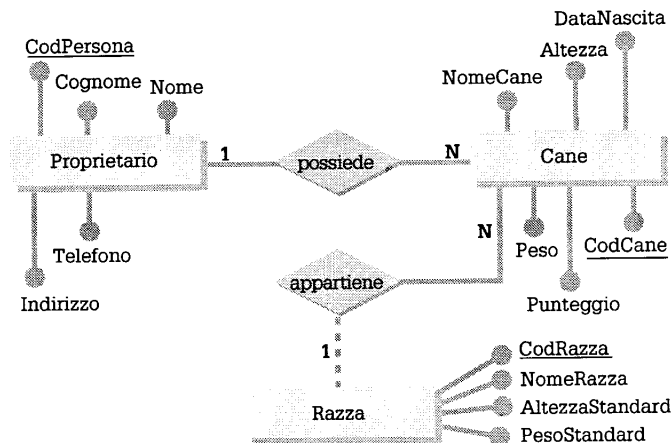
- V3:(Produce.DataProduzione < Prestato.DataPrestito)

3. Rappresenta tramite diagramma ER la seguente realtà di interesse: Amministrazione di un condominio. In essa:
- ogni appartamento è caratterizzato da una superficie espressa in metri quadrati, un numero di vani, un indirizzo e un interno per distinguere appartamenti nello stesso palazzo;
 - ogni appartamento ha un proprietario ed eventualmente uno o più inquilini;
 - ogni proprietario versa entro la fine del mese la quota condominiale, stabilita per ogni appartamento, all'amministratore.
- Deve essere possibile:
- Q1: inserire i pagamenti relativi a un determinato mese;
 - Q1: elencare le quote da versare relative agli appartamenti di un determinato proprietario;
 - Q2: elencare i proprietari che hanno versato la quota relativa a un determinato mese (e anno).
- Diagramma ER:



4. Rappresenta tramite diagramma la realtà di interesse: rappresentare i dati e le relazioni tra dati relativi alla gestione di una mostra canina. In essa:
- i cani devono appartenere a una razza canina caratterizzata dagli standard di altezza e peso;

- ogni cane ha un numero di gara e un punteggio attribuitogli dal giudice di gara;
 - vince il cane che ottiene il maggior punteggio tra quelli della propria razza.
- Deve essere possibile:
- Q1: assegnare un punteggio da 10 a 100 a un determinato cane individuato da un numero progressivo;
 - Q1: stilare una classifica dei cani all'interno di ogni razza;
 - Q2: stilare una classifica dal maggiore al minor punteggio indipendentemente dalla razza del cane.
- Diagramma ER:



Esempi di Vincoli di integrità:

V1:(Cane.Altezza < 110)

V2:(Cane.Peso < 150)

Vincolo interrelazionale, che esprime possibili variazioni del peso e dell'altezza di un cane rispetto agli standard di razza:

V4:(Razza.Peso - 10 ≤ Cane.Peso ≤ Raza.Peso + 10)

V5:(Razza.Altezza - 5 ≤ Cane.Altezza ≤ Raza.Altezza + 5)

Competenze

■ ESERCIZI

- Indica quale dei seguenti esempi sono astrazioni di generalizzazione e quale di aggregazione.
 - Animali: uccelli, pesci, leoni;
 - Animali: mammiferi, ovipari.
- Fai un esempio in cui si veda la differenza tra schema e istanza.
- Fai un esempio di astrazione di classificazione.
- Fai un esempio di astrazione di aggregazione.
- Fai un esempio di astrazione di generalizzazione.
- Rappresenta le seguenti classi come aggregazioni di altre classi:
 - mese;
 - anno;

- data;
 - indirizzo.
- Considera un aspetto della tua casa (ad esempio le stanze, la biblioteca personale, le persone, le pietanze, i vestiti) e arriva alla definizione del corrispondente concetto per mezzo di procedimenti di classificazione, di aggregazione, di generalizzazione, a partire da altri aspetti. Ad esempio: il frigorifero può essere definito come classificazione, aggregazione, generalizzazione.
 - Trovati i concetti nel precedente esercizio e partendo da questi, trova nuovi concetti più o meno generali utilizzando l'astrazione di generalizzazione.
 - Trova i concetti che risultino aggregazioni di cui il concetto originario fa parte, sempre a partire dall'esercizio 7.