

## 14. LA PROGETTAZIONE CONCETTUALE: IL DIAGRAMMA ER

La **progettazione concettuale** consiste nel riorganizzare tutti gli elementi che si hanno a disposizione dopo la fase di analisi della realtà di interesse (specifiche formali), per definire un **modello astratto o concettuale** della base di dati (**ossia indipendente dallo sviluppo successivo**).

Tale **modello concettuale** è destinato a diventare un documento ufficiale di riferimento e di comunicazione per i progettisti della successiva fase di **progettazione logica**.

Il documento concettuale in molti casi costituisce il tramite di verifica tra **committenti e progettisti**.

Si ribadisce che i termini **astratto e concettuale** si riferiscono al fatto che si evita il più possibile in tale fase di descrivere dettagli realizzativi. In particolare ci si astrae dal tipo di sistema e dal linguaggio che verranno utilizzati nella realizzazione. Si cerca invece di attribuire una “*certa struttura*” ai dati come vedremo dettagliatamente più avanti.

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

### LE ASTRAZIONI

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

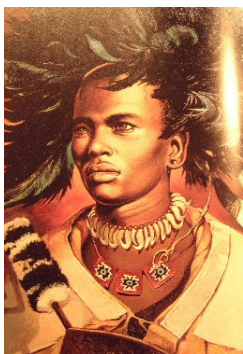
Nella progettazione di basi di dati vengono utilizzati principalmente **tre procedimenti di astrazione**:

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

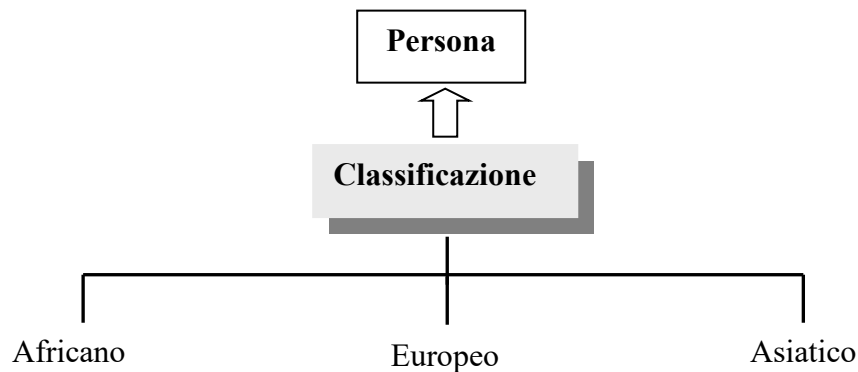
L'**astrazione per classificazione** permette di definire una classe a partire da un insieme di oggetti di cui si individuano le proprietà comuni.

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

*Esempio: analizziamo le seguenti immagini*



Se individuiamo le caratteristiche comuni ai soggetti esaminati (respirano, hanno una testa, due braccia e due gambe, parlano, camminano, etc.) e scartiamo quelle che li differenziano (lingua parlata, religione professata, colore della pelle, etc.) arriviamo a definire la classe “**Persona**”  
Possiamo rappresentare la classificazione con il seguente schema:

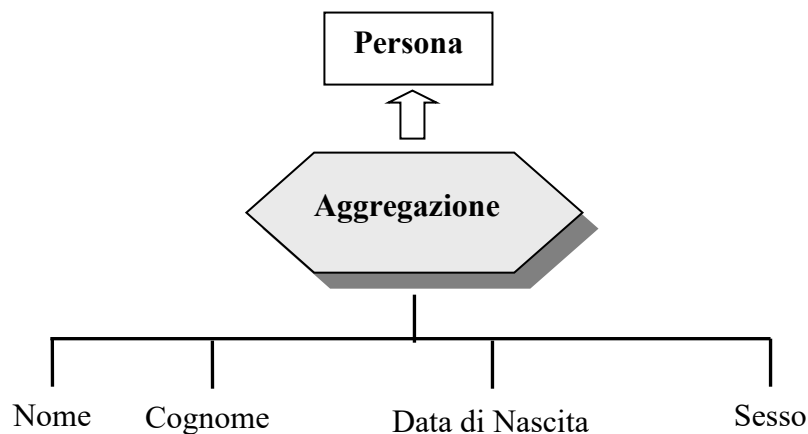


L'**astrazione per aggregazione** è il procedimento mentale che usiamo quando a partire da una o più classi generiamo una nuova classe. Le classi di partenza vengono dette **classi componenti o proprietà** in quanto la loro unione contribuisce a realizzare la classe di arrivo.

Proseguiamo con il nostro esempio.

Consideriamo le proprietà **Nome, Cognome, Data di Nascita, Sesso** delle persone e cerchiamo di individuare la classe in cui nome, cognome, data di nascita, sesso sono caratteristiche o parti componenti: è naturale arrivare all'individuazione della classe “**Persona**”.

Possiamo rappresentare l'aggregazione con il seguente schema:



**N.B.** L'*aggregazione* è un procedimento mentale diverso rispetto a quello utilizzato per la classificazione. Nell'*aggregazione* si parte da un insieme di classi per ottenerne un'altra, mentre nella *classificazione* si parte da un insieme di oggetti per ottenere una classe.

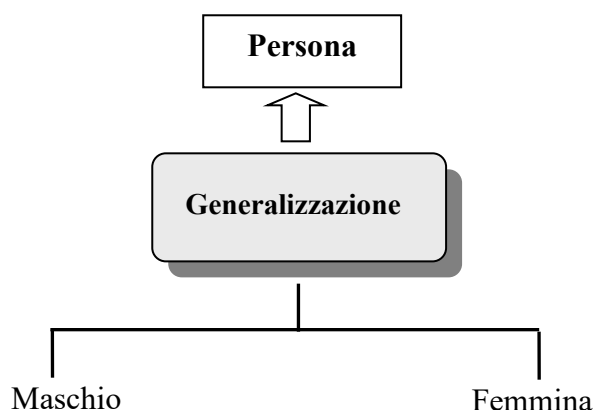
L'**astrazione per generalizzazione** è il procedimento mentale mediante il quale si può giungere alla definizione di una classe come **unione** di un insieme di classi ognuna delle quali è **inclusa** nella classe da definire.

**N.B.** Il legame che unisce le classi di partenza e la classe di generalizzazione è un legame di **sottoinsieme** ossia di inclusione ed è quindi un legame diverso rispetto al legame di aggregazione.

*Proseguiamo con il nostro esempio.*

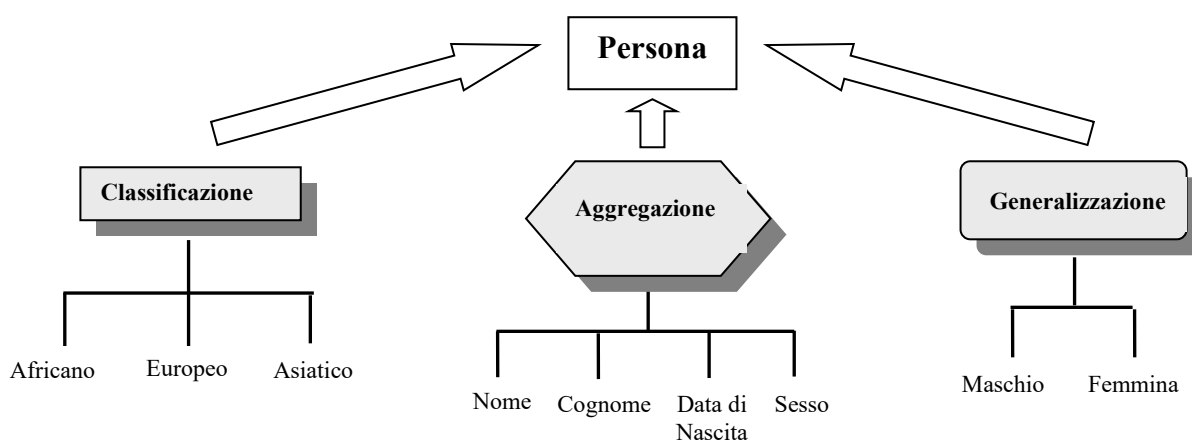
*Consideriamo le seguenti classi di partenza: Maschi e Femmine. Cerchiamo ora una classe che è l'unione delle due classi di partenza. Naturalmente ci viene in mente la classe "**Persona**" che è l'unione di queste due classi.*

*Possiamo rappresentare la generalizzazione con il seguente schema:*



Riassumendo secondo quanto abbiamo visto nei nostri esempi è possibile giungere alla definizione della stessa classe "Persona" utilizzando tre processi mentali completamente diversi.

Inoltre le tre astrazioni (processi mentali) sono **indipendenti una dall'altra** ossia non è possibile dar luogo ad una generalizzazione utilizzando una aggregazione o una classificazione e viceversa.



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.

## IL DIAGRAMMA ER

Il **diagramma Entità/Associazioni** (in *inglese* **E/R** da **Entità/Relationship**), introdotto nel 1976 da **Peter Chen**, è un **modello concettuale di tipo grafico** per la descrizione dei dati e delle loro interazioni presenti in una certa realtà di interesse.

E' uno strumento per l'analisi delle caratteristiche dei dati di una realtà indipendentemente dagli eventi che in essa accadono.

E' un modello concettuale dei dati *indipendente dalle applicazioni*.

Gli elementi di base di un **diagramma E/R** (che d'ora in poi indicheremo con la scrittura **ER**) sono:

- le *entità*;
- gli *attributi*;
- le *associazioni*;
- i *vincoli di integrità*;
- le *associazioni per generalizzazione (o ISA)*;
- le *associazioni per aggregazione (o HASA)*.

### ENTITA', ISTANZE di ENTITA' e suoi ATTRIBUTI

In un diagramma ER le **entità** corrispondono a **classi di oggetti** del mondo reale.

Esse sono definite specificando le proprietà che possiedono ossia gli **attributi** dei loro elementi.

Gli elementi di una entità vengono chiamate **istanze o occorrenze dell'entità**.

**N.B. 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 solo profumi allora è ovvio considerare "Profumo" come entità. Una entità può essere considerata tale soltanto in relazione al contesto nel quale la si considera.*

*Se invece consideriamo come realtà di interesse un supermercato, l'entità da considerare sarà "Articolo" e non più "Profumo".*

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

*Esempio:*

*Se consideriamo l'entità "Persona" nel contesto di un ufficio anagrafico i suoi possibili attributi sono: Nome, Cognome, Data di Nascita, Indirizzo, Città, Cap (definita **anagrafica breve**) ma anche Sesso, Email, Cellulare, etc.*

*Se consideriamo l'entità "BottigliaDiVino" nel contesto di un'enoteca i suoi possibili attributi sono: Denominazione, Produttore, Capacità, Prezzo, Vitigno.*

*Quest'ultimo esempio è interessante in quanto se di una certa bottiglia di vino sono disponibili nell'enoteca diverse istanze (cosa abbastanza frequente) allora con BottigliaDiVino possiamo intendere due concetti diversi: la singola bottiglia fisica oppure l'insieme delle bottiglie che hanno tutte la stessa caratteristica.*

*Ovviamente se intendessimo distinguere la singola bottiglia di vino all'interno della medesima caratteristica, dovremmo introdurre un altro attributo (ad esempio un progressivo o un codice) che ci permetta di distinguere una particolare bottiglia da tutte le altre istanze della medesima entità.*

Ogni **attributo** è specificato da:

- un **nome**;
- un **formato** che indica il tipo di valori che può assumere (es. Stringa, Booleano, Intero, Reale, Data, etc);
- una **dimensione** (non vale per i tipi Data e 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;
- una **opzionalità**, che indica la possibilità di non essere sempre valorizzato. Diremo che un attributo è **obbligatorio** se deve essere valorizzato con un valore non nullo (diverso da NULL), **opzionale (o facoltativo)** quando può essere valorizzato con valore nullo (uguale a NULL).  
*N.B. Il valore nullo in inglese NULL non va confuso con la stringa di caratteri blank o con un numero 0 bensì rappresenta un'informazione mancante, inapplicabile o sconosciuta.*

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

*N.B. Con **aggregato** ci riferiamo alla definizione di aggregazione precedentemente utilizzata ossia attributi che possono a loro volta considerarsi astrazioni di aggregazione di attributi semplici.*

*(Esempio: l'attributo Indirizzo come aggregazione di Via, NCivico, Cap, Città)*

Gli **attributi multipli** sono quegli attributi costituiti da una lista o elenco di lunghezza variabile di valori dello stesso tipo

*(Esempio: l'attributo **Hobby** dell'entità **Persona** oppure l'attributo **Genere** dell'entità **Libro**)*

**Quando modelliamo attraverso un diagramma ER una realtà di interesse per mezzo di entità e di attributi dobbiamo decidere se modellare un particolare concetto come entità o come attributo.**

Occorre scegliere:

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

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

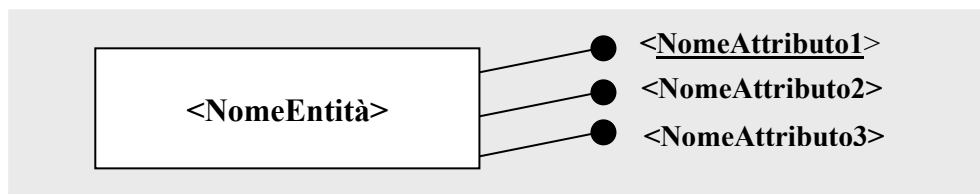
Se la risposta è *affermativa* creiamo una *entità*, *altrimenti* creiamo un *attributo*.

Si indica con il termine **chiave o chiave candidata** l'*insieme di uno o più attributi i cui valori* consentono di **distinguere** un'istanza da un'altra di una certa entità **in modo univoco**.

Tra le *chiavi candidate* quella con minor numero di attributi prende il nome di **chiave primaria o primary key** (sinteticamente indicata con **PK**)

**N.B.** Gli **attributi chiave** giocano un ruolo fondamentale in un **diagramma ER** in particolare nel *modello logico relazionale* che vedremo in seguito.

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



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 ed il nome dell'attributo;
- per gli **attributi chiave** si utilizza la stessa rappresentazione utilizzata per quelli *non chiave* avendo l'accortezza di sottolineare il nome oppure di cambiare il colore del piccolo cerchio;

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

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 delle *entità* o degli *attributi* composti da più parole

Per riferirci ad un attributo di una particolare entità possiamo fare riferimento alla seguente pseudoistruzione:

<NomeEntità>.<NomeAttributo>

## LE ASSOCIAZIONI (tra due o più entità)

L'**associazione** (in inglese *relationship*) è un legame esistente tra le istanze di due o più entità.

Così come le *entità* sono *classi di oggetti* del mondo reale, le **associazioni** sono **classi di interazioni o legami** tra gli oggetti del mondo reale.

Le **associazioni** sono **classi di interazioni o legami** tra gli oggetti del mondo reale eventualmente caratterizzate da proprietà specifiche; tali interazioni mettono in corrispondenza istanze di due (associazioni binarie) o più entità (associazioni ennarie).

Ogni associazione tra due entità (quindi binaria) ha due **versi** o **direzioni**: diretto e inverso.

Per **istanza di una associazione** si intende la combinazione di istanze delle entità che ne prendono parte (coppie ordinate).

**N.B. Solitamente i “sostantivi” del linguaggio naturale corrispondono alle entità, mentre i “verbi” corrispondono alle associazioni.**

*Esempio: Ipotizziamo che tra l'entità “Persona” e l'entità “Automobile” esista una associazione binaria di nome “Possiede” che descrive nel linguaggio naturale il legame secondo il quale “una persona possiede una o più automobili” e viceversa “un'automobile è posseduta da una persona”.*

*Quindi si può dire che tra l'entità “Persona” e l'entità “Automobile” esiste l'associazione binaria “**Possiede**” in cui*

- il verso diretto o associazione diretta è rappresentato anch'esso dall'azione “Possiede”
- il verso inverso o associazione inversa è rappresentato invece dall'azione “E'PossedutaDa”

*N.B. Decidiamo di denotare una qualsiasi associazione utilizzando SEMPRE (SE POSSIBILE) la forma ATTIVA del verbo (alla terza persona singolare).*

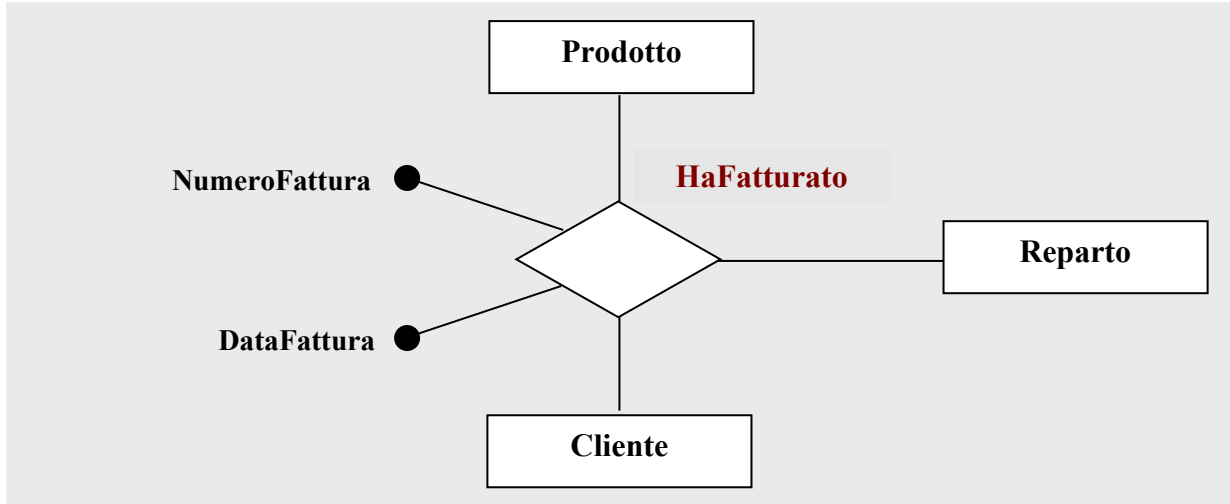
*L'associazione diretta di conseguenza sarà denotata con la medesima forma verbale (alla terza persona singolare) mentre l'associazione inversa sarà denotata SEMPRE (SE POSSIBILE) utilizzando la forma PASSIVA del verbo (alla terza persona singolare).*

*Ove non fosse possibile rispettare questa scelta notazionale si è obbligati IN OGNI CASO a denotare l'associazione binaria nel suo complesso e l'associazione diretta conseguente CON LA MEDESIMA azione anche se non fosse possibile utilizzare la forma ATTIVA.*

### Rappresentazione grafica di associazioni multiple n-arie e loro trasformazione in binarie

Vediamo ora come si possono rappresentare graficamente in un diagramma ER le **associazioni multiple o n-arie** ossia quelle associazioni che collegano tra loro le istanze di più di due entità.

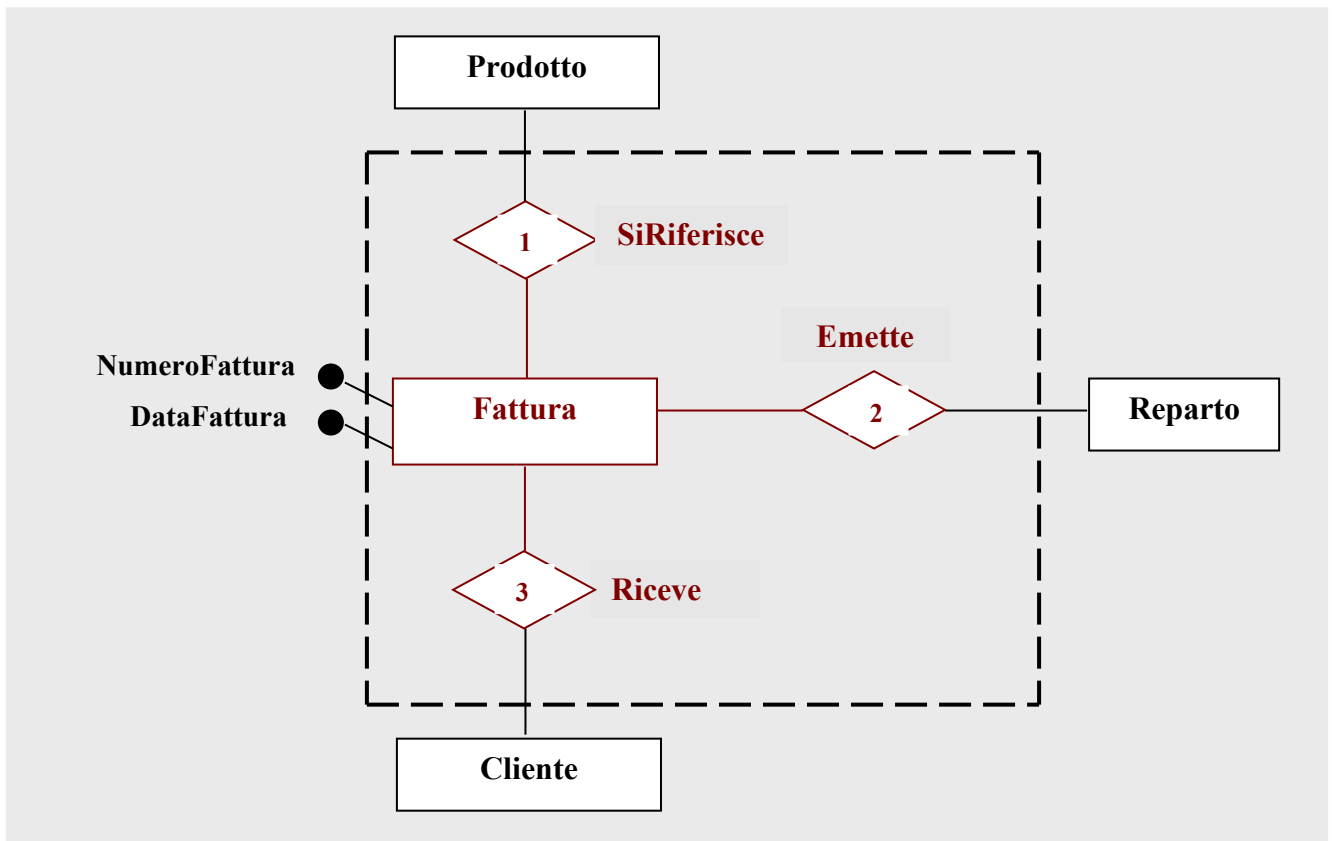
*Esempio: consideriamo la realtà di un supermercato in cui si vuole rappresentare il fatto che ogni reparto possa emettere fattura per i clienti. E' possibile utilizzare l'associazione ternaria (ossia che coinvolge tre entità) **HaFatturato** così schematizzabile:*



**N.B.** Le **associazioni multiple o n-arie** possono essere sempre trasformate in **associazioni binarie**, trasformando l'associazione in una opportuna entità ed introducendo tante associazioni tra questa nuova entità e le entità coinvolte nell'associazione multipla al fine di garantire l'originario collegamento tra di esse.

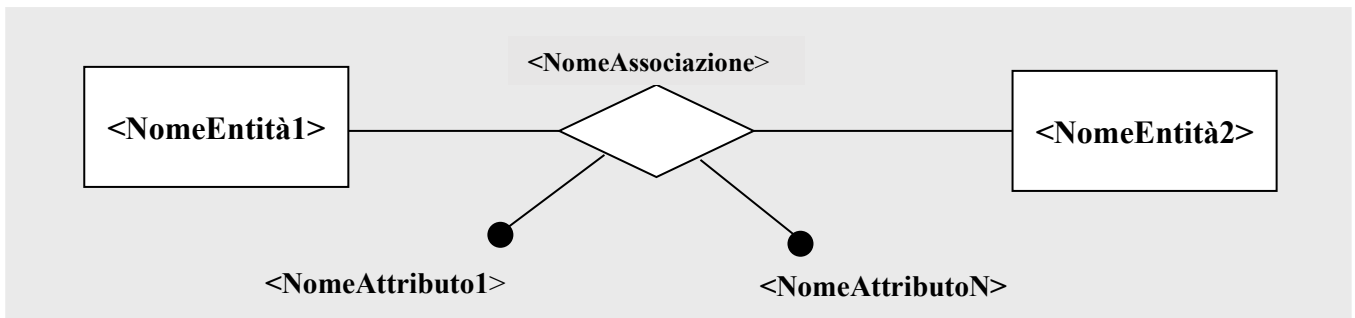
**Non appare riduttivo quindi considerare solo le associazioni binarie tra entità poiché qualsiasi associazione multipla potrà sempre essere trasformata in più associazioni binarie.**

*Esempio: Quindi la precedente associazione **HaFatturato** può essere trasformata nell'entità **Fattura** e sono state aggiunte **tre** associazioni binarie necessarie per garantire il collegamento tra le entità presenti.*



## Rappresentazione grafica delle associazioni binarie

Per rappresentare graficamente in un diagramma ER **un'associazione tra due entità o associazione binaria** si utilizza la seguente notazione dove per semplicità omettiamo gli attributi delle due entità.



Il *rombo* posto a metà della linea che unisce le due entità rappresenta la loro associazione.

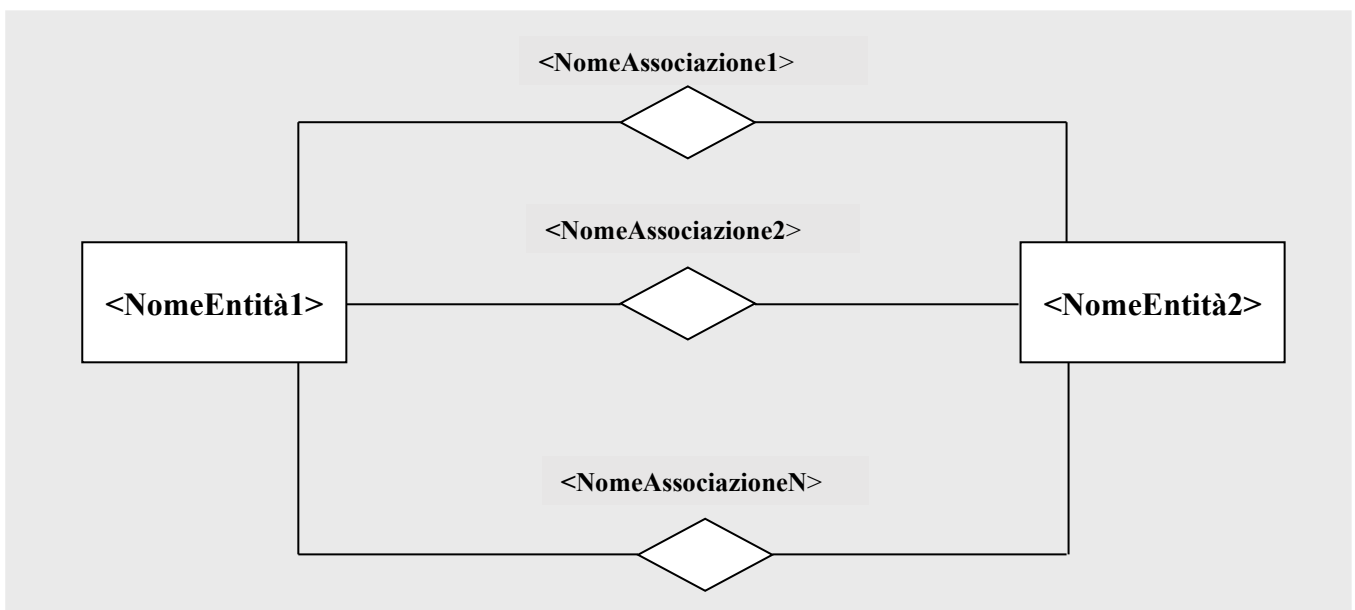
Per gli eventuali *attributi dell'associazione* si seguono le stesse regole degli attributi di una entità.

Per *convenzione* utilizziamo:

- *iniziali minuscole* per la descrizione dell'associazione
- *iniziali maiuscole* per i nomi degli *attributi* delle associazioni composti da più parole (eccetto naturalmente la prima).

## Rappresentazione grafica di più associazioni binarie tra due entità

Per rappresentare graficamente in un diagramma ER **più associazioni tra due entità** si utilizza la seguente notazione dove per semplicità **omettiamo** sia **gli attributi** delle due **entità**, sia gli eventuali **attributi** delle **associazioni**.





## Contenuto informativo di un diagramma ER

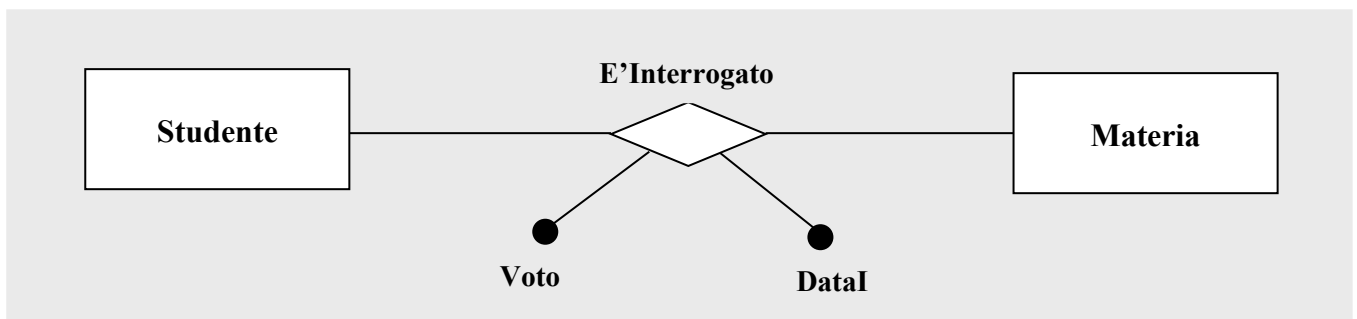
Come è facilmente possibile provare con un esempio, data una stessa realtà di interesse, è possibile descriverla con diversi schemi del modello ER; l'importante è che essi abbiano *il medesimo contenuto informativo*.

**DEF:** Si dice che due **diagrammi ER D1 e D2** hanno lo **stesso contenuto informativo**, quando per ogni interrogazione che possiamo fare sul diagramma **D1** esiste un'altra interrogazione che possiamo fare sul diagramma **D2** in grado di fornire la stessa risposta.  
In questo caso i due diagrammi ER **D1 e D2** si dicono **equivalenti**.

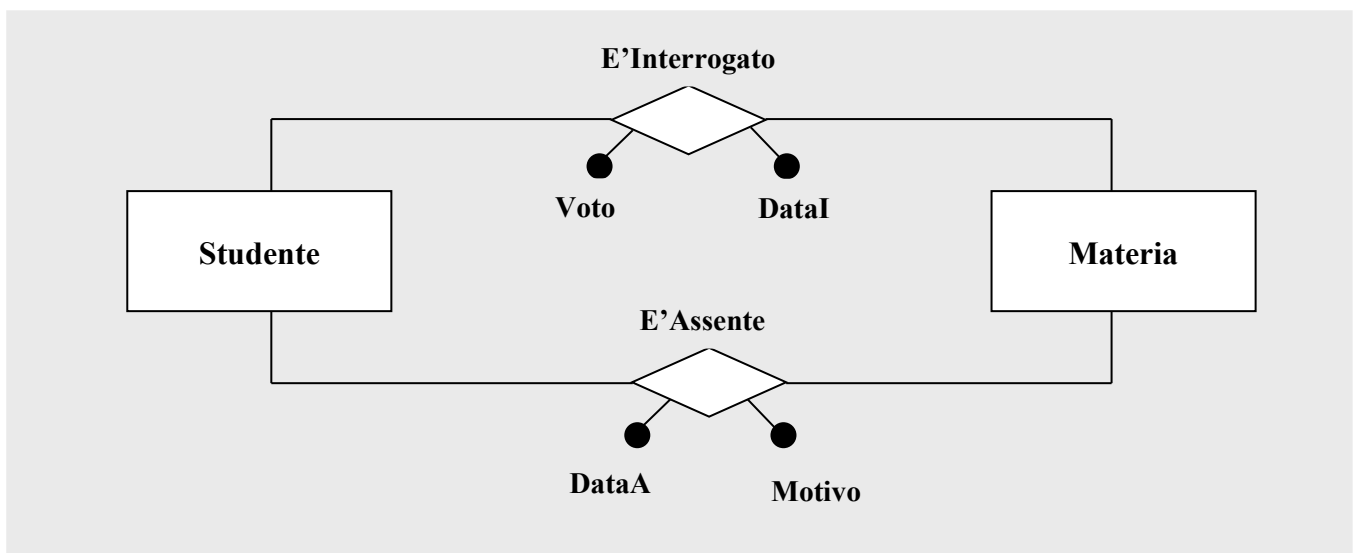
Il concetto che seguiremo d'ora in poi per scegliere quale diagramma ER utilizzare **tra due o più modelli equivalenti** è quello che privilegia la **chiarezza** ossia la **leggibilità**, la **comprensibilità** e l'**esaustività** rispetto al contenuto informativo della realtà che modella.

*Esempio: Consideriamo la realtà di una classe di alunni in cui si vogliono rappresentare le interrogazioni.*

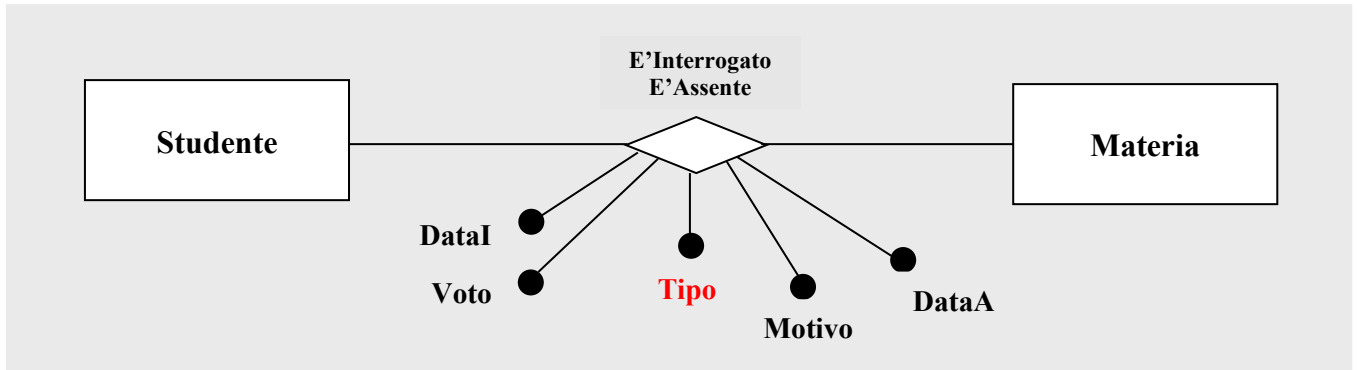
*E' possibile utilizzare il seguente diagramma ER*



*Supponiamo di volere informazioni anche sulle assenze: possiamo aggiungere al precedente diagramma un'altra associazione ottenendo il seguente diagramma ER **D1***



Una soluzione alternativa è offerta dal seguente diagramma ER **D2**



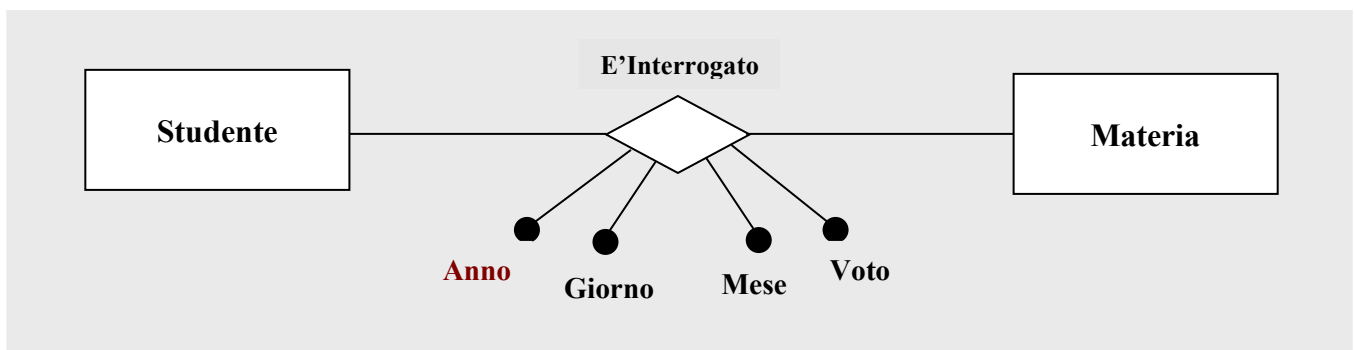
L'attributo *Tipo* indicherà di volta in volta se si tratta di interrogazione o di assenza.

In caso si trattasse di "assenza" gli attributi "Voto" "DataI" rimarrebbero non valorizzati (NULL), mentre se si trattasse di "interrogazione" gli attributi "Motivo" e "DataA" rimarrebbero non valorizzati.

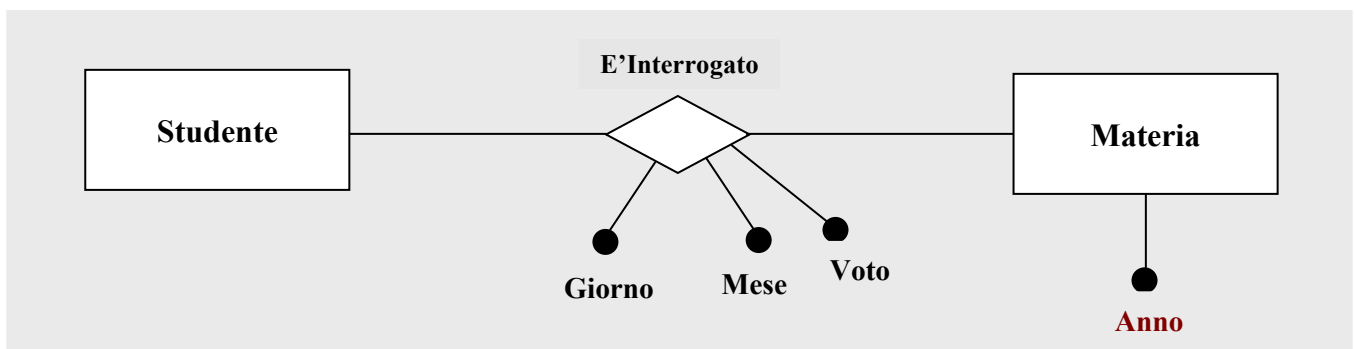
Ipoteticamente posti sul piatto di una bilancia in grado di valutarne il contenuto informativo essi risulterebbero **equivalenti** ma appare del tutto evidente di come il diagramma ER **D1** sia **molto più chiaro** rispetto al diagramma ER **D2** in quanto **separa concetti che sono intrinsecamente diversi** ("essere assente" è un concetto che non può essere accomunato a quello di "essere interrogato").

#### Attributo che passa dall'associazione all'entità

Consideriamo il seguente diagramma ER in cui abbiamo **tre attributi** per specificare la **data** dell'interrogazione (abbiamo scomposto un attributo considerato aggregato):

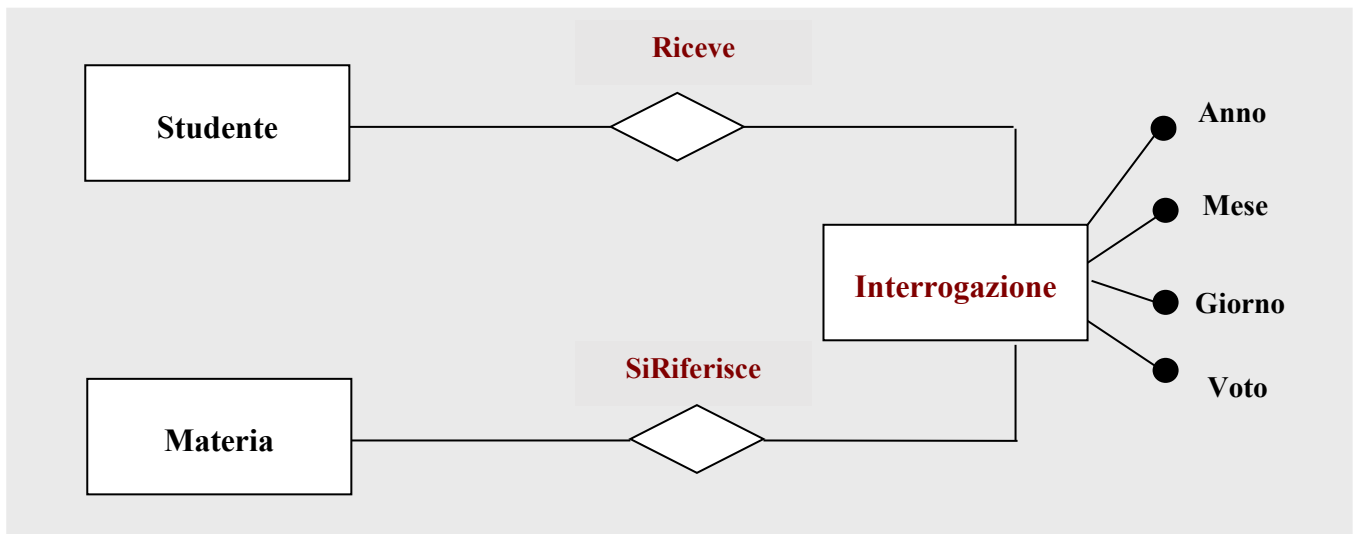


Se ci riferissimo alle interrogazioni di uno stesso anno, ossia se Anno avesse un valore fisso, il diagramma ER precedente diventerebbe:



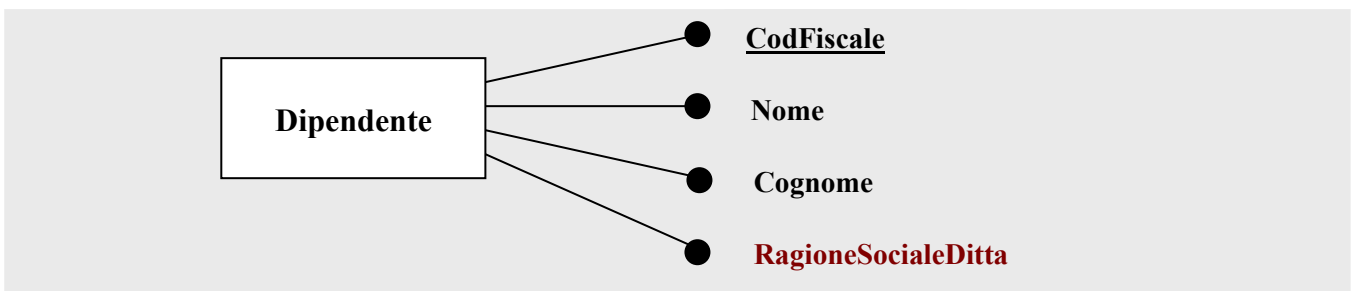
dove l'attributo Anno dell'associazione è diventato un attributo dell'entità Materia.

Per eliminare tutti gli attributi di un associazione occorre introdurre una nuova entità che contiene tali attributi. *In questo esempio allora avremmo*



### Attributo che diventa un'entità

Consideriamo il seguente diagramma ER in cui è presente l'entità *Dipendente* caratterizzato tra gli altri dall'attributo *RagioneSocialeDitta*:

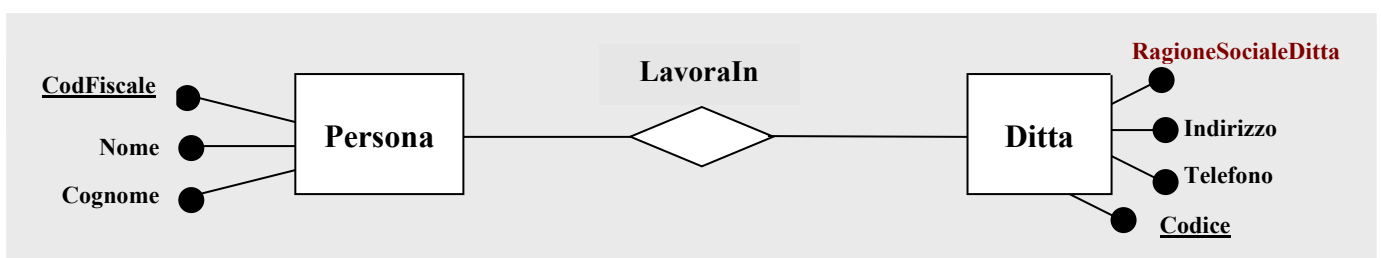


Se siamo interessati a conoscere solo la ragione sociale della ditta presso cui è impiegata quella persona allora l'attributo *RagioneSocialeDitta* è sufficiente.

Ma se occorre conoscere altre informazioni relative a quella ditta (ad esempio indirizzo, numero di telefono, indirizzo e-mail, sito web, etc.), allora occorrerà aggiungere altri attributi all'entità *Dipendente*.

Questa soluzione **non è assolutamente conveniente** in quanto replicherebbe gli stessi valori per gli attributi aggiunti relativi alla ditta per tutte le istanze dell'entità *Dipendente* che lavorano presso la medesima ditta. Inoltre una eventuale modifica relativa ad uno dei dati della ditta (ad esempio il numero di telefono) andrebbe replicato in tutte le istanze ove esso è presente.

La soluzione **più conveniente da adottare** in tale caso è quella di aggiungere un'altra entità, che nel nostro caso chiamiamo *Ditta*, da mettere in relazione con l'entità *Dipendente* la quale sia caratterizzata dai seguenti attributi:

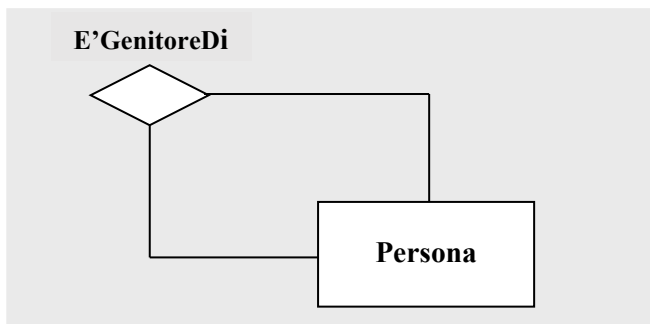


## Rappresentazione di un'associazione sulla stessa entità (o RICORSIVA): i ruoli

Quando un'associazione crea un legame tra due entità identiche (ossia la stessa entità) si ha il particolare caso di associazione sulla stessa entità.

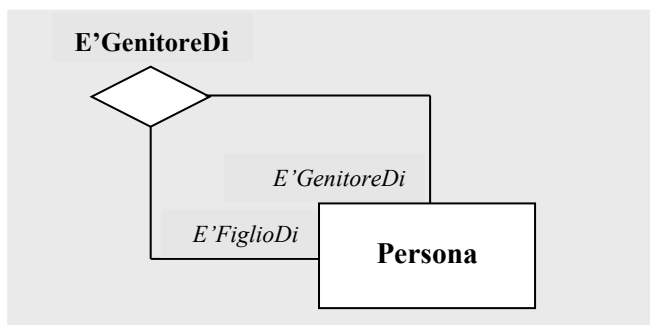
Ad esempio consideriamo l'entità *Persona* e supponiamo di volere rappresentare l'associazione "E'GenitoreDi".

Questo classico esempio di associazione si rappresenta graficamente così:



Poiché come abbiamo detto un'associazione è una *classe di interazioni o legami* che collegano le istanze di due o più *entità* dobbiamo distinguere *in ogni interazione* che costituisce un'istanza dell'entità *Persona*, chi è il genitore e chi è il figlio.

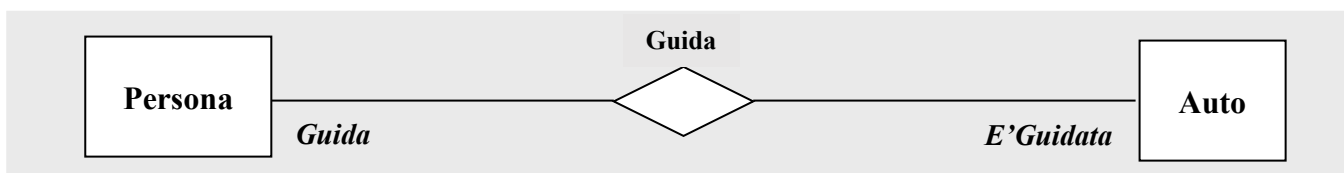
Graficamente occorre distinguere qual è il ramo che rappresenta il genitore e qual è il ramo che rappresenta il figlio.



dove "E'GenitoreDi" ed "E'FiglioDi" sono anche detti **ruolo** dell'entità nell'ambito dell'associazione (che determina anche la **direzione** o **verso** delle associazioni dirette e/o inverse che vedremo in seguito).

In questo diagramma una qualsiasi istanza dell'entità *Persona* svolge due diversi ruoli nell'ambito dell'associazione "E'GenitoreDi": il **ruolo** di genitore ed il **ruolo** di figlio.

**N.B.** Il concetto di **ruolo** se necessario, vista la sua funzione chiarificatrice, può essere esteso a **qualsiasi associazione binaria** come nel seguente diagramma ER:



### Associazioni dirette ed inverse

L'associazione "Guida" nell'esempio precedente ha quindi due **versi**:

- uno che va da *Persona* ad *Auto* (relativo all'associazione "Guida" che possiamo definire *DIRETTA*);
- uno che va da *Auto* a *Persona* (relativo all'associazione "E'Guidata" che possiamo definire *INVERSA*).

Data un'associazione binaria **A** tra le entità X ed Y, si attribuisce un *verso* da X ad Y che definisce l'**associazione diretta** ed un verso da Y ad X (che indicheremo con  $A^{-1}$ ) che definisce l'**associazione inversa**.

Possiamo allora dire che la coppia di istanze (y, x) con  $y \in Y$  ed  $x \in X$  appartiene ad  $A^{-1}$  se e solo se la coppia (x, y) appartiene ad A.

**N.B. Occorre sottolineare che A ed  $A^{-1}$  sono una l'inversa dell'altra e definiscono la stessa associazione salvo che per l'ordine delle coppie di istanze delle entità X ed Y che ne fanno parte.**

La scelta dunque del verso relativo all'associazione diretta e del verso relativo all'associazione inversa è puramente arbitrario.

E' possibile fare riferimento ai **versi** dell'associazione diretta ed inversa utilizzando i **ruoli**.

### Associazioni totali e parziali

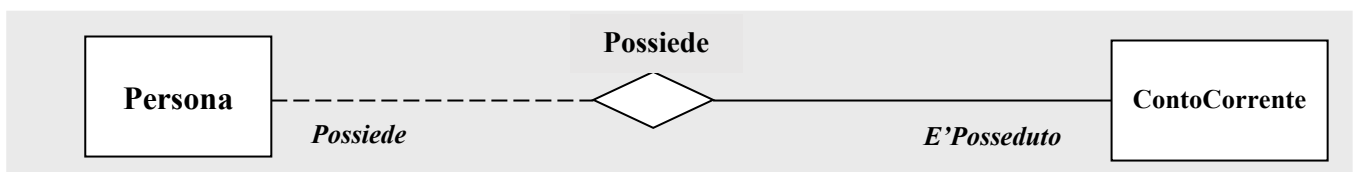
Un'associazione binaria **A** tra le entità X ed Y si dice **totale** quando il legame tra le istanze delle due entità deve essere sempre presente (ossia ad ogni elemento di X deve corrispondere almeno un elemento di Y o viceversa ad ogni elemento di Y deve corrispondere almeno un elemento di X), altrimenti si dice **parziale** (ossia può esistere un elemento di X a cui non corrisponde alcun elemento di Y o viceversa un elemento di Y a cui non corrisponde alcun elemento di X)

Per rappresentare graficamente un'associazione parziale utilizzeremo **una linea tratteggiata**.

Può succedere che entrambe le associazioni (diretta e inversa) siano totali oppure parziali ma può anche accadere che una sola associazione (diretta o inversa) sia totale mentre l'altra no.

Ciò dipende esclusivamente dalla realtà di interesse cui la base dati fa riferimento.

*Esempio: Consideriamo il seguente diagramma ER*



*In tale associazione notiamo che:*

. l'associazione diretta "Possiede" è rappresentata con linea tratteggiata è **parziale** poiché non tutte le persone possiedono un conto corrente bancario;

. l'associazione inversa "E'Posseduto" è rappresentata con linea continua è **totale** poiché ogni conto corrente bancario deve essere intestato ad almeno una persona.

## Tipologia di Associazioni binarie

**Definizione:** La **molteplicità** di un'associazione binaria qualsiasi fra le entità X ed Y indica una istanza dell'entità X con quante istanze dell'entità Y è TEORICAMENTE collegata e viceversa.

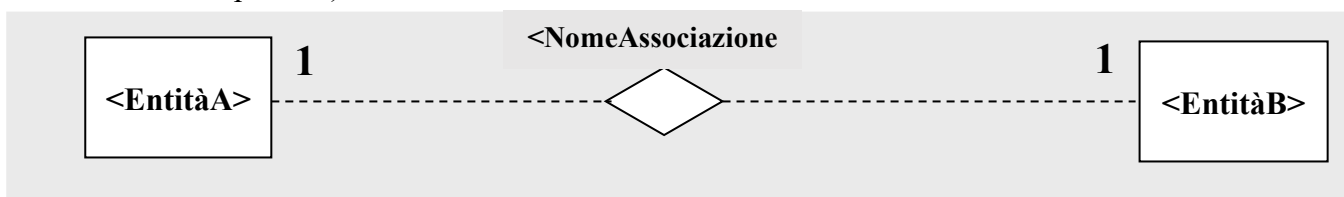
**Definizione:** La **cardinalità** di un'associazione binaria qualsiasi fra le entità X ed Y indica il numero delle coppie di elementi di X e di Y che in un certo istante sono REALMENTE state collegate dall'associazione.

**Un'associazione binaria tra due entità A e B può avere molteplicità:**

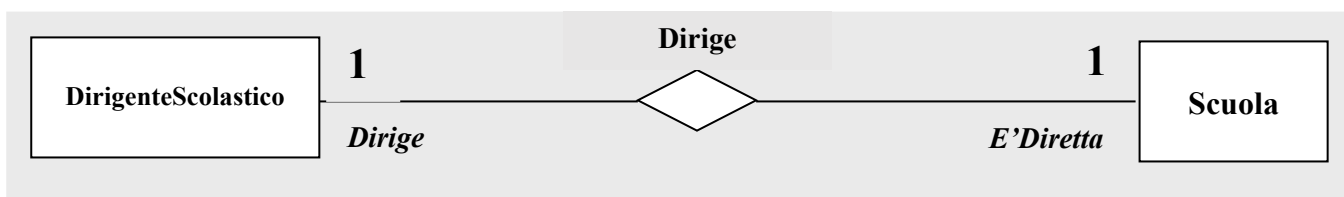
a) **uno a uno** (ossia **1:1**) quando ad un'istanza dell'entità A può/deve corrispondere nessuna/una istanza dell'entità B e viceversa quando ad un'istanza dell'entità B può/deve corrispondere nessuna/una istanza istanza dell'entità A

**N.B. In questo caso sia l'associazione diretta, sia l'associazione inversa avranno molteplicità 1:1**

Graficamente tale tipo di associazione viene rappresentata così (in caso di associazione diretta ed inversa entrambe parziali):

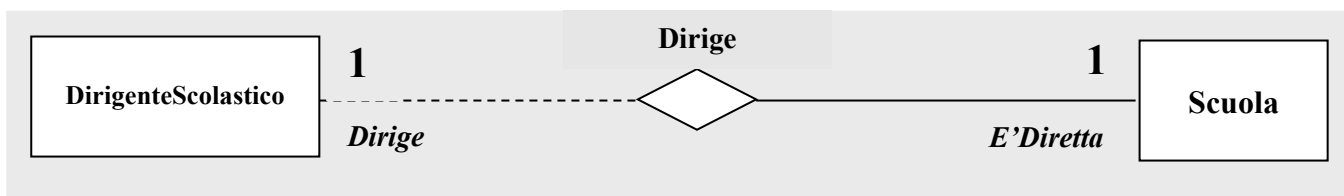


*Esempio: un'associazione uno a uno è quella che esiste tra le entità "DirigenteScolastico" e "Scuola". Graficamente esplicitando i ruoli il diagramma ER è il seguente.*



**Regola di lettura:** Un Dirigente scolastico **deve** dirigere **una** sola scuola (per l'associazione diretta) e viceversa **una** scuola **deve** essere diretta da **un** solo dirigente scolastico (per l'associazione inversa).

*N.B. Questa associazione potrebbe avere l'associazione diretta **parziale** nel caso venissero considerati anche i presidi in pensione*



b) **uno a molti** (ossia **1:N**) quando ad un'istanza dell'entità A può/deve corrispondere nessuna/una o più istanze dell'entità B ed ad ogni istanza dell'entità B può/deve corrispondere nessuna o una sola istanza dell'entità A (la scelta di A o B come entità di partenza essendo del tutto casuale).

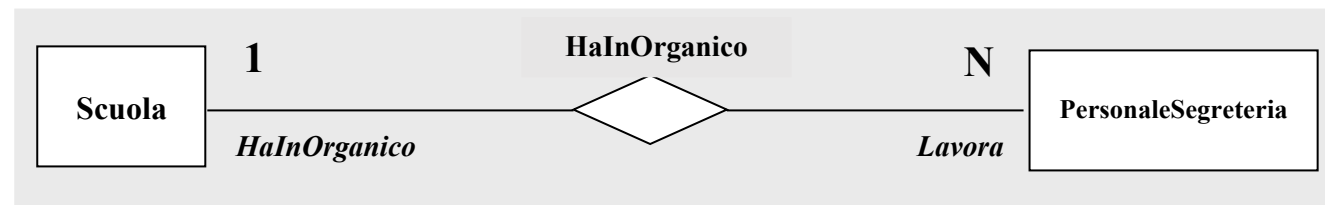
**N.B. In questo caso l'associazione diretta avrà molteplicità 1:N mentre l'associazione inversa avrà molteplicità 1:1**

Graficamente tale tipo di associazione viene rappresentata così (in caso di associazione diretta ed inversa entrambe parziali):



*Esempio: un'associazione **uno a molti** è quella che esiste tra le entità “Scuola” e “PersonaleSegreteria”.*

*Graficamente esplicitando i ruoli il diagramma ER è il seguente.*



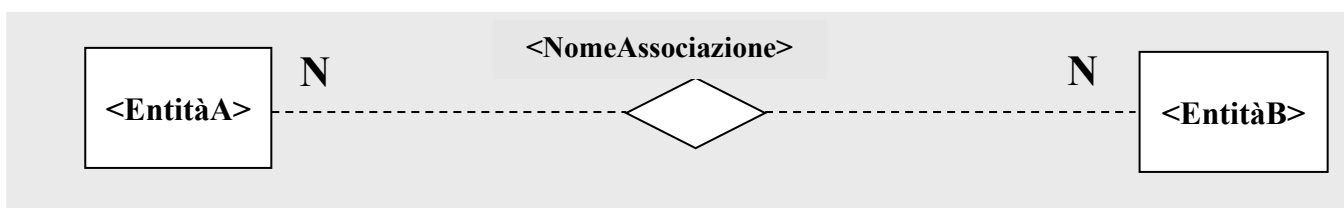
**Regola di lettura** Una Scuola deve avere in organico **uno o più** personale di segreteria (per l'associazione diretta) e viceversa **una** persona di segreteria deve lavorare in **una** sola scuola (per l'associazione inversa).

**N.B.** Per un'associazione di molteplicità N:1 il discorso è l'inverso di quello fatto nel caso di molteplicità 1:N

**N.B.** In questo caso l'associazione diretta avrà molteplicità 1:1 mentre l'associazione inversa avrà molteplicità 1:N

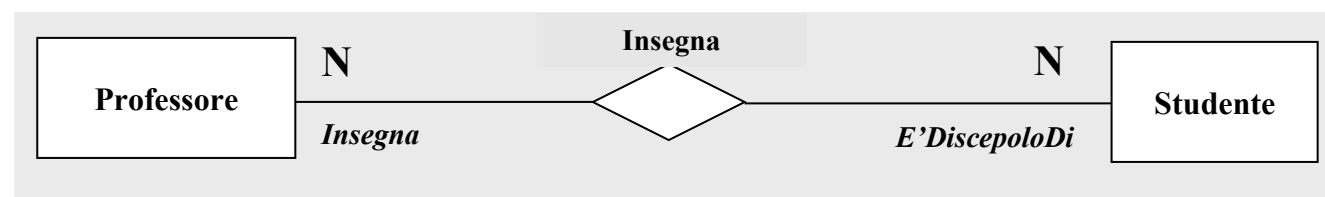
c) **molti a molti** (ossia N:N): quando ad un'istanza dell'entità A può/deve corrispondere nessuna/una o più istanze dell'entità B e viceversa ad un'istanza dell'entità B può/deve corrispondere nessuna/una o più istanze dell'entità A.

Graficamente tale tipo di associazione viene rappresentata così (in caso di associazione diretta ed inversa entrambe parziali) :



*Esempio: Sia assegnata l'associazione binaria “Insegna” tra le entità “Professore” e “Studente”. con molteplicità **molti a molti**.*

*Graficamente esplicitando i ruoli il diagramma ER è il seguente.*



**Regola di lettura** Un professore deve insegnare ad **uno o più** studenti (per l'associazione diretta) e viceversa **uno** studente deve essere discepolo di **uno o più** professori (per l'associazione inversa).

**N.B.**

**Le regole di lettura servono per effettuare un controllo sulla bontà della modellazione effettuata. Infatti se 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 effettuato un buon lavoro di analisi della realtà di interesse.**

Per riferirci ad un attributo di una particolare associazione possiamo fare riferimento alla seguente pseudo istruzione basata sulla *dot.notation*:

**<NomeAssociazione>.<NomeAttributo>**



## I VINCOLI DI INTEGRITA' DI UN DIAGRAMMA ER

E' stato detto che una base di dati definisce un insieme di **regole** che devono essere soddisfatte da tutte le istanze delle categorie.

In un diagramma ER le **regole** a livello concettuale possono essere definite precisando i **vincoli di integrità**.

Un **vincolo di integrità** (in breve **V.d.I.**) è un'asserzione ovvero un predicato che deve essere soddisfatto da tutte le istanze di una entità o di una associazione presente nella base dati

I **vincoli di integrità** sono così chiamati perché devono essere rispettati pena la consistenza stessa dei dati.

*(Se ad esempio l'attributo Stipendio dell'entità Dipendente assumesse valori negativi potrebbe causare grossi problemi al corretto funzionamento dei programmi che lo utilizzano in quanto trattasi di un'evidente incongruenza)*

I **vincoli di integrità** sono di due tipi:

1 – **vincoli di integrità IMPLICITI**: sono quei vincoli imposti dalla stessa natura dei dati che possono essere rappresentati direttamente nel diagramma ER con particolari "accorgimenti" grafici.

Possono essere a loro volta divisi in:

1.1 – **vincoli di chiave primaria**: questi vincoli impongono che tutte le istanze di una entità debbano essere diverse tra loro e si rappresentano graficamente sottolineando i relativi attributi che ne fanno parte;

1.2 – **vincoli referenziali**: Questi vincoli derivano direttamente dalla TOTALITA' di un'associazione (diretta o inversa) e si rappresentano graficamente con linee continue di collegamento con le entità coinvolte.

*(N.B. Date due entità A e B ed un'associazione tra loro supposta TOTALE da A verso B. Questa eventualità IMPONE che non si possa inserire un'istanza in A o che non esista un'istanza in A che non sia associata ad una istanza di B. Ecco il vincolo (obbligo) da rispettare)*

2 – **vincoli di integrità ESPLICITI**: sono quei vincoli che occorre esplicitare con apposite dichiarazioni NON essendo possibili desumerli graficamente dal diagramma ER.

*(Esempio: il valore dell'attributo Stipendio dell'entità Dipendente non può essere negativo né maggiore di 1800.00 euro).*

Per rappresentare tale tipo di vincoli ricorriamo ad una NOTAZIONE che utilizzi la seguente pseudoistruzione:

**V<NumProg>: (<Espressione>)**

Dove

<NumProg> è un numero progressivo del vincolo;

<Espressione> è una qualsiasi espressione in pseudolinguaggio naturale o di tipo logico-matematico che specifichi il vincolo stesso

*Esempio:*

*Per esprimere il vincolo sullo Stipendio dell'entità Dipendente prima ipotizzato potremmo scrivere:*

**V1: (0 < Persona.Stipendio ≤ 1800.00)**

*Per esprimere il vincolo che "non si può assumere un dipendente di Età inferiore a 16 anni" scriveremo utilizzando la funzione YEAR():*

**V2: ( YEAR(Dipendente.DataAssunzione) - YEAR(Dipendente.DataNascita) >16 )**

## ASSOCIAZIONI PER GENERALIZZAZIONE: LE GERARCHIE ISA

**Definizione:** Un'associazione per generalizzazione nel modello ER è esattamente un'astrazione per generalizzazione così come definita in precedenza.

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

Tale associazione è anche chiamata **associazione ISA** dall'inglese "is a" ossia in italiano "è un".

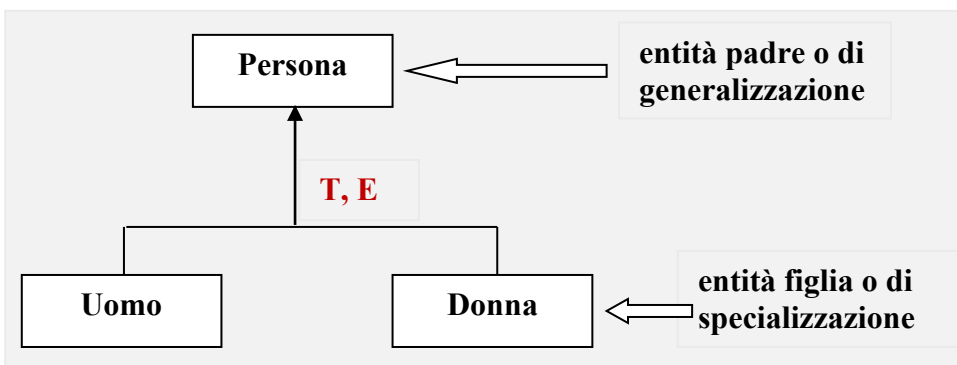
Si parla di **generalizzazione totale** (indicata con **T**) quando ogni occorrenza o istanza dell'entità padre è un'occorrenza di almeno una delle entità figlie

(altrimenti viene detta non totale o **parziale** ed indicata con  $\overline{\mathbf{T}}$ ):

Si parla di **generalizzazione esclusiva** (indicata con **E**) quando ogni occorrenza o istanza dell'entità padre è un'occorrenza di al massimo una delle entità figlie

(altrimenti viene detta non esclusiva o **sovrapposta** ed indicata con  $\overline{\mathbf{E}}$ ):

*Esempio:* Come abbiamo visto in precedenza l'entità "Persona" è la generalizzazione delle entità "Uomo" e "Donna".



La generalizzazione dell'esempio precedente tra le entità "Persona" vista come specializzazione delle entità "Uomo" e "Donna" risulta:

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

Ricordiamo che secondo l'astrazione per generalizzazione, le classi "Uomo" e "Donna" sono sottoinsiemi della classe "Persona" ossia sono in essa contenute.

In un'associazione per generalizzazione sussiste il principio di **ereditarietà** che permette di associare alle entità figlie attributi che non compaiono tra gli attributi della classe padre

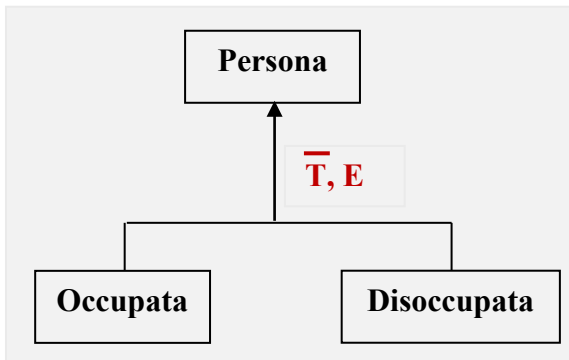
**Principio di ereditarietà:** In una qualunque associazione per generalizzazione (ISA) ogni **proprietà** dell'entità padre è anche **proprietà** delle entità figlie e viceversa ogni **proprietà** delle entità figlie è anche **proprietà** delle entità padre (ma non delle delle altre entità figlie partecipanti alla generalizzazione) dove per *proprietà* intendiamo:

- gli attributi,
- le associazioni;
- le generalizzazioni.

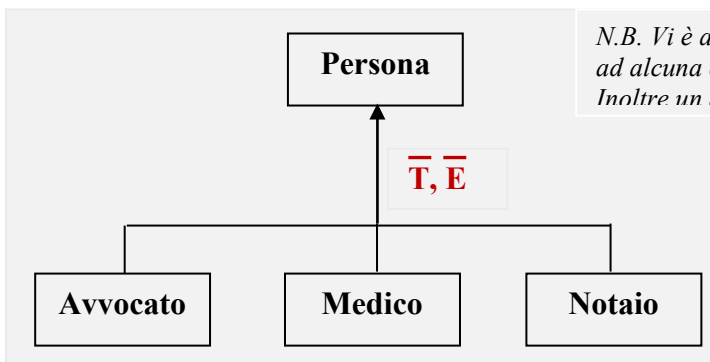
**N.B. Il principio di ereditarietà è uno strumento molto importante perché ci consente di non duplicare l'informazione e quindi di evitare la ridondanza.**

**N.B. Non tutte le generalizzazioni sono TOTALI ed ESCLUSIVE potendo esserci casi di ISA che prevedono le altre tre possibili combinazioni (parziale ed esclusiva, parziale e sovrapposta e totale e sovrapposta,)**

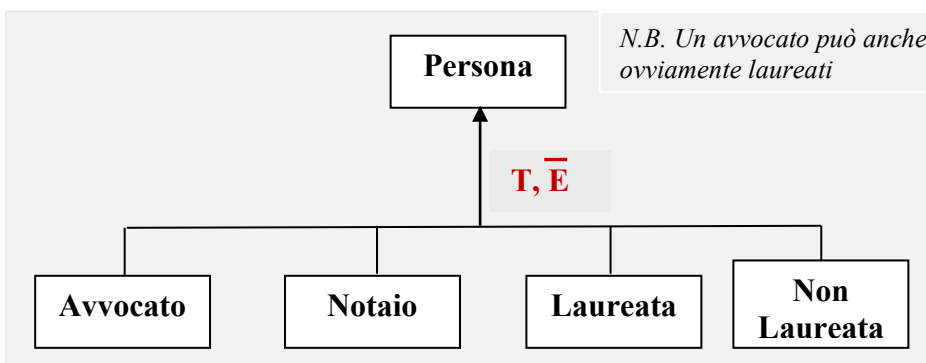
*Esempi riepilogativi applicati alla medesima entità Persona*



*N.B. Vi è almeno un'istanza dell'entità Persona che non appartiene ad alcuna entità figlia (esempio un pensionato)*



*N.B. Vi è almeno un'istanza dell'entità Persona che non appartiene ad alcuna entità figlia (esempio un non laureato). Inoltre un avvocato può anche essere notaio*

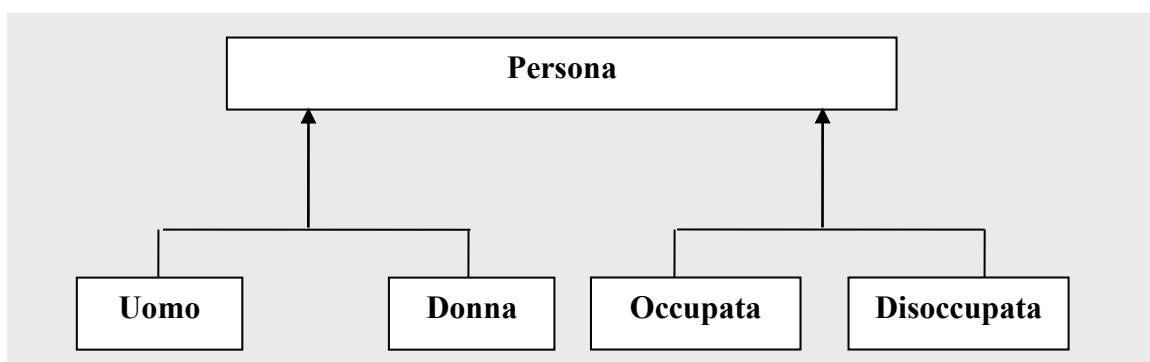


*N.B. Un avvocato può anche essere notaio e sono entrambi ovviamente laureati*

**Una stessa entità può essere generalizzazione di diversi insiemi di entità.**

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 dalle entità figlie Occupata e Disoccupata.*



## Strategie per la “trasformazione” di un’ISA all’interno di un diagramma ER

Per “trasformare” una associazione per generalizzazione nel modello relazionale possiamo seguire tre strategie diverse:

**a) accorpamento delle entità figlie nell’entità padre**

(n.b. strategia valida qualunque sia la tipologia di ISA individuata);

**b) accorpamento dell’entità padre nelle entità figlie**

(n.b. strategia valida solo se l’ISA è TOTALE);

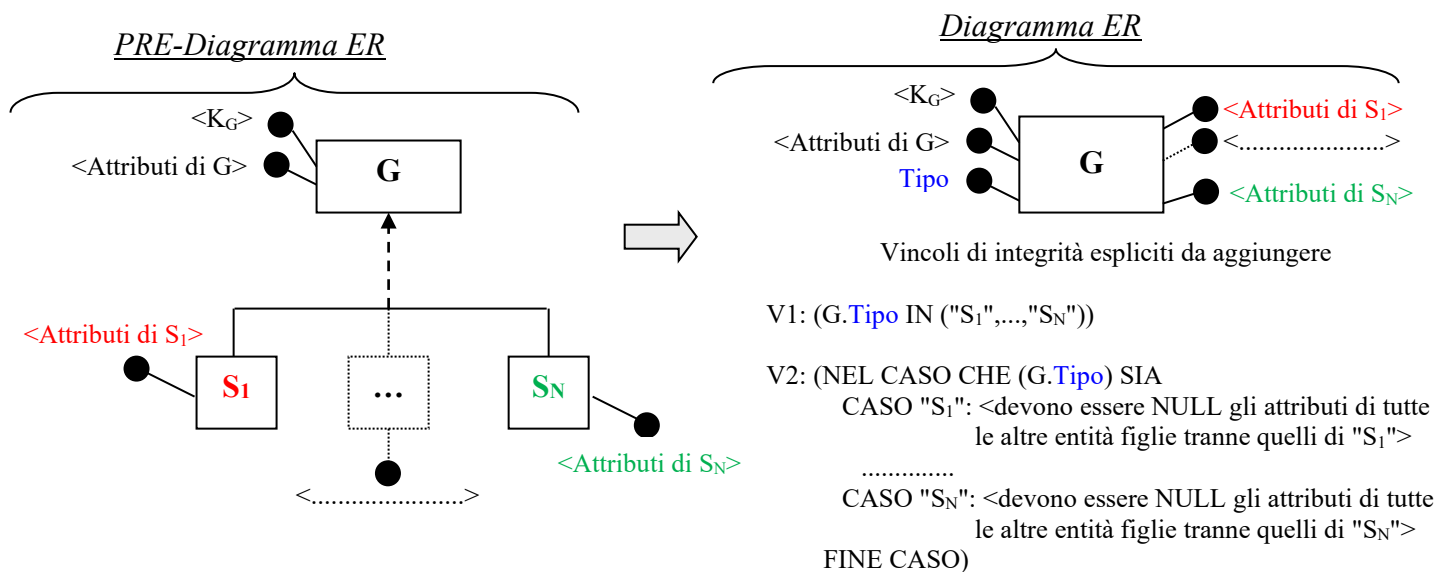
**c) sostituzione della generalizzazione con associazioni binarie di molteplicità 1:1**

(n.b. valida solo se l’ISA è ESCLUSIVA);

**a) Nell’accorpamento delle entità figlie nell’entità padre:** le entità figlie  $S_1, S_2, \dots, S_N$  vengono eliminate ed i loro attributi e le associazioni cui partecipano, vengono aggiunti all’entità padre  $G$ . All’entità padre viene aggiunto un altro attributo che serve per distinguere il tipo di ogni **ennupla** del padre ossia per distinguere se ciascuna ennupla appartiene a  $S_1, S_2, \dots, S_N$ . A seconda del valore di questo attributo saranno annullati (NULL) tutti quegli attributi aggiunti all’entità padre appartenenti alle altre entità figlie.

**OSS:** Tale traduzione conviene quando le operazioni sulla base dati non fanno molta distinzione tra ennuple di una figlia o di un’altra e tra gli attributi di una figlia ed un’altra.

In questo caso avremmo minimizzato gli accessi alla memoria anche se con uno spreco maggiore della stessa visto che vi saranno valori nulli per alcuni attributi di volta in volta



Osservazioni sui vincoli espliciti imposti:

Il vincolo V1 può essere espresso anche attraverso l’utilizzo della disgiunzione logica (OR).

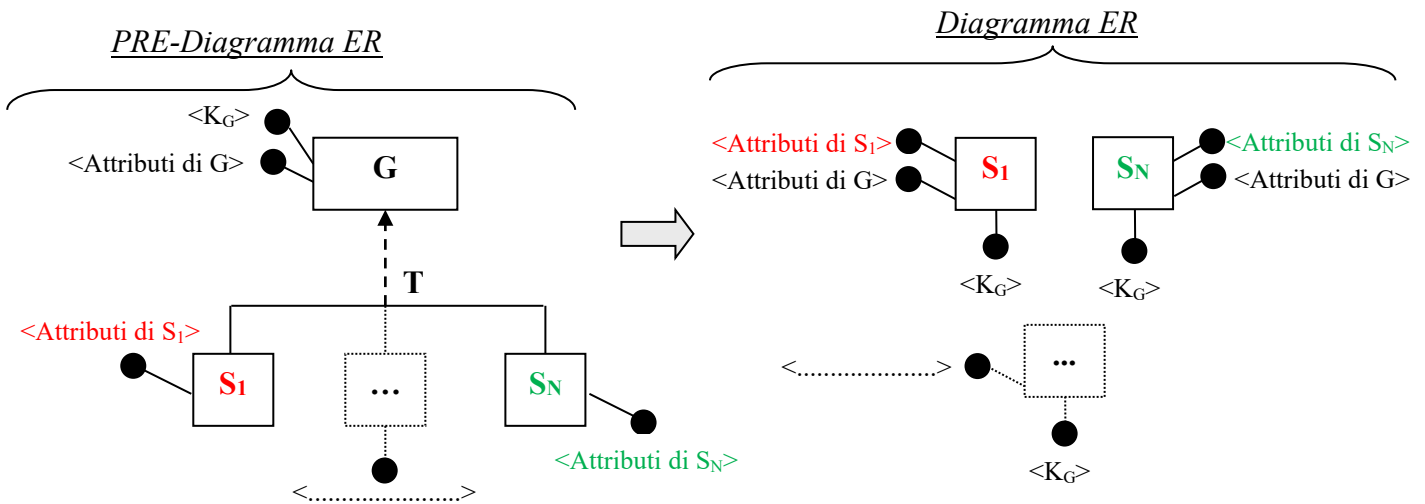
I valori con i quali si decide di identificare le entità figlie sono lasciati al progettista (suggerimento: i nomi delle stesse entità figlie)

Ovviamente per esprimere il vincolo V2 è possibile anche usare pseudoistruzioni selettive annidate.

**N.B.:** Tale “trasformazione” è possibile solo quando la generalizzazione è **TOTALE** ossia se ogni ennupla dell’entità padre **G** è una ennupla **di almeno una** delle entità figlie tra **S<sub>1</sub>** o di **S<sub>2</sub>** o di **S<sub>N</sub>**

**b) Nell’accorpamento dell’entità padre nelle entità figlie:** l’entità padre **G** viene eliminata ed i suoi attributi e le associazioni (anche eventuali altre ISA) alle quali essa partecipa, vengono aggiunti alle figlie **S<sub>1</sub>, S<sub>2</sub>, ..., S<sub>N</sub>**

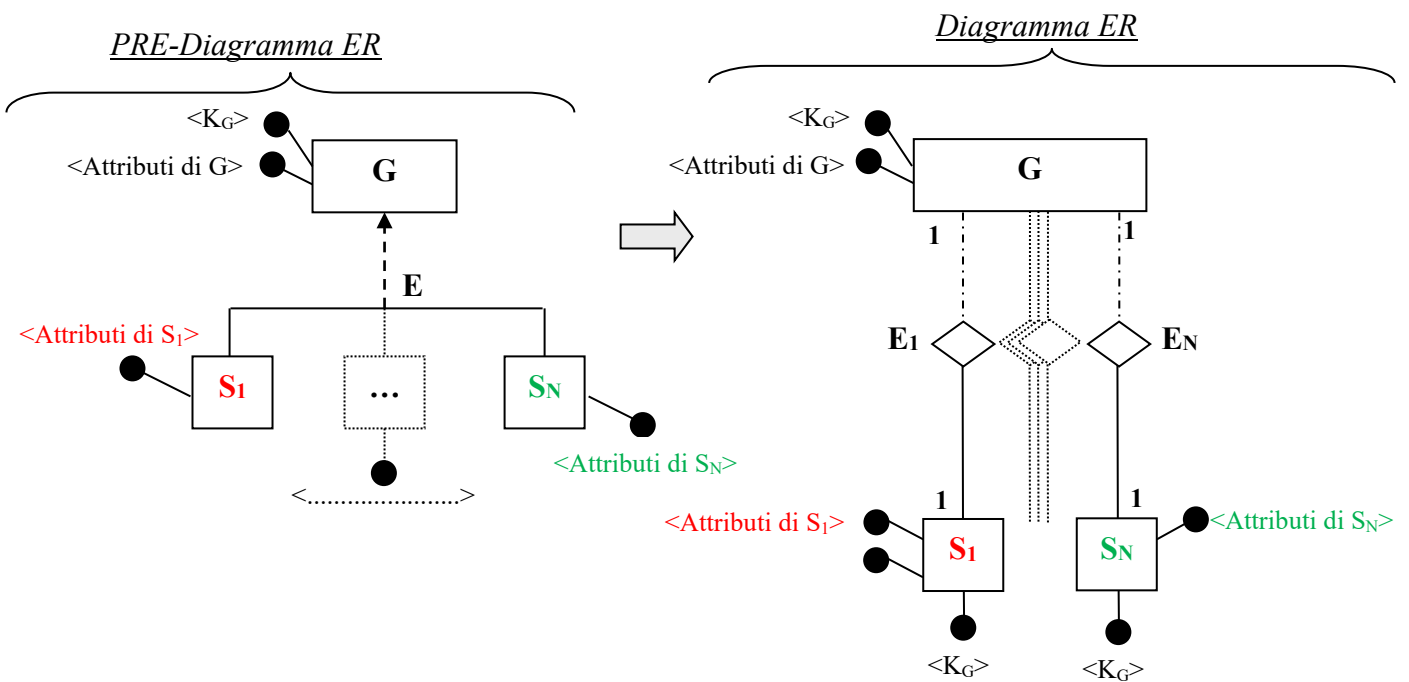
**OSS:** Tale traduzione conviene quando ci sono operazioni sulla base dati che coinvolgono solo ennuple di una figlia o solo ennuple di un’altra figlia, distinguendo tra le entità figlie.



**N.B.:** Tale “trasformazione” è possibile solo quando la generalizzazione è **ESCLUSIVA** ossia se ogni ennupla dell’entità padre **G** è una ennupla **di al massimo una** delle entità figlia tra **S<sub>1</sub>** o di **S<sub>2</sub>** o di **S<sub>N</sub>**

**c) Nella sostituzione della generalizzazione con associazioni binarie di molteplicità 1:1:** in questa traduzione si sostituisce la generalizzazione con tante associazioni di molteplicità 1:1 quante sono le entità figlie partecipanti.

**N.B. la molteplicità 1:1 delle associazioni inserite è un’ovvia conseguenza dell’esclusività dell’ISA**

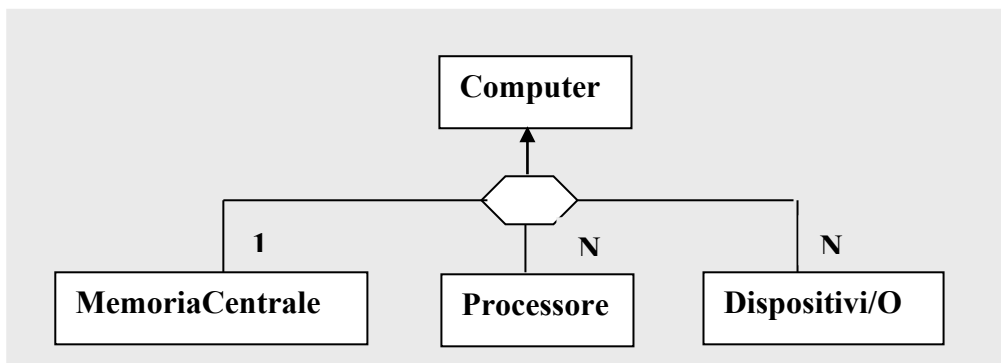


## ASSOCIAZIONE PER AGGREGAZIONE: LE ASSOCIAZIONI HASA

**Definizione:** Un'associazione per aggregazione esplicita nel modello ER è esattamente un'astrazione per aggregazione così come definita in precedenza.

Essa viene anche chiamata **associazione HASA** dall'inglese “has an” ossia in italiano “ha un”. Le entità componenti contribuiscono a formare l'entità **composizione** o **contenitore**.

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



Un'aggregazione generalmente viene classificata in **LASCA** e **STRETTA**

Un'aggregazione **LASCA** indica che l'elemento “contenuto” potrà esistere indipendentemente dall'entità “contenitore”.

Possiamo quindi dire che ad un elemento di un'entità contenuto potrà non corrispondere alcun elemento dell'entità contenitore.

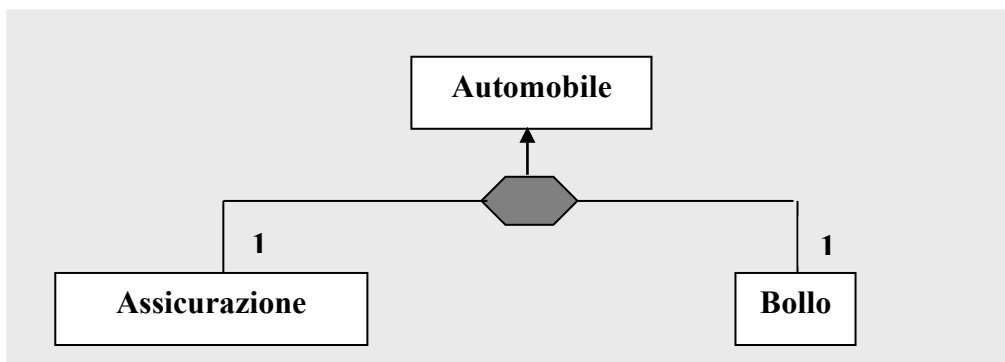
*Graficamente* è rappresentata con un esagono chiaro vuoto (come nell'esempio precedente)

Un'aggregazione **STRETTA** indica che l'elemento “contenuto” non ha vita propria ma esiste unicamente perché è stata creata l'entità “contenitore”.

Possiamo quindi dire che a ogni elemento di un'entità contenuto deve corrispondere almeno un elemento dell'entità contenitore.

*Graficamente* è rappresentata con un esagono pieno (come nell'esempio successivo)

*Esempio: Mettiamo in relazione un'automobile al relativo bollo ed assicurazione. Il bollo per un'auto non ha più senso di esistere senza l'auto su cui agisce.*



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

## RAPPRESENTAZIONE DI INFORMAZIONI NON ALFANUMERICHE: IMMAGINI, AUDIO E VIDEO

E' frequente la richiesta di rappresentare in una base di dati informazioni multimediali quali immagini e suoni.

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 considerato sono tipi primitivi: Interi, Stringhe, Reali, etc.

Per rappresentare quindi informazioni diverse dai tipi primitivi occorre fare riferimento ai **file** che contengono tali informazioni. Il problema allora diventa **come individuare il file** contenente le informazioni *non alfanumeriche* richieste. La **visualizzazione** di tali file sarà un problema da risolvere a livello di programmazione e non a livello concettuale.

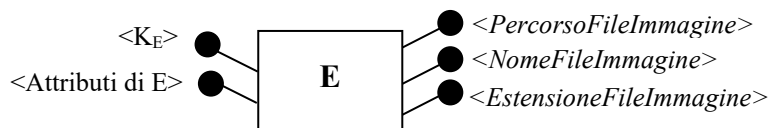
Supponiamo allora che:

- i file siano **memorizzati** in directory di cui occorre individuare il *path* o *percorso* completo;
- i file abbiano una **estensione** che ne specifica il formato;
- i file abbiano **nomi** autoesplicativi del loro contenuto ossia leggendo il nome del file deve essere facile capire cosa contiene.

Occorrerà allora introdurre nuovi attributi per implementare le precedenti ipotesi:

- un attributo che specifichi il **percorso** completo del file (ad esempio *PercorsoFile*);
- un attributo che specifichi l'**estensione** del file (ad esempio *EstensioneFile*);
- un attributo che specifichi il **nome** del file (ad esempio *NomeFile*).

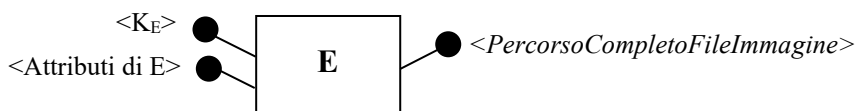
*Esempio: Se all'entità E occorre salvare sul database un'immagine relativa al file "C:\immagini\profilo.jpg" possiamo gestire tale informazione attraverso i seguenti tre attributi:*



con

*PercorsoFileImmagine = "C:\immagini\  
NomeFileImmagine = "profilo"  
EstensioneFile Immagini = "jpg"*

*In caso volessimo gestire il tutto con un minor numero di attributi a livello di database (spostando una parte della logica applicativa a livello software) potremmo anche prevedere la soluzione:*



con

*PercorsoCompletoFileImmagine = "C:\immagini\profilo.jpg"*