

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 raccolta delle richieste (utente), 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**.

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

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.

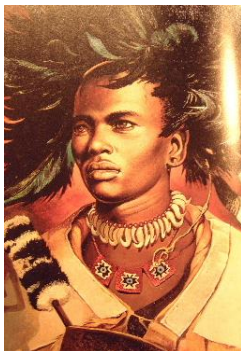
Nella progettazione di basi di dati vengono utilizzati principalmente **3 procedimenti di astrazione**:

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

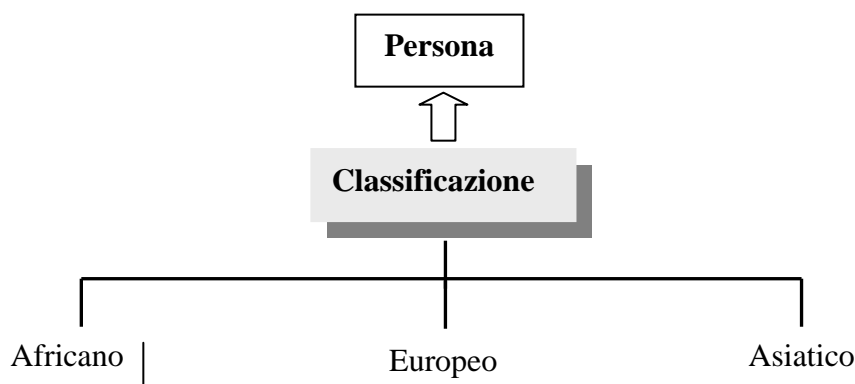
L'**astrazione per classificazione** permette di definire una classe a partire da un insieme di oggetti di cui si individuano le proprietà comuni.

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:

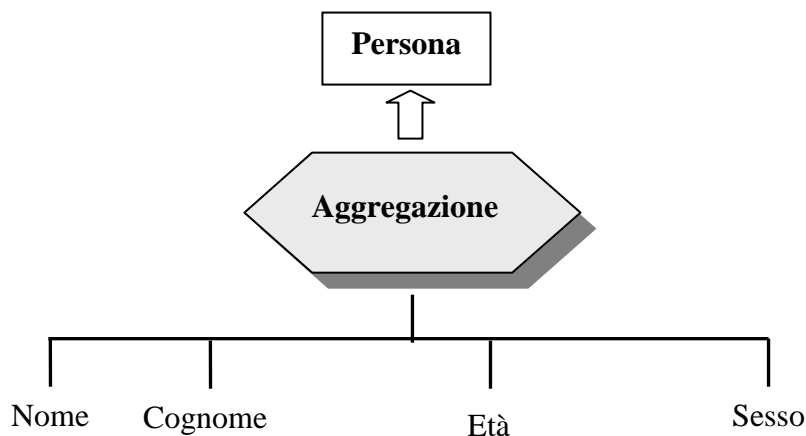


L'**astrazione per aggregazione** è il procedimento mentale che usiamo quando a partire da una o più classi generiamo una nuova classe. Le classi di partenza vengono dette **classi componenti o proprietà** in quanto la loro unione contribuisce a realizzare la classe di arrivo.

Proseguiamo con il nostro esempio.

Consideriamo le proprietà **Nome, Cognome, Età, Sesso** delle persone e cerchiamo di individuare la classe in cui nome, cognome, età, 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.

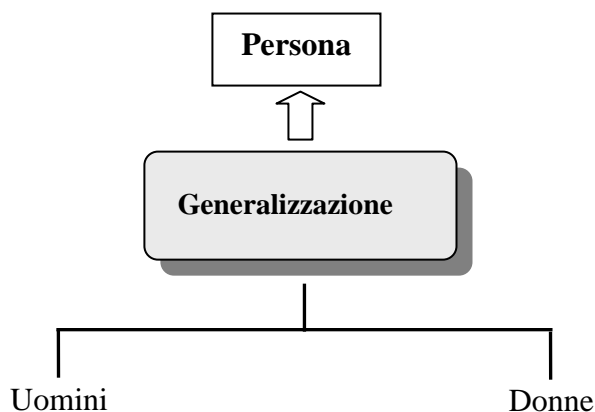
L'**astrazione per generalizzazione** è il procedimento mentale mediante il quale si può giungere alla definizione di una classe come **unione** di un insieme di classi ognuna delle quali è **contenuta** nella classe da definire.

N.B. 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.

Proseguiamo con il nostro esempio.

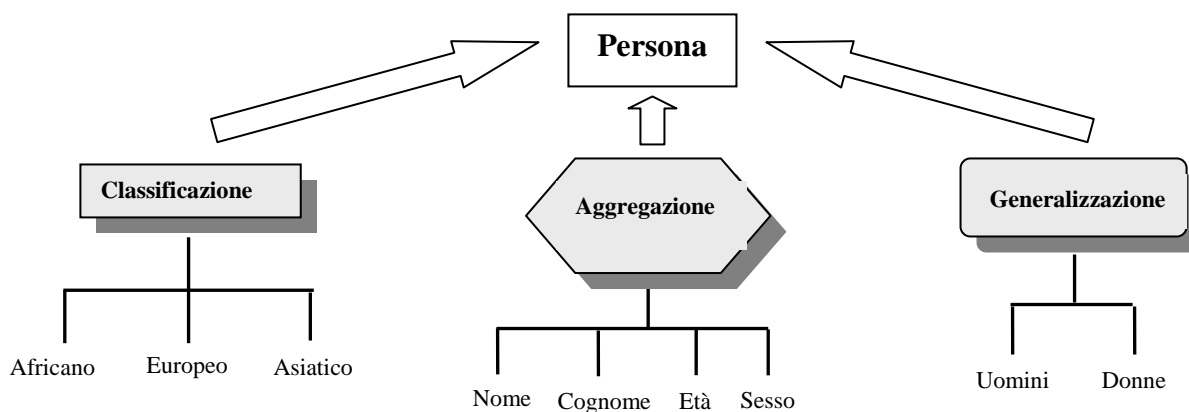
*Consideriamo le seguenti classi di partenza: Uomini e Donne. Cerchiamo ora una classe che è l'unione delle due classi di partenza. Naturalmente ci viene in mente la classe "**Persona**" che è l'unione di queste due classi.*

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 grafico per la descrizione dei dati e delle loro relazioni in una certa realtà di interesse.

E' uno strumento per l'analisi delle caratteristiche 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;*
- *la generalizzazione;*
- *l'aggregazione.*

ENTITA', ISTANZE E LORO ATTRIBUTI

Le **entità** corrispondono a **classi di oggetti** del mondo reale. Esse sono definite specificando le proprietà ossia gli **attributi** dei loro elementi.

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

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, Età, Sesso.

Se consideriamo l'entità "BottigliaDiVino" nel contesto di un'enoteca i suoi possibili attributi sono: Nome, 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 la singola bottiglia di vino dovremmo introdurre un altro attributo (ad esempio un progressivo) 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 un elenco di lunghezza variabile di attributi dello stesso tipo

(Esempio: l'attributo Hobby dell'entità Persona)

Quando modelliamo 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?*

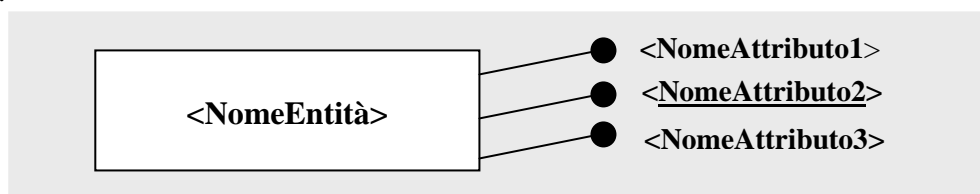
Se la risposta è *affermativa* creiamo una *entità*, *altrimenti* creiamo un *attributo*.

Si indica con il termine **chiave o chiave candidata** l'insieme di uno o più attributi che 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**.

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

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. Ogni associazione tra due entità ha due **versi**.

L'**istanza di una associazione** è una combinazione di istanze delle entità che ne prendono parte.

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

Esempio: Tra l'entità "Persona" e l'entità "Automobile" esiste una 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".

*Quindi si può dire che tra l'entità "Persona" e l'entità "Automobile" esiste l'associazione **Possiede** mentre tra l'entità "Automobile" e l'entità "Persona" esiste l'associazione **E'PossedutaDa***

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.

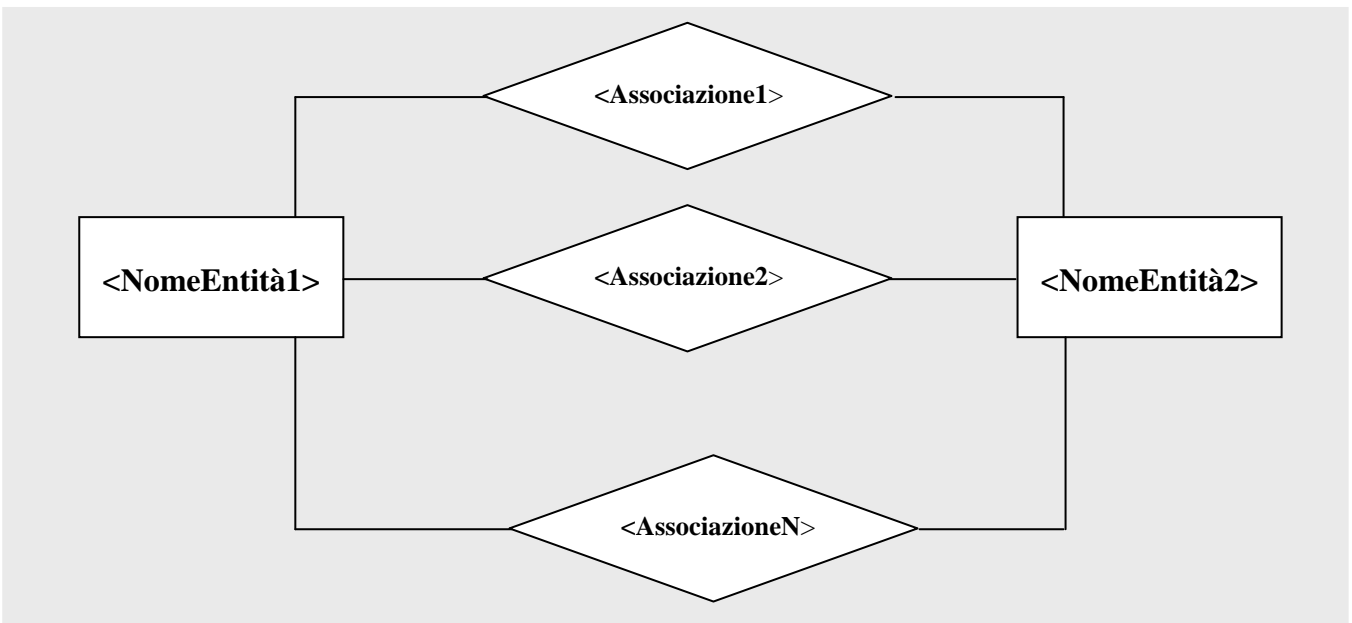


Il rombo posto a metà della linea che unisce le due entità rappresenta la loro associazione. Per gli eventuali *attributi dell'associazione* si seguono le stesse regole degli attributi di una entità. Per *convenzione* utilizziamo:

- *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 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.



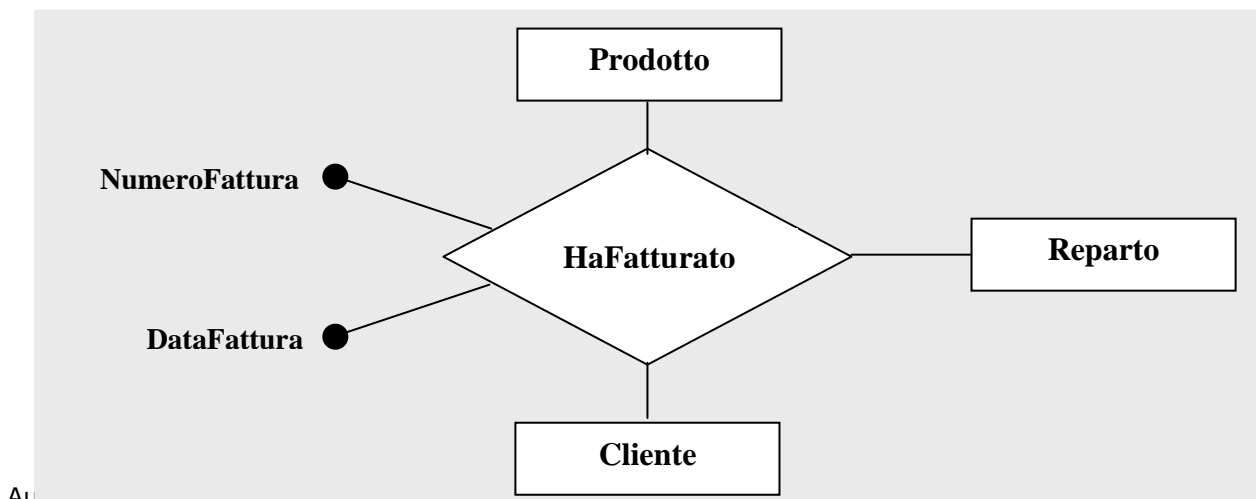
Rappresentazione grafica di associazioni multiple n-arie

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

Per spiegare bene ciò opportuno utilizzare un esempio esplicativo.

Esempio: consideriamo la realtà di un supermercato in cui si vuole rappresentare il fatto che ogni reparto possa emettere fattura per i clienti.

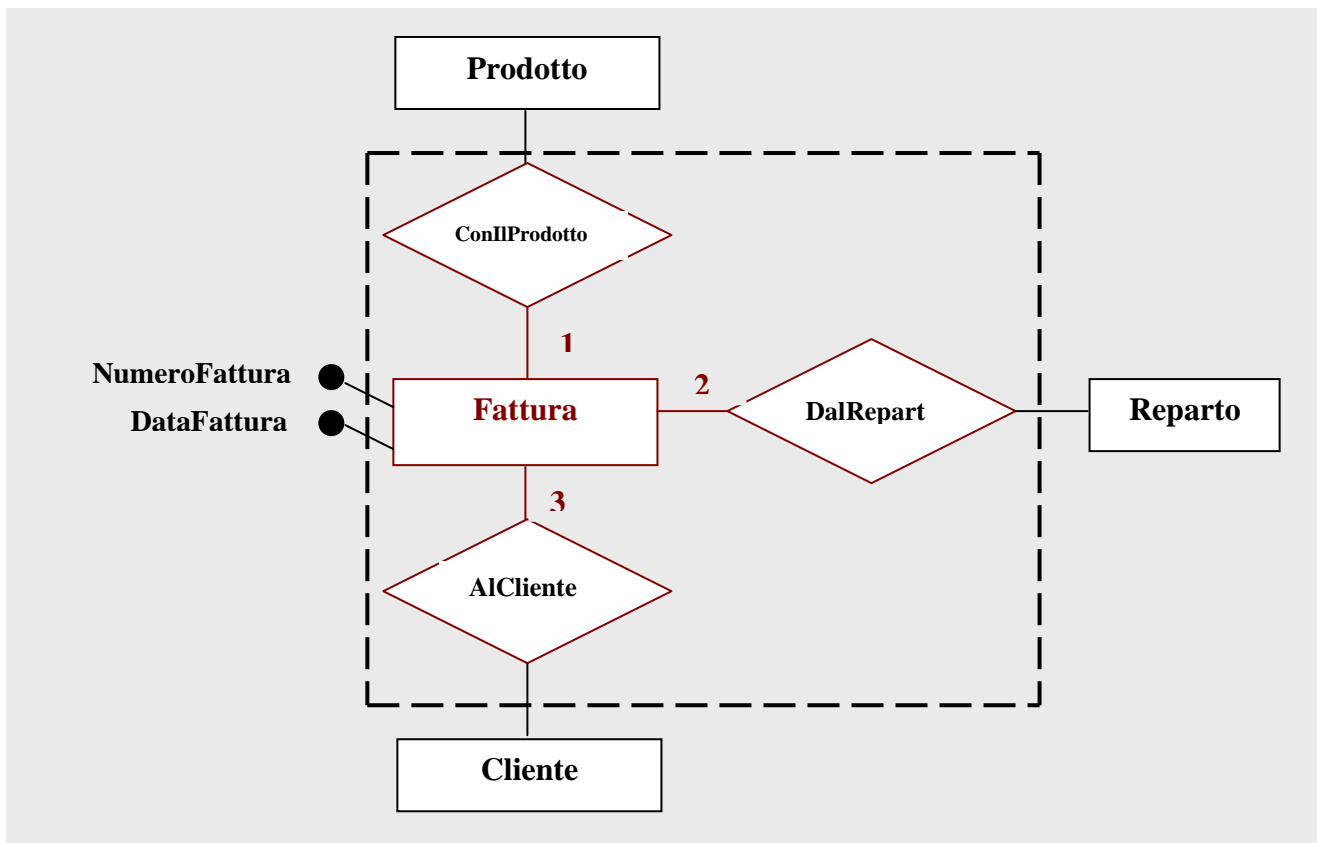
E' possibile utilizzare l'associazione ternaria (ossia che coinvolge tre entità) HaFatturato così schematizzabile:



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.

N.B. 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.*



Contenuto informativo di un diagramma ER

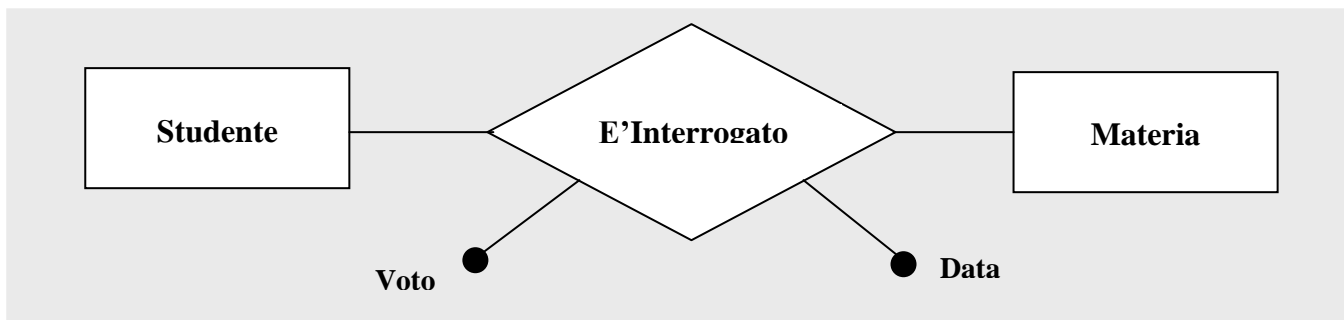
Come è facilmente possibile provare con un esempio, data una stessa realtà di interesse, è possibile descriverla con diversi schemi del modello ER; l'importante è che essi abbiano *il medesimo contenuto informativo*.

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'interrogazione che possiamo fare sul diagramma **D2** in grado di fornire la stessa risposta. In questo caso i diagrammi D1 e D2 si dicono **equivalenti**.

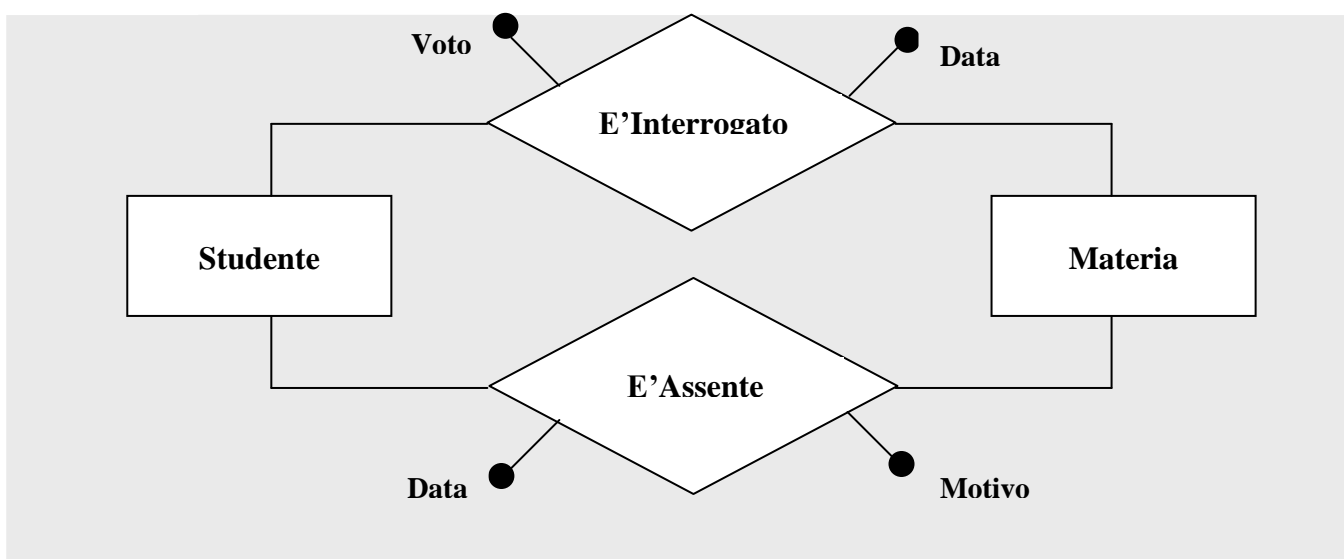
Il concetto che seguiremo d'ora in poi per scegliere quale diagramma ER utilizzare tra due o più modelli equivalenti è quello che privilegia la **chiarezza** ossia la leggibilità, la comprensibilità e l'eshaustività rispetto al contenuto informativo della realtà che modella.

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

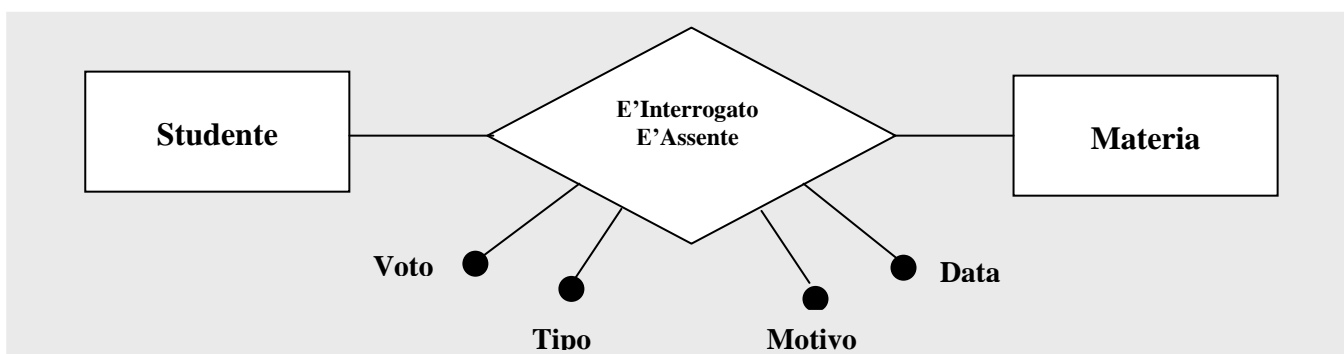
E' possibile utilizzare il seguente diagramma ER



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**

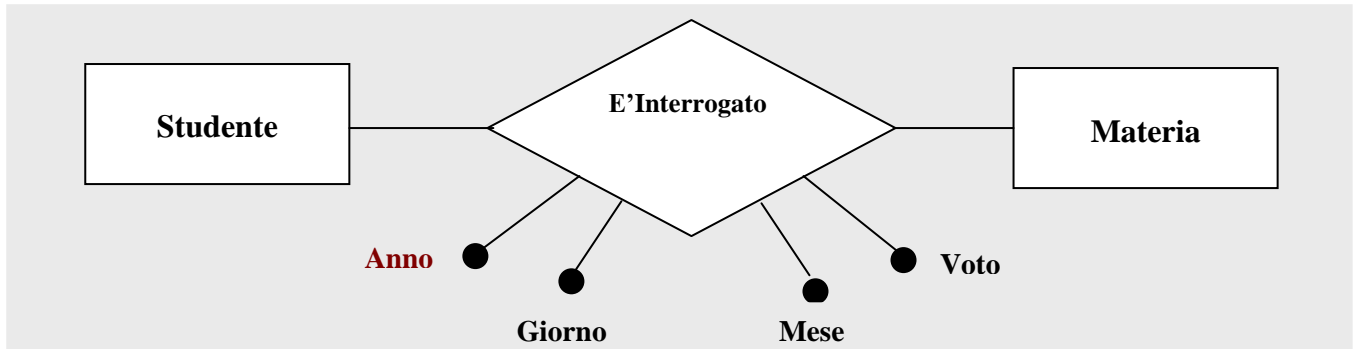


L'attributo **Tipo** indicherà di volta in volta se si tratta di interrogazione o di assenza. In caso si trattasse di assenza l'attributo **Voto** rimarrebbe non valorizzato, mentre se si trattasse di interrogazione l'attributo **Motivo** rimarrebbe non valorizzato.

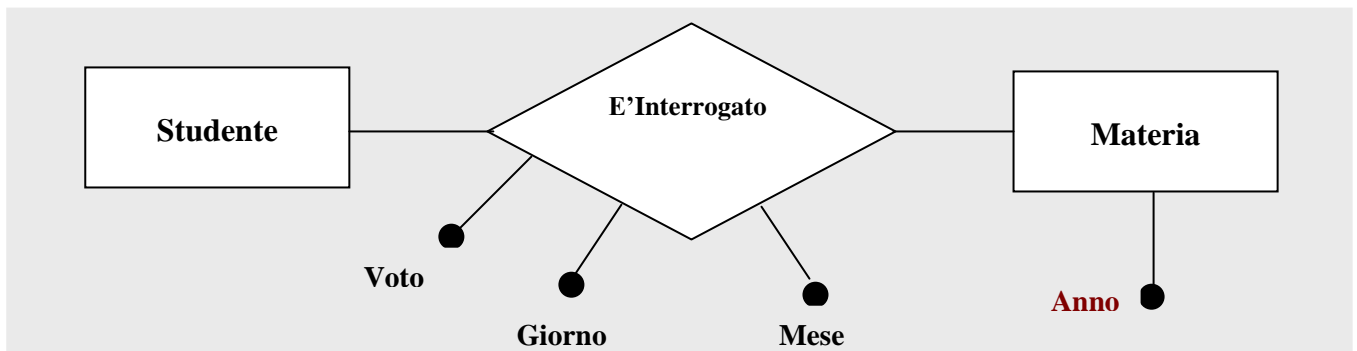
Ipoteticamente posti sul piatto di una bilancia in grado di valutarne il contenuto informativo essi risulterebbero **equivalenti** ma appare del tutto evidente di come il diagramma ER **D1** sia **molto più chiaro** rispetto al diagramma ER **D2** in quanto separa concetti che sono intrinsecamente diversi ("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:

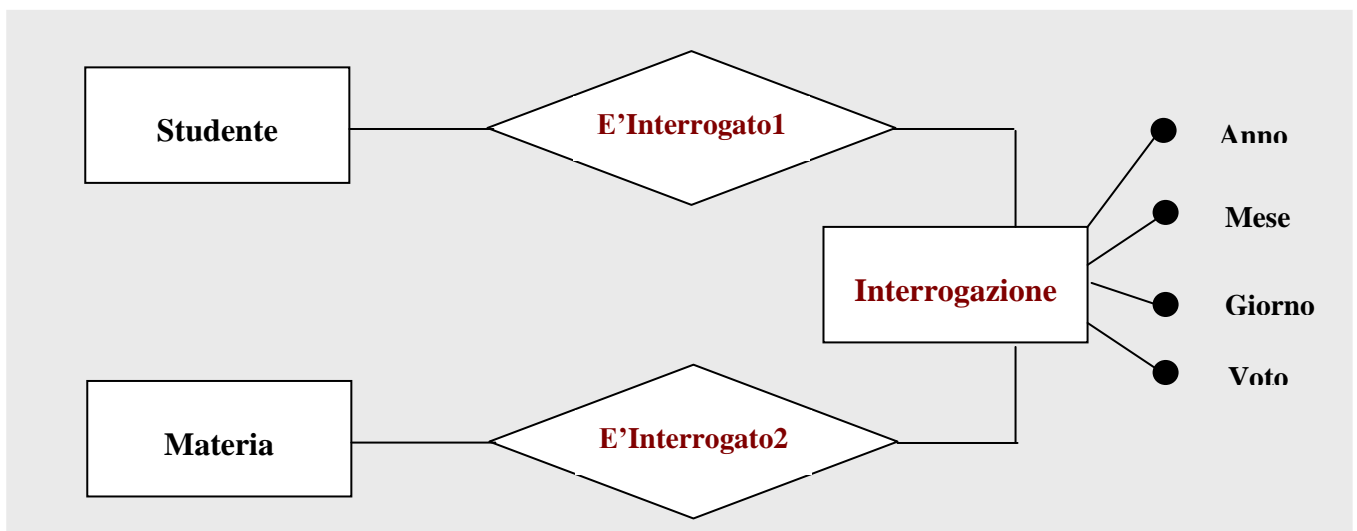


Se ci riferissimo alle interrogazioni di uno stesso anno, ossia se Anno avesse un valore fisso, il diagramma ER precedente diventerebbe:



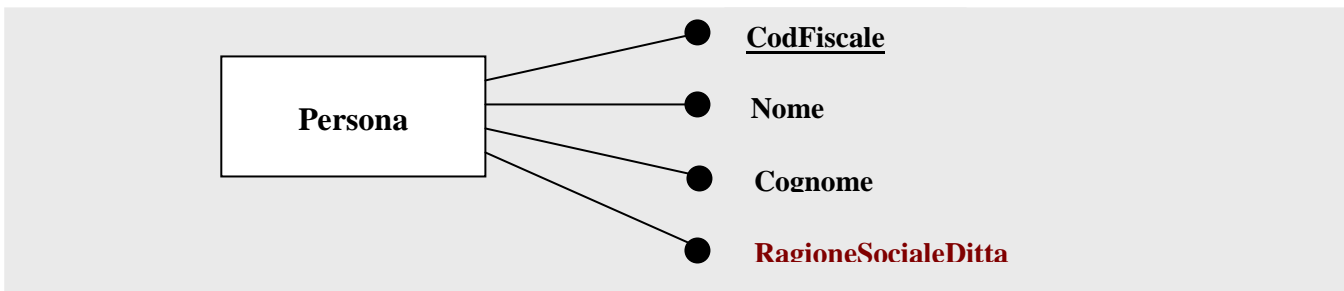
dove l'attributo Anno dell'associazione è diventato un attributo dell'entità Materia.

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 *Persona* caratterizzato tra gli altri dall'attributo *RagioneSocialeDitta*:

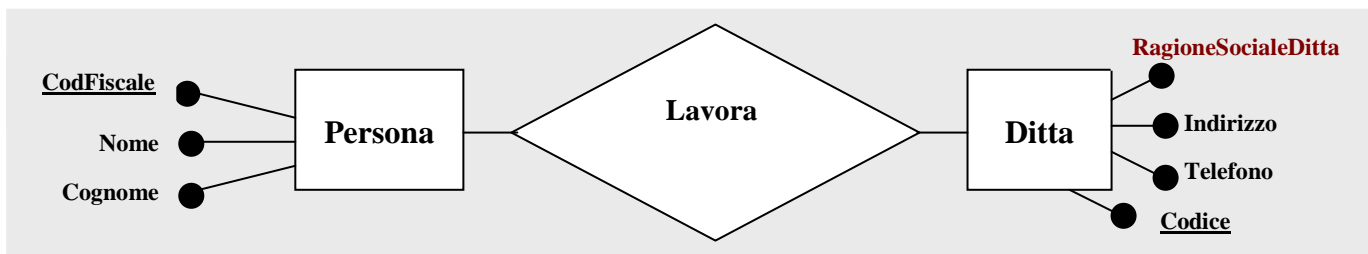


Se siamo interessati a conoscere solo la ragione sociale della ditta presso cui è impiegata quella persona allora l'attributo *RagioneSocialeDitta* è sufficiente.

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 *Persona*.

Questa soluzione non è assolutamente conveniente i quanto replicherebbe gli stessi valori per gli attributi aggiunti relativi alla ditta per tutte le istanze dell'entità *Persona* 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.

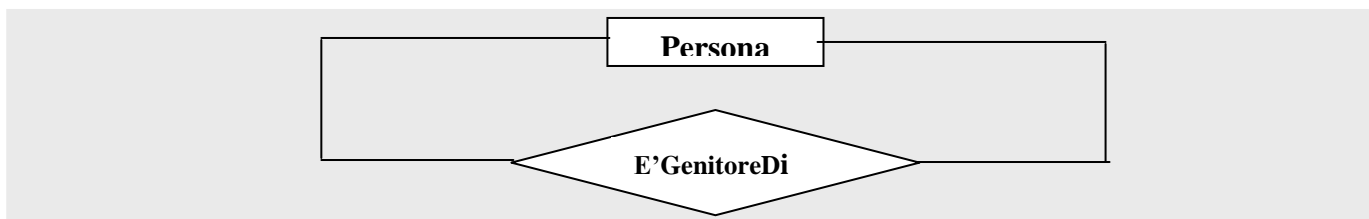
La soluzione conveniente da adottare in tale caso è quella di aggiungere un'altra entità, che nel nostro caso chiamiamo *Ditta*, da mettere in relazione con l'entità *Persona* la quale sia caratterizzata dai seguenti attributi:



Rappresentazione di un'associazione sulla stessa entità: i ruoli

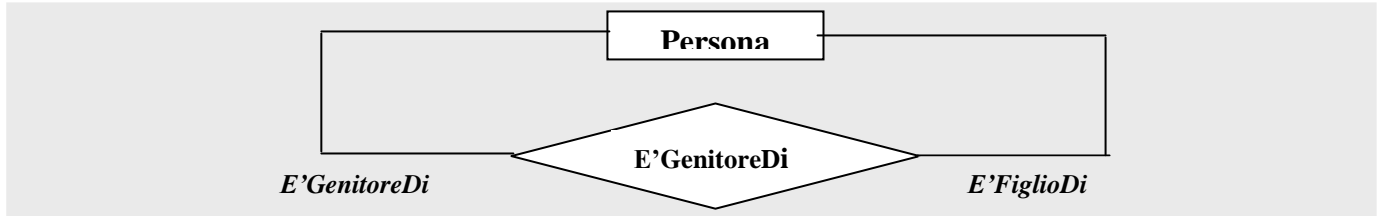
Quando un'associazione crea un legame tra due entità identiche (ossia la stessa entità) si ha il particolare caso di associazione sulla stessa entità.

Ad esempio consideriamo l'entità *Persona* e supponiamo di volere rappresentare l'associazione *E'GenitoreDi*. Questo classico esempio di associazione si rappresenta graficamente così:



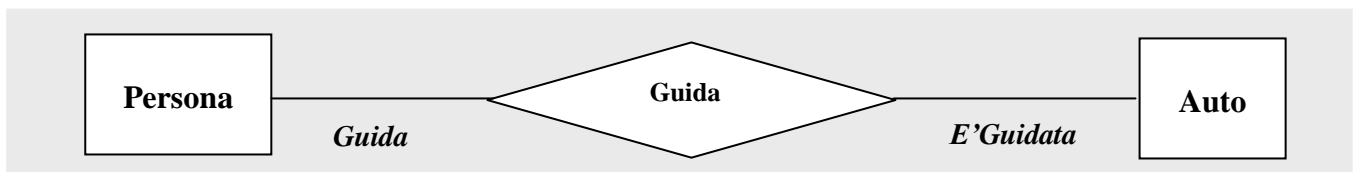
Poiché come abbiamo detto un'associazione è una *classe di fatti* che collegano due o più *entità* dobbiamo distinguere *in ogni fatto* che costituisce un'istanza dell'entità *Persona*, chi è il genitore e chi è il figlio.

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



dove *E'GenitoreDi* e *E'figlioDi* sono anche detti **ruolo** dell'entità nell'ambito dell'associazione. In questo diagramma l'entità *Persona* ha 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 da *Persona* ad *Auto* (relativo all'associazione che possiamo chiamare *diretta*) e l'altro da *Auto* a *Persona* (relativo all'associazione che possiamo chiamare *inversa*).

Data un'associazione **A** tra le entità X ed Y, si attribuisce un *verso* da X ad Y che definisce l'associazione **diretta** da X ad Y.
 Definiamo **inversa** l'associazione da Y ad X che indicheremo con A^{-1} .
 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** ossia sono la stessa associazione salvo che per l'ordine delle coppie.

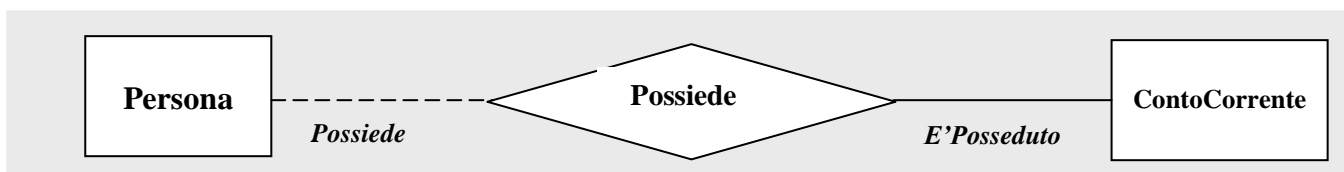
La scelta dunque del verso relativo all'associazione diretta e del verso relativo all'associazione inversa è puramente arbitrario. E' possibile indicare l'associazione diretta e quella inversa con i nomi dei ruoli.

Associazioni totali e parziali

Un'associazione **A** tra le entità X ed Y si dice **totale** quando il legame tra le 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 l'associazione diretta sia totale mentre l'inversa sia parziale oppure viceversa la diretta parziale mentre l'inversa totale

Esempio: Consideriamo il seguente diagramma ER



In tale associazione notiamo che:

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

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

Tipi di Associazioni

Definizione: La **molteplicità** di un'associazione fra le entità X ed Y indica quante istanze dell'entità Y possono trovarsi in relazione con una istanza dell'entità X e viceversa.

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

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

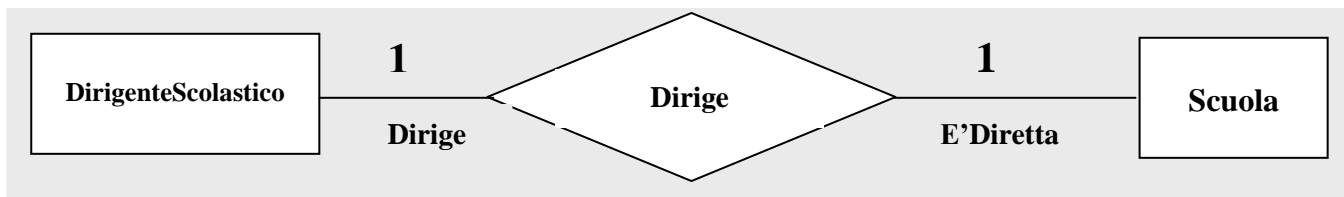
a) **uno a uno** (ossia **1 : 1**) quando ad un'istanza dell'entità A può/deve corrispondere nessuna/una istanza dell'entità B e viceversa quando ad un'istanza dell'entità B può/deve corrispondere nessuna/una istanza istanza dell'entità A

In modo equivalente si può dire che un'associazione binaria è **uno a uno** quando sia l'associazione diretta sia quella inversa sono totali ed univoche

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

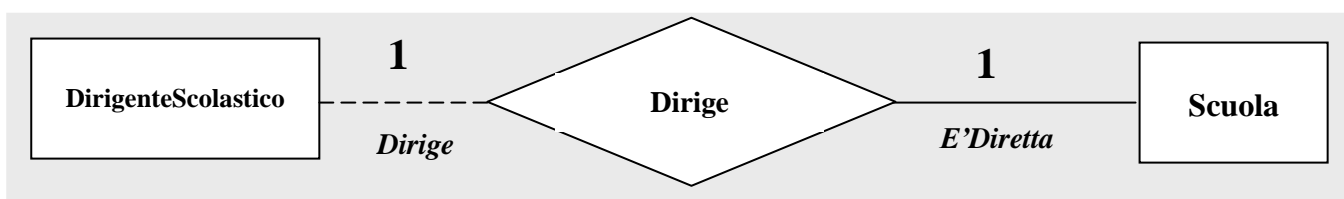


Esempio: un'associazione **uno a uno** è quella che esiste tra le entità *DirigenteScolastico* e *Scuola*. Graficamente esplicitando i ruoli il diagramma ER è il seguente.



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** o **semplice** (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).

In modo equivalente si può dire che un'associazione binaria è **uno a molti** se è multipla l'associazione diretta o l'associazione inversa ma non entrambe.

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).

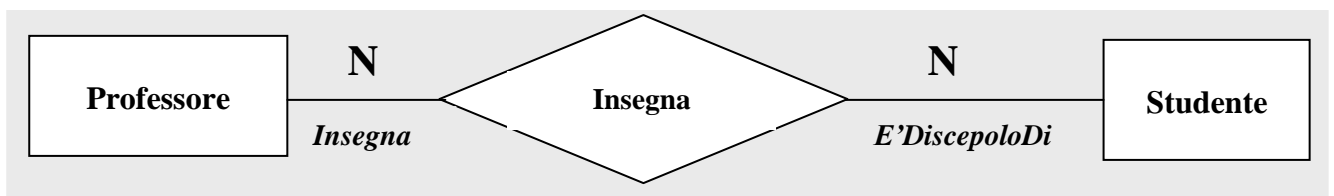
N.B. Per un'associazione di molteplicità N:1 il discorso è l'inverso di quello fatto nel caso di molteplicità 1:N

c) **molti a molti** o **complessa** (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. In modo equivalente si può dire che un'associazione binaria è **molti a molti** quando sia l'associazione diretta sia quella inversa sono totali ed multiple.

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



*Esempio: un'associazione **molti a molti** è quella che esiste tra le entità Professore e Studente. 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*:

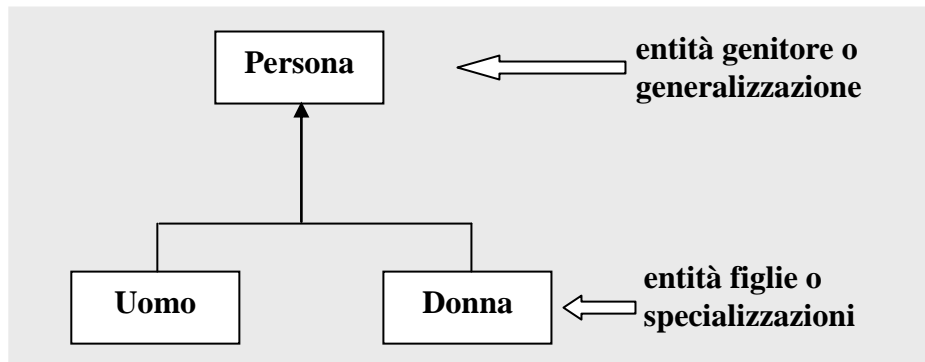
<NomeAssociazione>.<NomeAttributo>

Rappresentazione dell'astrazione per generalizzazione. 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".

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



Ricordiamo che nell'astrazione per generalizzazione le classi Uomo e Donna sono sottoinsiemi della classe Persona. Dobbiamo associare alle entità figlie attributi che non compaiono tra gli attributi della classe genitore.

Per questo introduciamo la definizione di **ereditarietà**.

In una generalizzazione ogni **proprietà** dell'entità padre è anche **proprietà** delle entità figlie e viceversa ogni **proprietà** delle entità figlie è anche **proprietà** delle entità padre dove per *proprietà* intendiamo:

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

L'ereditarietà è uno 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 viene detta **parziale**):

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

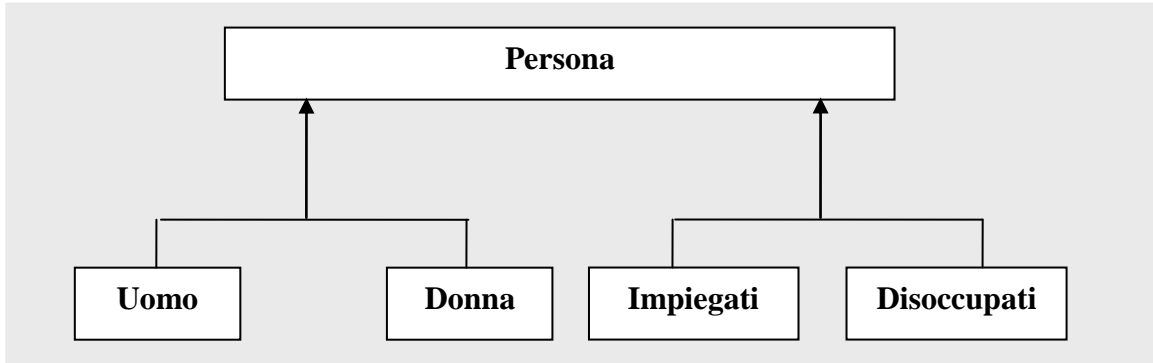
Esempio: La generalizzazione tra *Persona* come *Uomo* e *Donna* è:

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

N.B. Non tutte le generalizzazioni sono totali ed esclusive potendo esserci casi che prevedono tutte le possibili eventualità (né totale né esclusiva, totale ma non esclusiva, esclusiva ma non totale)

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 figlie Uomo e Donna e dall'altra parte dalle entità figlie Impiegati e Disoccupati.



Strategie per la sostituzione di un'ISA all'interno di un diagramma ER

Per tradurre una associazione per generalizzazione nel modello relazionale possiamo seguire tre strade diverse:

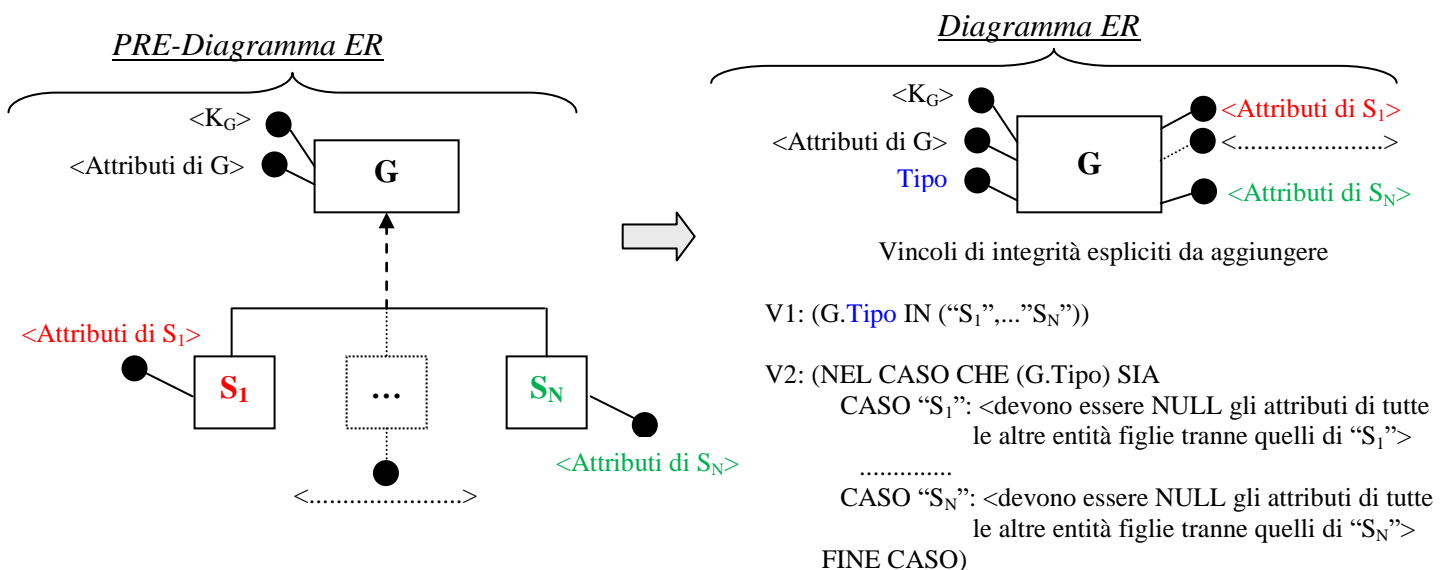
- a) **accorpamento** delle entità figlie nell'entità padre (**Valida qualunque sia l'ISA**);
- b) **accorpamento** dell'entità padre nelle entità figlie (**Valida solo se ISA TOTALE**);
- c) **sostituzione** della generalizzazione con associazioni binarie di tipo 1:1 (**Valida solo se ISA ESCLUSIVA**);

a) Nell'**accorpamento delle figlie nel padre**: le entità S_1, S_2, \dots, S_N vengono eliminate ed i loro attributi e le associazioni cui partecipano, vengono aggiunti all'entità padre G .

Inoltre 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

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.

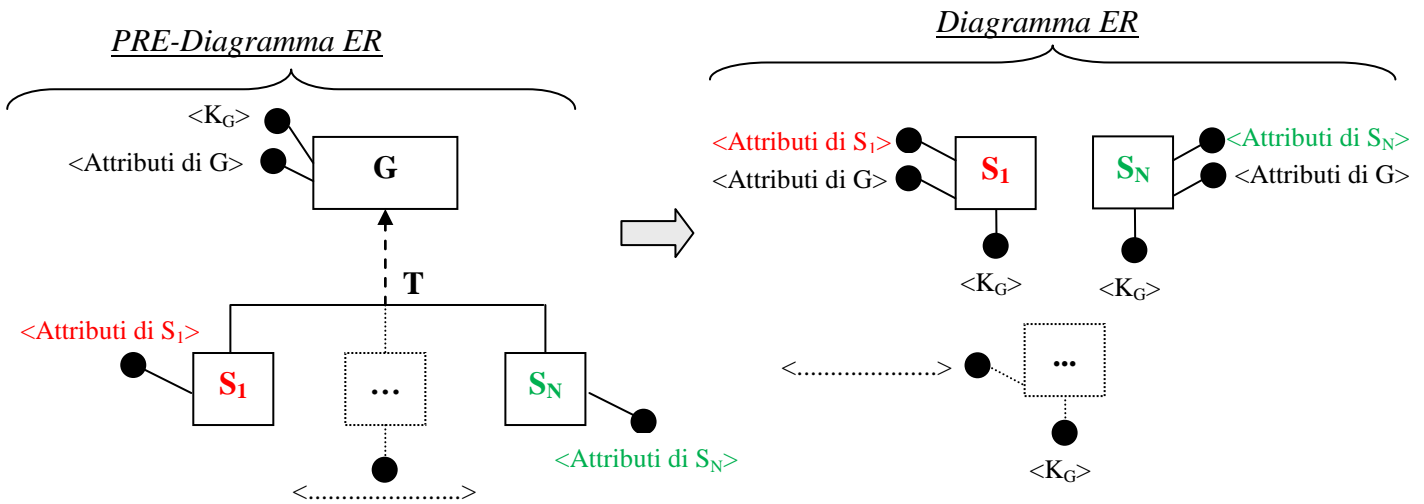
In questo caso avremmo minimizzato gli accessi alla memoria anche se con uno spreco maggiore della stessa visto che vi saranno valori nulli per alcuni attributi di volta in volta



N.B.: Tale traduzione è possibile solo quando la generalizzazione è **TOTALE** ossia se ogni ennupla di G è una ennupla di almeno una figlia tra S_1 o di S_2 o di S_N

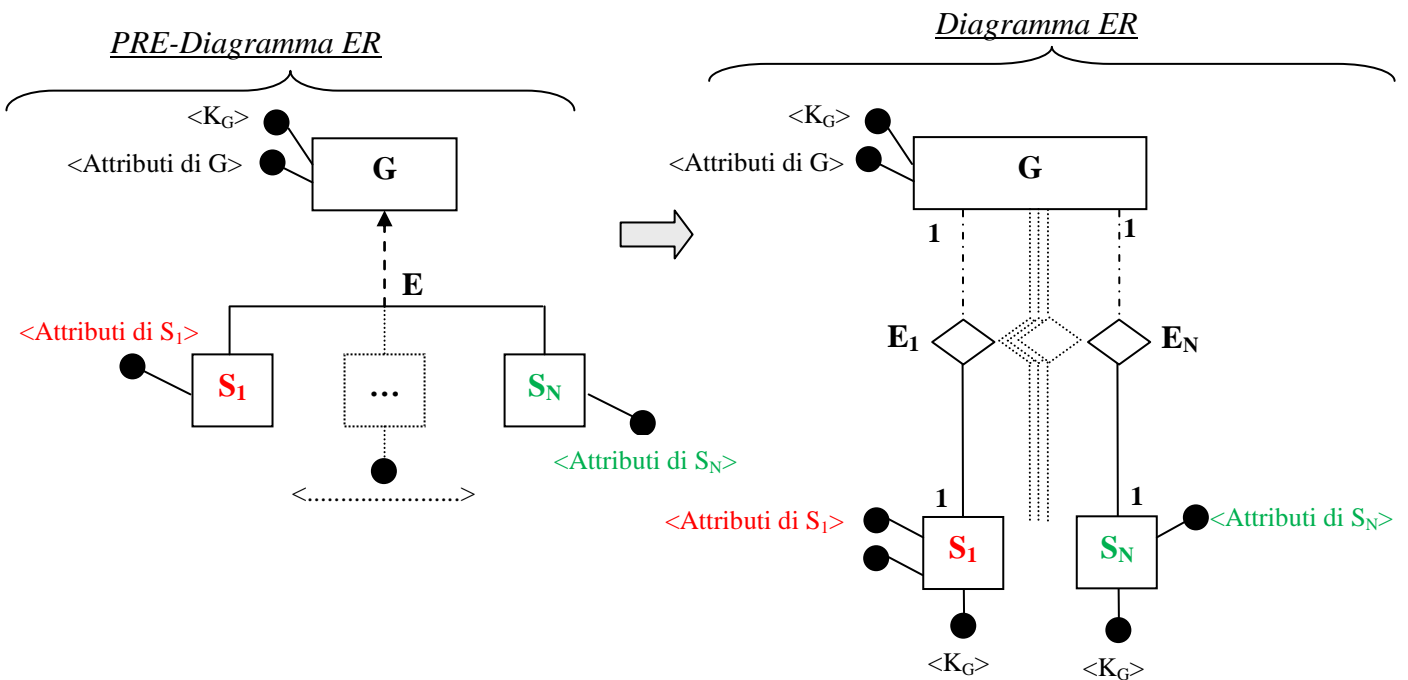
b) Nell'**accorpamento del padre nelle figlie**: l'entità padre G viene eliminata ed i suoi attributi e le associazioni alle quali essa partecipa, vengono aggiunti alle figlie S_1, S_2, \dots, S_N

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 traduzione è possibile solo quando la generalizzazione è **ESCLUSIVA** ossia se ogni ennupla di G è una ennupla di al massimo una figlia tra S_1 o di S_2 o di S_N

c) Nella **sostituzione della generalizzazione con associazioni 1:1**: in questa traduzione si trasforma la generalizzazione in tante associazioni di tipo 1.1 quante sono le entità figlie.



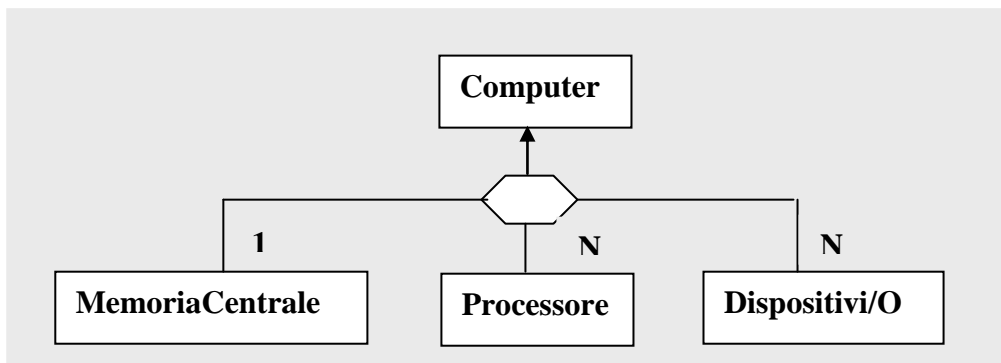
Rappresentazione dell'astrazione per aggregazione. Associazioni HASA

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**

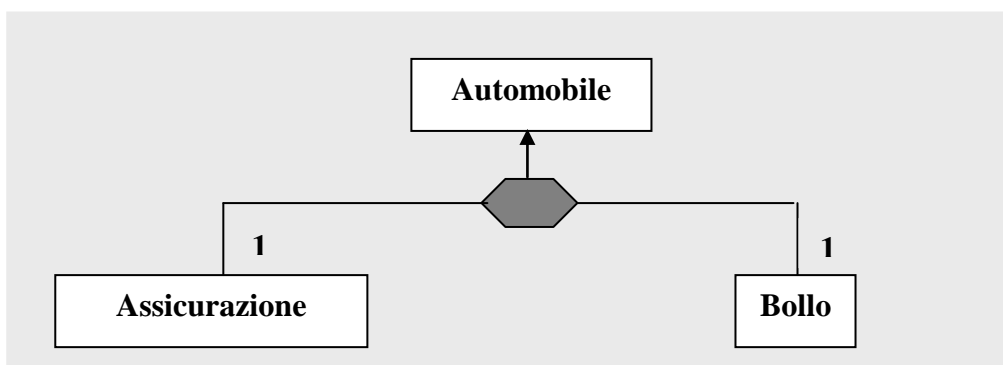
Un'aggregazione **lasca** indica che l'elemento "contenuto" potrà esistere indipendentemente dall'entità "contenitore". Possiamo quindi dire che ad un elemento di un'entità contenuto potrà non corrispondere alcun elemento dell'entità contenitore.

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.

I VINCOLI DI INTEGRITÀ 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** è un'asserzione ovvero un predicato che deve essere soddisfatto dalle istanze delle categorie.

Essi sono di due tipi:

1 – **vincoli di integrità impliciti**: sono quei vincoli imposti dalla stessa struttura dei dati e si dividono a loro volta in:

1.1 – **vincoli di chiave primaria**: questi vincoli impongono che le istanze di una categoria debbano essere tutte diverse tra loro;

1.2 – **vincoli referenziali**: date due entità A e B ed 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 ad un elemento in B. Può anche essere visto come un vincolo sulla totalità dell'associazione.

2 – **vincoli di integrità espliciti**: sono quelli che occorre esplicitare con apposite dichiarazioni (ad esempio il valore dell'attributo *Età* non può essere negativo né maggiore di 120).

I **vincoli di integrità** sono così chiamati perché devono essere rispettati pena l'integrità stessa dei dati (se ad esempio l'attributo *Età* assumesse valori negativi o maggiori di 120 ciò potrebbe causare grossi problemi al corretto funzionamento dei programmi che lo utilizzano).

Riguardo la rappresentazione grafica dei vincoli di integrità

1 – i **vincoli di integrità impliciti**: si rappresentano direttamente con simboli del diagramma ER e precisamente:

1.1 – i **vincoli di chiave primaria**: si rappresentano **sottolineando** i relativi attributi;

1.2 – **vincoli referenziali**: si rappresentano con le **linee continue delle associazioni** tra le entità (rappresentazione della totalità di un'associazione).

2 – i **vincoli di integrità espliciti**: per rappresentare tale tipo di vincoli ricorriamo ad una nostra sintassi che utilizzi la seguente pseudoistruzione

V<NumProg>: (<Espressione>)

dove

<NumProg> è un numero progressivo del vincolo

<Espressione> è una qualsiasi espressione in pseudolinguaggio naturale che specifichi il vincolo

Esempio:

Per esprimere il vincolo sull'Età prima ipotizzato potremmo scrivere:

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)

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.

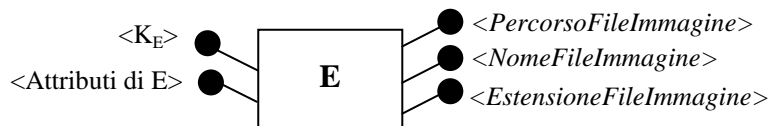
Supponiamo allora che:

- 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*).

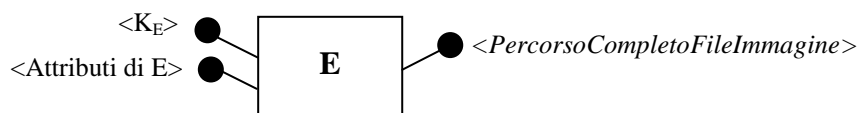
Esempio: Se all'entità E occorre salvare sul database un'immagine relativa al file "C:\immagini\profilo.jpg" possiamo gestire tale informazione attraverso i seguenti tre attributi:



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"