PROGETTAZIONE DI BASI DI DATI

1. Preliminari
2. Modelli concettuali
3. Modello Entity-Relationship
4. Traduzione ER $\rightarrow$ relazionale
5. Metodologia

PRELIMINARI

Progettare una base di dati: definirne il **contenuto** e la **struttura** che esso deve avere

Metodologie di progettazione: le basi di dati sono sempre più **complesse** e sofisticate $\Rightarrow$ è necessario un approccio sistematico

Fasi della progettazione:

1. raccolta ed **analisi dei requisiti**
2. progettazione **concettuale**
3. progettazione **logica**
4. progettazione **fisica**

Ogni fase si basa su un **modello**

FASI DELLA PROGETTAZIONE

1. *raccolta ed analisi dei requisiti*

   - requisiti **informativi**: caratteristiche dei dati

   - requisiti sui **processi**: operazioni sui dati

   - requisiti sui **vincoli di integrità**: proprietà dei dati e delle operazioni

   - **disambiuguazione** delle specifiche dell'utent

2. *progettazione concettuale*

   - **modello concettuale**: descrizione ad alto livello indipendente dall'implementazion

   - risultato: **schema concettuale**

3. *progettazione logica*

   - traduzione dello schema concettuale nel modello dei dati del DBMS

   - risultato: **schema logico** nel DDL del DBMS

4. *progettazione fisica*

   - strutture di memorizzazione e accesso ai dati

   - risultato: **schema fisico**

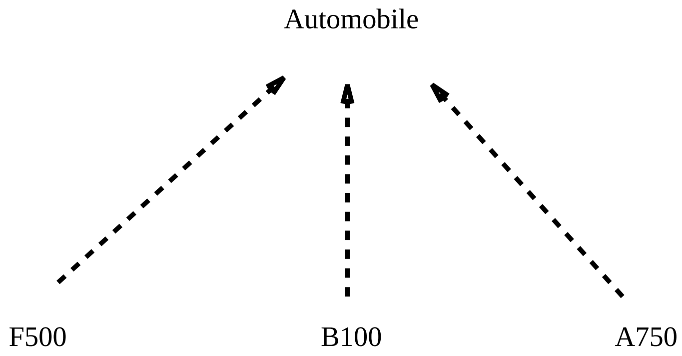
MODELLI CONCETTUALI

Concetto di astrazione: procedimanto mentale che si adotta quando si concentra l'attenzione su alcune caratteristiche di un insieme di entità, trascurando le altre giudicate non rilevanti

4 tipi:

- astrazione di **classificazione**
- astrazione di **aggregazione**
- astrazione di **generalizzazione**
- astrazione di **associazione**

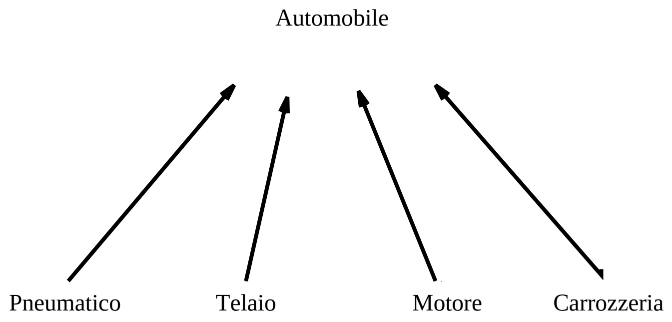
Astrazione di classificazione: definizione di una **classe** a partire da un insieme di oggetti aventi **proprietà comuni**



`F500` è un'**istanza della** classe `Automobile`

Relazione di **instance_of**

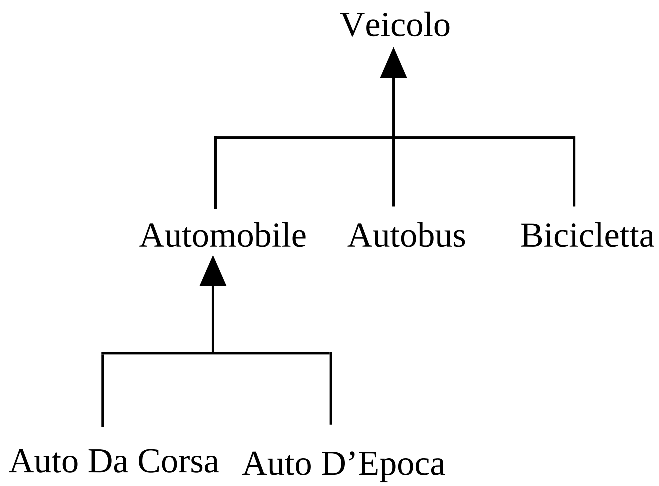
Astrazione di aggregazione: definizione di una classe a partire da un insieme di classi che costituiscono le sue **componenti**



Il `Pneumatico` è una **parte dell'**automobile

Relazione di **part_of**

Astrazione di generalizzazione: definizione di una classe (**superclasse**) da un insieme di classi aventi proprietà comuni (**sottoclassi**)



Le istanze di `Automobile` sono un **sottoinsieme** delle istanze di `Veicolo`

Relazione di **subset_of** o **is_a**

<u>Nota</u>: le sottoclassi possono possedere caratteristiche proprie

Data una classe C, generalizzazione (superclasse) delle sottoclassi $C_1, \ldots, C_n$, si hanno **2** tipi di generalizzazione:

- **totale**: ogni istanza di C è istanza di almeno una classe C_i, $i \in [1, n]$

$$\bigcup_{i=1}^{n} C_i = C$$

- **parziale**: esiste almeno una istanza di C che non è istanza di alcuna classe C_i, $i \in [1, n]$

$$\bigcup_{i=1}^{n} C_i \neq C$$

Esempio: gli aerei sono veicoli, ma non esiste la sottoclasse `Aereo` di `Veicolo`

Data una classe C, generalizzazione (superclasse) delle sottoclassi $C_1, \ldots, C_n$, si hanno altri **2** tipi di generalizzazione:

- **esclusiva**: ogni istanza di C è istanza di non più di una classe C_i, $i \in [1, n]$

$$\bigcap_{i=1}^{n} C_i = \emptyset$$

- **condivisa**: esiste almeno una istanza di C che è istanza di più di una classe C_i, $i \in [1, n]$

$$\bigcap_{i=1}^{n} C_i \neq \emptyset$$

Esempio: la relazione che lega la classe `Persona` alle sue sottoclassi `Uomo` e `Donna` è esclusiva

Astrazione di associazione: definizione di un **collegamento** fra due o piú classi

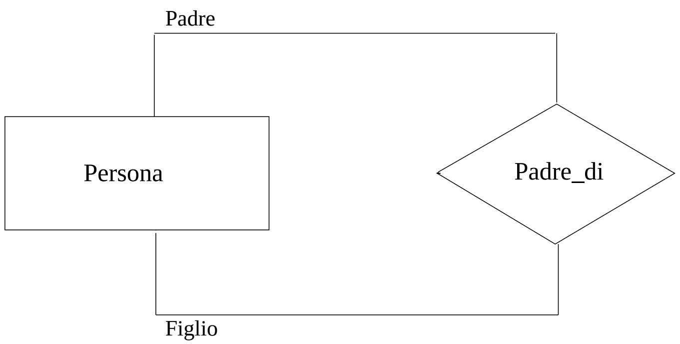
Le associazioni possono avere **proprietà**

Esempio: `data` (in cui si è presa la residenza) può essere una proprietà dell'associazione `Risiede`

Grado: numero di classi che partecipano ad un'associazione

- **unarie**: grado 1
- **binarie**: grado 2
- **n-arie**: grado > 2

Ruolo: funzione che un'entità (istanza di una classe) esercita nell'ambito di un'associazione



Nota: associazione unaria $\Rightarrow$ ruolo obbligatorio

Vincoli di cardinalità: numero minimo e massimo di istanze dell'associazione a cui un'istanza della classe può partecipare

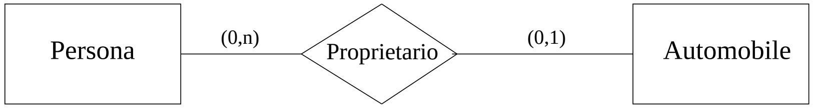
Valori più comuni:

- cardinalità minima (**c_min**): 0, 1
- cardinalità massima (**c_max**): n, ovvero qualunque intero > 1

Data C una classe e A un'associazione:

- **c_min=**0 $\Rightarrow$ esistono istanze di C che non partecipano ad alcuna istanza di A
- **c_min=**1 $\Rightarrow$ ogni istanza di C partecipa almeno ad una istanza di A
- **c_max=**1 $\Rightarrow$ ogni istanza di C può partecipare al più ad una istanza di A
- **c_max=**n $\Rightarrow$ non esiste limite al numero massimo di istanze di A a cui ogni istanza di C può partecipare
- **c_max= c_min=**1 $\Rightarrow$ ogni istanza di C partecipa ad una ed una sola istanza di A
- **c_max=**0, **c_min=**n $\Rightarrow$ ogni istanza di C può partecipare ad un numero qualsiasi di istanze di A

Esempio

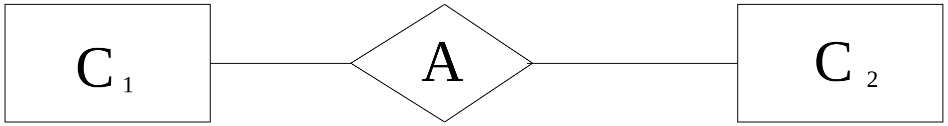


- **c_min** di `Automobile` rispetto a `Proprietario` è 0 ⇒ esistono automobili non possedute da alcuna persona

- **c_min** di `Persona` rispetto a `Proprietario` è 0 ⇒ esistono persone che non posseggono alcuna automobile

- **c_max** di `Persona` rispetto a `Proprietario` è n ⇒ ogni persona può essere proprietaria di un numero arbitrario di automobili

- **c_max** di `Automobile` rispetto a `Proprietario` è 1 ⇒ ogni automobile può avere al più un proprietario



- **c_max** di $\mathtt{C}_1$ e $\mathtt{C}_2$ rispetto ad A è 1 (**c_min**=1) ⇒ `A` è una associazione **uno a uno**

- **c_max** di $\mathtt{C}_1$ rispetto ad A è n e **c_max** di $\mathtt{C}_2$ rispetto ad A è 1, allora A è una associazione **uno a molti**

- **c_max** di $\mathtt{C}_1$ rispetto ad A è 1 e **c_max** di $\mathtt{C}_2$ rispetto ad A è n, allora A è una associazione **molti a uno**

- **c_max** di $\mathtt{C}_1$ e di $\mathtt{C}_2$ rispetto ad A è pari ad n, allora A è una associazione **molti a molti**

Analogo per associazioni unarie dove $\mathtt{C}_1$=$\mathtt{C}_2$ e i ruoli discriminano

MODELLO ENTITÀ-RELAZIONE

Uno dei modelli **più utilizzati** nell'ambito della progettazione concettuale

Ha rappresentazione grafica: **diagramma ER**

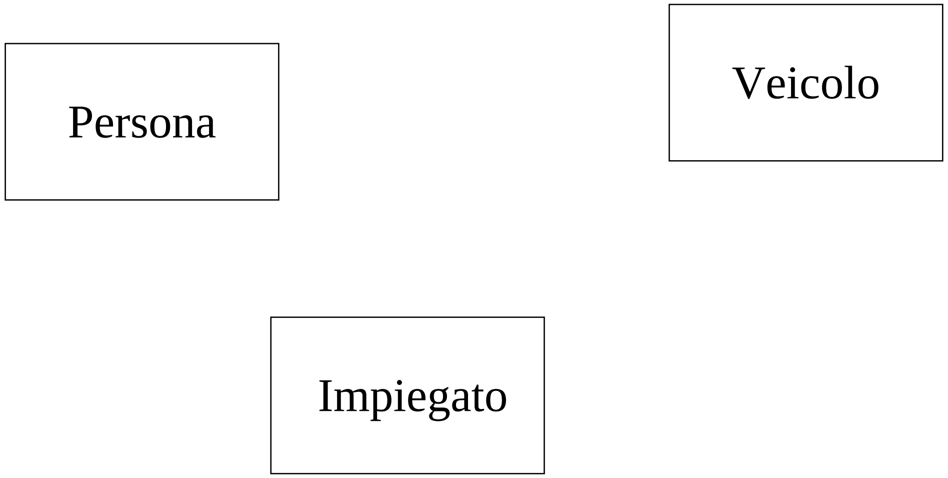
<u>Concetti fondamentali</u>:

- **entità**
- **relazione**
- **attributo**

Entità: insieme di oggetti della realtà che possiedono caratteristiche comuni

Istanze di entità: oggetti appartenenti a una certa entità

Graficamente:



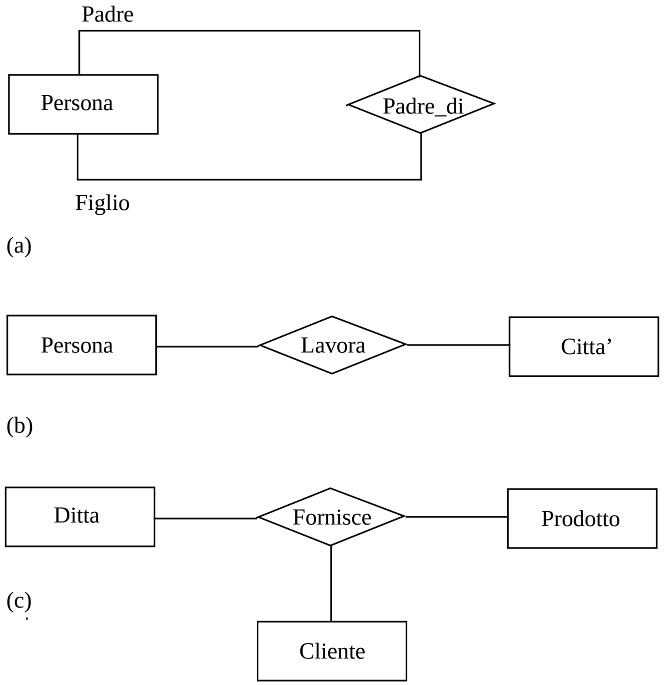
Relazione: legame logico fra entità

Istanze di relazione: combinazione delle istanze delle entità che prendono parte ad una relazione

Graficamente:

p istanza di `Persona`
c istanza di `Città`
(p, c) istanza di `Risiede`

Esempio



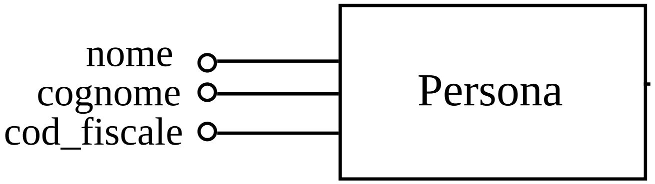
(a) relazione unaria

(b) relazione binaria

(c) relazione n-aria

Attributo: proprietà elementare posseduta da un'entità o da un insieme di relazioni

Graficamente:

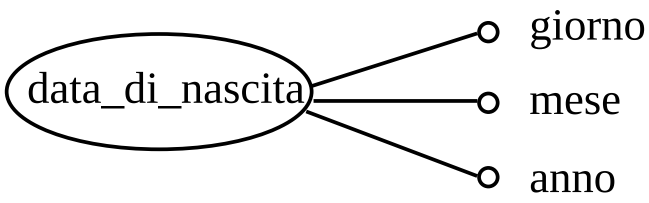


.

`nome`, `cognome`, `cod_fiscale` sono attributi di `Persona`

Un attributo può essere:

- **monovalore**: può assumere al più un valore (`cod_fiscale`)
- **multivalore**: può assumere più valori (`titolo_di_studio`)
- **composito**: possiede dei sottoattributi (`data_di_nascita` con sottoattributi `giorno`, `mese`, `anno`)



Dominio di un attributo: insieme di valori legali per l'attributo

Domini possibili:

- interi, reali, booleani, caratteri
- intervalli di interi e di caratteri
- stringhe di caratteri
- domini definiti dall'utente

<u>Notazione</u>:

- v_i, v_j intervallo compreso fra v_i e v_j
- $(v_i, \ldots v_j)$ insieme di valori possibili $v_i, \ldots, v_j$

Distinguiamo i domini in:

- **semplici**: domini degli attributi non compositi
- **compositi**: domini degli attributi compositi ovvero prodotto Cartesiano degli insiemi di valori associati ai domini componenti

  Se $D = D_1 \times D_2 \times \ldots D_n$ allora $\langle d_1, \ldots, d_n \rangle$ t.c. $d_i \in D_i$ valore possibile

Esempio

Si consideri il diagramma ER della relazione `Risiede` ed in particolare l'entità `Persona`

Dichiarazione di `Persona`:

```
nome: stringa(20)
cognome: stringa(20)
cod_fiscale: stringa(16)
data_di_nascita: giorno × mese × anno
titolo_di_studio: stringa(50)
```

dove i domini `giorno`, `mese`, ed `anno` sono:

`giorno` = 1, ..., 31
`mese` = { Gen,Feb,Mar,Apr,Mag,Giu, Lug,Ago,Set,Ott,Nov,Dic}
`anno` = 1900, ..., 1997

Costrutti di base: entità relazione attributo

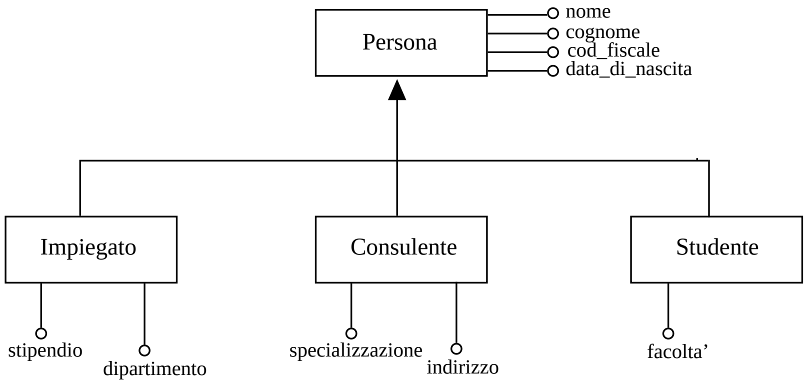
Altri costrutti:

gerarchie di generalizzazione

Generalizzazione: una entità E è una **generalizzazione** delle entità $E_1, \ldots E_n$ se ogni istanza delle entità $E_1, \ldots E_n$ è anche un'istanza di E

E entità **padre**
$E_1, \ldots E_n$ entità **figlie**

<u>Esempio</u>

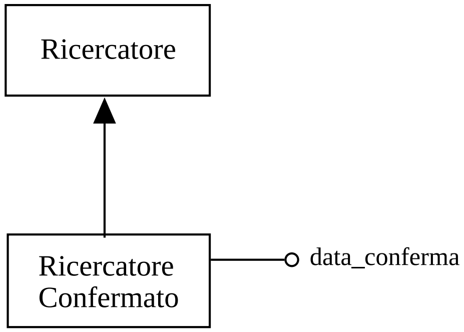


Caso particolare di generalizzazione (parziale ed esclusiva): relazione di **sottoinsieme**

Definire una relazione di sottoinsieme tra una entità E_1 ed una entità E_2 significa specificare che ogni istanza di E_1 è anche istanza di E_2

Ogni entità può avere al più una entità figlia

<u>Esempio</u>



Ricordiamo le gerarchie di generalizzazione di di relazioni

<u>Esempio</u>: `Risiede_temp` sottoinsieme di `Risiede`

Vincoli di integrità:

- **impliciti**: automaticamente verificati dal sistema. Ogni occorrenza di una base di dati relativa ad uno schema ER li deve verificare

- **espliciti**: definiti esplicitamente da chi progetta lo schema ER

  vincoli di **cardinalità**

  vincoli di **identificazione**

Vincoli impliciti:

- ogni istanza di relazione deve riferirsi ad istanze di entità presenti nell'occorenza della base di dati

- istanze diverse della stessa relazione devono riferirsi a differenti combinazioni di istanze delle entità partecipanti alla relazione

- se una entità E_1 è definita come generalizzazione di una entità E_2, l'insieme delle istanze di E_2 deve essere contenuto in quello delle istanze di E_1.

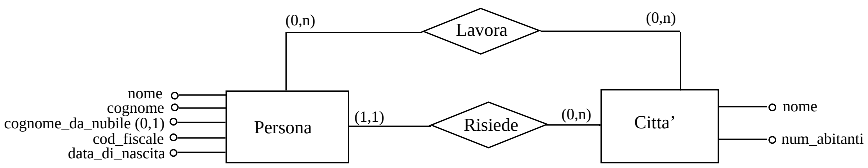
  Vincoli simili esistono anche per gli attributi e le relazioni legate da gerarchie di generalizzazione

Vincoli espliciti di cardinalità:

- su **relazioni**: cardinalità minima e massima sulla partecipazione di una entità ad una relazione. Valori di default: (0,n)

- su **attributi**: numero minimo e massimo di valori dell'attributo che possono essere associati ad una istanza della corrispondente relazione od entità. Valori di default: (1,1)

Esempio



Vincoli espliciti di identificazione: insieme di attributi e/o entità che identificano univocamente le istanze dell'entità. Sono detti **identificatori** o **chiavi**

Possono essere:

- **interni**: uno o più attributi dell'entità

- **esterni**: uno o più entità collegate all'entità a cui si riferiscono

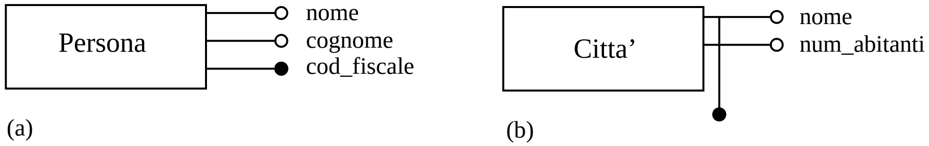
- **misti**: attributi o entità

- **semplici**: un elemento

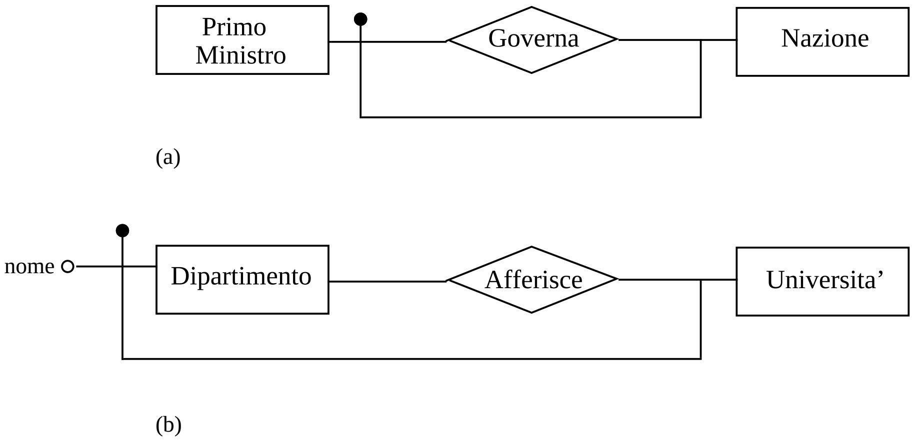
- **compositi**: più di un elemento

- **minimali**: qualsiasi sottoinsieme proprio non è un identificatore

Esempio



(a) identificatore interno semplice
(b) identificatore interno composito



(a) identificatore esterno
(b) identificatore misto

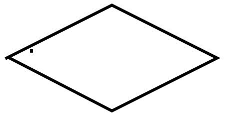
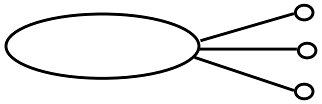
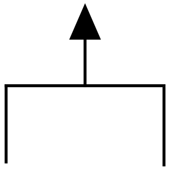
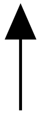
Riassumendo:

- Astrazione di **classificazione**: entità, relazioni, attributi

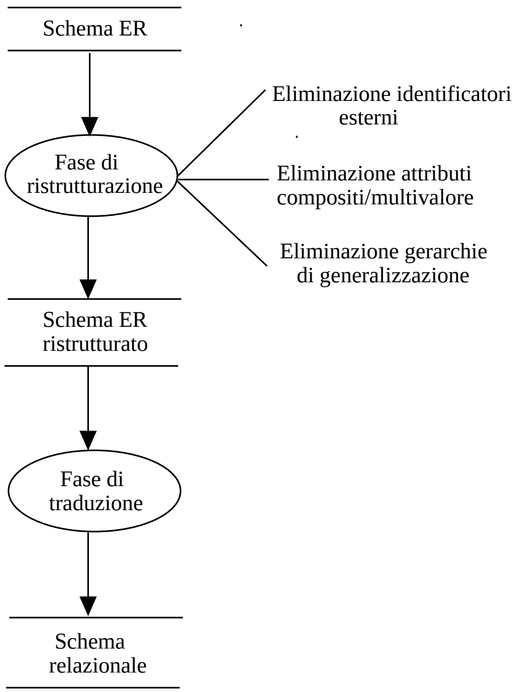
- Astrazione di **aggregazione**:
  - entità come aggregazione di attributi
  - relazione come aggregazione di attributi ed entità
  - attributo composito come aggregazione di sottoattributi

- Astrazione di **generalizzazione**: gerarchie di generalizzazione tra entità relazioni e attributi

- Astrazione di **associazione**: relazione

Componente	Simbolo
Entita’	
Relazione	
Attributo	
Attributo composito	
Gerarchia di generalizzazione	
Relazione di sottoinsieme	
Identificatore	
Vincolo di cardinalita’	(c_min,c_max)

E-R → RELAZIONALE



2 fasi principali:

1. fase di **ristrutturazione**

2. fase di **traduzione**

Fase di ristrutturazione

Eliminazione dallo schema ER di tutti i costrutti che non possono essere direttamente rappresentati nel modello relazionale

1. eliminazione degli identificatori esterni

2. eliminazione degli attributi compositi

3. eliminazione delle gerarchie di generalizzazione

RISULTATO: **schema ER ristrutturato**

Fase di traduzione

Traduzione con **regole di trasformazione** di entità, attributi e associazioni dello schema ER in relazioni del modello relazionale:

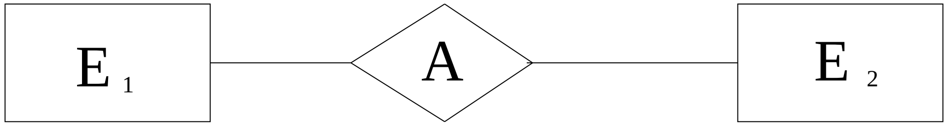
Schema ER $\Rightarrow$ Schema relazionale

entita'	relazioni
attributi	
associazioni	

RISULTATO: **schema relazionale**

Fase di ristrutturazione

1. eliminazione degli identificatori esterni



E_1 ha identificatore (chiave) misto od esterno rappresentato da E_2 attraverso A

2 casi:

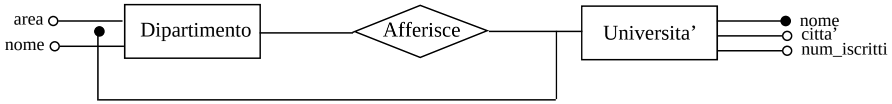
a. E_2 ha un identificatore interno

b. E_2 ha da un identificatore misto od esterno

CASO a: E_2 ha un identificatore interno

L'identificatore di E_1è trasformato in un identificatore interno aggiungendo agli attributi di E_1 l'identificatore interno di E_2. L'associazione A può essere eliminata

Esempio



(a)

(b)

CASO b: E_2 ha da un identificatore misto od esterno

E_2 è a sua volta caratterizzata da un identificatore esterno o misto costituito da E_3

2 casi:

1. E_3 ha un identificatore interno $\Rightarrow$ l'eliminazione dell'identificatore esterno di E_1 avviene come segue:

   - trasformazione dell'identificatore di E_2 in un equivalente identificatore interno

   - trasformazione dell'identificatore di E_1 in un equivalente identificatore interno

   i passi precedenti sono eseguiti con la procedura del caso a)

2. E_3 ha un identificatore esterno o misto $\Rightarrow$ applico ricorsivamente il passo 1 precedente per E_3

2. eliminazione di attributi compositi e multivalore

Perchè? Il modello relazionale consente solo la specifica di attributi semplici e monovalore

2 casi:

a. attributi compositi

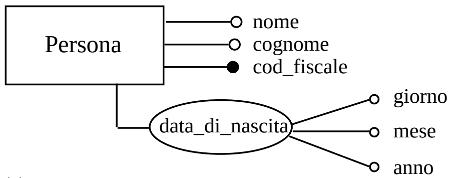
b. attributi multivalore

CASO a: attributi compositi

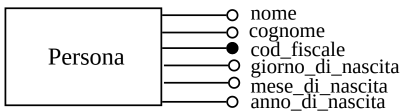
Posso procedere in due modi (ricorsivamente):

1. considerando tutti i sottoattributi come attributi di E
2. eliminando i sottoattributi e considerando l'attributo composito come un attributo semplice (ridefinizione del dominio)

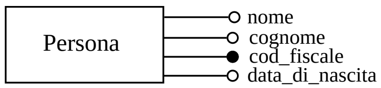
Esempio



(a)



(b)

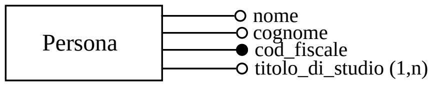


(c)

CASO b: attributi multivalore

Definizione di una nuova entità collegata all'entità di partenza con un'associazione che modella l'attributo multivalore mediante un attributo a valore singolo

Esempio



(a)

(b)

3. eliminazione delle gerarchie di generalizzazione

<u>Perchè</u>? Le gerarchie di generalizzazione non sono supportate dal modello relazionale

E_1 generalizzazione di $E_2, \ldots, E_n$

<u>3 alternative</u>:

a. eliminazione delle entità figlie

b. eliminazione dell'entità padre

c. sostituzione della generalizzazione con associazioni

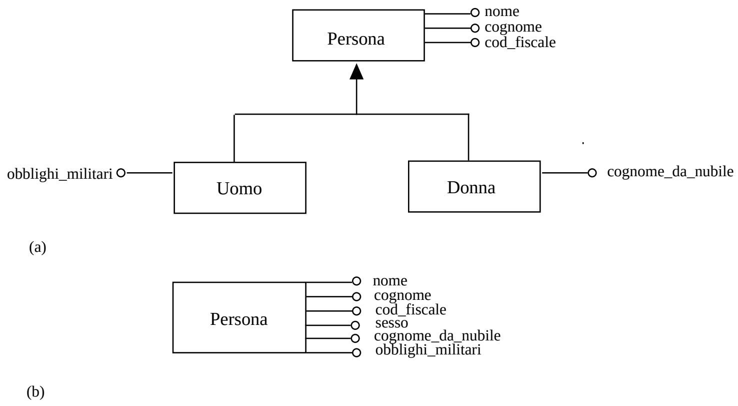
<u>CASO a</u>: eliminazione delle entità figlie

$E_2, \ldots, E_n$ vengono eliminate e i loro attributi vengono inseriti in E_1. Ad E_1 viene aggiunto un attributo a per tenere traccia delle entità figlie

- generalizzazioni totali $\Rightarrow$ a mai nullo

- generalizzazioni parziali $\Rightarrow$ a nullo per istanza dell'entità padre che non appartiene a nessuna istanza entità figlia

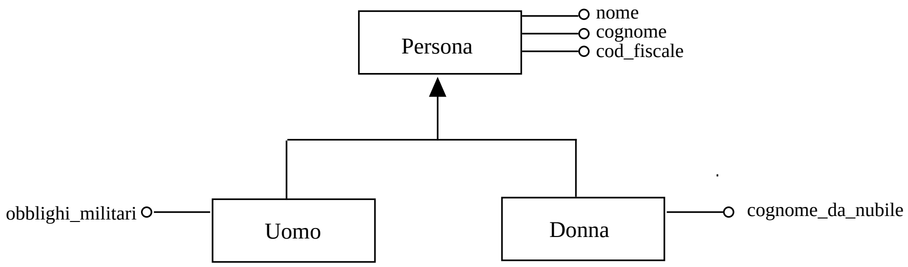
<u>Esempio</u>



CASO b: eliminazione dell'entità padre

E_1 viene eliminata e i suoi attributi vengono inseriti in $E_2, \ldots, E_n$. Solo nel caso di generalizzazioni totali

Esempio



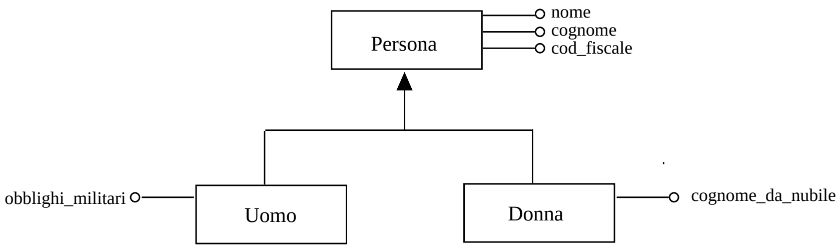
(a)

(c)

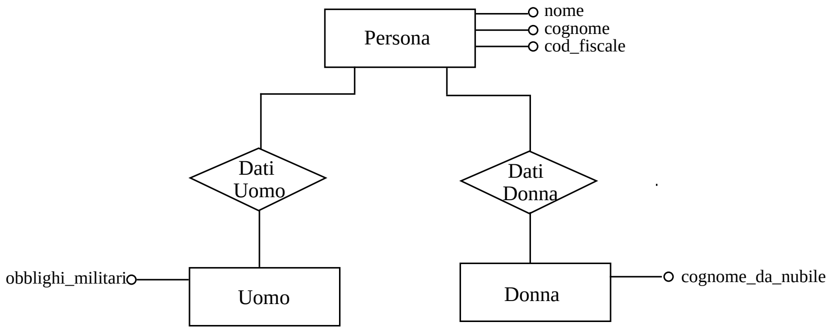
CASO c: sostituzione della generalizzazione con associazioni

$E_1, \ldots, E_n$ invariate. La gerarchia di generalizzazione è sostituita da un'associazione uno a uno, ognuna delle quali lega l'entità padre con una diversa entità figlia

Esempio



(a)



(d)

Quale scegliere?

a. *eliminazione delle entità figlie* ⇒ spreco di memoria per la presenza di valori nulli.

   Conviene: se le operazioni non fannno distinzione fra le varie entità

b. *eliminazione dell'entità padre* ⇒ risparmio di memoria, solo generalizzazioni totali.

   Conviene: se esistono operazioni che si riferiscono ad istanze di una particolare entità figlia

c. *sostituzione della generalizzazione con associazioni* ⇒ risparmio di memoria.

   Conviene: se esistono operazioni che si riferiscono alternativamente a entità padre o figlie

Fase di traduzione

schema ER ⇒ schema relazionale

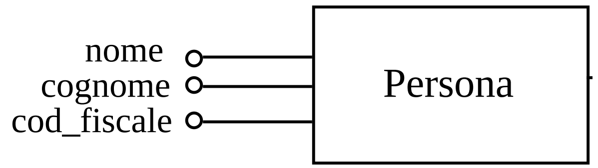
1. Per ogni entità genero una relazione che ha un attributo per ogni attributo dell'entità

entità ⇒ relazione

attributo di entità ⇒ attributo di relazione

identificatore di entità ⇒ chiave di relazione

Esempio:



`Persona(nome, cognome, cod_fiscale)`

2. traduzione delle associazioni dipende da:

- numero di entità partecipanti

- vincoli di cardinalità

2 alternative:

1. l'associazione viene rappresentata inserendo opportuni attributi in una delle relazioni rappresentanti le entità che partecipanti

2. l'associazione stessa viene modellata con una relazione

Vediamo per ogni tipo di associazione

Associazione binaria uno a uno

L'associazione viene modellata mediante attributi inseriti nelle relazioni che modellano le entità partecipanti.

2 casi:

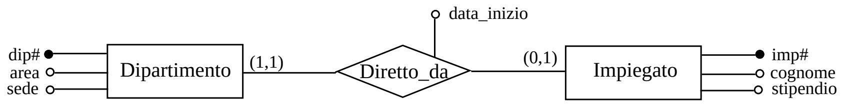
a. partecipazione obbligatoria di una sola entità

b. partecipazione opzionale od obbligatoria di entrambe le entità

CASO a: partecipazione obbligatoria di una sola entità

La relazione che rappresenta l'entità per cui l'associazione è obbligatoria contiene come chiave esterna la chiave della relazione che rappresenta l'altra entità e gli attributi della relazione

Esempio



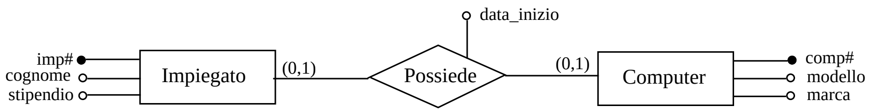
`Dipartimento(Dip#,Area,Sede, Data_inizio,Imp#)` chiave `Dip#`

`Impiegato(Imp#,Cognome,Stipendio)` chiave `Imp#`

CASO b: partecipazione opzionale od obbligatoria di entrambe le entità

Come il caso a, ma la relazione può essere scelta indistintamente

Esempio



1. `Impiegato(Imp#,Cognome,Stipendio, Data_inizio,Comp#)` chiave `Imp#` `Computer(Comp#,Modello,Marca)` chiave `Comp#`

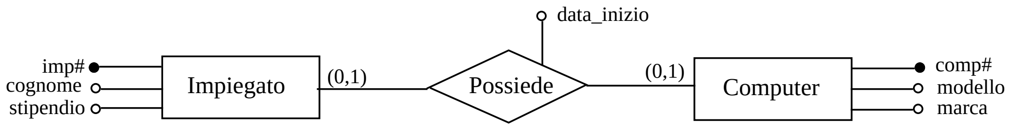
2. `Impiegato(Imp#,Cognome,Stipendio)` chiave `Imp#` `Computer(Comp#,Modello,Marca,Data_inizio, Imp#)` chiave `Comp#`

Caso particolare

Partecipazione di entrambe le entità opzionale
⇒ **relazione nuova** per modellare l'associazione contenente:

- le chiavi delle entità partecipanti
- attributi delle associazioni

Esempio



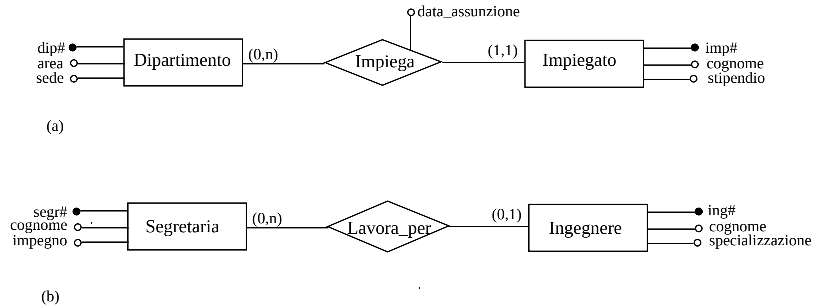
`Impiegato(Imp#,Nome,Stipendio)` chiave `Imp#`
`Computer(Comp#,Modello,Marca)` chiave `Comp#`
`Possiede(Comp#,Imp#,Data_inizio)`, chiave: indifferentemente `Comp#` o `Imp#`

Vantaggio: mai valori nulli
Svantaggio: una relazione in più

Associazione binaria uno a molti

Inserimento nella relazione dell'entità dal lato uno della chiave della relazioni delle entità dal lato n e degli attributi dell'associazione

Esempio



(a) `Dipartimento(Dip#,Area,Sede)` chiave `Dip#`
`Impiegato(Imp#,Cognome,Stipendio, Data_assunzione, Dip#)` chiave `Imp#`

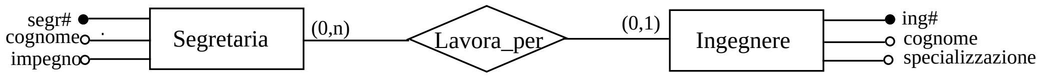
(b) `Segretaria(Segr#,Cognome,Impegno)` chiave `Segr#`
`Ingegnere(Ing#,Cognome,Specializzazione, Segr#)` chiave `Ing#`

Caso particolare

Partecipazione opzionale dell'entità dal lato uno
⇒ **relazione nuova** per modellare l'associazione contenente:

- le chiavi delle entità partecipanti
- attributi delle associazioni

Esempio

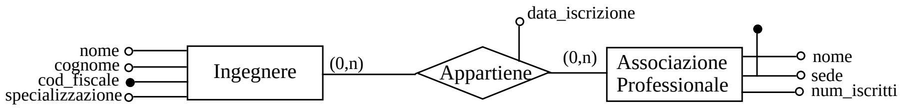


`Segretaria(Segr#,Cognome,Impegno)` chiave `Segr#`
`Lavora_per(Segr#,Ing#)` chiave `Ing#`
`Ingegnere(Ing#,Cognome,Specializzazione)` chiave `Ing#`

Associazione binaria molti a molti

Nuova relazione con attributo le chiavi di entrambe le entità che partecipano all'associazione e gli attributi dell'associazione

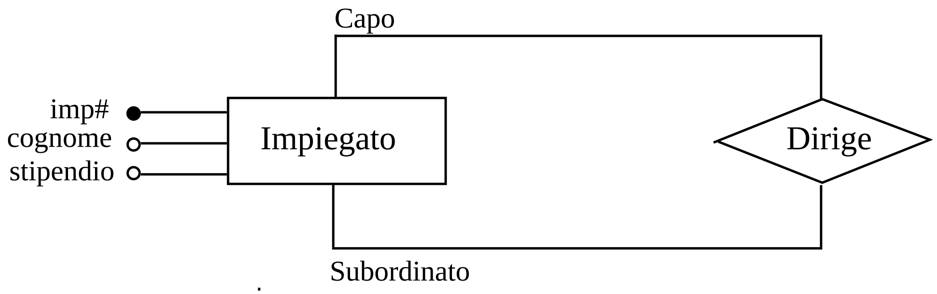
Esempio



`Ingegnere(Nome,Cognome,Cod_fiscale, Specializzazione)` chiave `Cod_fiscale`
`Associazione_Professionale(Nome,Sede, Num_iscritti)` chiave `(Nome,Sede)`
`Appartiene(Nome_ass,Sede_ass, Cod_fiscale_iscritto,Data_iscrizione)` chiave `(Nome_ass,Sede_ass,Cod_fiscale_iscritto)`

Associazione unaria

Come associazioni binarie con attributi distinti per ruoli distinti



- se uno a molti o uno a uno:
  `Impiegato(Imp#,Cognome,Stipendio,Capo#)`
  chiave `Imp#`

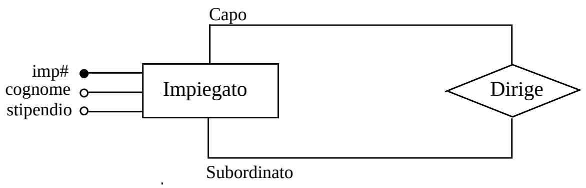
- se molti a molti:
  `Impiegato(Imp#,Cognome,Stipendio)`
  chiave `Imp#`
  `Dirige(Capo#,Subordinato#)`

Relazione nuova nel caso uno a uno e uno a molti, 2 alternative per scegliere la **chiave** della relazione che rappresenta l'associazione:

- se la relazione è uno a uno, la chiave è uno qualsiasi dei due attributi corrispondenti ai ruoli giocati dall'entità nell'associazione

- se la relazione è uno a molti, la chiave è costituita dall'attributo che corrisponde al ruolo dalla parte uno della relazione

Esempio



`Impiegato(Imp#,Cognome,Stipendio)` chiave `Imp#`
`Dirige(Capo#,Subordinato#)`

Per determinare la chiave di `Dirige`:

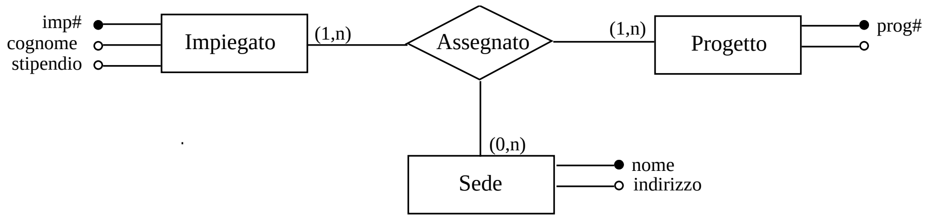
- se un impiegato può avere più capi ed un capo può avere più subordinati, la chiave è `Capo#` e `Subordinato#`
- se un impiegato ha uno ed un solo capo, la chiave può essere indifferentemente `Capo#` o `Subordinato#`
- se un impiegato può avere un solo capo la chiave è `Subordinato#`
- se un impiegato può avere un solo subordinato la chiave è `Capo#`

Associazione n-aria

Nuova relazione contenente:

- chiavi delle entità partecipanti (chiave dell'associazione)
- attributi della relazione

Esempio



`Impiegato(Imp#,Cognome,Stipendio)` chiave `Imp#`
`Progetto(Prog#,Budget)` chiave `Prog#`
`Sede(Nome,Indirizzo)` chiave `Nome`
`Assegnato(Imp#,Prog#,Nome_sede)`
chiave `Imp#,Prog#,Nome_sede`

$\boxed{\text{METODOLOGIA DI TRADUZIONE}}$

2 passi fondamentali:

PASSO 1 generazione delle relazioni corrispondenti alle entità dello schema ER e degli attributi delle relazioni generate

PASSO 2 generazione delle relazioni corrispondenti alle associazioni presenti nello schema ER che non sono state mappate nelle relazioni generate al passo 1

PASSO 1

(a) entità $\Rightarrow$ relazione
attributi entità $\Rightarrow$ attributi relazione
identificatore entità $\Rightarrow$ chiave relazione

(b) associazione A **binaria uno a uno** tra E_1 ed E_2
$\Rightarrow$ nella relazione che rappresenta E_1 si aggiunge:

- chiave di E_2
- attributi di A

E_1 è tale che:

- E_1 partecipa obbligatoriamente
- E_1 ed E_2 obbligatorie $\Rightarrow$ scelta indifferente

(c) associazione A **binaria uno a molti** tra E_1 ed E_2, E_1 dal lato uno
$\Rightarrow$ nella relazione che rappresenta E_1 si aggiunge:

- chiave di E_2
- attributi di A

(d) associazione A **unaria uno a uno** entità E
$\Rightarrow$ nella relazione che rappresenta E si aggiunge:

- uno dei ruoli
- attributi di A

(e) associazione A **unaria uno a molti** entità E
$\Rightarrow$ nella relazione che rappresenta E si aggiunge:

- ruolo del lato molti
- attributi di A

PASSO 2

(a) associazione A **binaria molti a molti** o associazione A **n-aria**
$\Rightarrow$ nuova relazione contenente:

- chiavi delle relazioni delle entità partecipanti o sottoinsieme

- attributi di A

(b) associazione A **unaria molti a molti**
$\Rightarrow$ nuova relazione contenente:

- un attributo per ogni ruolo

- attributi di A