



# Capitolo 6 (Progettazione di basi di dati)

## Progettazione di basi di dati

È una delle attività del processo di sviluppo dei sistemi informatici, va quindi inquadrata in un contesto più generale

### Il ciclo di vita dei sistemi informativi:

- Insieme e sequenzializzazione delle attività svolte da analisti, progettisti, utenti, nello sviluppo e nell'uso dei sistemi informatici
- Attività iterativa, ogni attività dà luogo ad un output, che diventa input di un'attività successiva. Può anche tornare indietro in caso di errore

Studio di fattibilità

Raccolta e analisi dei requisiti

Progettazione

Realizzazione

Validazione e collaudo

Funzionamento

## Fasi (tecniche) del ciclo di vita

**Studio di fattibilità:** definizione di costi e priorità di realizzazione dei vari componenti del sistema

**Raccolta e analisi dei requisiti:** studio delle proprietà e delle funzionalità che il sistema informativo dovrà avere. Richiede un'interazione con gli utenti del sistema e produce una descrizione informale di dati e operazioni

**Progettazione:** si divide in

- *progettazione dei dati:* si definisce la struttura e l'organizzazione dei dati
- *progettazione delle applicazioni:* si definiscono le caratteristiche dei programmi applicativi

Queste descrizioni sono formali con specifici modelli a differenza delle precedenti

Nonostante entrambe queste caratteristiche siano importanti, il ruolo centrale ce l'hanno i dati: i dati sono più stabili

**Realizzazione:** implementazione del sistema informativo secondo le strutture definite in precedenza. E' costruita e popolata la base di dati

**Valutazione e collaudo:** verifica il corretto funzionamento del sistema informativo

**Funzionamento:** il sistema diventa operativo

## Metodologie di progettazione e basi di dati

Per garantire prodotti di buona qualità è opportuno seguire una **metodologia di progetto**.

Una metodologia di progettazione consiste in:

- una **decomposizione** dell'intero progetto in passi successivi indipendenti (articolazione delle attività in fasi)
- una serie di **strategie** da seguire nei vari passi e alcuni **criteri** per la scelta in caso di alternative

- alcuni **modelli di rappresentazione** per descrivere i dati in ingresso e uscita dalle varie fasi

Le proprietà che una metodologia devono garantire sono:

- **Generalità:** rispetto alle applicazioni e ai sistemi utilizzati
- **Qualità del prodotto:** in termini di correttezza, completezza ed efficienza
- **Facilità d'uso:** di strategie e modelli presi in considerazione

I prodotti delle varie fasi sono schemi di alcuni **modelli di dati**:

- **Schema concettuale (Progettazione concettuale):**

Rappresenta le specifiche informali con una descrizione formale.

Lo schema concettuale fa riferimento a un **modello concettuale** dei dati: ci permettono di descrivere l'organizzazione dei dati a un alto livello di astrazione, senza tenere conto degli aspetti implementativi. Si cerca di rappresentare il **contenuto informativo** della base di dati

- **Schema logico (Progettazione logica):**

E' la traduzione del modello concettuale in termini del modello di rappresentazione dei dati adottato dal sistema di gestione di base di dati a disposizione. Lo schema logico fa riferimento a un **modello logico** dei dati: la descrizione dei dati risulta indipendente dai dettagli fisici, ma concreta poiché è disponibile nei sistemi di gestione di basi di dati. Le scelte progettuali riguardano l'ottimizzazione delle operazioni sui dati: si fa uso di tecniche per verificare la

qualità dello schema. Nel caso del modello relazione la tecnica che si usa è la **normalizzazione**

- **Schema fisico (Progettazione fisica):**

Lo schema logico è completato dalla specifica dei parametri fisici di memorizzazione dei dati. Lo schema fisico fa riferimento a un **modello fisico** che dipende dallo sistema di gestione di basi di dati scelto

I requisiti di base di una base di dati sono:

- **Specifiche sui dati** → Riguardano il contenuto della base di dati
- **Specifiche sulle operazioni** → Riguardano l'uso che utenti e applicazioni fanno della base di dati

A seconda del tipo di progettazione si usano uno o l'altra specifica, o entrambe

## Modello dei dati

Insieme di costrutti utilizzati per organizzare i dati di interesse e descriverne la struttura in modo che essa risulti comprensibile a un elaboratore

Componenti fondamentale sono **meccanismi di strutturazione** (o **costruttori di tipo**): Permettono di definire nuovi tipi sulla base di tipi predefiniti (elementari) e costruttore di tipo

Come nei linguaggi di programmazione esistono meccanismi che permettono di definire nuovi tipi, ogni modello dei dati prevede alcuni costruttori

Ad esempio, il **modello relazionale** prevede il costruttore **relazione**, che permette di definire insiemi di record omogenei

**Relazione**: è rappresentata per mezzo di una tabella, le cui righe rappresentano specifici record e le cui colonne corrispondono ai campi dei record

(scriverlo prima)

## Schemi e istanze

In ogni base di dati esistono:

- Lo **schema**, sostanzialmente invariante nel tempo, che ne descrive la struttura poiché costituisce le *caratteristiche dei dati* (aspetto intensionale)
  - Nel modello relazionale, le intestazioni delle tabelle
- L'**istanza**, i *valori attuali*, che possono cambiare anche molto rapidamente nel tempo (aspetto estensionale)
  - Nel modello relazionale, il "corpo" di ciascuna tabella, quindi le righe della tabella

(scriverlo prima)

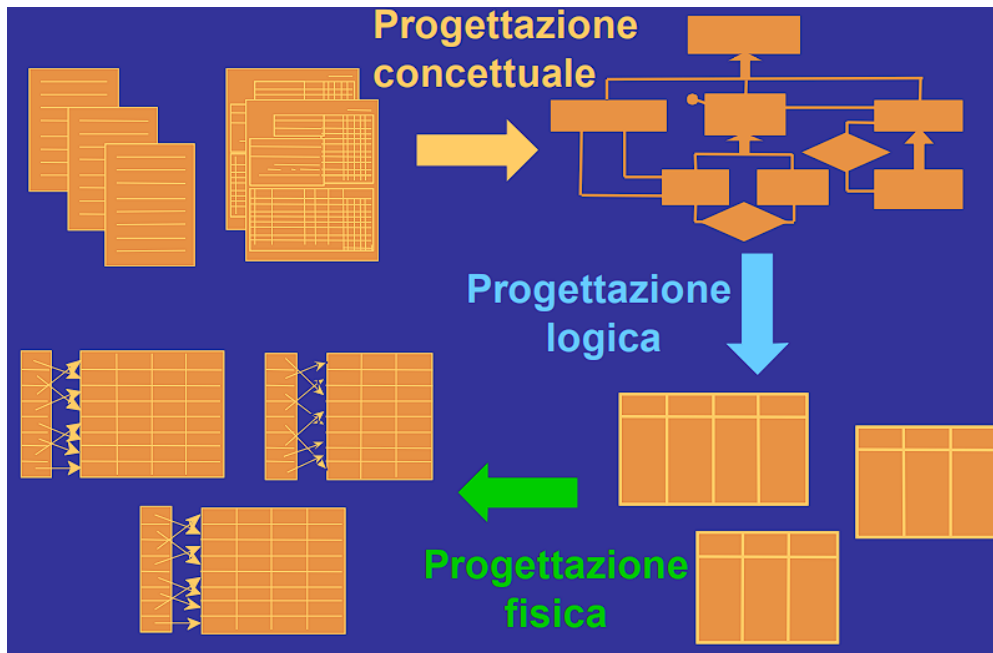
## Due tipi principali di modelli

**Modelli logici:** utilizzati nei DBMS esistenti per l'organizzazione dei dati

- Utilizzati dai programmi
- Indipendenti dalle strutture fisiche
- Esempi: **relazionale**, reticolare, gerarchico, a oggetti

**Modelli concettuali:** permettono di rappresentare i dati in modo indipendente da ogni sistema

- Cercano di descrivere i concetti del mondo reale
- Sono utilizzati nelle fasi preliminari di progettazione
- Il più noto è il modello **Entità-Relazioni (Entity-Relationship)**



## Modelli concettuali

I modelli concettuali servono per ragionare sulla realtà di interesse, indipendentemente dagli aspetti realizzativi

Permettono di rappresentare le classi di dati di interesse e le loro correlazioni definendo **costrutti** che prescindono dai criteri di organizzazione dei dati

I costrutti vengono utilizzati per definire **schemi** che descrivono l'organizzazione e la struttura delle **occorrenze** (da non confondere con l'istanza (insieme di tuple) nel modello relazionale) dei dati, cioè i valori assunti dai dati al variare del tempo

Per ogni costrutto è presente una determinata rappresentazione grafica (utile anche per la documentazione e comunicazione)

## Modello Entità-Relazione

Il modello Entità Relazione è il modello concettuale più diffuso

Ne esistono molte versioni, (più o meno) diverse l'una dall'altra

Costrutti del modello E-R:

- Entità
- Associazione
- Attributo
- Identificatore
- Generalizzazione
- ...

Uno schema E-R, graficamente:



## Entità



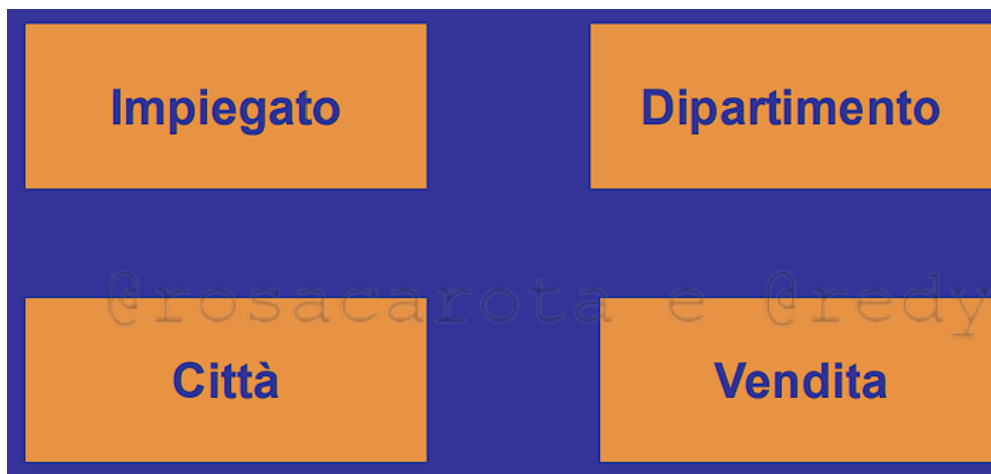
**L'entità** è una classe di oggetti (fatti, persone, cose) della applicazione di interesse con proprietà comuni (omogenei) e con esistenza "autonoma"

Ogni entità ha un **nome** che la identifica univocamente

Esempi: impiegato, città, conto corrente, ordine, fattura

- **Occorrenza** (o **istanza**) di entità:
  - Elemento della classe (l'oggetto, la persona, ..., non i dati)
- Nello schema concettuale rappresentiamo le **entità**, non le singole occorrenze ("astrazione") (in poche parole l'occorrenza di entità non è un valore che identifica l'oggetto, ma è l'oggetto stesso (per esempio non è il codice fiscale dell'impiegato che lo identifica, ma è l'impiegato in carne ed ossa)

Rappresentazione grafica di entità:



## Associazioni



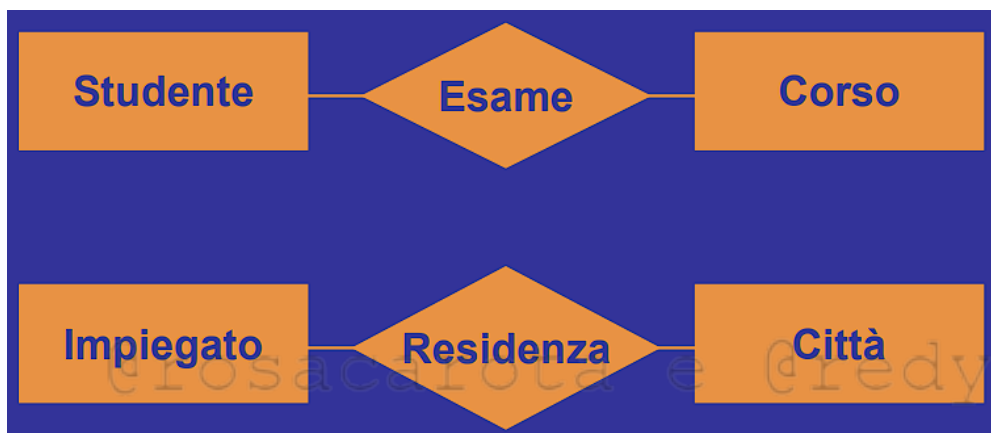


Le **associazioni** (o **relazioni**) sono dei legami logici fra due o più entità, rilevante nell'applicazione di interesse

Ogni associazione ha un **nome** che la identifica univocamente (sostantivi)

Esempi: residenza (fra persone e città), esame (fra studente e corso)

Rappresentazione grafica di associazioni:



Una **occorrenza** di un'associazione

- **Binaria**: è una coppia di occorrenze di entità, una per ciascuna entità coinvolta
- **N-aria**: è una n-upla di occorrenze di entità, una per ciascuna entità coinvolta

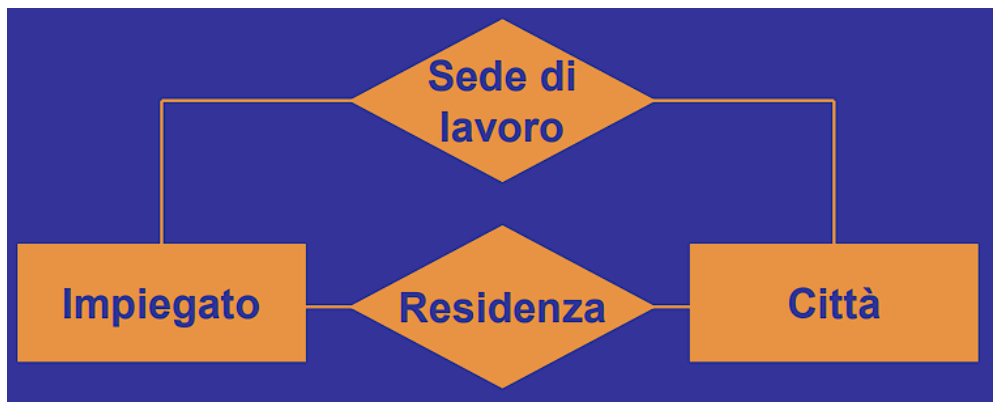
Possono esistere associazioni diverse che coinvolgono la stessa entità

Nell'ambito di un'associazione non ci possono essere occorrenze (coppie, ennuple) ripetute

Esistono anche associazioni **ricorsive**, cioè un'associazione tra un'entità e se stessa

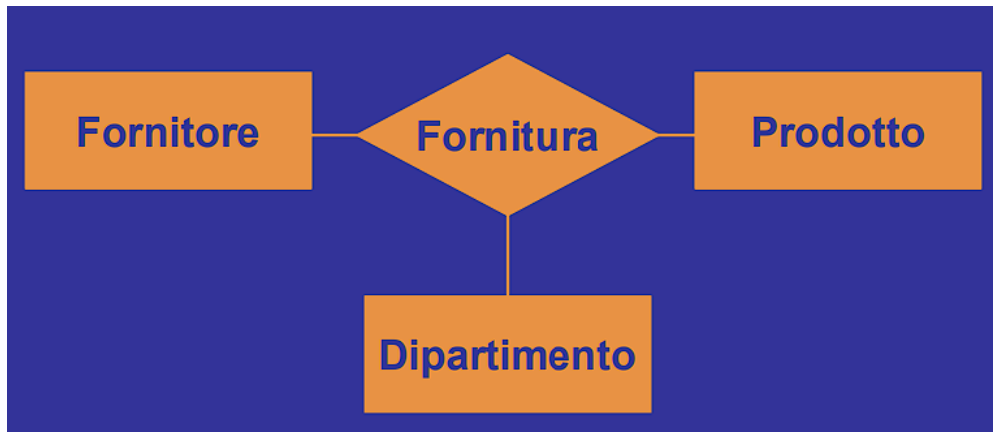
Di solito è necessario stabilire due **ruoli** che l'entità coinvolta rappresenta nella relazione

Due associazioni sulle stesse entità:

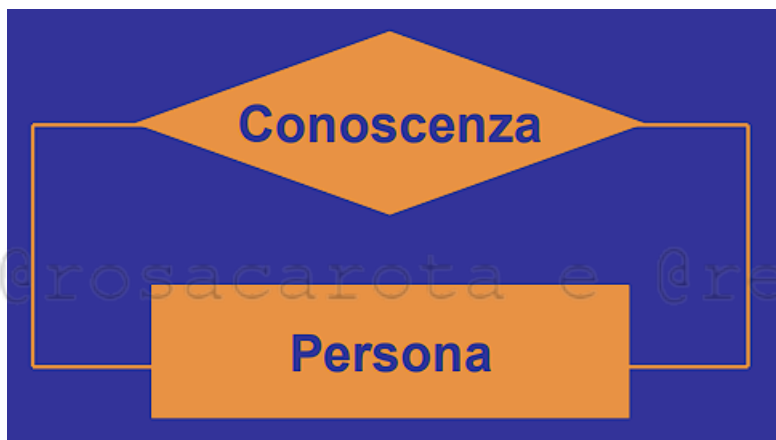


@rosacarota e @redyz13

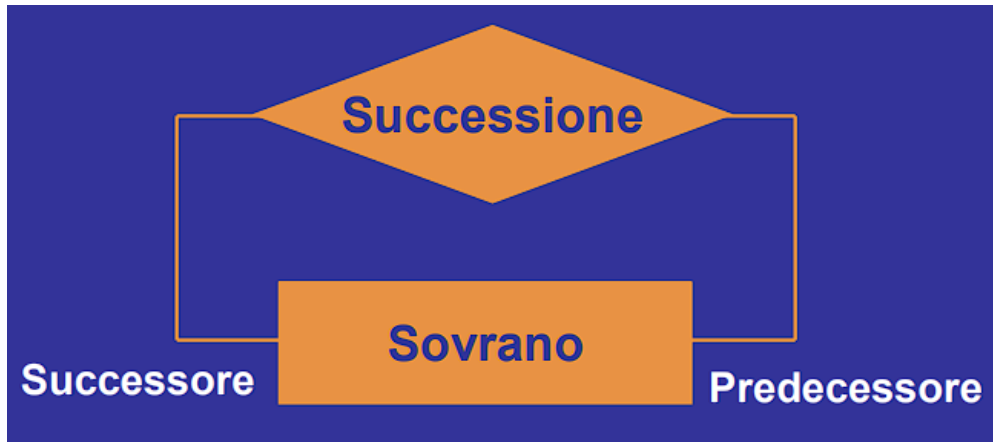
Associazione n-aria:



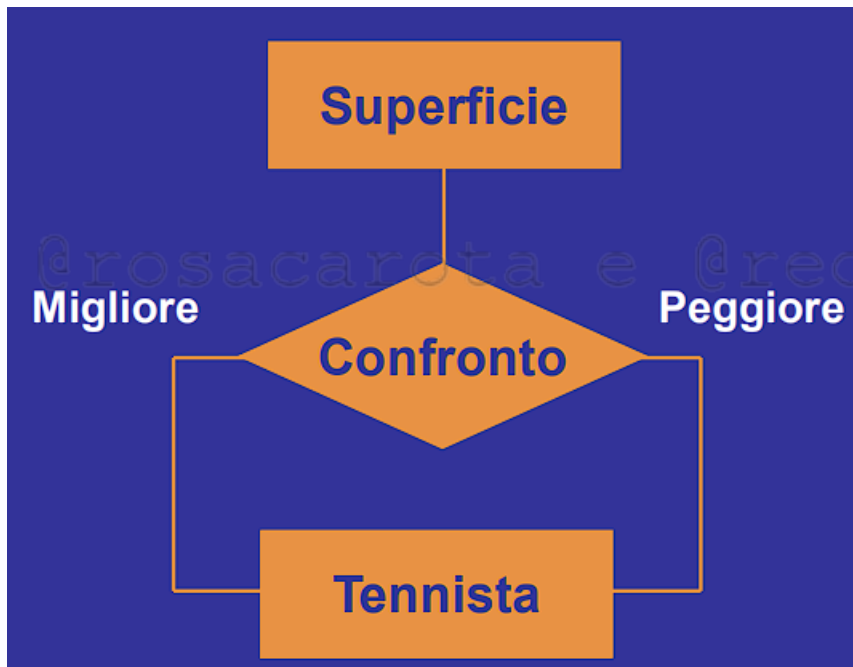
Associazione ricorsiva:



Associazione ricorsiva con "ruoli":



Associazione ternaria ricorsiva:



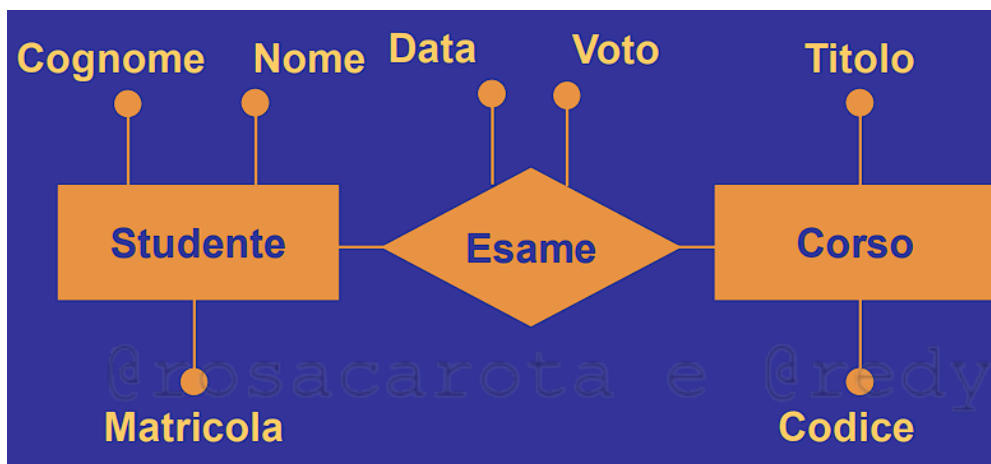
## Attributo



L'**attributo** è una proprietà elementare di un'entità o di un'associazione, di interesse ai fini dell'applicazione

Associa ad ogni occorrenza di entità o associazione un valore appartenente a un insieme detto **dominio** dell'attributo

Attributi, rappresentazione grafica:



## Attributi composti

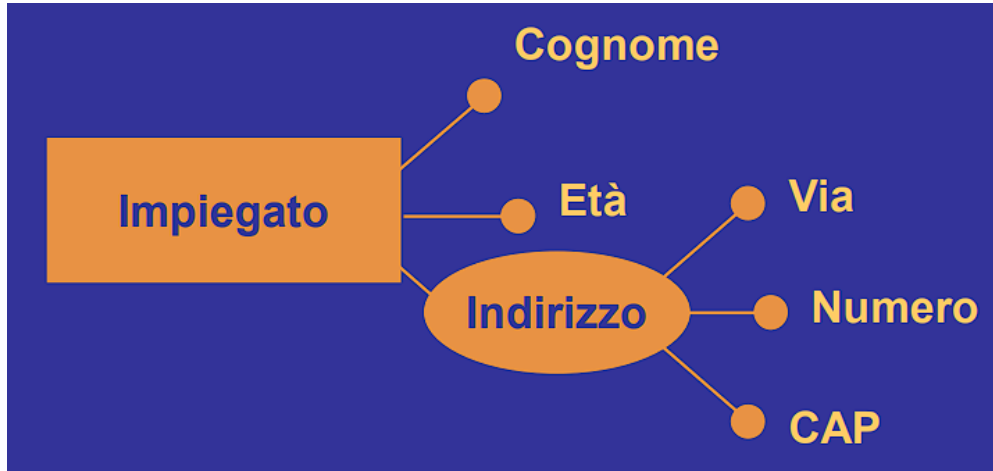


Gli **attributi composti** raggruppano attributi di una medesima entità o associazione che presentano affinità nel loro significato o uso

Esempio:

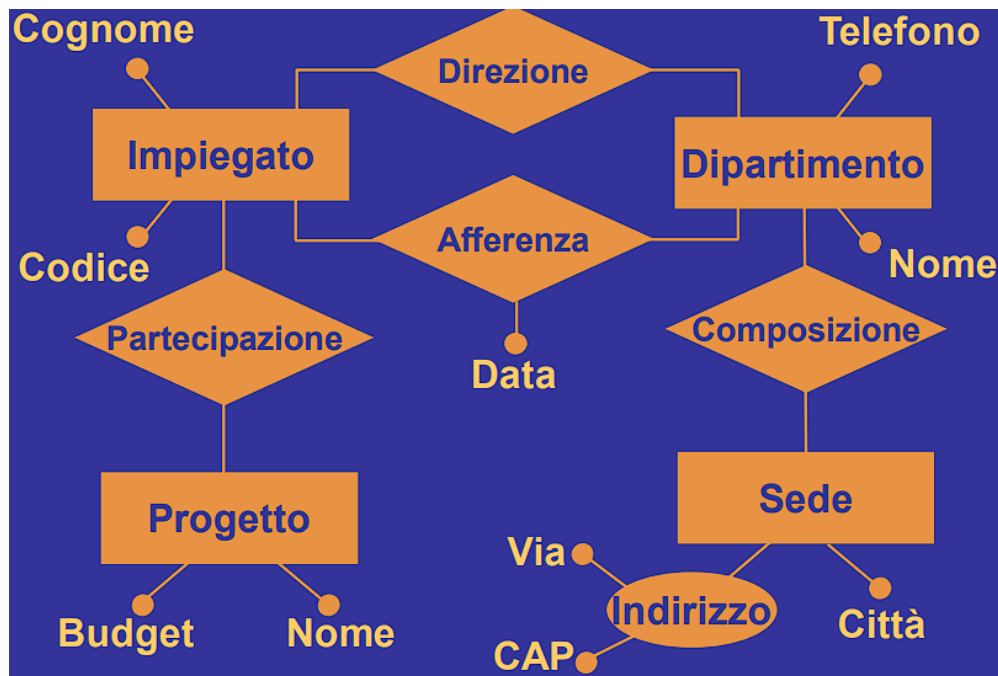
- Via, numero civico e CAP formano un indirizzo

Rappresentazione grafica:



Esempio di schema con l'utilizzo di entità, associazioni e attributi:

@rosacarota e @redyz13



## Altri costrutti del modello E-R

Cardinalità:

- Di associazione
- Di attributo

Identificatore:

- Interno
- Esterno

Generalizzazione

## Cardinalità di associazione



La **cardinalità di associazione** è un valore associato a ogni entità che partecipa ad una associazione. Specifica il numero minimo e massimo di occorrenze dell'associazione a cui ciascuna occorrenza di un'entità può partecipare.

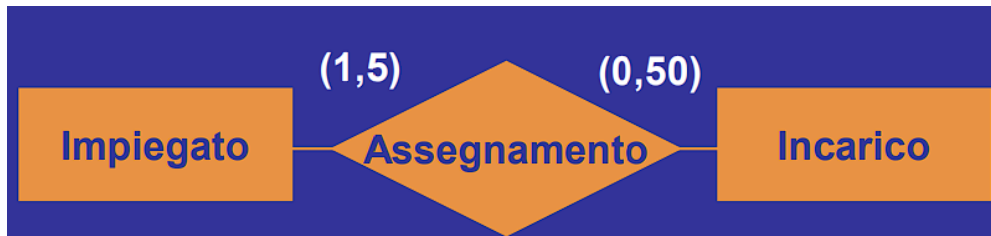
Dicono quante volte un'occorrenza dell'entità può essere legata a occorrenze delle altre entità.

Per semplicità usiamo solo tre simboli:

- 0 e 1 per la cardinalità minima:
  - 0 = "partecipazione **“opzionale”**
  - 1 = "partecipazione **“obbligatoria”**
- 1 e N per la massima:
  - 1 associa a una occorrenza dell'entità una sola occorrenza delle altre entità che partecipa alla relazione
  - N non pone alcun limite

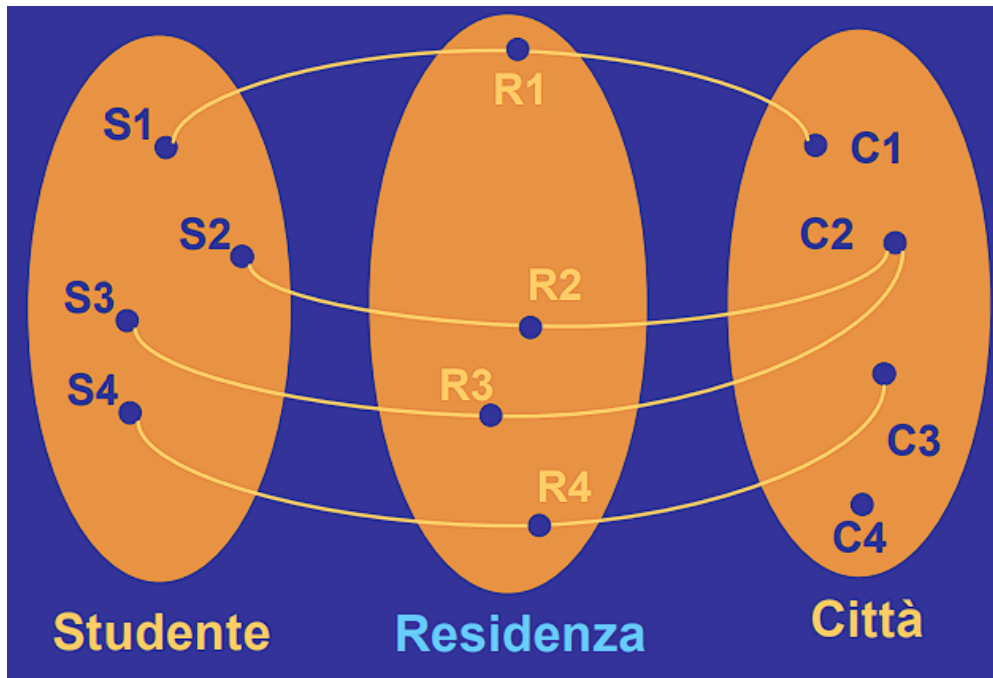
La cardinalità minima deve essere minore o uguale della cardinalità massima.

Esempio di cardinalità:



Occorrenze di Residenza:





Cardinalità di Residenza:



### Tipi di associazioni

Con riferimento alle cardinalità **massime**, abbiamo associazioni:

- **Uno a uno (1,1)**
- **Uno a molti (1,N)**

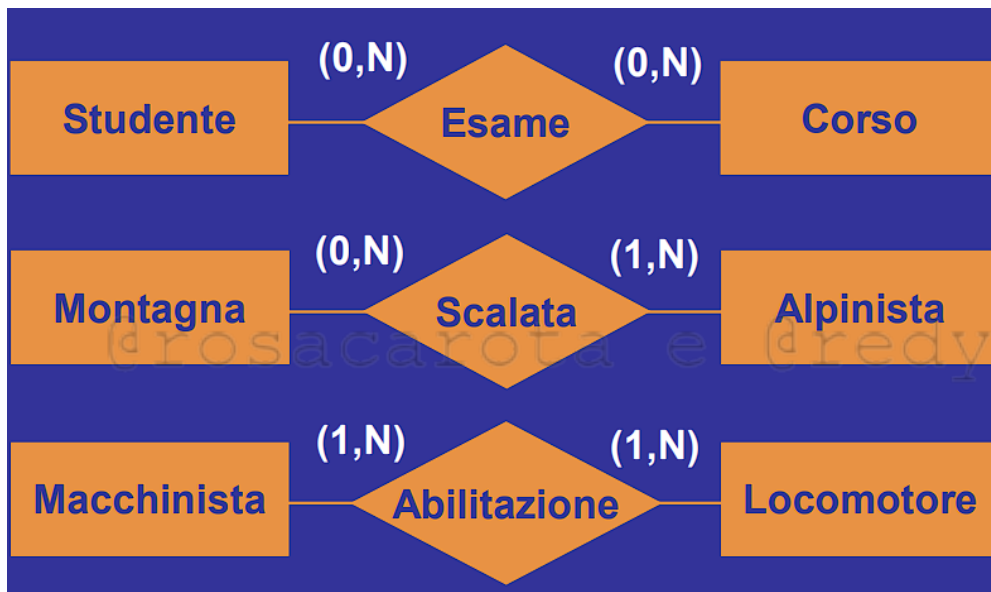
- **Molti a molti (N,N)**

La cardinalità minima si usa solo per dire se la partecipazione dell'occorrenza è **opzionale** o **obbligatoria**

Nelle associazioni n-arie le entità coinvolte partecipano quasi sempre con una cardinalità massima pari a N

Nel caso partecipasse con cardinalità massima pari a 1 significa che ogni sua occorrenza può essere legata a una sola occorrenza della relazione e quindi a un'unica ennupla di occorrenze delle altre entità

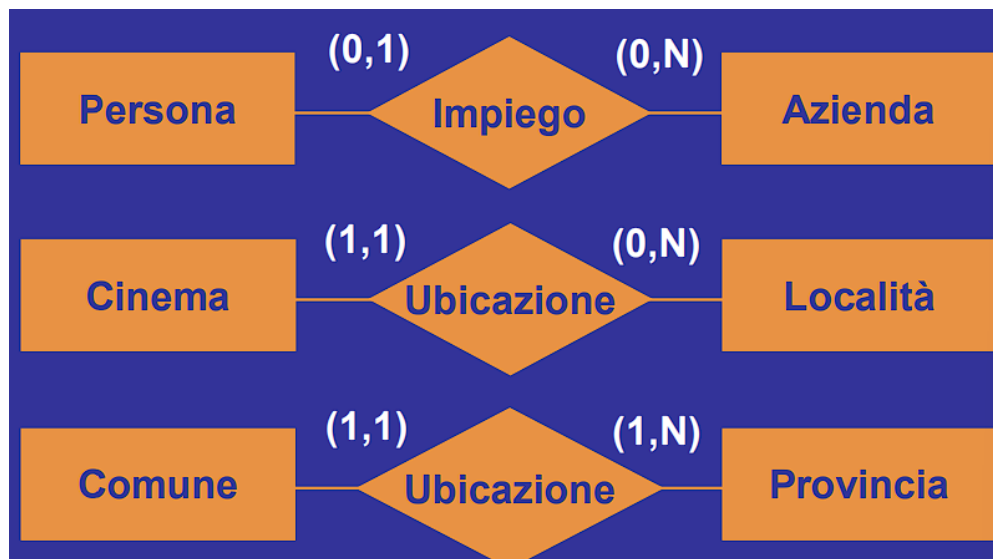
Associazioni "molti a molti":



Due avvertenze:

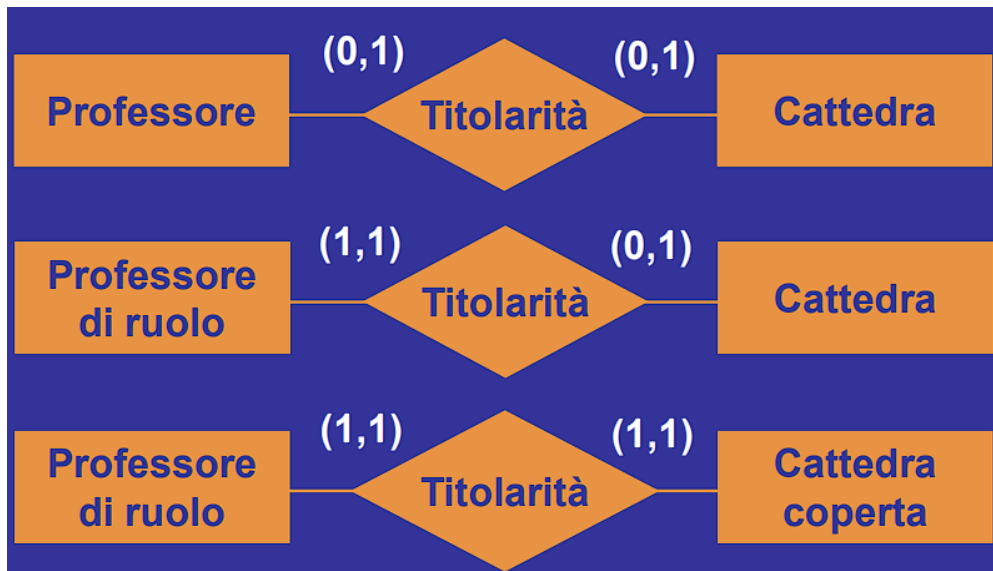
- Attenzione al "verso" nelle associazioni uno a molti
- Le associazioni obbligatorie-obbligatorie sono molto rare

Associazioni "uno a molti":



Associazioni "uno a uno":

@rosacarota e @redyz13



Scambiando i due massimi delle varie associazioni uno a molti, è possibile capire chi ha più collegamenti

@rosacarota e @redyz13

### Cardinalità di attributi



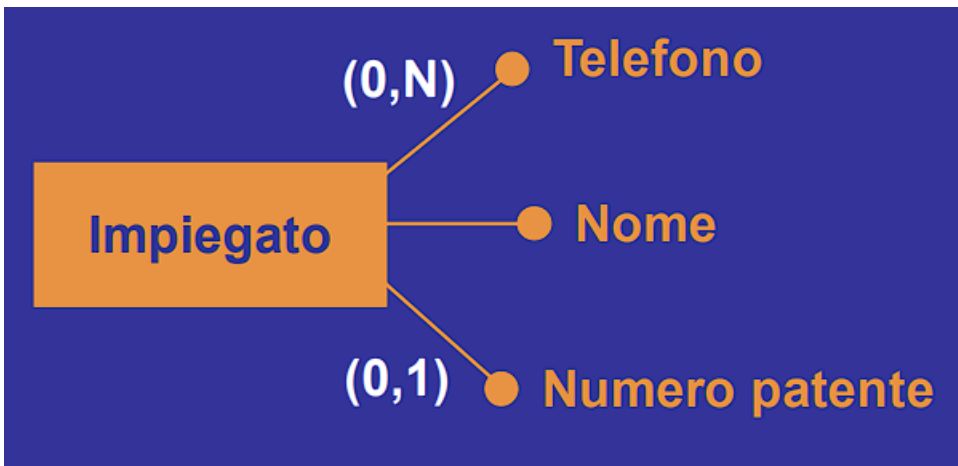
Descrivono il numero minimo e massimo di valori dell'attributo associati a ogni occorrenza di entità o relazione

Nella maggior parte dei casi la cardinalità di un attributo è pari a (1,1) e viene omessa

La cardinalità per un attributo si ha per definirne 3 tipi:

- Cardinalità  $(0, \_)$  → **Attributo opzionale** (informazione incompleta)
- Cardinalità  $(1, \_)$  → **Attributo obbligatorio**
- Cardinalità  $(\_, N)$  → **Attributo multivalore**

Rappresentazione grafica:



@rosacarota e @redyz13  
**Identificatore di una entità**



Sono specificati per ciascuna entità di uno schema e identificano in maniera univoca le occorrenze di entità

Esso è costituito da:

- Attributi dell'entità
  - **Identificatore interno:** formato soltanto dagli attributi di un'entità. Può esserne uno o possono essere più

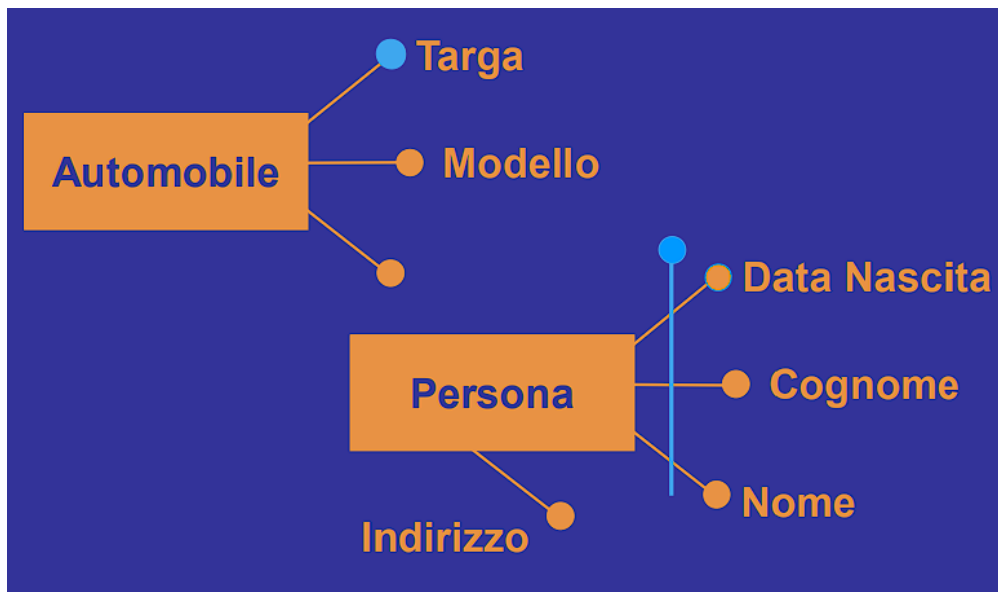
attributi messi insieme ognuno dei quali ha cardinalità (1,1)

- Attributi + Entità esterne attraverso associazioni
  - **Identificatore esterno:** l'identificazione di un'entità si ha utilizzando altre entità. (Si usano alcuni attributi dell'entità e attributi di un'altra entità collegata tramite associazione)
    - Un'entità E può essere identificata da altre entità **solo se** tali entità sono coinvolte in una relazione a cui E partecipa con **cardinalità (1,1)**
    - Un'identificazione esterna può coinvolgere un'entità che a sua volta è identificata esternamente, ma non devono essere generati cicli di identificazione
    - L'entità che è identificata esternamente è detta **entità debole**

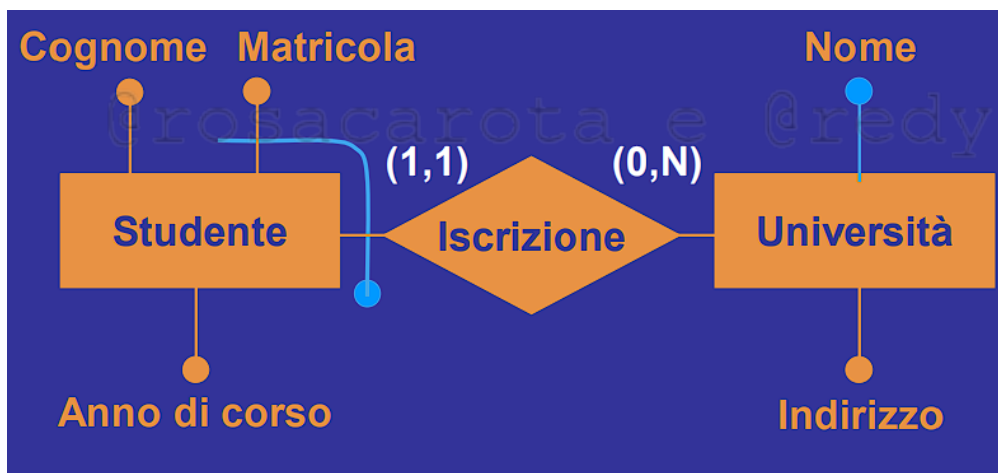
Ogni entità deve avere almeno un identificatore (interno o esterno), ma ne può avere in genere anche di più: in questo caso, gli attributi e le entità coinvolte in alcune identificazioni (tranne una) possono essere opzionali

Identificatori interni:

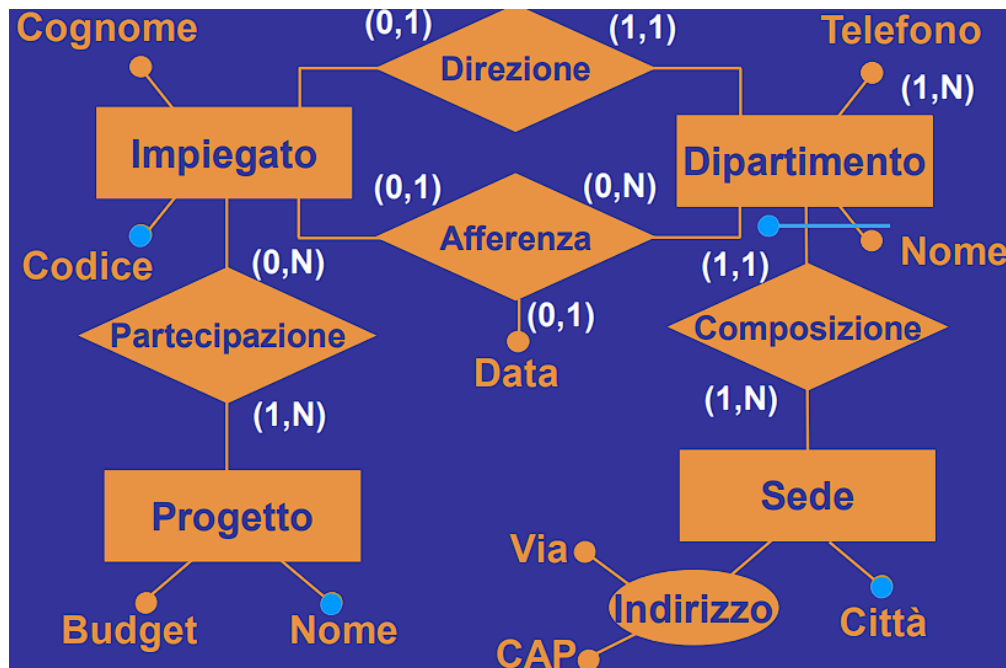
@rosacarota e @redyz13



Identificatore esterno:



Stiamo usando Matricola (da Studente) e Nome (da Università)



## Generalizzazione



Rappresentiamo legami tra un entità  $E$ , detta **entità genitore**, e uno o più entità  $E_1, \dots, E_n$ , dette **entità figlie**, di cui  $E$  è più generale, nel senso che le comprende come caso particolare

Si dice quindi che:

- $E$  è **generalizzazione** di  $E_1, \dots, E_n$
- $E_1, \dots, E_n$  sono **specializzazioni** (o sottotipi) di  $E$



Proprietà:

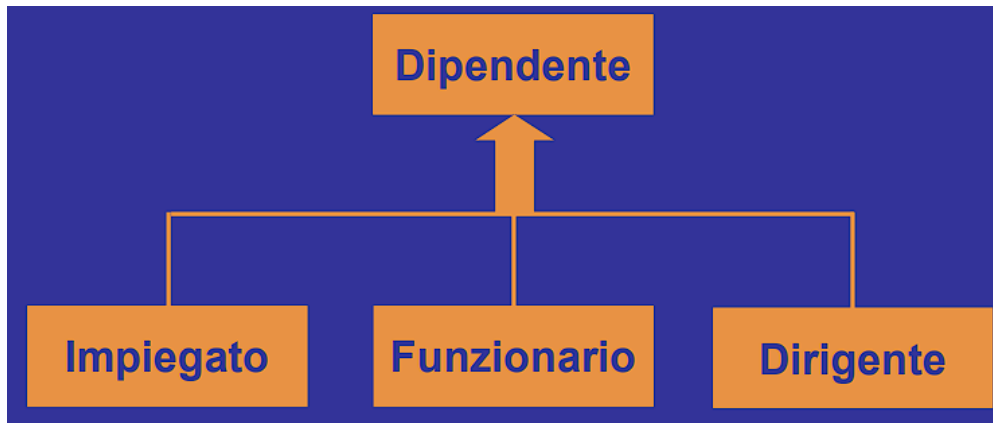
- Ogni occorrenza di un'entità figlia è anche un'occorrenza dell'entità genitore
- **Ereditarietà:** Ogni proprietà dell'entità genitore (attributi, identificatori, relazioni, e altre generalizzazioni) è anche una proprietà delle entità figlie. Le proprietà non sono esplicitate

### Tipi di generalizzazione

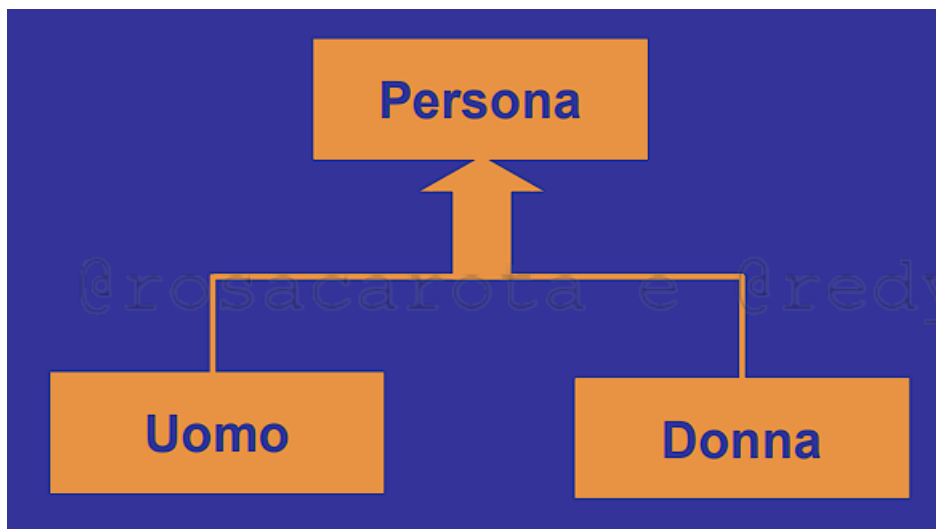
- **Totale:** ogni occorrenza dell'entità genitore è una occorrenza di almeno una delle entità figlie
- **Parziale:** non tutte le occorrenze dell'entità genitore sono occorrenze delle entità figlie
- **Esclusiva:** ogni occorrenza dell'entità genitore è al più un'occorrenza di una delle entità figlie
- **Sovrapposta:** le occorrenze del padre possono essere contemporaneamente occorrenze di più figlie

Le generalizzazioni sovrapposte possono essere trasformate in esclusive aggiungendo una o più entità figlie per rappresentare i concetti che costituiscono le intersezioni delle entità che sovrappongono

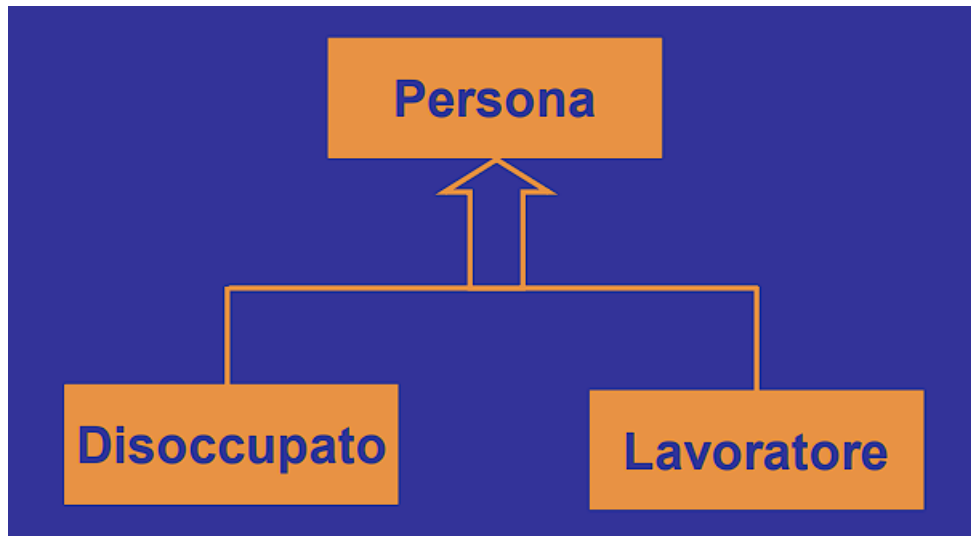
Rappresentazione grafica:



Generalizzazione totale:



Generalizzazione parziale:



### Altre proprietà

Possono esistere gerarchie a più livelli e multiple generalizzazioni allo stesso livello

Un'entità può essere inclusa in più gerarchie, come genitore e/o come figlia

Se una generalizzazione ha solo un'entità figlia si parla di **sottoinsieme**

Alcune configurazioni non hanno senso

Il genitore di una generalizzazione totale può non avere identificatore

### Documentazione associata agli schemi concettuali

Risulta necessario corredare lo schema E-R a una documentazione di supporto, utile anche per rappresentare proprietà dei dati

non possono essere descritte tramite i costrutti dello schema

## Regole aziendali

Una “regola” indica qualunque informazione che definisce o vincola qualche aspetto di un’applicazione

Business rules (regola aziendali):

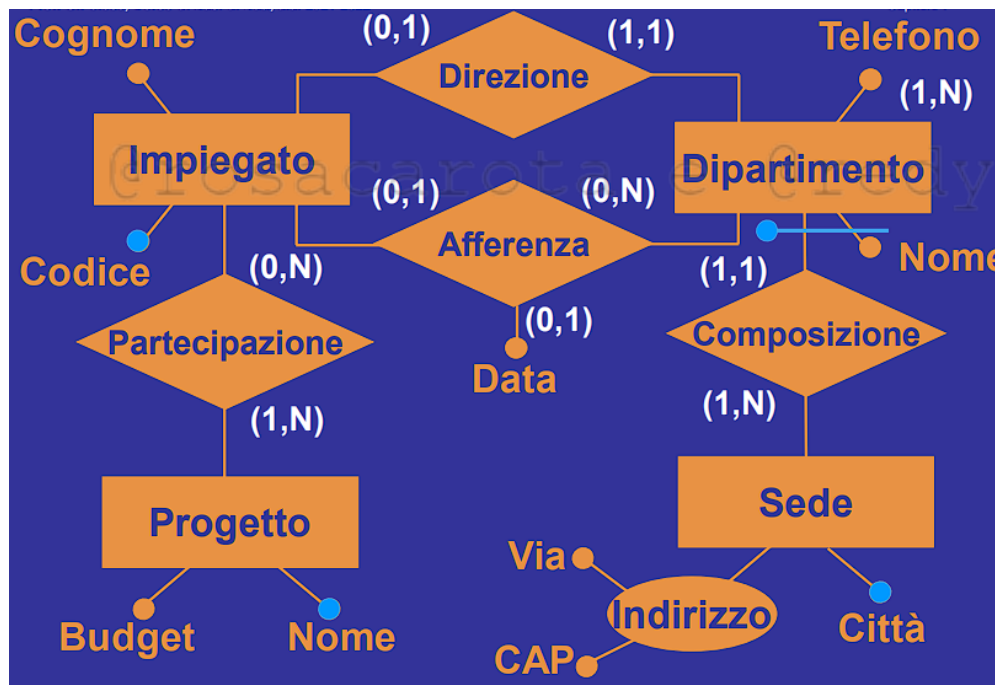
- La **descrizione di un concetto** importante per l’applicazione:
  - La definizione di un’entità, di un attributo o di una relazione del modello E-R
- Un **vincolo di integrità** sui dati dell’applicazione
  - Espresso tramite costrutto o la descrizione di un vincolo non esprimibile tramite i costrutti del modello
  - Sono espresse sotto forma di **asserzioni**: affermazioni che devono essere sempre verificate nella base di dati (devono essere atomiche, non possono essere decomposte in frasi che sono asserzioni a loro volta
  - Vanno enunciate in maniera dichiarativa: “<concetto> deve/non deve <espressione su concetti>”  
dove i concetti compaiono nello schema ER oppure a concetti derivabili da essi
- Una **derivazione**
  - Un concetto che può essere ottenuto, attraverso un calcolo aritmetico per esempio, da altri concetti dello schema
  - Possono essere espresse specificando le operazioni (aritmetiche o altro) che permettono di ottenere il concetto derivato
  - Vanno enunciate come: “<concetto> si ottiene <operazione su concetti>

Nella traduzione in una base di dati le regole che esprimono vincoli o derivazioni devono essere codificate

La documentazione dei vari concetti rappresentati in uno schema può essere prodotta facendo uso di un **dizionario dei dati**

Esso è composto da due tabelle:

- La prima descrive le entità dello schema con il nome, una definizione informale in linguaggio naturale, l'elenco di tutti gli attributi (con eventuali descrizioni associate) e i possibili identificatori
- L'altra tabella descrive le relazioni con il nome, una loro descrizione informale, l'elenco degli attributi (con eventuali descrizioni) e l'elenco delle entità coinvolte insieme alla loro cardinalità di partecipazione



Dizionario dei dati (entità):

Entità	Descrizione	Attributi	Identificatore
Impiegato	Dipendente dell'azienda	Codice, Cognome, Stipendio	Codice
Progetto	Progetti aziendali	Nome, Budget	Nome
Dipartimento	Struttura aziendale	Nome, Telefono	Nome, Sede
Sede	Sede dell'azienda	Città, Indirizzo	Città

Dizionario dei dati (associazioni)

Relazioni	Descrizione	Componenti	Attributi
Direzione	Direzione di un dipartimento	Impiegato, Dipartimento	
Afferenza	Afferenza a un dipartimento	Impiegato, Dipartimento	Data
Partecipazione	Partecipazione a un progetto	Impiegato, Progetto	
Composizione	Composizione dell'azienda	Dipartimento, Sede	