

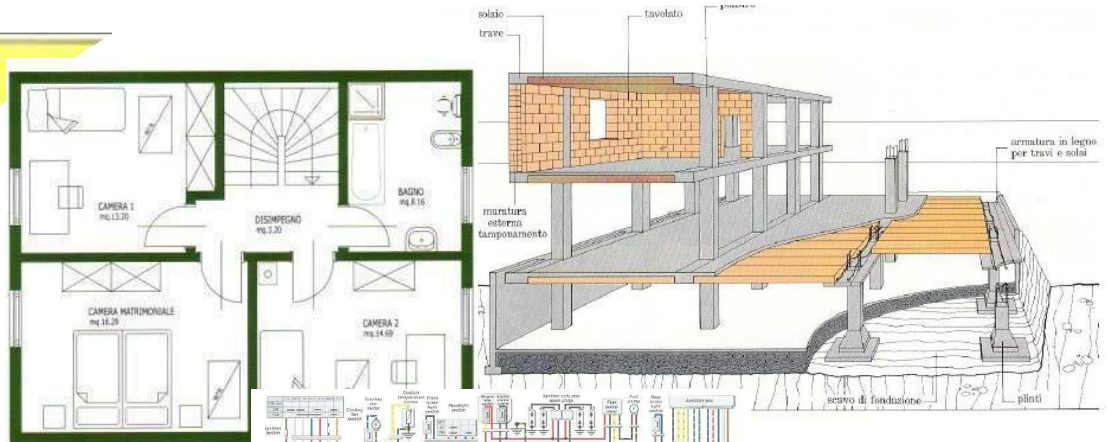


## Progettazione di un database

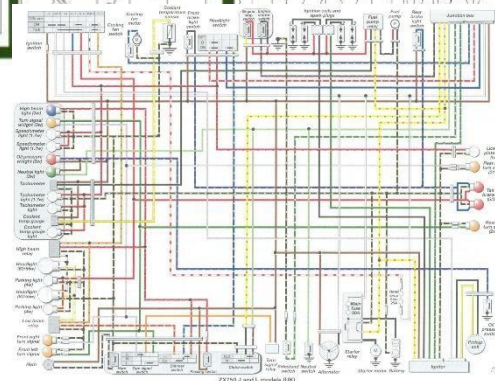
Joseph Giovanelli, Annalisa Franco, Dario Maio  
Università di Bologna

# Un'analogia

- In ogni settore delle scienze applicate la progettazione è svolta in fasi, ciascuna avente un preciso obiettivo; indica in sostanza l'attività che è alla base della costruzione/realizzazione di qualsiasi manufatto complesso, sia esso materiale o soltanto concettuale attraverso la stesura di un progetto. Via via la comprensione dei documenti di specifiche prodotti richiede una peculiare competenza.

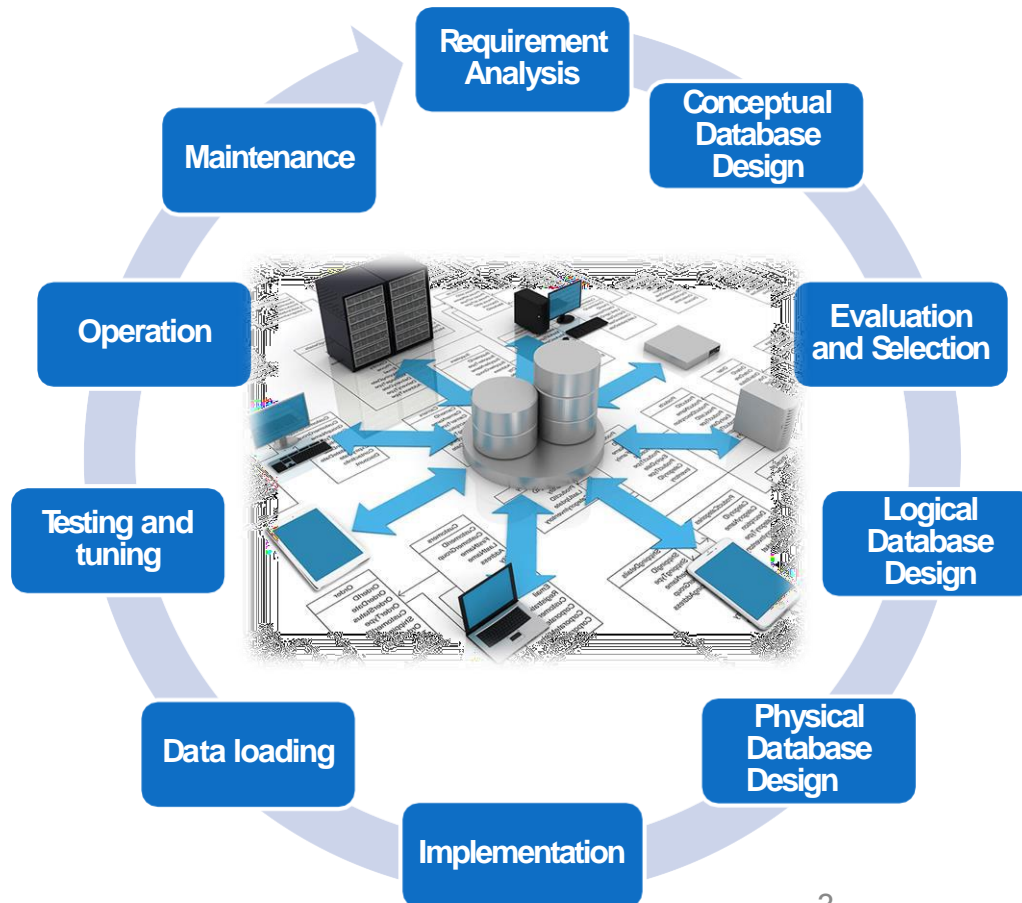


In ogni fase si adottano modelli di rappresentazione, si producono schemi di tipo concettuale, logico e fisico.



# Database Development Life Cycle

- **Database Development Life Cycle (DDLC)**: sottoinsieme del ciclo di sviluppo del software del sistema informativo, dedicato agli aspetti della base di dati e condotto facendo ricorso a opportuni metodi e strumenti.



Le fasi di progettazione e realizzazione possono essere reiterate sulla base di un processo incrementale di comprensione ed evoluzione dei requisiti, avvalendosi anche di tecniche di rapid prototyping e di validazione da parte dell'utente.

# Cosa e come modellare

- Il processo di analisi è incrementale e porta per passi successivi alla stesura di un insieme di documenti in grado di rappresentare un modello dell'organizzazione e comunicare, in modo non ambiguo, una descrizione esauriente, coerente e realizzabile dei vari aspetti statici, dinamici e funzionali di un SI.
- Aspetti ontologici da modellare:
  - ▮ conoscenza concreta (oggetti e fatti specifici, le loro caratteristiche e le loro interrelazioni);
  - ▮ conoscenza astratta (fatti generali che descrivono la conoscenza concreta e regolano il modo in cui essa può evolvere);
  - ▮ conoscenza procedurale (modalità per modificare la conoscenza concreta e ricavare altri fatti);
  - ▮ dinamica (modalità per l'evoluzione nel tempo della conoscenza);
  - ▮ comunicazione (modalità per accedere alla conoscenza).

# Oggetti, funzioni e stati



## Oggetti

- Possono essere descritti a partire da termini molto generici (es. edificio) fino ad arrivare a livello di dettaglio specifici (es. Palazzo Mazzini Marinelli, via Sacchi, 3, Cesena)



## Funzioni

- Possono essere espresse inizialmente in modo vago (es. controllare il livello di gas nocivi nell'aria) e successivamente precisate (es. la programmazione del livello di soglia per l'allarme della centralina è attivata, dopo aver digitato un Pin, premendo il pulsante P)



## Stati

- Possono essere descritti a un elevato livello di astrazione (es. la centralina è in stato di errore) o specificati in maggior dettaglio (es. è acceso il segnalatore d'errore nel sensore S)

# Meccanismi di astrazione

Molteplici  
sono le  
relazioni in  
gioco fra  
oggetti,  
funzioni e  
stati e  
molteplici i  
livelli di  
possibile  
dettaglio

- L'analista deve far ricorso a tecniche che gli consentano di **organizzare e interrogare la conoscenza** sul problema via via acquisita.
- I principali meccanismi di astrazione usati durante il processo di analisi per costruire una base di conoscenza sul problema sono:
  - ▮ **classificazione**
  - ▮ **generalizzazione**
  - ▮ **aggregazione**
  - ▮ **proiezione**

*N.B Nel seguito, negli esempi, l'attenzione sarà focalizzata principalmente sulla modellazione di oggetti.*

# Classificazione

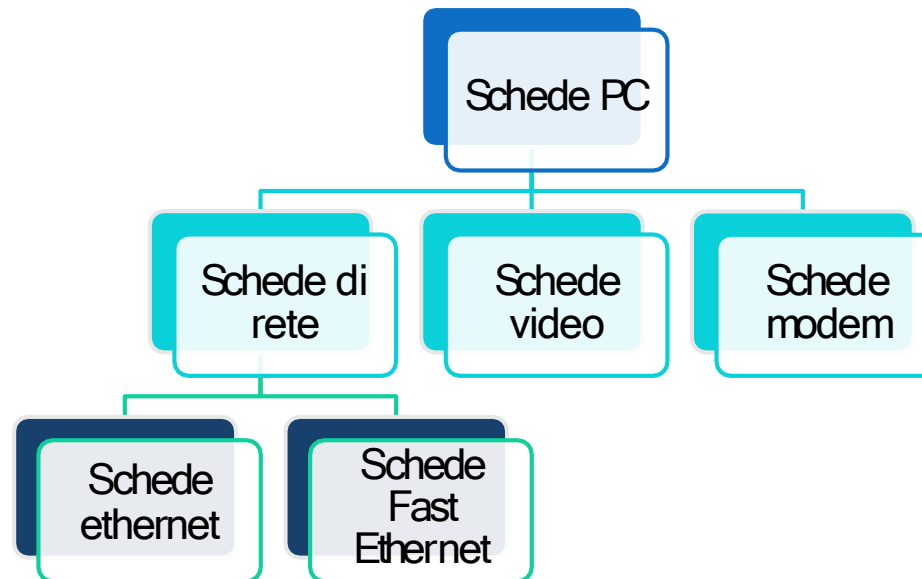
Raggruppa gli oggetti in classi in base alle loro proprietà

Computer



# Generalizzazione

Cattura le relazioni di tipo «è un» ovvero permette di astrarre le caratteristiche comuni fra più classi definendo superclassi



Ciò che caratterizza una scheda per PC è comune anche a ogni suo sottoinsieme, ovvero ogni sottoclasse eredita dalla superclasse ma può anche avere caratteristiche proprie.

La **specializzazione** è il processo inverso della generalizzazione.



# Copertura delle generalizzazioni

**Confronto fra unione delle specializzazioni e classe generalizzata**

**TOTALE**

La classe generalizzata è l'unione delle specializzazioni

**PARZIALE**

La classe generalizzata contiene l'unione delle specializzazioni

**Confronto fra le classi specializzate**

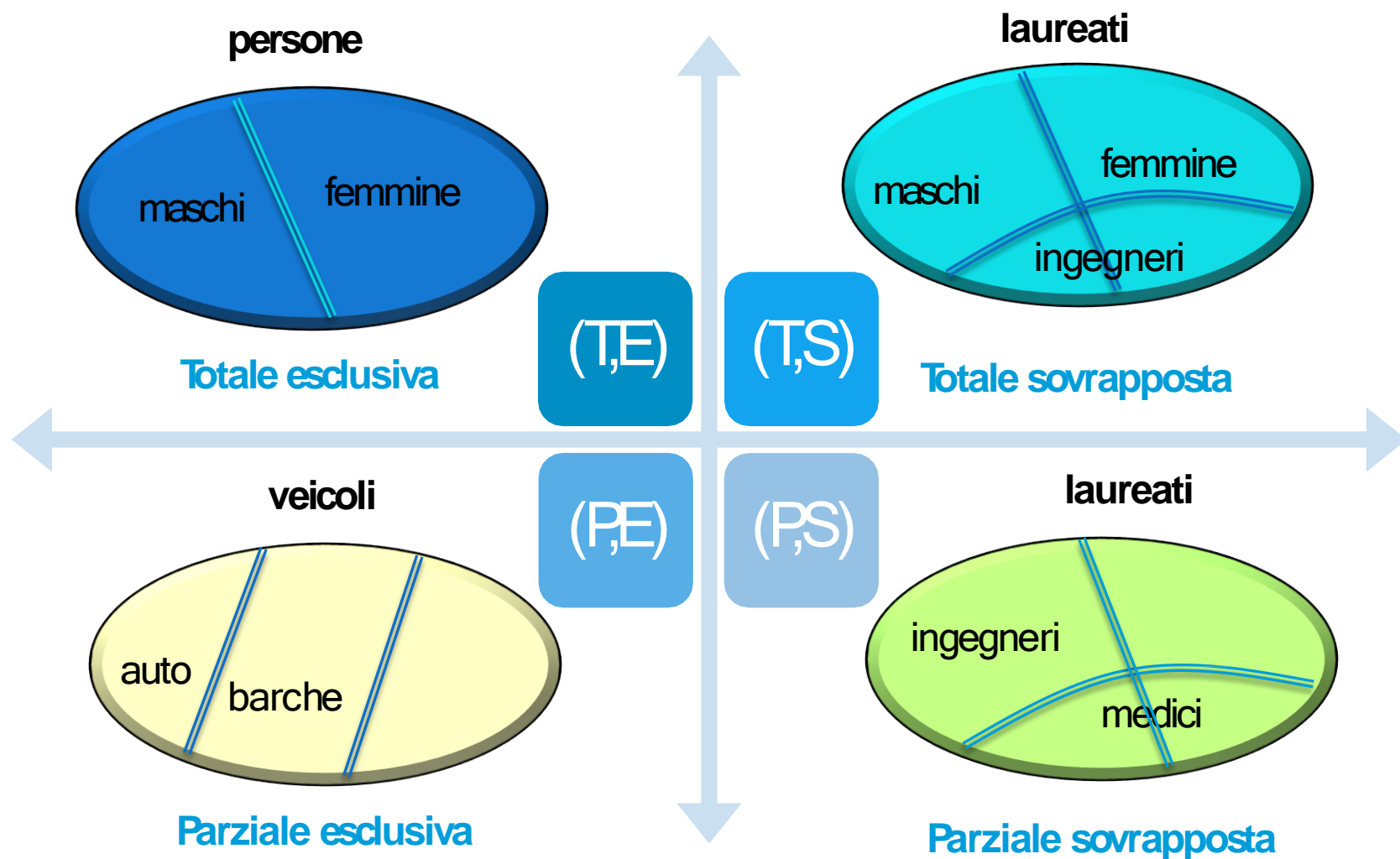
**ESCLUSIVA**

Gli insiemi delle specializzazioni sono fra loro disgiunti

**SOVRAPPOSTA**

Può esistere un'intersezione non vuota fra insiemi delle specializzazioni

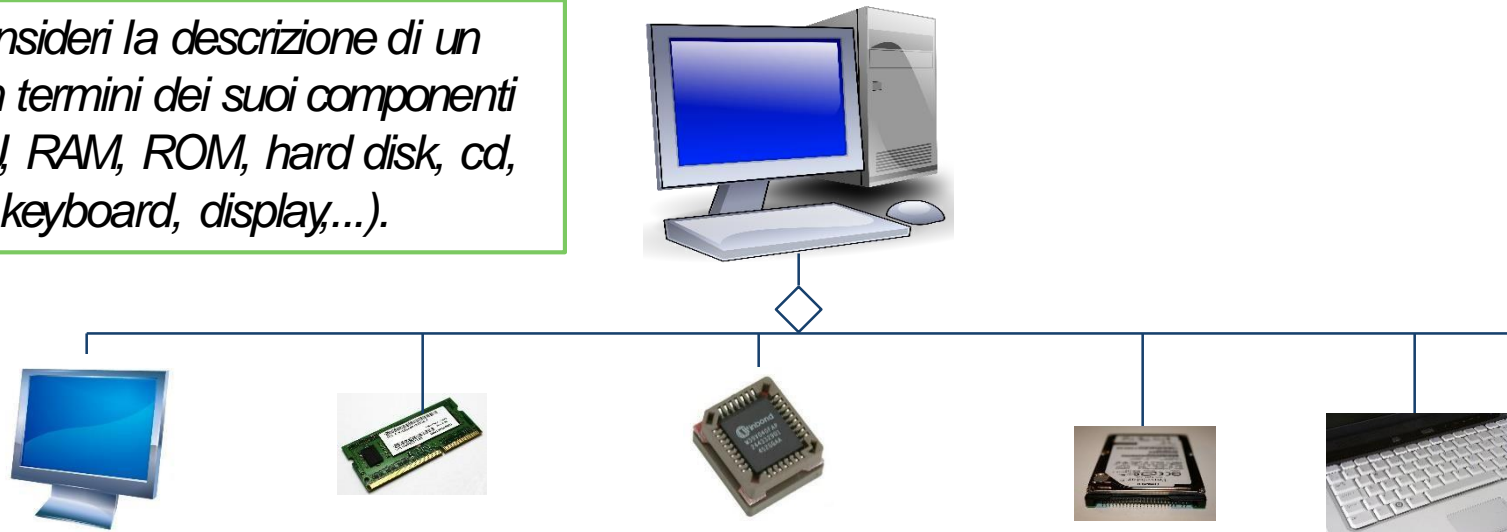
# Proprietà di copertura - esempi



# Aggregazione

L'aggregazione esprime le relazioni parte di che sussistono tra oggetti, tra funzioni, o fra stati

*Si consideri la descrizione di un PC in termini dei suoi componenti (CPU, RAM, ROM, hard disk, cd, dvd, keyboard, display,...).*



# Proiezione

Cattura la vista delle relazioni strutturali fra gli oggetti, le funzioni, gli stati

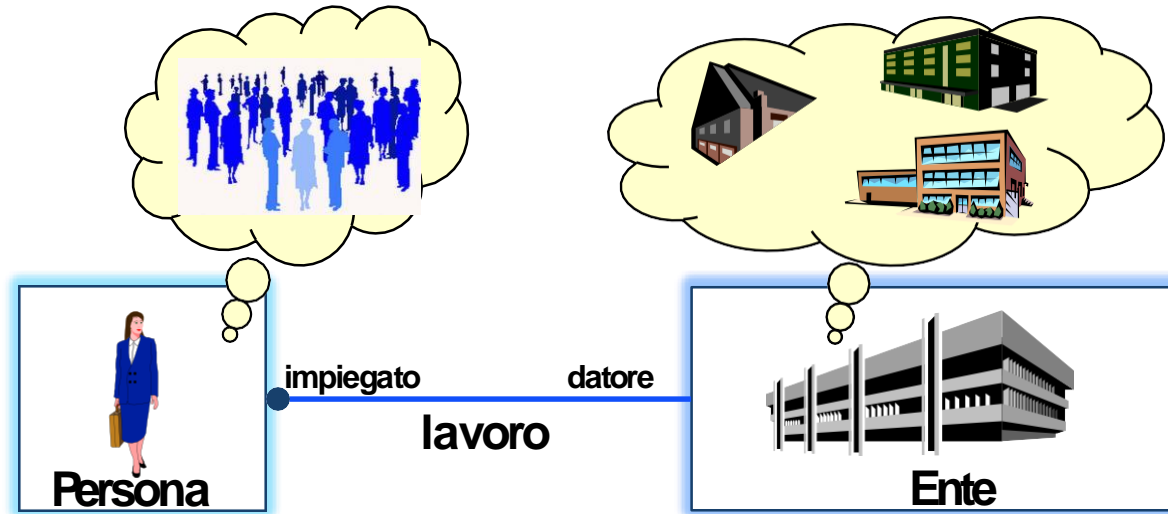


Ad esempio, nel descrivere il funzionamento di un certo personal computer può essere necessario distinguere il punto di vista dell'operatore, dell'installatore, del programmatore.



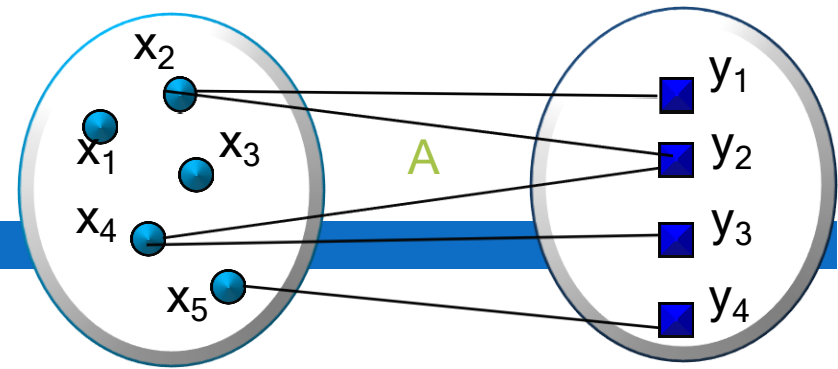
# Associazioni

- Oltre ai meccanismi citati è importante modellare le associazioni che sussistono fra le varie classi.



- Le associazioni (corrispondenze tra classi) sono di fatto aggregazioni di cui le classi sono le componenti.
- Dal punto di vista della modellazione, è importante caratterizzare queste corrispondenze in termini di vincoli di cardinalità (un impiegato in quanti enti può lavorare?).

# Vincoli di cardinalità



- Sia  $A$  un'associazione fra  $C1$  e  $C2$ 
  - ▮  $\text{min-card}(C1, A)$ : cardinalità minima di  $C1$  in  $A$ 
    - è il *minimo numero di corrispondenze* nell'associazione  $A$  alle quali ogni istanza di  $C1$  deve partecipare;
  - ▮  $\text{max-card}(C1, A)$ : cardinalità massima di  $C1$  in  $A$ 
    - è il *massimo numero di corrispondenze* nell'associazione  $A$  alle quali ogni istanza di  $C1$  può partecipare.
- Vincoli di cardinalità minima
  - ▮ *partecipazione opzionale*:  $\text{min-card}(C1, A) = 0$ 
    - alcuni elementi di  $C1$  *possono non essere associati* tramite  $A$  a elementi di  $C2$ ;
  - ▮ *partecipazione obbligatoria (totale)*:  $\text{min-card}(C1, A) > 0$ 
    - a ogni elemento di  $C1$  *deve essere associato*, tramite  $A$ , almeno un elemento di  $C2$ .

# Vincoli di cardinalità: esempi

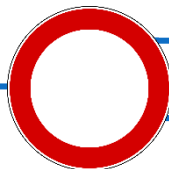
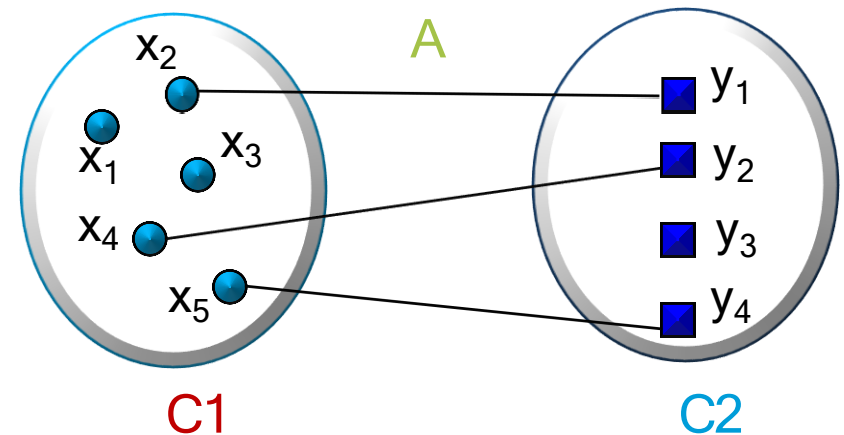
- Sia ASSEGNAZIONE un'associazione fra DIPENDENTE e MANSIONE.
- Nel particolare dominio applicativo in esame sono vere le seguenti affermazioni:
  - ▮ un dipendente svolge almeno una mansione e al massimo 3;
  - ▮ una medesima mansione può essere ancora non assegnata o essere svolta da più dipendenti, al massimo 5.
- Si ha:
  - ▮  $\text{min-card}(\text{DIPENDENTE}, \text{ASSEGNAZIONE}) = 1$
  - ▮  $\text{max-card}(\text{DIPENDENTE}, \text{ASSEGNAZIONE}) = 3$
  - ▮  $\text{min-card}(\text{MANSIONE}, \text{ASSEGNAZIONE}) = 0$
  - ▮  $\text{max-card}(\text{MANSIONE}, \text{ASSEGNAZIONE}) = 5$

# Associazioni binarie *uno a uno*

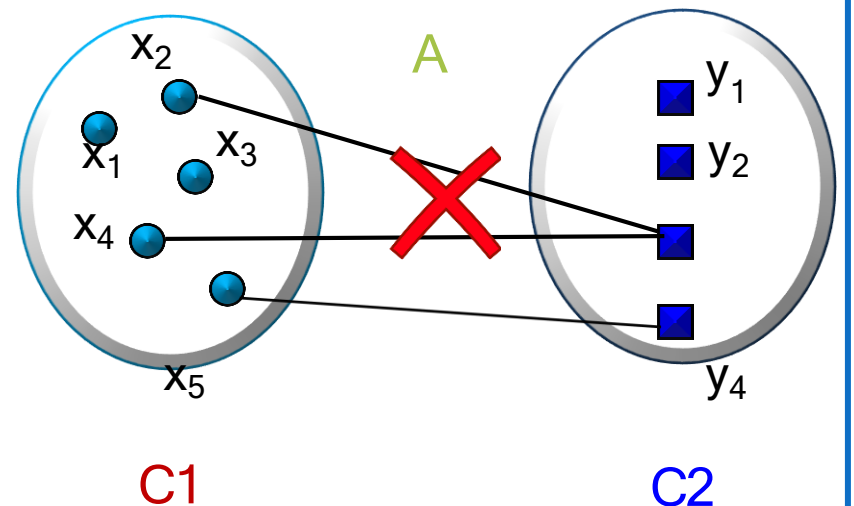
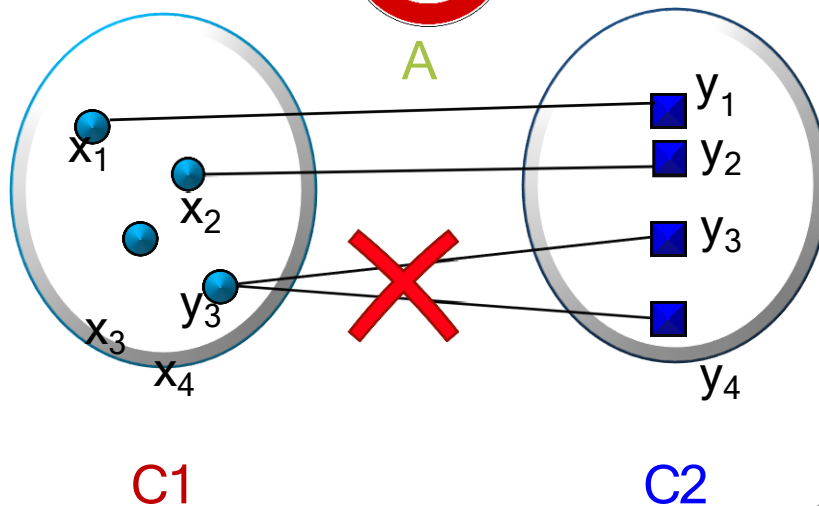
**uno a uno** (one-to-one)

$\text{max-card}(\text{C1}, A) = 1$

$\text{max-card}(\text{C2}, A) = 1$



Associazioni che non soddisfano i vincoli





# Associazioni binarie *uno a molti*

**uno a molti** (one-to-many)

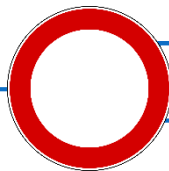
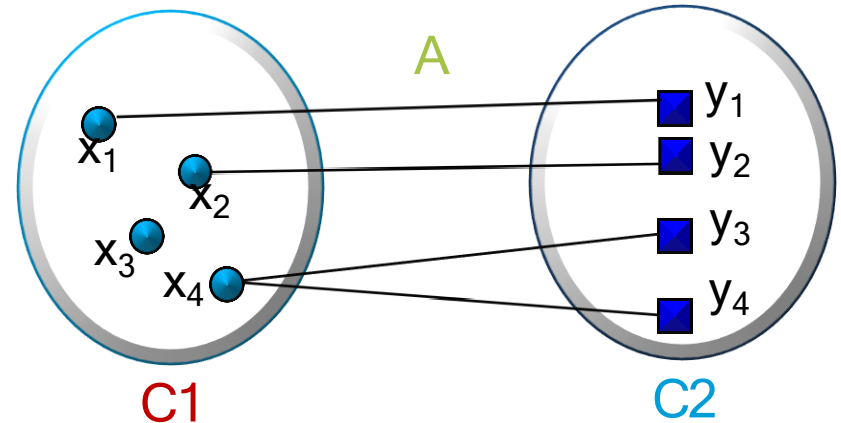
$\text{max-card}(\text{C1}, A) = N$

$\text{max-card}(\text{C2}, A) = 1$

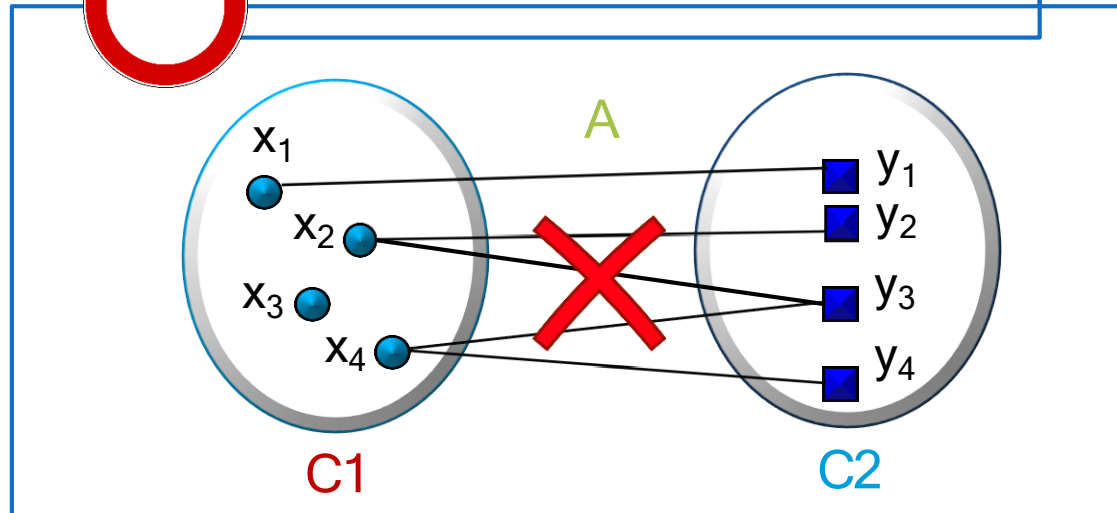
oppure

$\text{max-card}(\text{C1}, A) = 1$

$\text{max-card}(\text{C2}, A) = N$



Associazioni che non soddisfano i vincoli

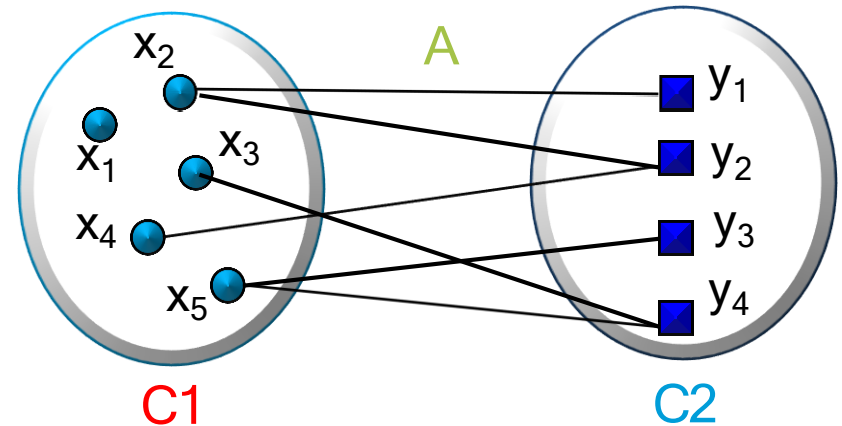


# Associazioni binarie *molti a molti*

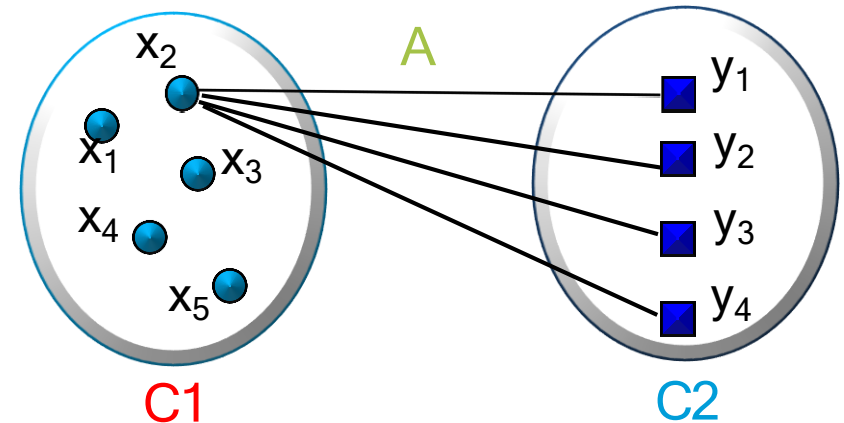
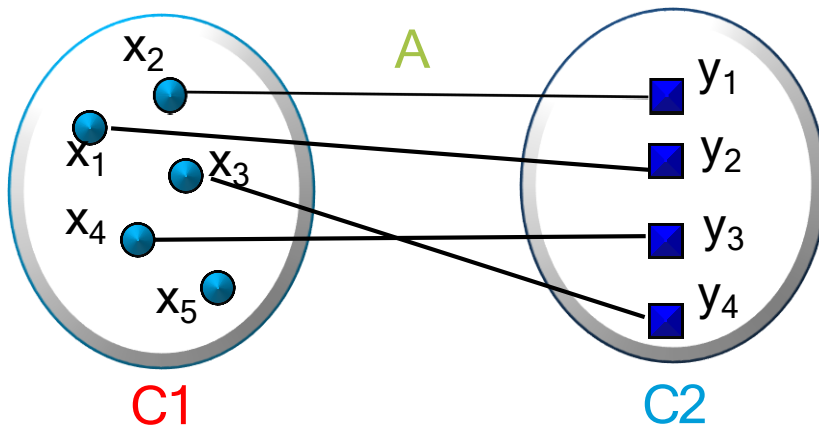
**molti a molti** (many-to-many)

$\text{max-card}(\text{C1}, A) = N$

$\text{max-card}(\text{C2}, A) = M$



Tutte queste associazioni soddisfano i vincoli

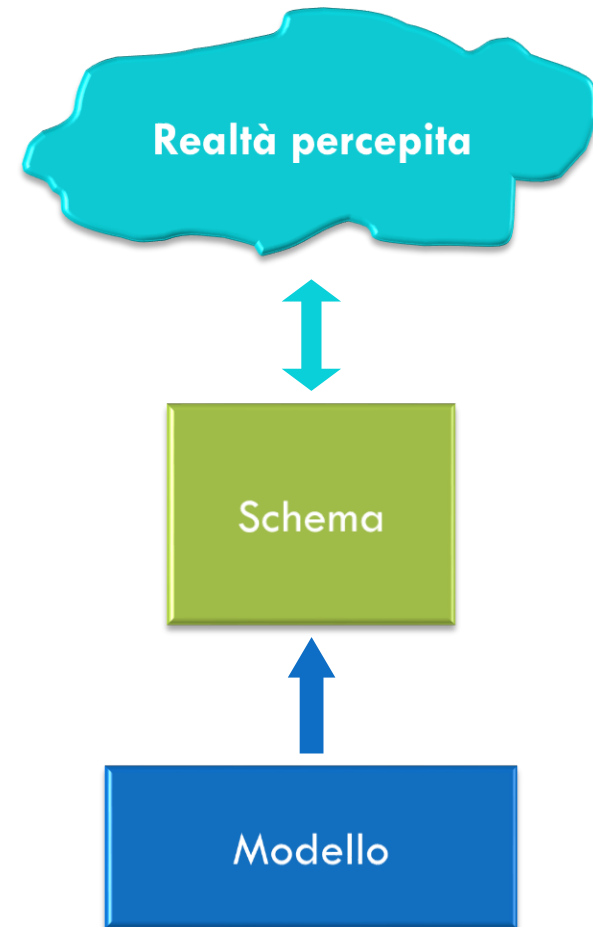


# Modelli dei dati: logici vs concettuali

- Un modello dei dati è una collezione di concetti che sono utilizzati per descrivere i dati, le loro associazioni, e i vincoli che questi devono rispettare.
- Un ruolo di primaria importanza nella definizione di un modello dei dati è svolto dai meccanismi che possono essere usati per strutturare i dati.
  - ▮ **Modelli logici**: utilizzati nei DBMS per l'organizzazione dei dati;
    - utilizzati dai programmi - indipendenti dalle strutture fisiche.
  - ▮ **Modelli concettuali**: permettono di rappresentare i dati in modo indipendente da ogni particolare sistema;
    - cercano di descrivere i concetti del mondo reale;
    - sono utilizzati nelle fasi preliminari di progettazione.
- Il più noto modello concettuale in ambito DB è il modello E/R.

# I modelli concettuali dei dati

- Lo scopo è pervenire a uno schema che rappresenti la realtà di interesse in modo indipendente dal DBMS.
- Si cerca un livello di astrazione “intermedio” tra sistema e utenti, che sia al tempo stesso:
  - ▮ flessibile
  - ▮ intuitivo
  - ▮ espressivo
- I modelli concettuali prevedono tipicamente una rappresentazione grafica, che risulta anche utile come strumento di documentazione e comunicazione.

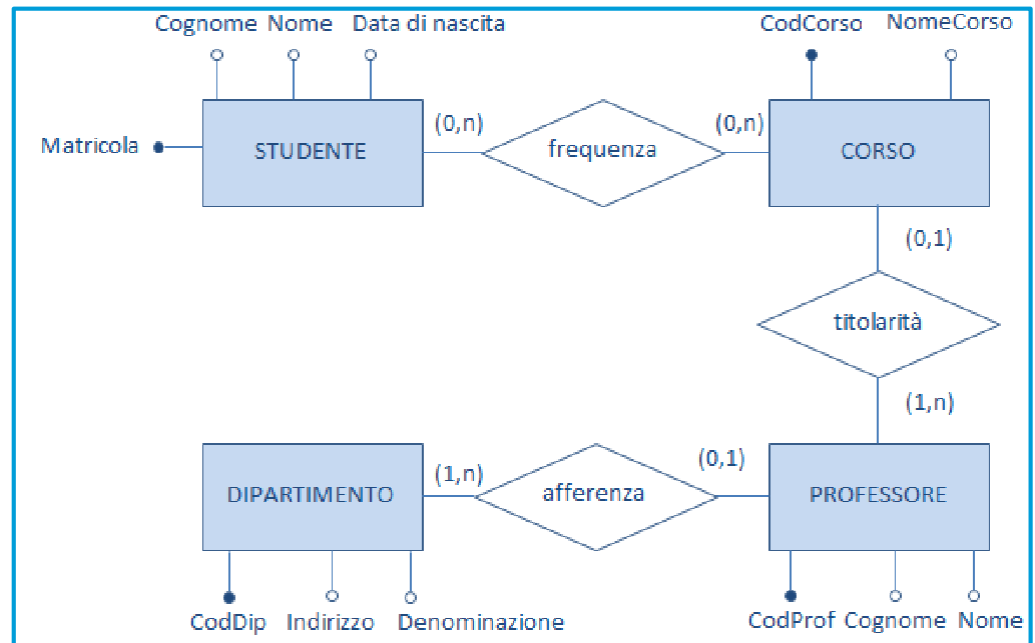


# Esempio di modellazione E/R

- Si fa inoltre uso di un glossario dei termini (detto anche dizionario dati), con finalità di memorizzare in modo uniforme e non ambiguo i vari tipi di dati riscontrati durante l'analisi e le relazioni che sussistono fra essi nonché le altre risorse coinvolte nel sistema informativo.

Uno schema E/R

**N.B.** Snapshot di un anno accademico



# Domande?

---

