



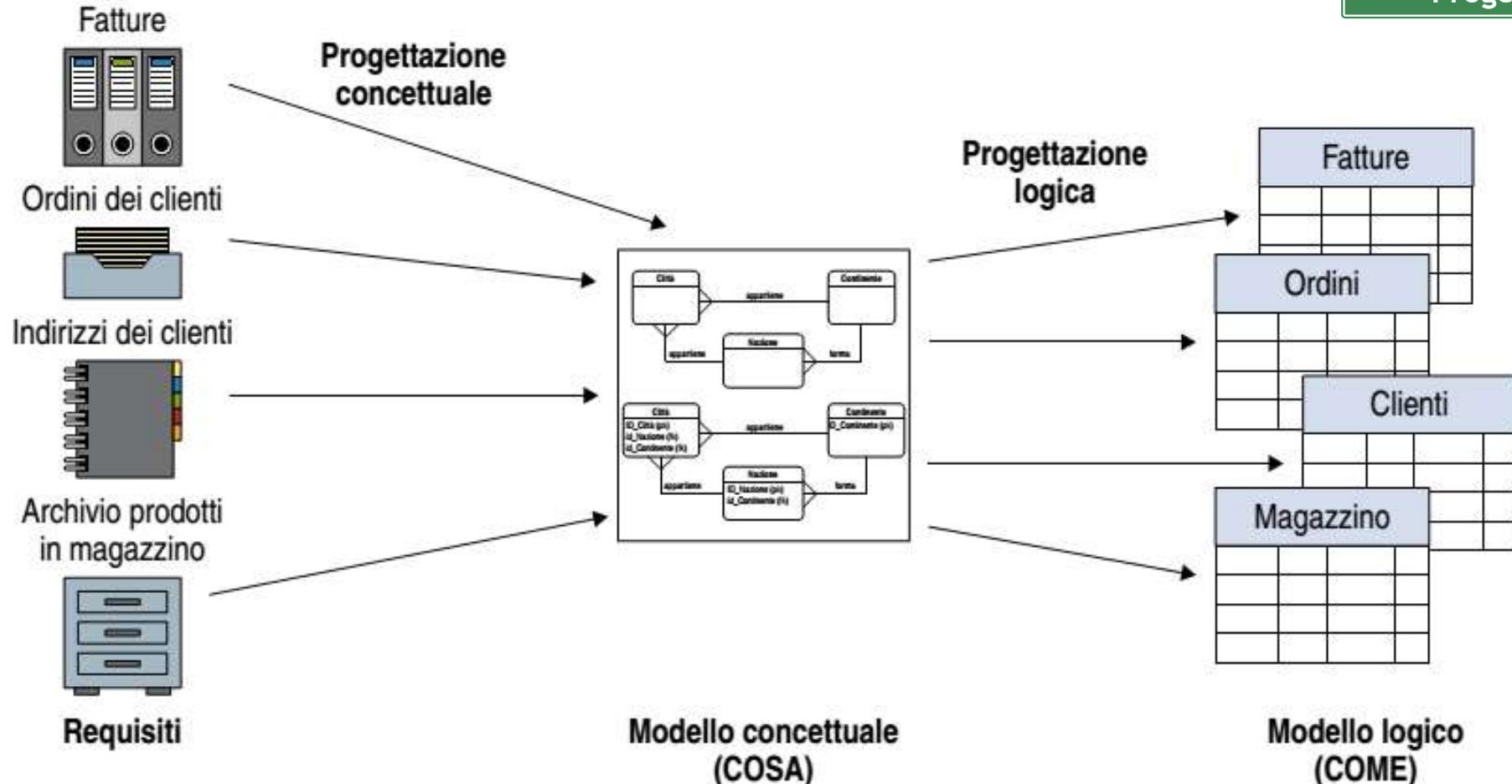
Progettazione concettuale

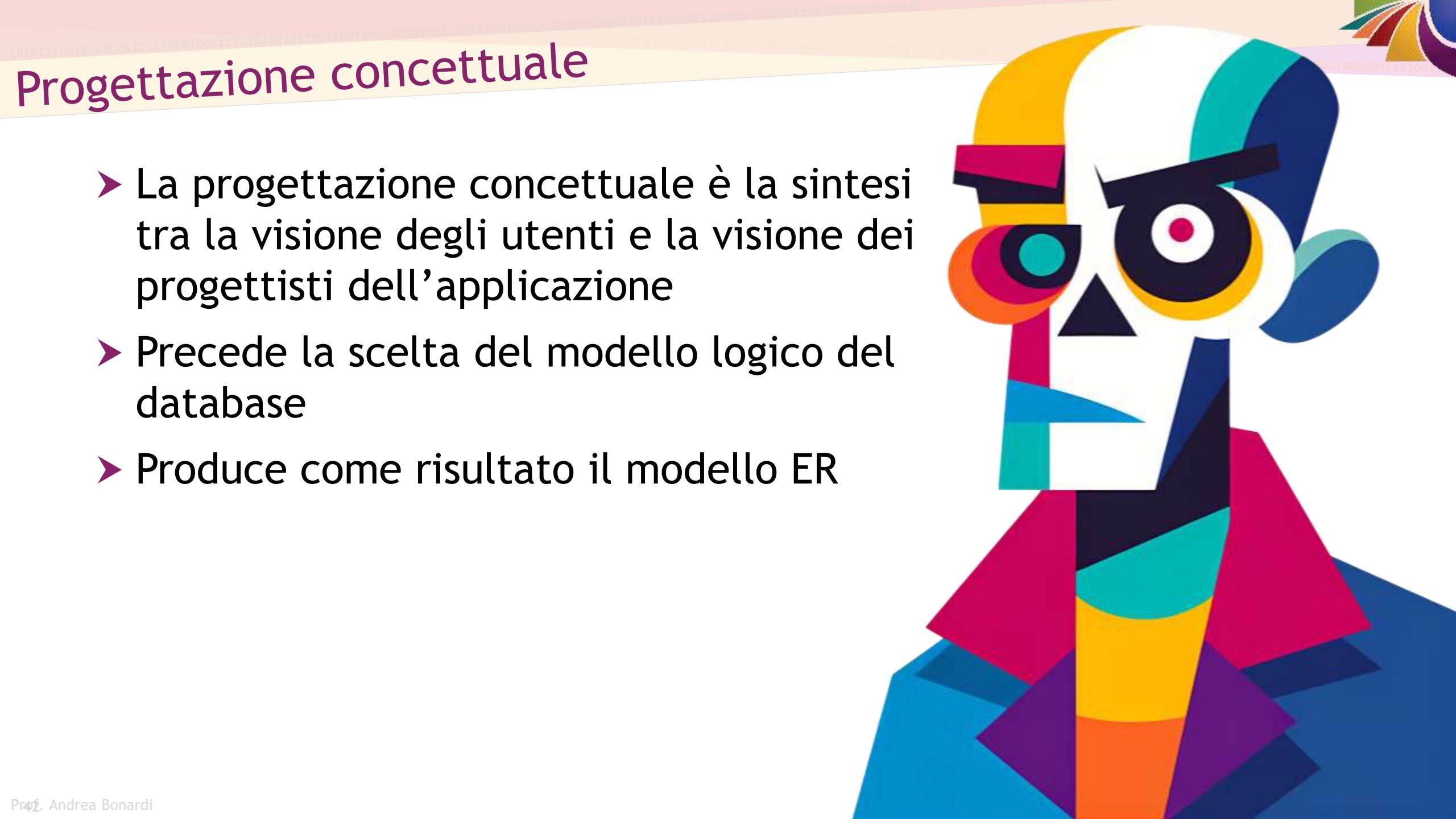
Modello ER

Progettazione di un database



Progettazione di un database





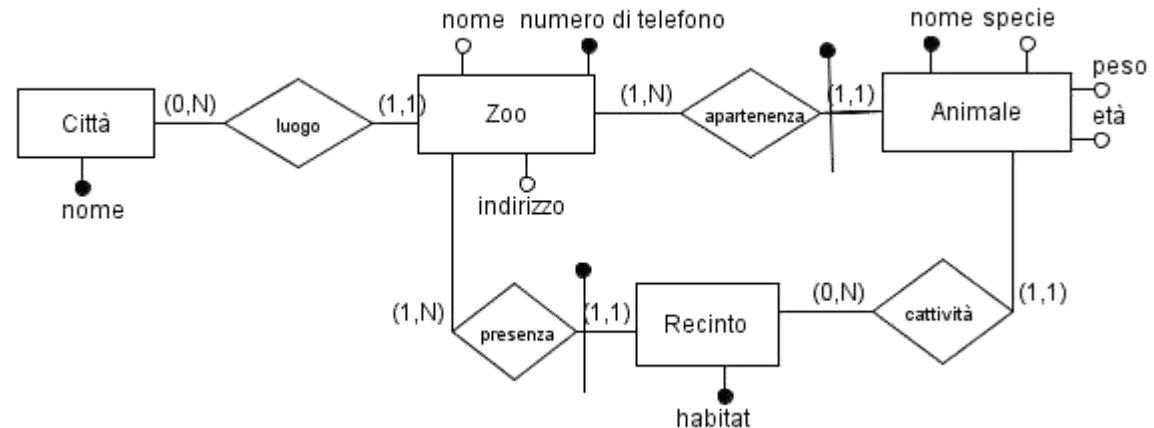
Progettazione concettuale

- La progettazione concettuale è la sintesi tra la visione degli utenti e la visione dei progettisti dell'applicazione
- Precede la scelta del modello logico del database
- Produce come risultato il modello ER

Modello entità/associazioni



- Il modello ER (*Entity-Relationship*) fu introdotto nel 1976 da Peter Chen
 - Consente di modellare il mondo reale
 - È semplice e facile da capire

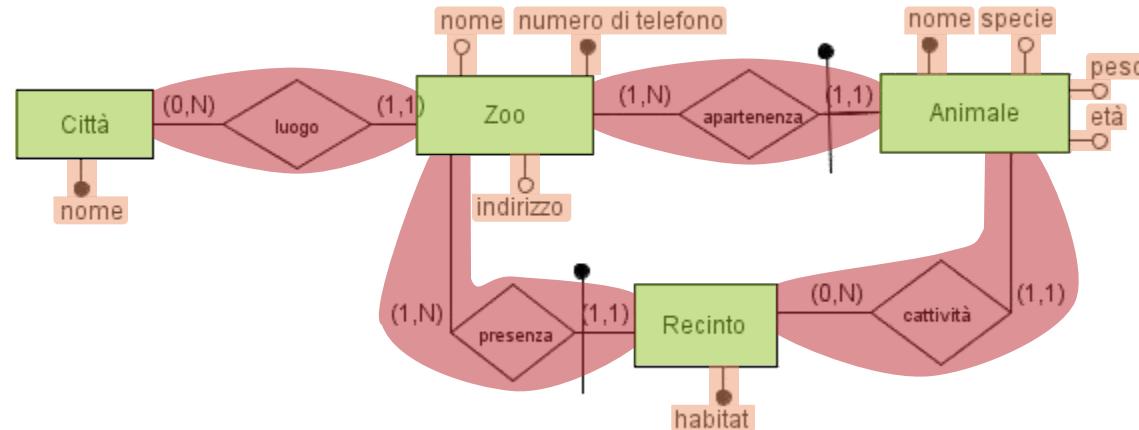




Modello entità/associazioni

➤ Il modello ER utilizza:

- Entità
- Attributi
- Associazioni





Modello entità/associazioni

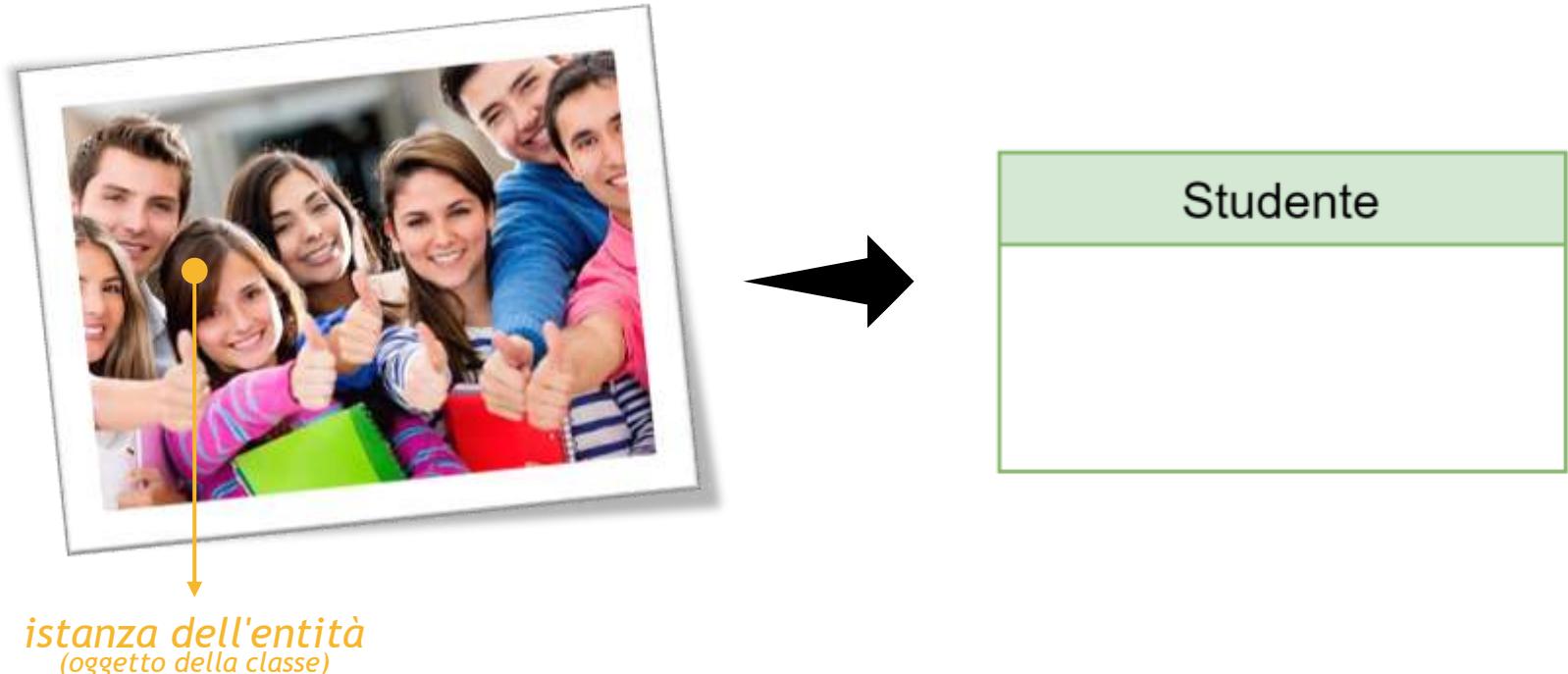
- Noi utilizzeremo una notazione alternativa, detta **crow's foot**





Entità

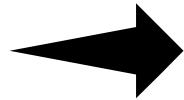
- Le entità rappresentano **classi di oggetti** (concreti o astratti) aventi proprietà comuni
- Si rappresentano con rettangoli, contenenti il nome dell'entità (al singolare)





Attributi

- Le entità⁽¹⁾ possono essere descritte usando una serie di **attributi** (o proprietà)
- Tutte le istanze della stessa entità hanno gli stessi attributi



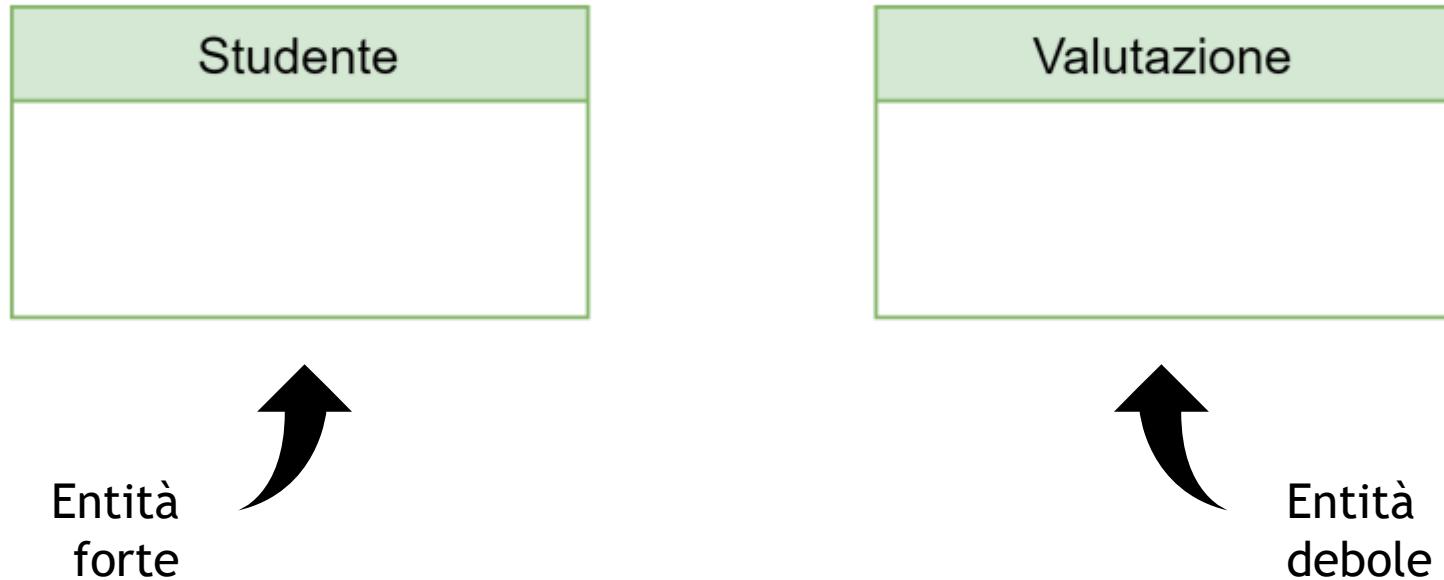
Studente
matricola
nome
cognome
data di nascita

⁽¹⁾ ma anche alcune associazioni



Entità deboli ed entità forti

- Le entità che richiedono altre entità per essere identificate sono dette **deboli**; altrimenti si dicono **forti**.





Chiave primaria

- Si indica con il termine **chiave** o **chiave primaria (*primary key*)** un attributo, o un insieme minimale di attributi, che permette di distinguere le istanze di una stessa entità.
- L'attributo chiave viene sottolineato oppure affiancato dalla sigla (pk)

Studente	
	<u>matricola</u> (pk)
	nome
	cognome
	data di nascita



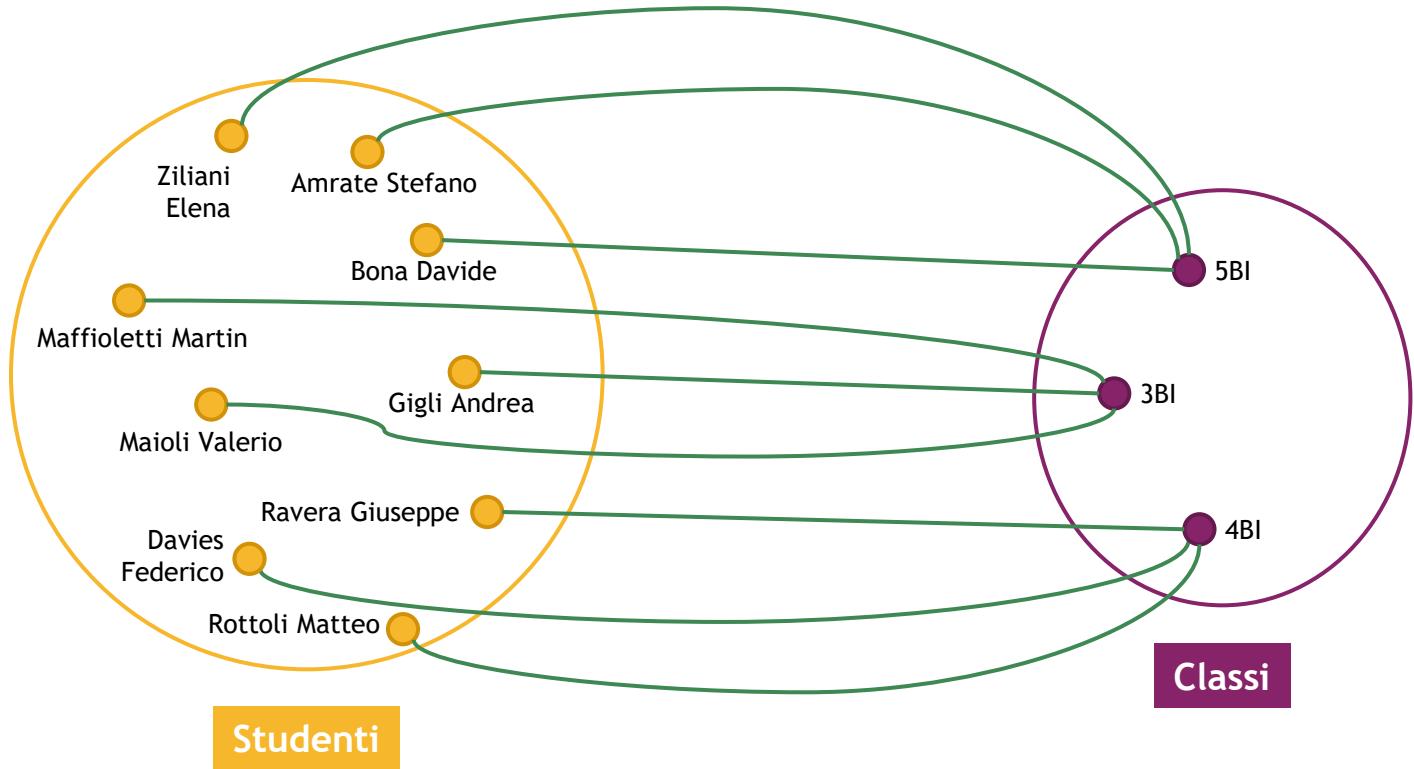
Attributi

- Un attributo è **opzionale** quando può non essere valorizzato (avere valore *null*)
- Un attributo è **derivato** quando il suo valore può essere calcolato con altri attributi (es. età)
 - Gli attributi derivati non vanno aggiunti al modello.



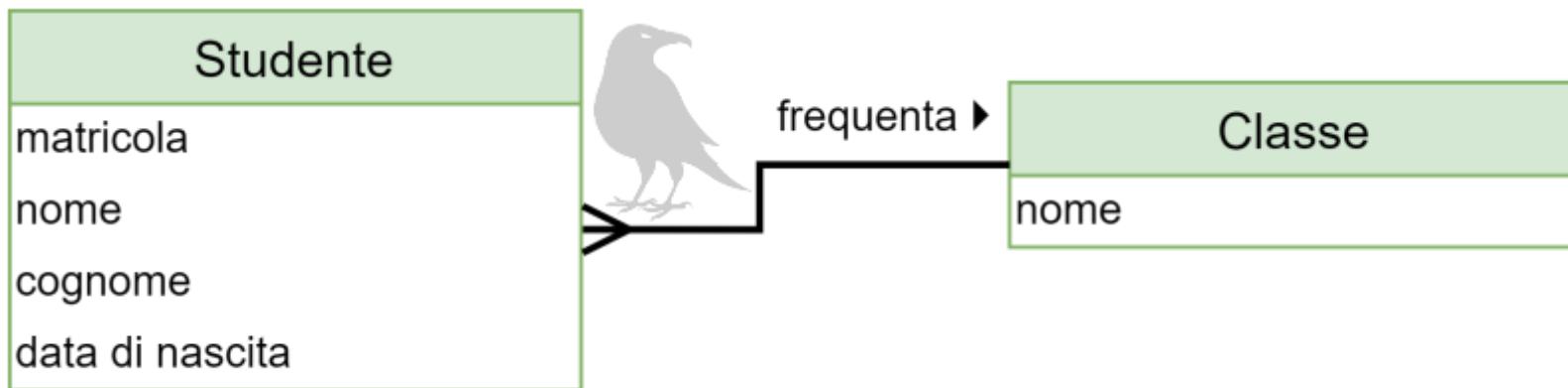
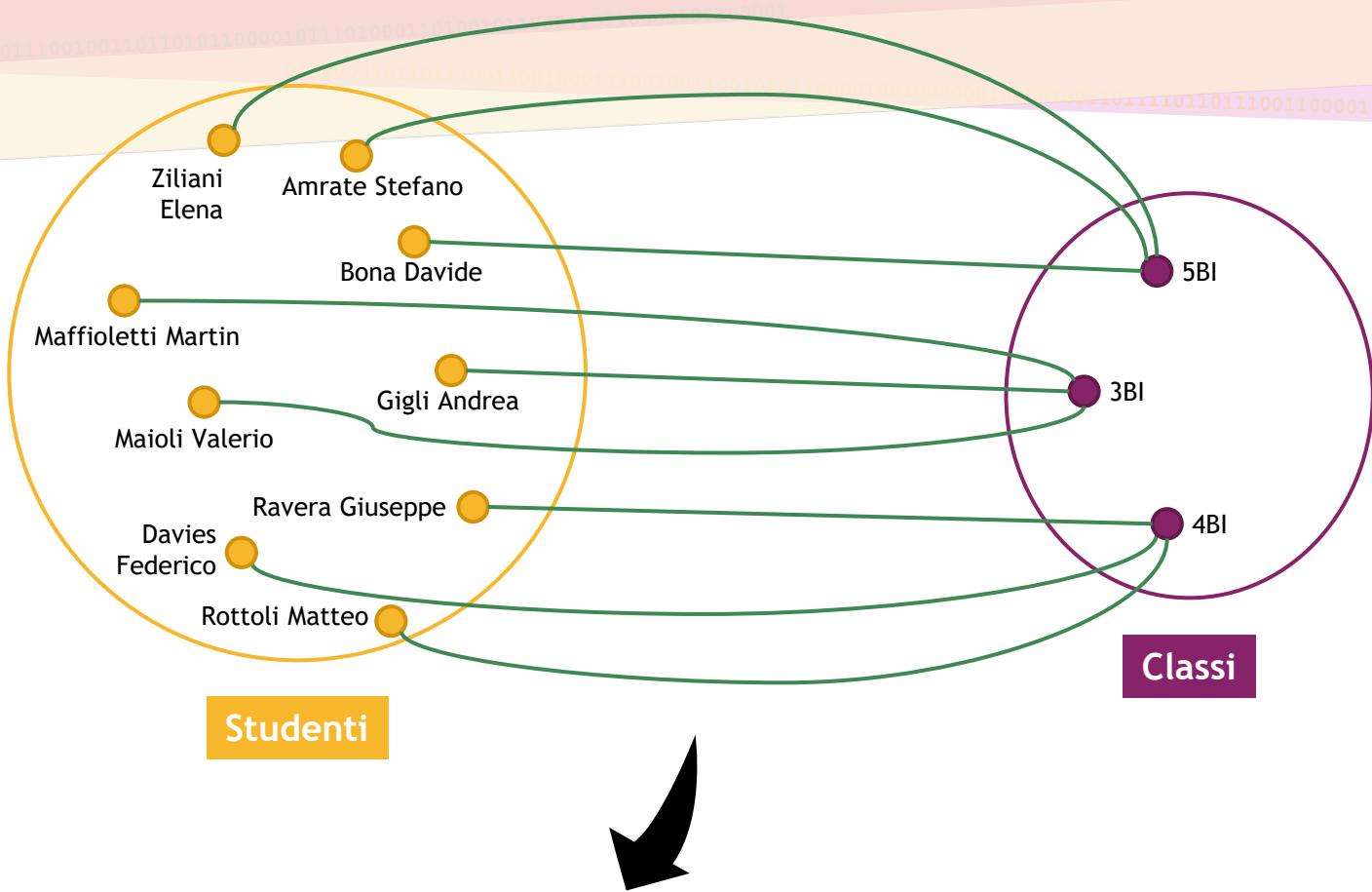
Associazioni

► Le associazioni rappresentano i legami logici tra entità.





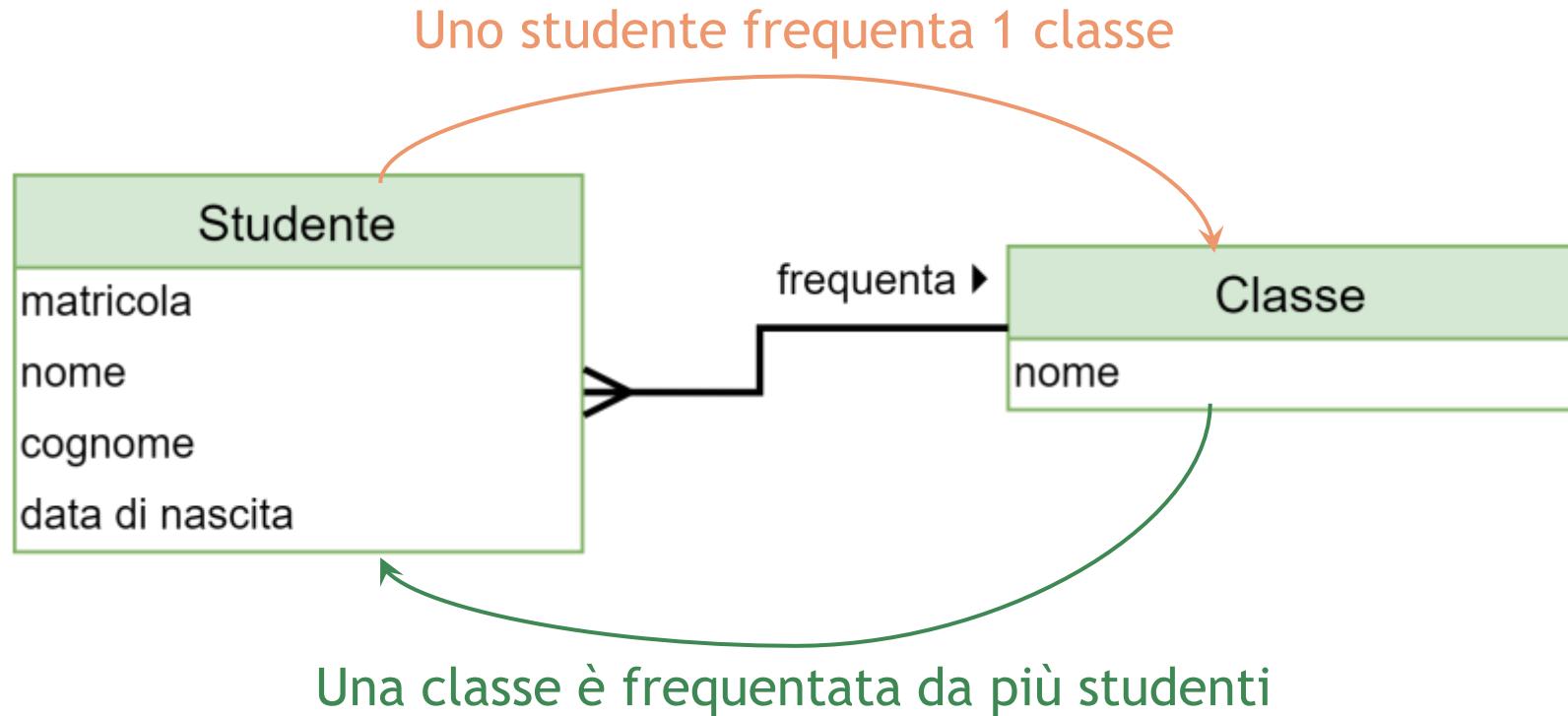
Associazioni





Associazioni

- Un'associazione ha una **descrizione** e può essere letta in due versi.





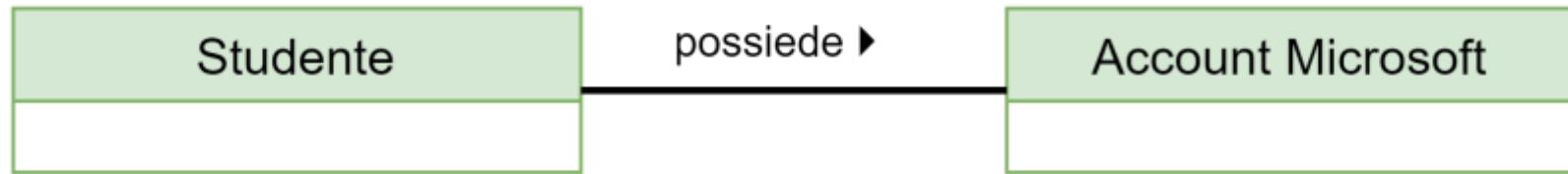
Associazioni

- Le associazioni binarie (tra due entità), con riferimento alle **cardinalità massime** delle istanze delle entità coinvolte, vengono così classificate:
 - Associazioni uno-a-uno (1:1)
 - Associazioni uno-a-molti (1:N)
 - Associazioni molti-a-molti (N:M)



Associazioni uno-a-uno (1:1)

- Quando a ogni istanza della prima entità corrisponde al più un'istanza della seconda entità e viceversa.

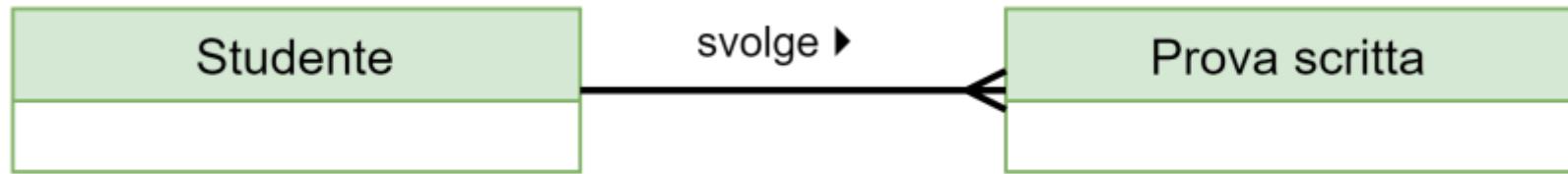


Uno studente possiede un account Microsoft,
Un account Microsoft è di un singolo studente



Associazioni uno-a-molti (1:N)

- Quando esiste un'istanza della prima entità cui corrisponde più di un'istanza della seconda entità, ma a ogni istanza della seconda entità corrisponde al più un'istanza della prima entità.

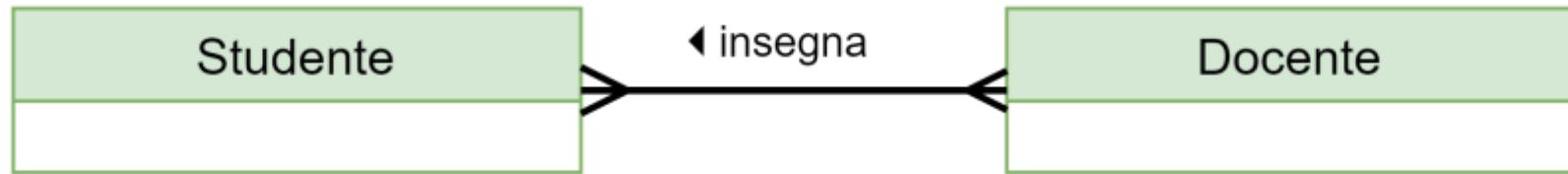


Uno studente svolge più prove scritte,
una prova scritta è di un singolo studente



Associazioni multi-a-molti (N:M)

- Quando esiste un'istanza della prima entità cui corrisponde più di un'istanza della seconda entità e viceversa.

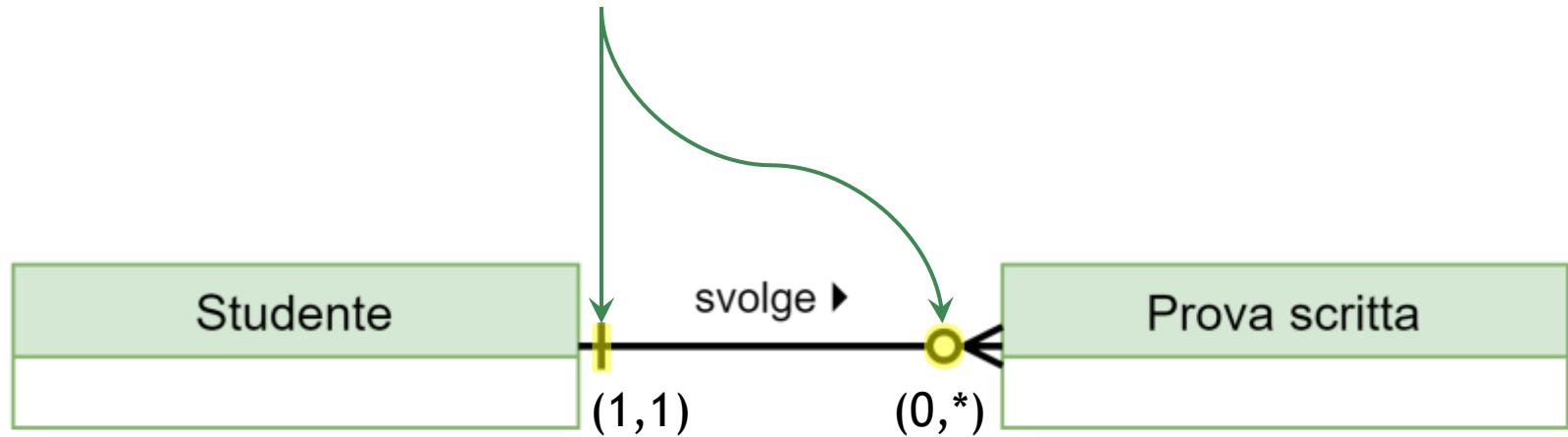


Un docente insegna a più studenti,
uno studente viene istruito da più docenti



Associazioni

- Nelle associazioni è possibile evidenziare anche le cardinalità minime.

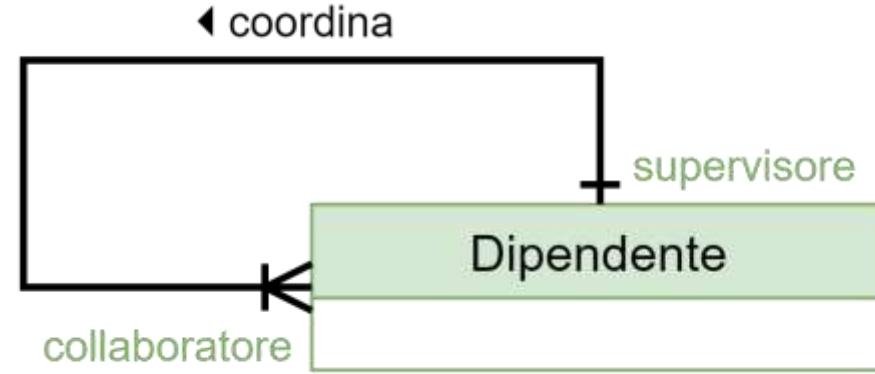


Uno studente svolge zero o più prove scritte,
una prova scritta è di uno e un solo studente

Associazioni ricorsive



- Sono associazioni tra un'entità e se stessa.

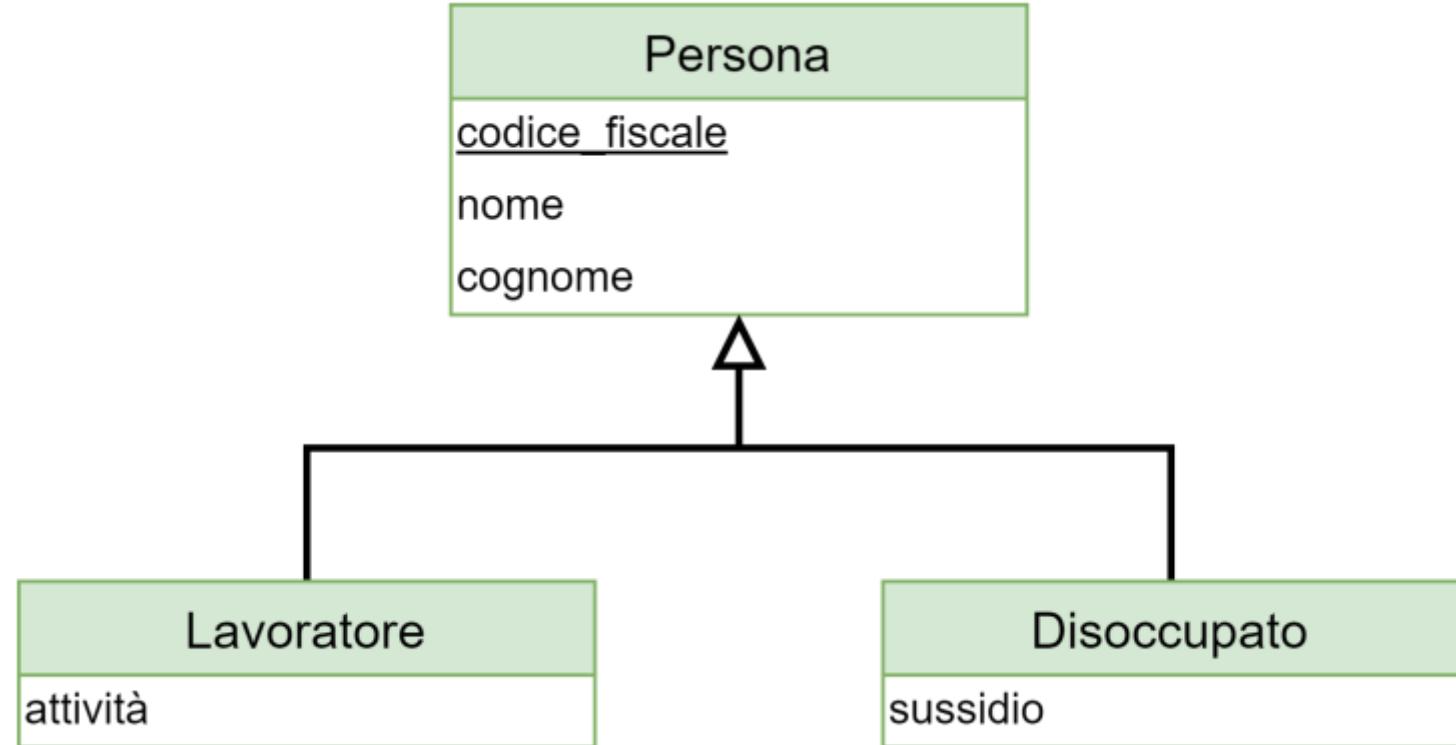


Un dipendente (supervisore) coordina uno o più dipendenti (collaboratori),
un dipendente (collaboratore) viene supervisionato da un altro dipendente (supervisore)



Gerarchie

- ▶ Tra entità possono esserci legami gerarchici, come nella OOP.





Gerarchie

➤ Classificazione

	Totale Ogni oggetto della superclasse appartiene a una sottoclasse	Parziale Possono esistere oggetti della superclasse che non sono in alcuna sottoclasse
Esclusiva Le sottoclassi non hanno oggetti in comune	<pre> graph TD Persona[Persona] --> Minorenne[Minorenne] Persona --> Maggiorenne[Maggiorenne] </pre>	<pre> graph TD Persona[Persona] --> Lavoratore[Lavoratore] Persona --> Disoccupato[Disoccupato] </pre>
Sovrapposta Le sottoclassi possono avere oggetti in comune	<pre> graph TD Persona[Persona] --> Genitore[Genitore] Persona --> Figlio[Figlio] </pre>	<pre> graph TD Persona[Persona] --> Studente[Studente] Persona --> Musicista[Musicista] </pre>



Modello entità/associazioni

- Software per realizzare diagrammi ER:
<https://app.diagrams.net/>

- Completo
ma non
immediato

