

Metodologie e Modelli Progettuali

Perché?

- Proviamo a **modellare un'applicazione** definendo direttamente lo **schema logico** della base di dati:
 - **da dove** cominciamo?
 - rischiamo di **perderci** subito **nei dettagli**
 - dobbiamo pensare subito a come **correlare** le varie **tabelle** (chiavi, etc.)
 - il modello relazionale è “**rigido**”

Progettazione delle basi di dati

- È una delle attività del processo di sviluppo dei sistemi informativi
- Va quindi inquadrata in un contesto più generale:
 - il **ciclo di vita** (*lifecycle*) dei sistemi informativi:
 - **insieme e organizzazione temporale delle attività** svolte da **analisti, progettisti, utenti**, nello **sviluppo** e **nell'uso** dei sistemi informativi
 - **attività iterativa**, quindi ciclo

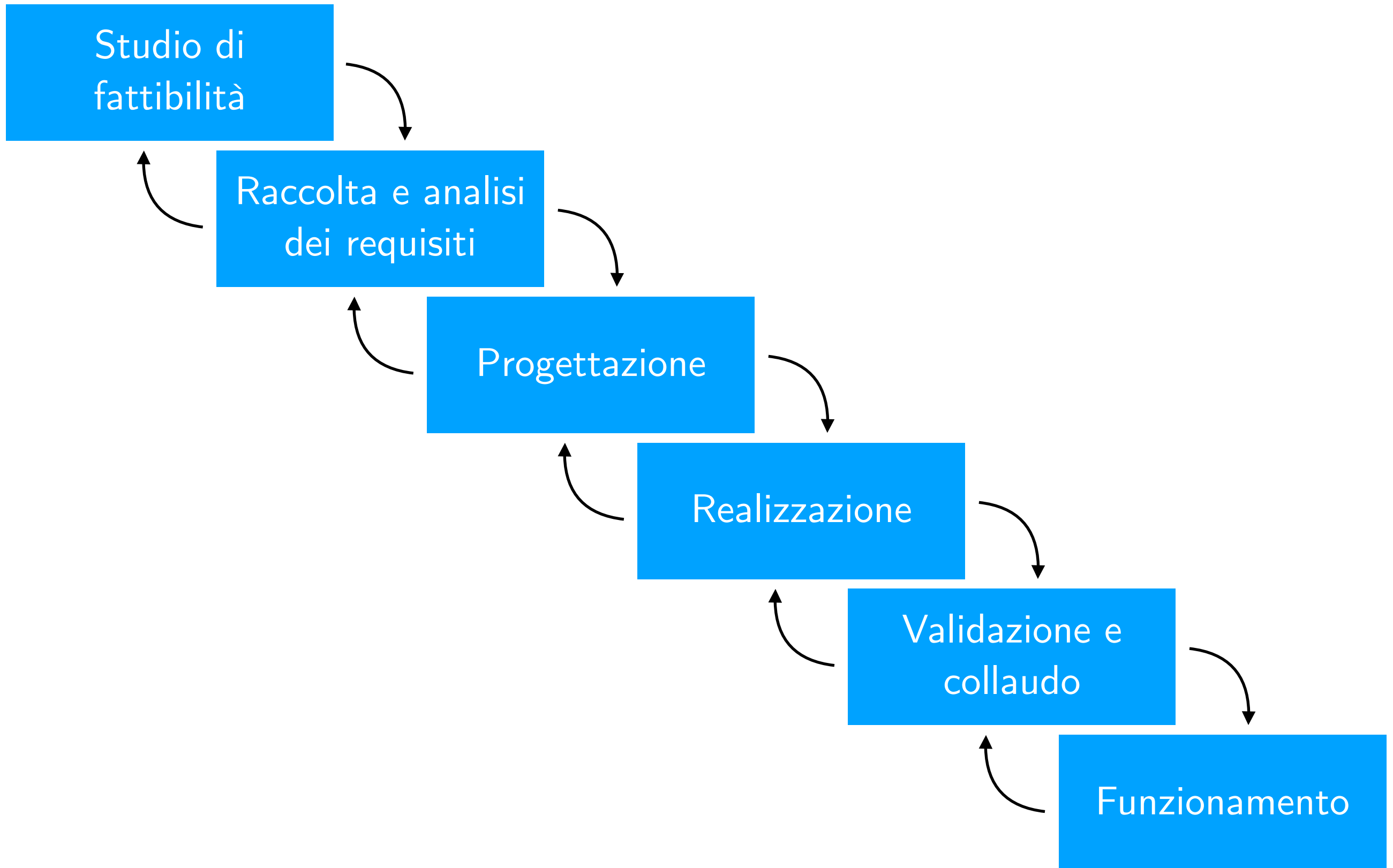
Un buon progetto

- I passi del ciclo di vita per essere “ben fatti” richiedono in generale un linguaggio/modello per descrivere il sistema da progettare
- Per le basi di dati, quindi, la metodologia di progetto deve essere basata su
 - modelli per rappresentare i dati che siano facili da usare
 - decomposizione delle attività in fasi (e/o livelli)
 - strategie e criteri di scelta nei vari passi

Modello per il ciclo di vita

- Il primo modello da scegliere è quello per il ciclo di vita
- Esistono vari modelli: il più vecchio è il **modello a cascata** (*waterfall model*)
- Nel modello a cascata **le fasi sono ordinate e “non ripetibili”**

Fasi del ciclo di vita



Fasi del ciclo di vita

- Studio di fattibilità: definizione costi e priorità
- Raccolta e analisi dei requisiti: studio delle proprietà del sistema
- Progettazione: dati e funzioni
- Realizzazione: implementazione
- Validazione e collaudo: sperimentazione
- Funzionamento: il sistema diventa operativo in produzione (*shipping*)

Raccolta e analisi dei requisiti

- Ci sono **due sotto-fasi**:
 - **acquisizione dei requisiti**: il reperimento dei requisiti è un'attività difficile e non standardizzabile
 - **analisi dei requisiti**: l'attività di analisi inizia con i primi requisiti raccolti e spesso indirizza verso altre acquisizioni
 - **Linguaggi per definire i requisiti**, ad esempio in **UML**

Come si acquisiscono i requisiti?

- Direttamente dagli **utenti**:
 - interviste
 - documentazione apposita
- Da **documentazione esistente**:
 - normative (leggi, regolamenti di settore)
 - regolamenti interni, procedure aziendali
 - realizzazioni preesistenti

Interazione con gli utenti

- Problemi
 - **utenti diversi** possono fornire **informazioni diverse**
 - **utenti a livello più alto** hanno spesso una visione **più ampia** ma **meno dettagliata**
 - spesso l'acquisizione dei requisiti avviene “**per raffinamenti successivi**”
- Spunti:
 - effettuare **spesso** verifiche di comprensione e coerenza
 - verificare anche per mezzo di **esempi** (generalisti e relativi a casi limite)
 - **richiedere definizioni** e classificazioni
 - far evidenziare gli **aspetti essenziali** rispetto a quelli **marginali** (ranking dei requisiti)

Interazione con gli utenti tramite documentazione

- Regole generali:
 - **standardizzare la struttura delle frasi**
 - **separare le frasi sui dati da quelle sulle funzioni**
 - **organizzare termini e concetti**
 - costruire un **glossario dei termini**
 - **unificare i termini** (individuare i sinonimi)
 - **rendere esplicito il riferimento** fra termini
 - **riorganizzare le frasi per concetti**

Esempio

Società di formazione (1)

Si vuole realizzare una base di dati per una società che eroga corsi: di ogni corso vogliamo rappresentare i dati dei partecipanti e dei docenti. Per gli studenti (circa 5000), identificati da un codice, si vuole memorizzare il codice fiscale, il cognome, l'età, il sesso, il luogo di nascita, il nome dei loro attuali datori di lavoro, i posti dove hanno lavorato in precedenza insieme al periodo, l'indirizzo e il numero di telefono, i corsi che hanno già frequentato (le materie sono in tutto circa 200) e il giudizio finale.

Esempio

Società di formazione (2)

Rappresentiamo anche i corsi attualmente attivi e, per ogni giorno, i luoghi e le ore dove sono tenute le lezioni. I corsi hanno un codice, un titolo e possono avere varie edizioni con date di inizio e fine e numero di partecipanti. Se gli studenti sono liberi professionisti, vogliamo conoscere l'area di interesse e, se lo possiedono, il titolo. Per quelli che lavorano alle dipendenze di altri, vogliamo conoscere invece il loro livello e la posizione ricoperta.

Esempio

Società di formazione (3)

Per gli insegnanti (circa 300), rappresentiamo il cognome, l'età, il posto dove sono nati, il nome del corso che insegnano, quelli che hanno insegnato nel passato e quelli che possono insegnare. Rappresentiamo anche tutti i loro recapiti telefonici. I docenti possono essere dipendenti interni della società o collaboratori esterni.

Glossario dei termini

Termine	Descrizione	Sinonimi	Collegamenti
Partecipante	Persona che partecipa ai corsi.	Studente	Corso Società
Docente	Docente dei corsi. Può essere esterno.	Insegnante	Corso
Corso	Corso organizzato dalla società. Può avere più edizioni	Materia	Docente
Datore di lavoro	Ente presso cui i partecipanti lavorano o hanno lavorato.	Posto	Partecipante

**Strutturazione dei requisiti
in gruppi di frasi omogenee**

Frase di carattere generale

Si vuole realizzare una base di dati per una società che eroga corsi: di ogni corso vogliamo rappresentare i dati dei partecipanti e dei docenti.

Frase relative ai partecipanti

Per i partecipanti (circa 5000), identificati da un codice, rappresentiamo il codice fiscale, il cognome, l'età, il sesso, la città di nascita, i nomi dei loro attuali datori di lavoro e di quelli precedenti (insieme alle date di inizio e fine rapporto), le edizioni dei corsi che stanno attualmente frequentando e quelli che hanno frequentato nel passato, con la relativa votazione finale in decimi.

Frase relative ai datori di lavoro

Relativamente ai datori di lavoro presenti e passati dei partecipanti, rappresentiamo il nome, l'indirizzo e il numero di telefono.

Frase relative ai corsi

Per i corsi (circa 200), rappresentiamo il titolo e il codice, le varie edizioni con date di inizio e fine e, per ogni edizione, rappresentiamo il numero di partecipanti e il giorno della settimana, le aule e le ore dove sono tenute le lezioni.

Frase relative a tipi specifici di partecipanti

Per i partecipanti che sono liberi professionisti, rappresentiamo l'area di interesse e, se lo possiedono, il titolo professionale. Per i partecipanti che sono dipendenti, rappresentiamo invece il loro livello e la posizione ricoperta.

Fraasi relative ai docenti

Per i docenti (circa 300), rappresentiamo il cognome, l'età, la città di nascita, tutti i numeri di telefono, il titolo del corso che insegnano, di quelli che hanno insegnato nel passato e di quelli che possono insegnare. I docenti possono essere dipendenti interni della società di formazione o collaboratori esterni.

Progettazione

- La **progettazione** è una fase del ciclo di vita
- Per un sistema software la progettazione consta fondamentalmente di **due aspetti**:
 - **progettazione dei dati**
 - nel caso di sistemi informativi, il progetto dei dati ha un **ruolo centrale**
 - **progettazione delle applicazioni**

Progettare per livelli di astrazione

- **Livello concettuale.** Esprime i requisiti di un sistema in una descrizione adatta all'analisi dal punto di vista esterno
- **Livello logico.** Evidenzia l'organizzazione dei dati dal punto di vista del loro contenuto informativo, descrivendo la struttura di ciascun record e i collegamenti tra record diversi.
- **Livello fisico.** A questo livello la base di dati è vista come un insieme di blocchi fisici su disco. Qui viene decisa l'allocazione dei dati e le modalità di memorizzazione dei dati sul disco.

Requisiti della base di dati

Progettazione
concettuale

CHE COSA: analisi dei requisiti

Schema concettuale

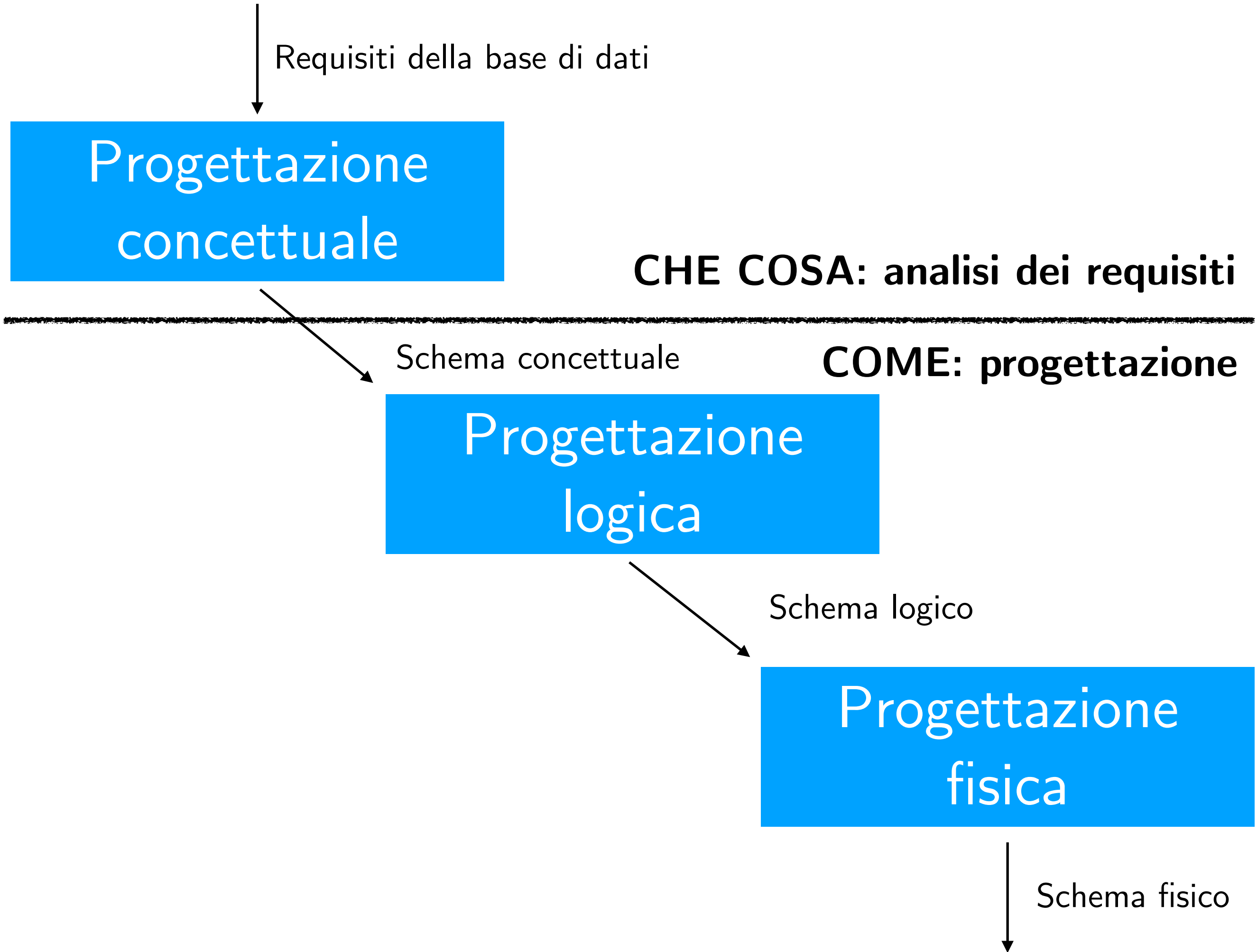
COME: progettazione

Progettazione
logica

Schema logico

Progettazione
fisica

Schema fisico



Modello dei dati

- Insieme di **costrutti** utilizzati per **organizzare i dati** di interesse e **descrivere la dinamica**
- Componente fondamentale: **meccanismi di strutturazione** (o **costruttori di tipo**)
- Come nei linguaggi di programmazione esistono meccanismi che permettono di definire nuovi tipi, così ogni modello dei dati prevede alcuni **costruttori**
- Ad esempio, il **modello relazionale** prevede il **costruttore relazione**, che permette di definire insiemi di record omogenei

Schemi e istanze

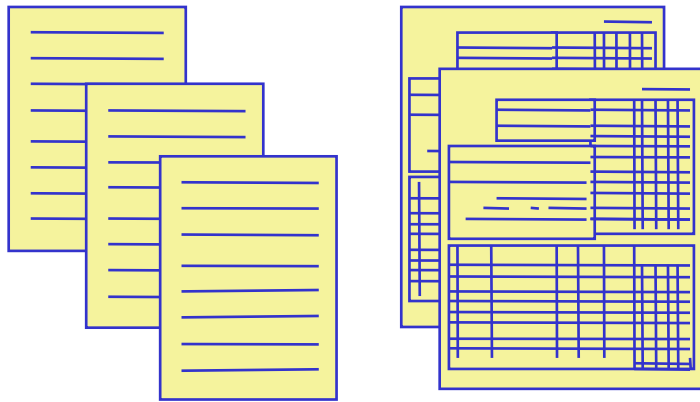
- In ogni **base di dati** esistono:
 - lo **schema**, sostanzialmente invariante nel tempo, che ne descrive la struttura
 - nel modello relazionale, le **intestazioni** delle tabelle
 - l'**istanza**, i valori attuali, che possono cambiare anche molto rapidamente
 - nel modello relazionale, il **corpo** di ciascuna tabella

Principali Tipi di Modelli

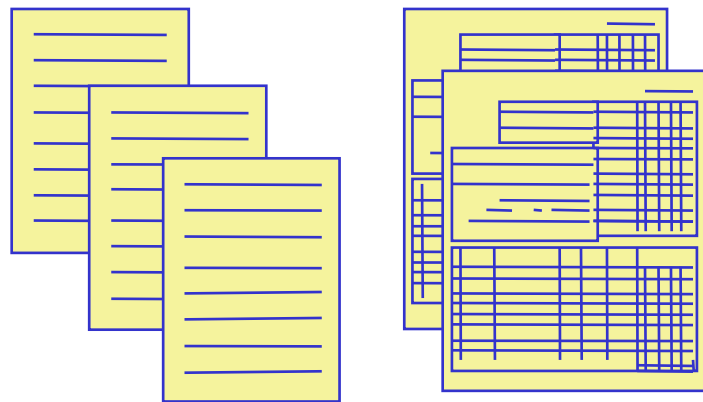
- **Modelli logici:** utilizzati nei DBMS esistenti per l'organizzazione dei dati
 - utilizzati dai programmi
 - indipendenti dalle strutture fisiche
 - esempi: **relazionale**, reticolare, gerarchico, a oggetti
- **Modelli concettuali:** permettono di rappresentare i dati in modo indipendente da ogni sistema
 - cercano di descrivere i concetti del mondo reale
 - sono utilizzati nelle fasi preliminari di progettazione
 - il più noto è il **modello Entità-Relazione** (***Entity-Relationship***)
 - useremo il termine in inglese per non confondersi con la relazione del modello relazionale

Passaggi tra Modelli

Passaggi tra Modelli



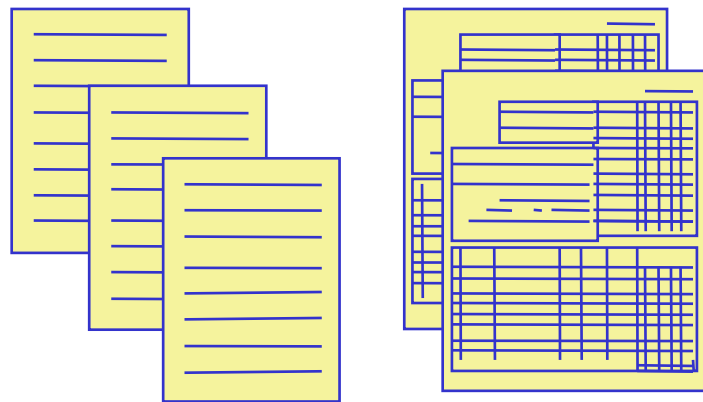
Passaggi tra Modelli



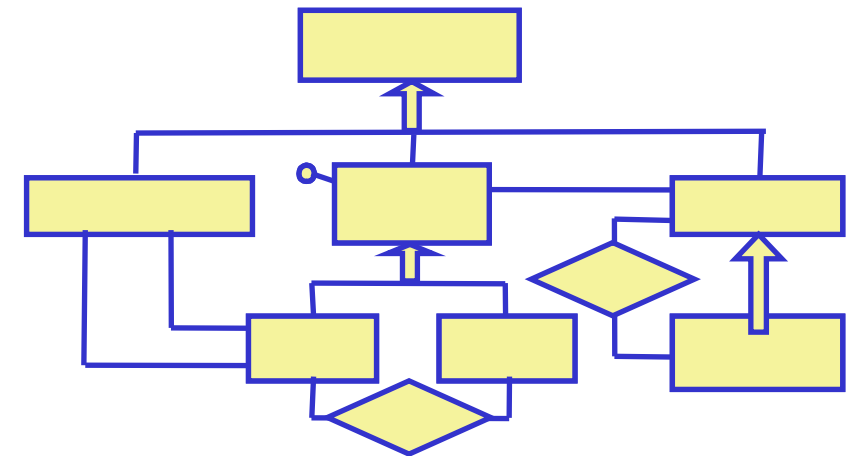
**Progettazione
concettuale**



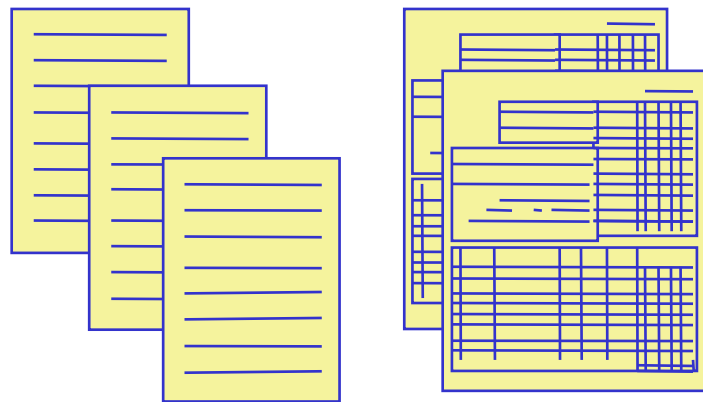
Passaggi tra Modelli



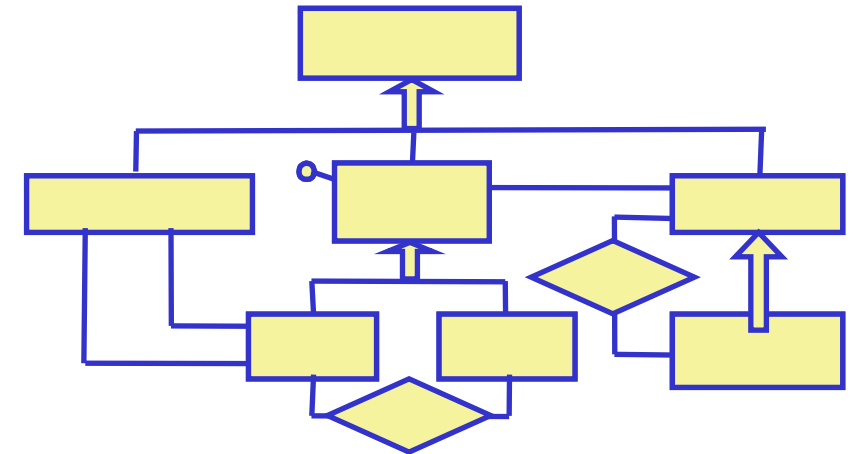
**Progettazione
concettuale**



Passaggi tra Modelli



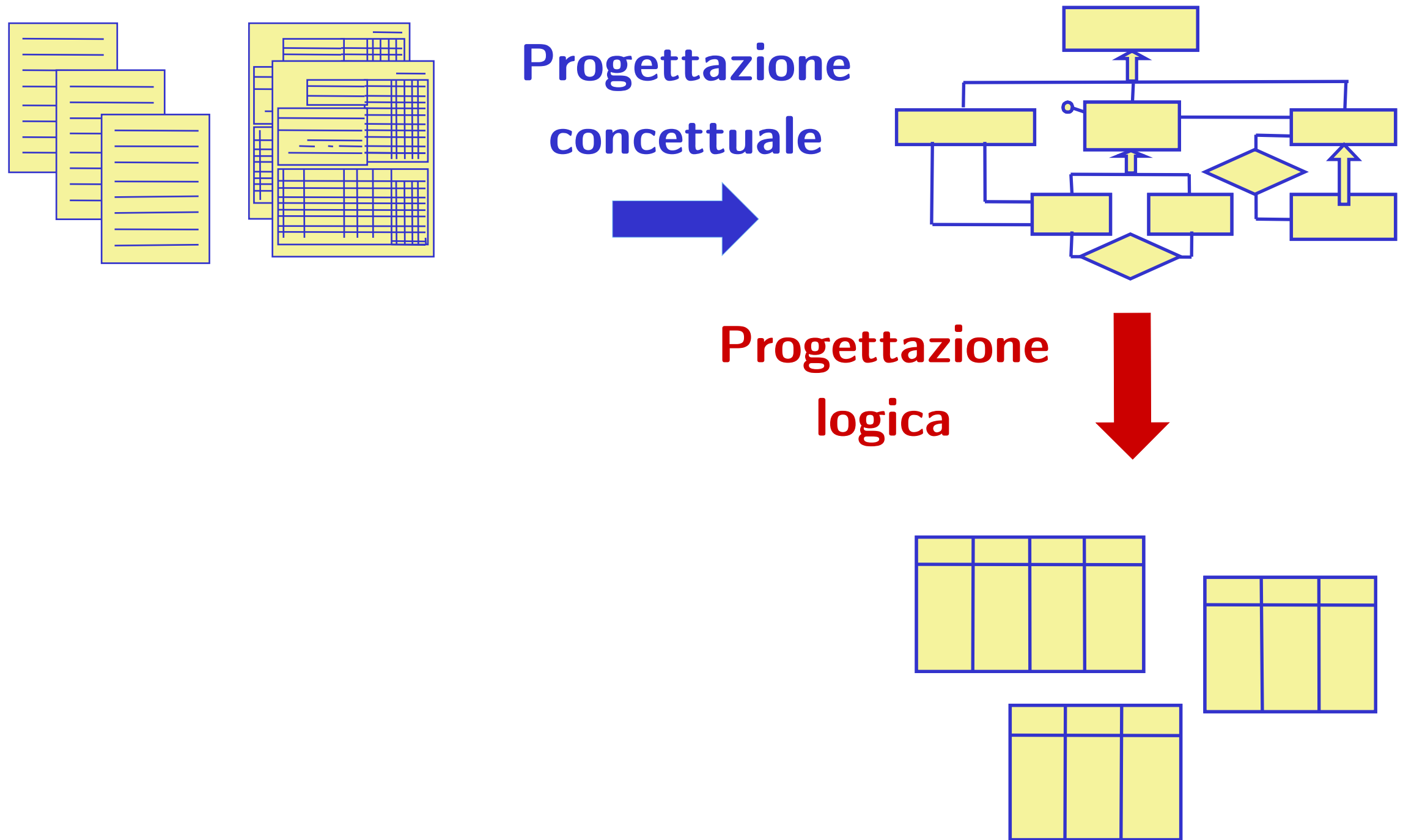
**Progettazione
concettuale**



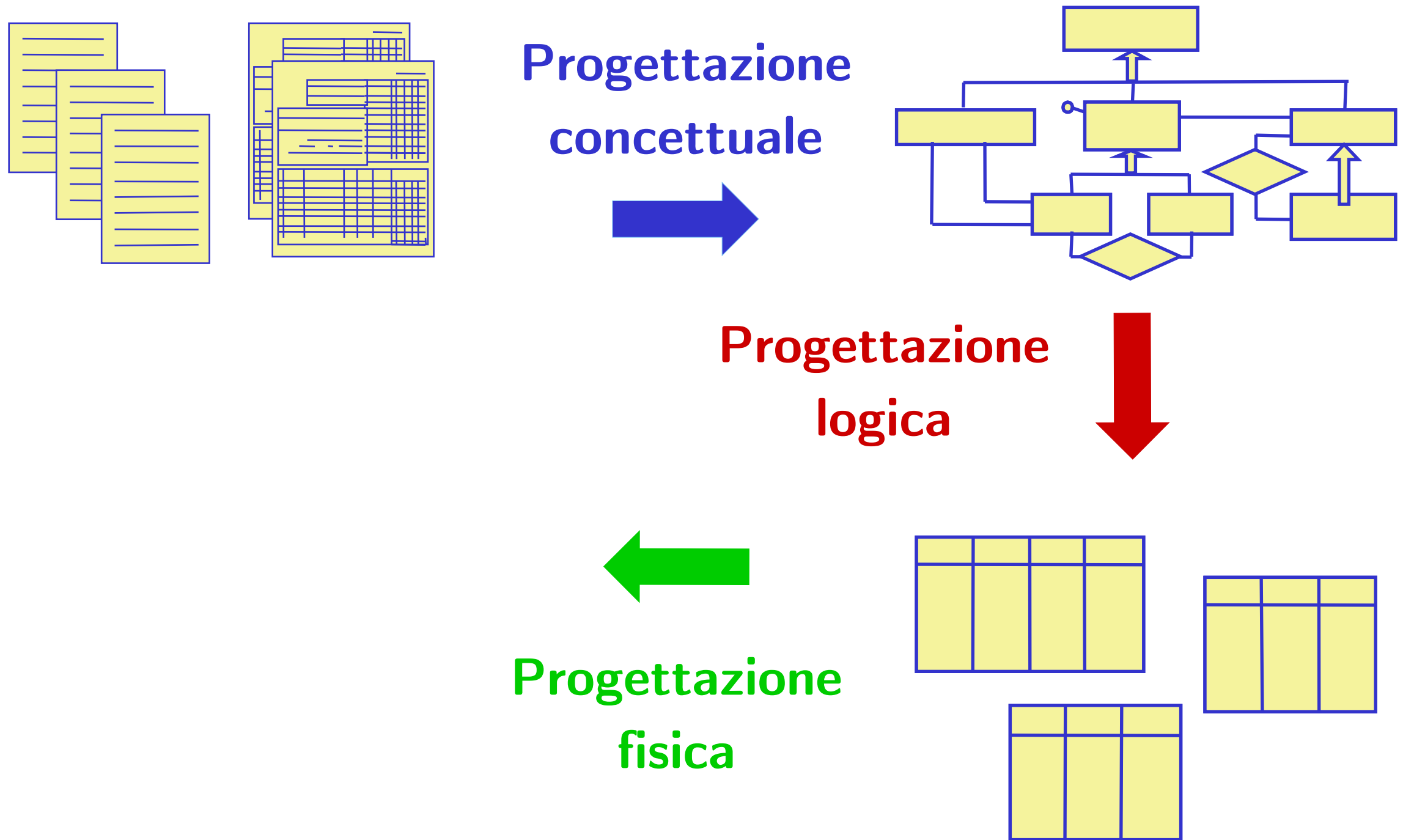
**Progettazione
logica**



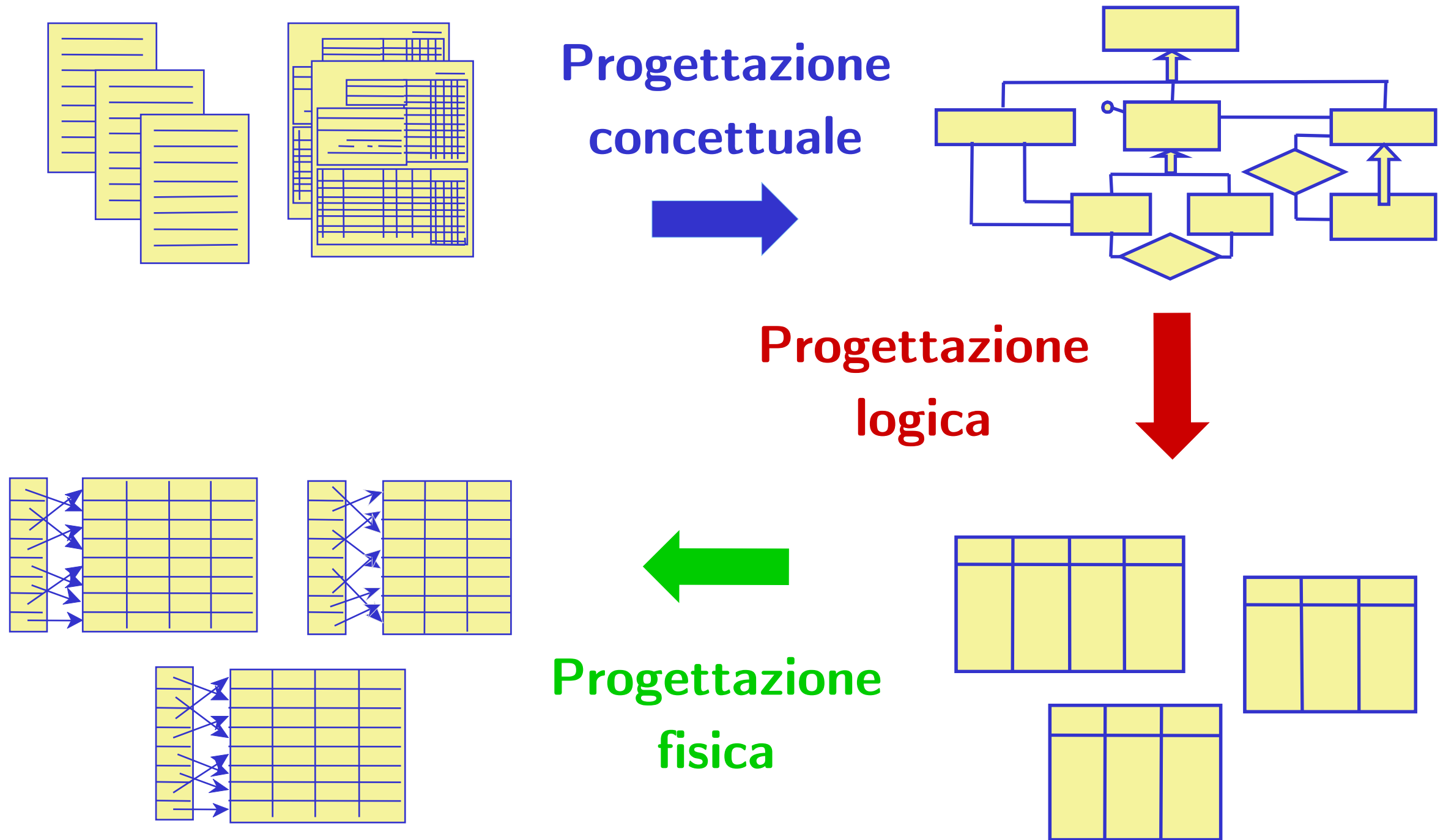
Passaggi tra Modelli



Passaggi tra Modelli



Passaggi tra Modelli



Modello E-R

- Il **modello E-R** (*Entity-Relationship*, P. P. Chen 1976) si è ormai affermato nelle metodologie di progetto e nei sistemi software di ausilio alla progettazione anche se in una versione leggermente diversa da quella originaria



Peter Pin-Shan Chen
1947

Costrutti del Modello E-R

- **Costrutti di base:**

- Entità
- *Relationship*
- Attributo

- **Altri costrutti:**

- Identificatore
- Generalizzazione
- ...

Entità

- **Classe di oggetti** (fatti, persone, cose) della applicazione di **interesse** con proprietà **comuni** e con esistenza “**autonoma**”
- Esempi:
 - impiegato, città, conto corrente, ordine, fattura
- **Occorrenza (o istanza) di entità**
 - **elemento della classe** (l'oggetto, la persona, ..., non un valore dei dati legati all'oggetto)
 - Per esempio, un “impiegato”, non so nulla di lui, ma esiste con proprietà note

Rappresentazione grafica delle entità

Impiegato

Dipartimento

Città

Vendita

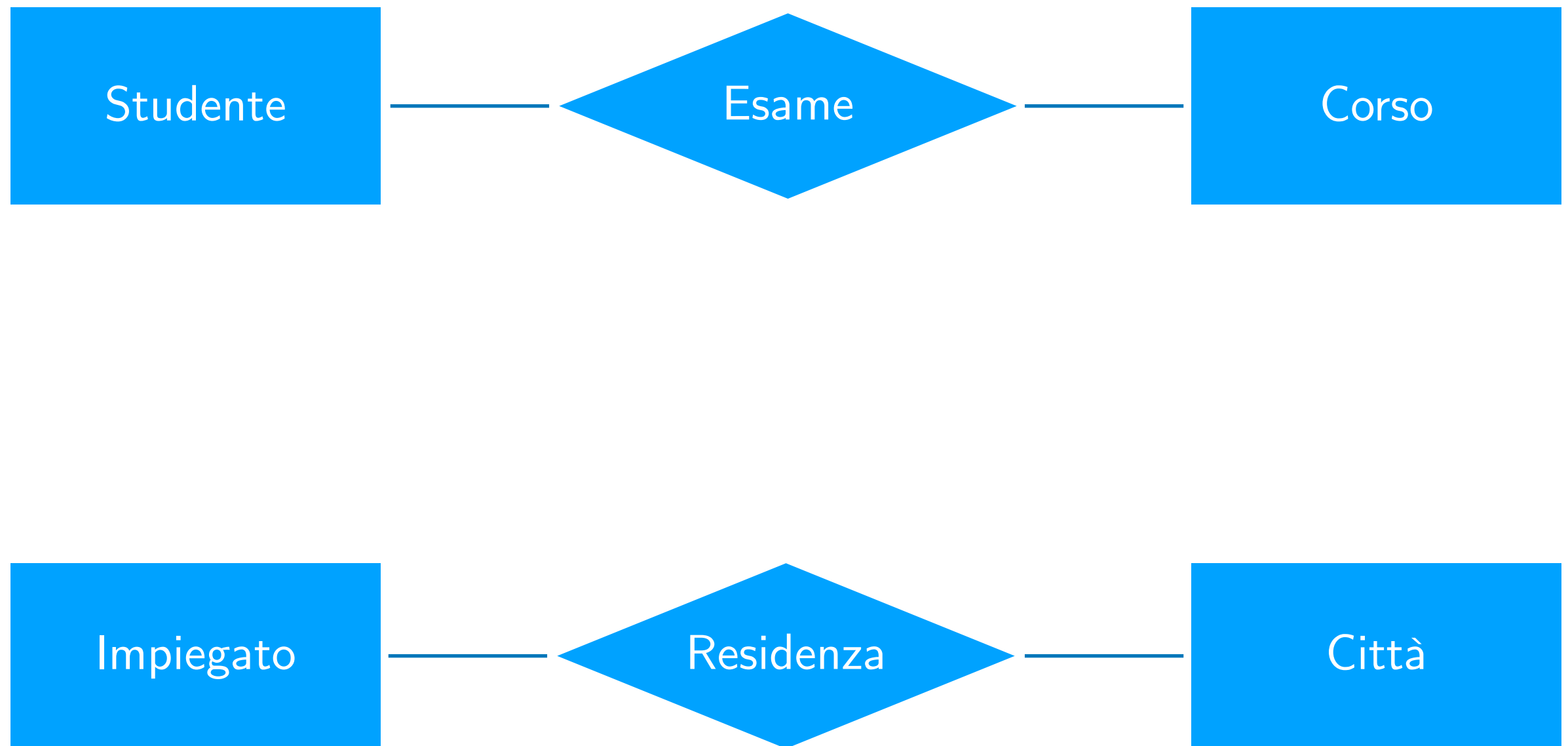
Caratteristiche delle entità

- Ogni entità ha un **nome** che la identifica **univocamente** nello schema:
 - nomi **espressivi**
 - opportune **convenzioni**
 - singolare

Relationship

- **Legame logico** fra due o più entità, rilevante nell'applicazione di interesse
- Esempi:
 - Residenza (fra persona e città)
 - Esame (fra studente e corso)
- Chiamata anche:
 - **relazione, correlazione, associazione**

Rappresentazione grafica delle *relationship*



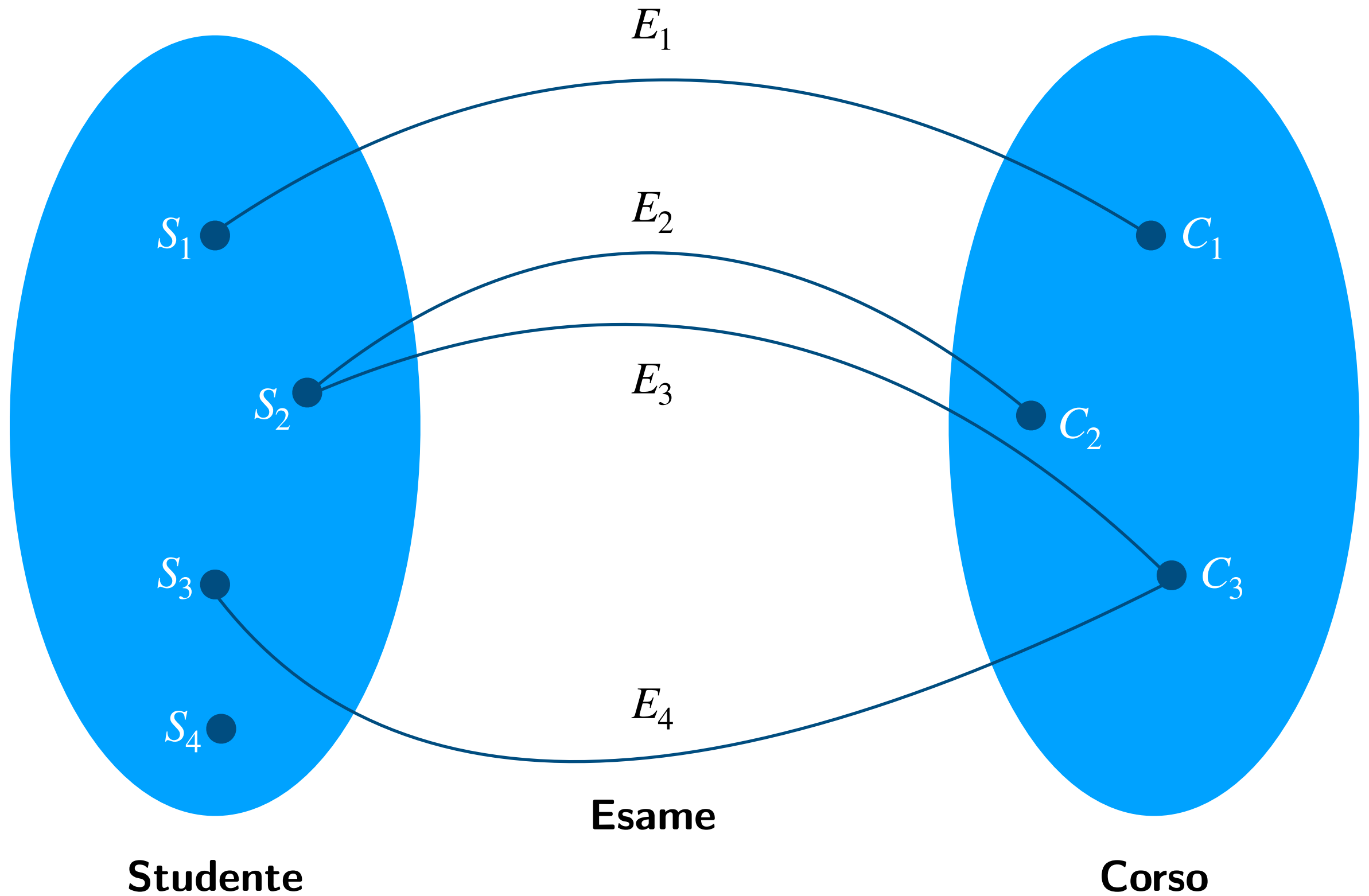
Caratteristiche delle *relationship*

- Ogni *relationship* ha un **nome** che la identifica **univocamente** nello schema:
 - nomi **espressivi**
 - opportune **convenzioni**
 - singolare
 - **sostantivi invece di verbi** (se possibile)
 - per non dare un verso alla *relationship*

Occorrenze delle *relationship*

- Una occorrenza di una *relationship* binaria è coppia di occorrenze di entità, una per ciascuna entità coinvolta
- Una **occorrenza** di una *relationship* n -aria è una **n -upla di occorrenze di entità, una per ciascuna delle n entità coinvolte**
- Nell'ambito di una *relationship* non ci possono essere **occorrenze** (coppie, n -uple) **ripetute**

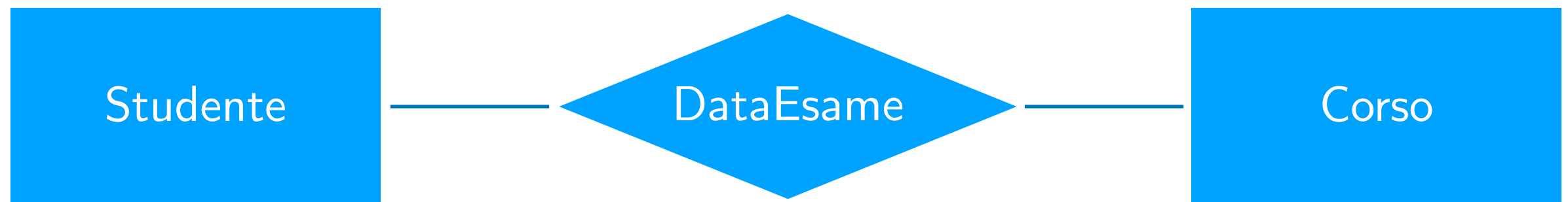
Esempi di occorrenze



Esempio

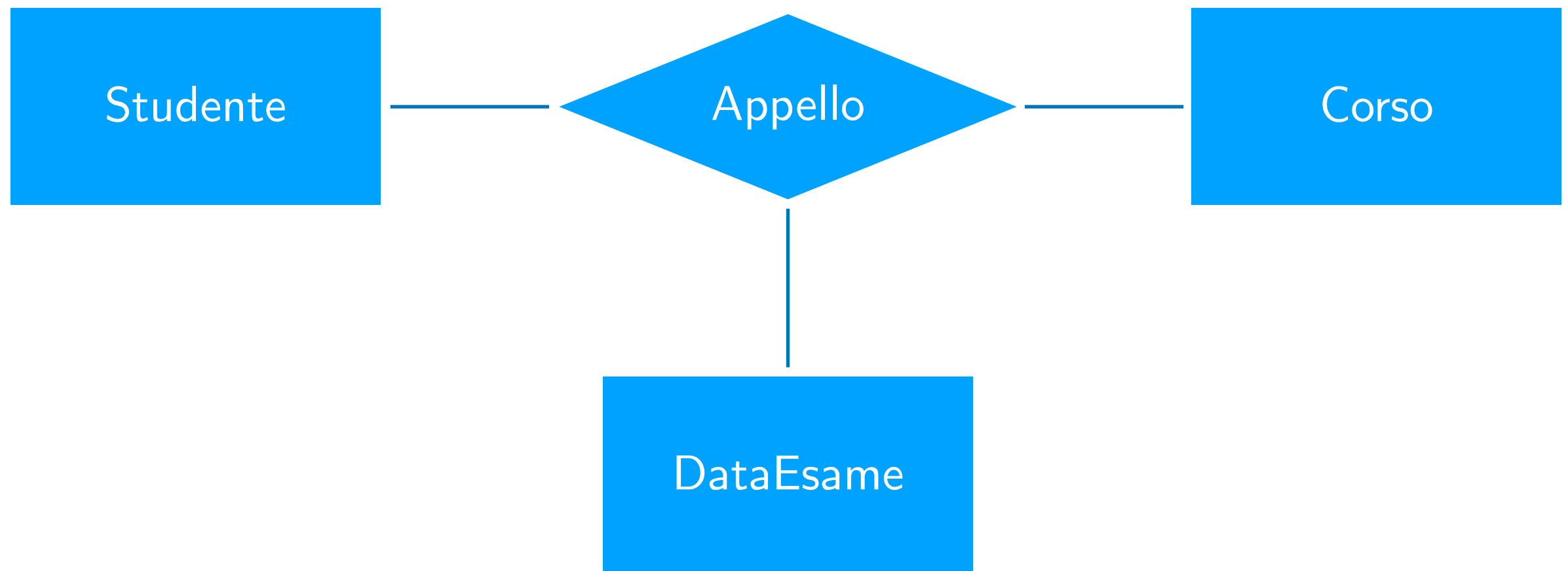
- Vogliamo progettare una base di dati per il **libretto elettronico**
- **Che uso** vogliamo fare di questa base di dati?
 - Supporto al servizio **statini** con la possibilità di calcolare **statistiche**

Prima Rappresentazione

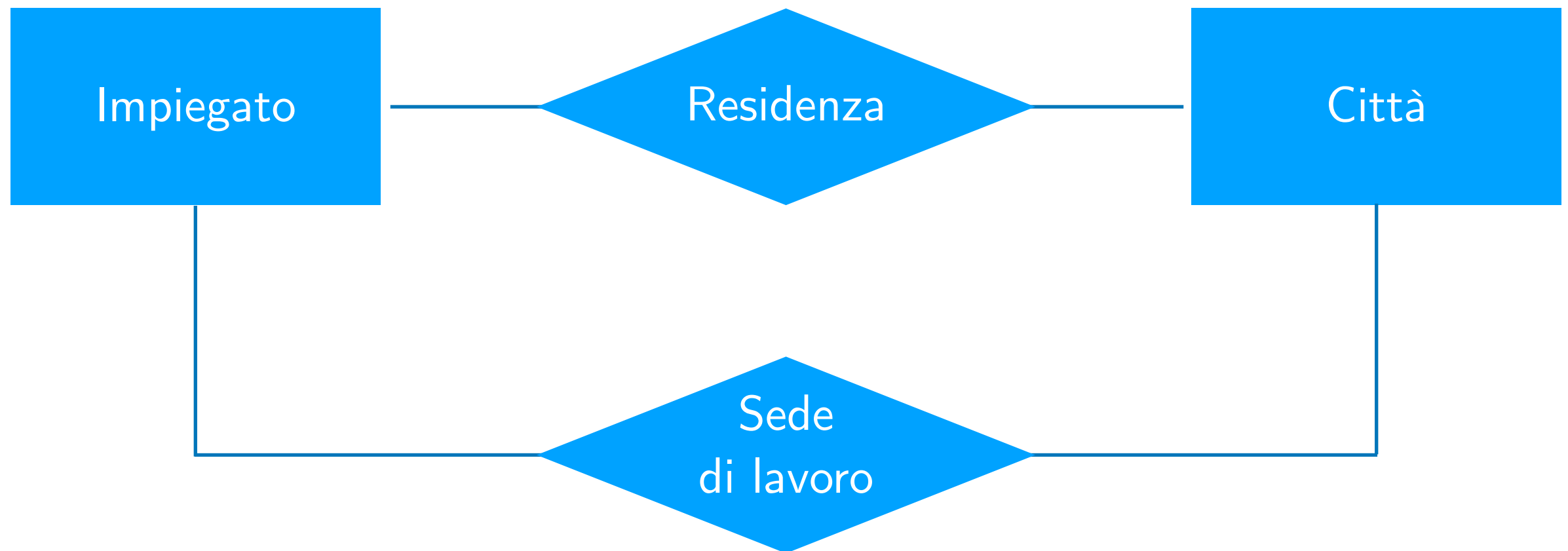


- E le **statistiche**?
 - Per esempio, numero di studenti di un corso che sostengono l'esame in un dato appello?

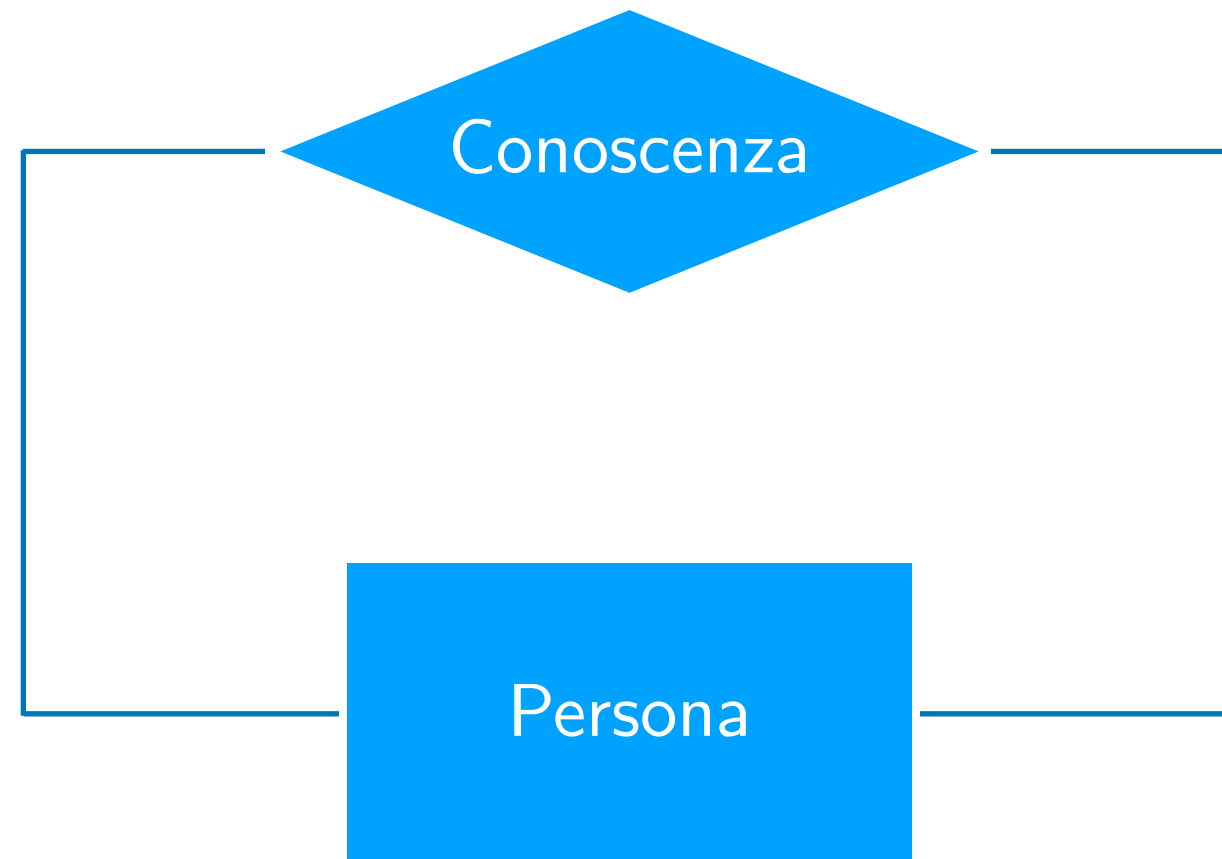
Seconda Rappresentazione



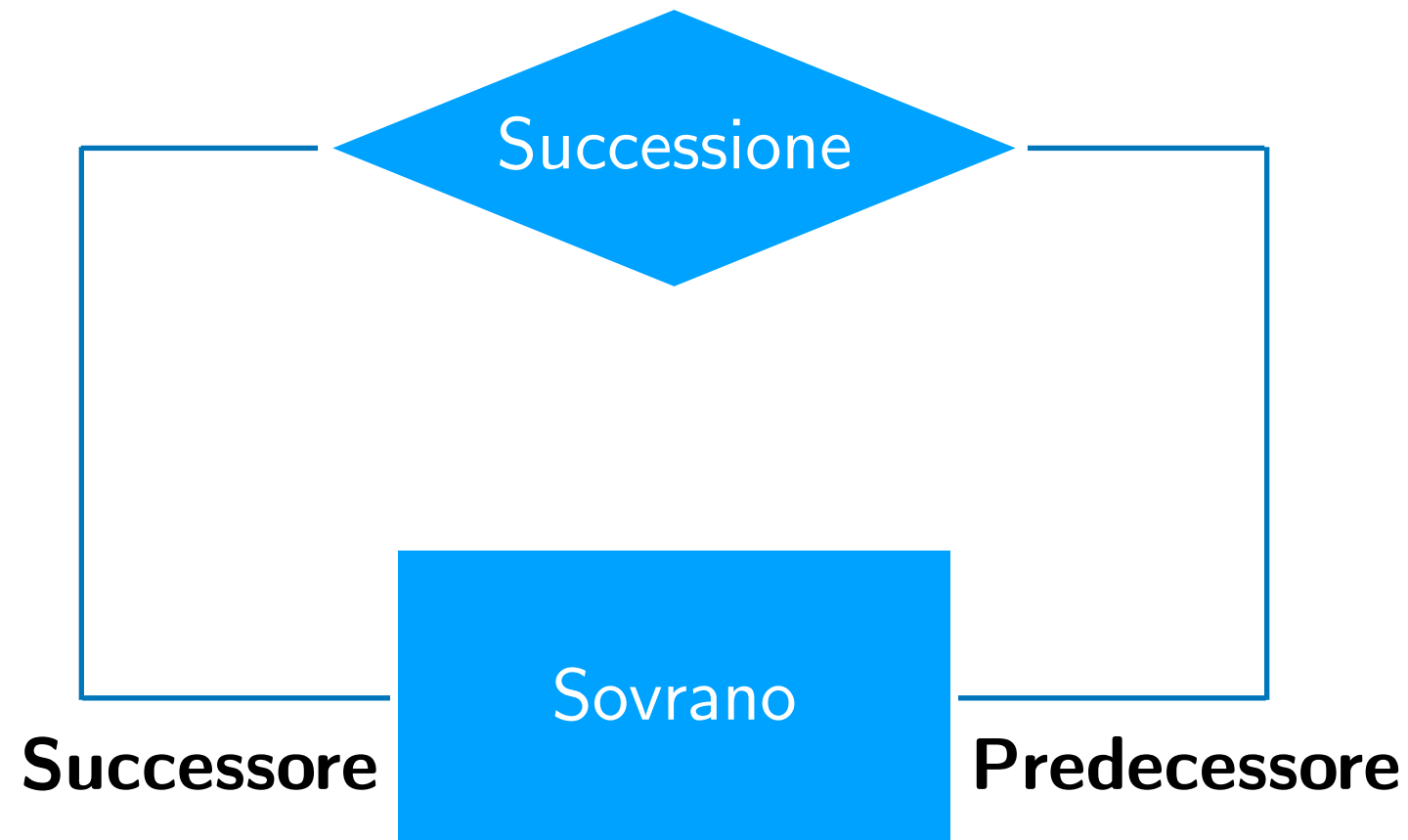
Relationship diverse sulle stesse entità



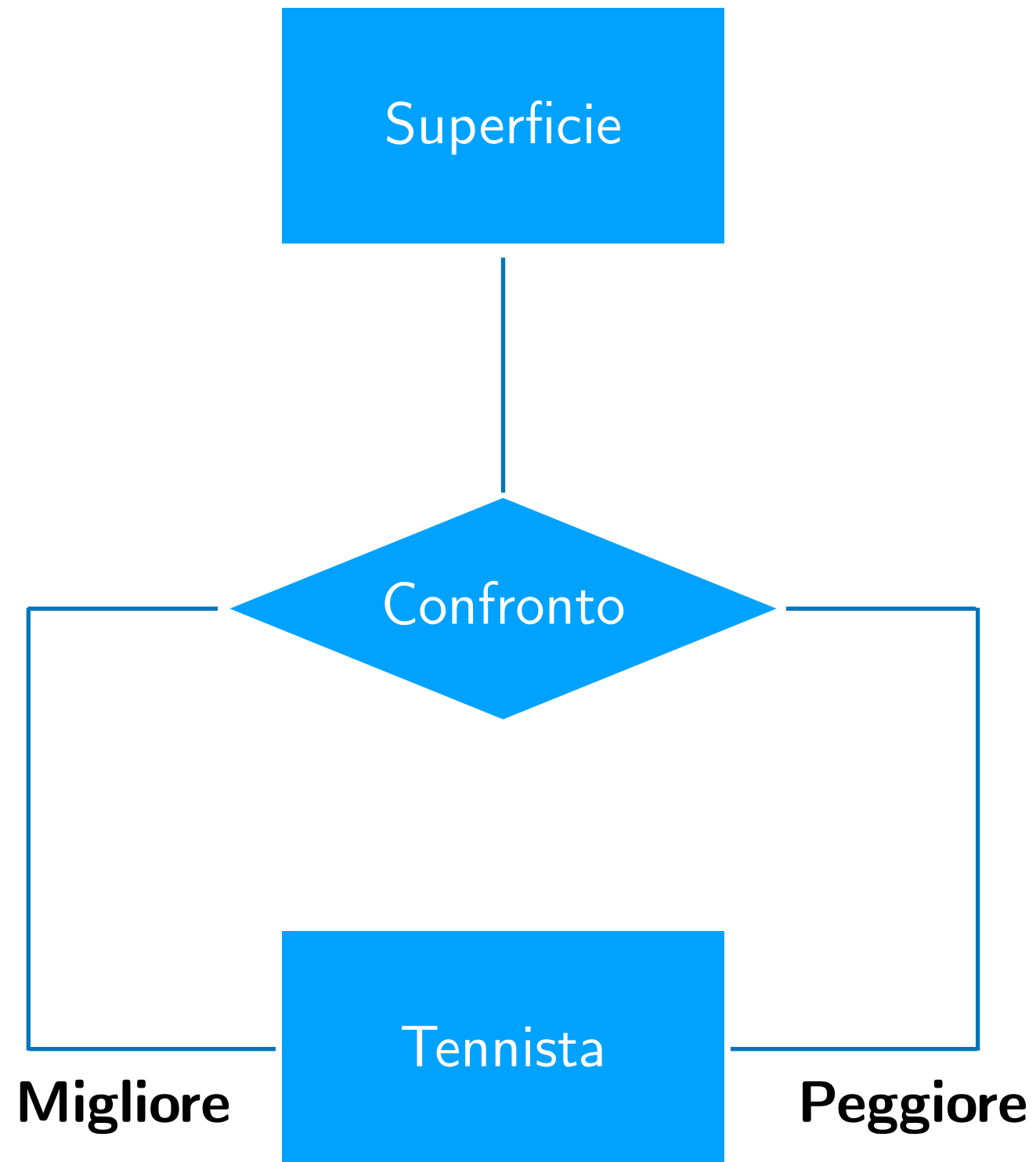
Relationship Ricorsiva



Relationship Ricorsiva con ruoli



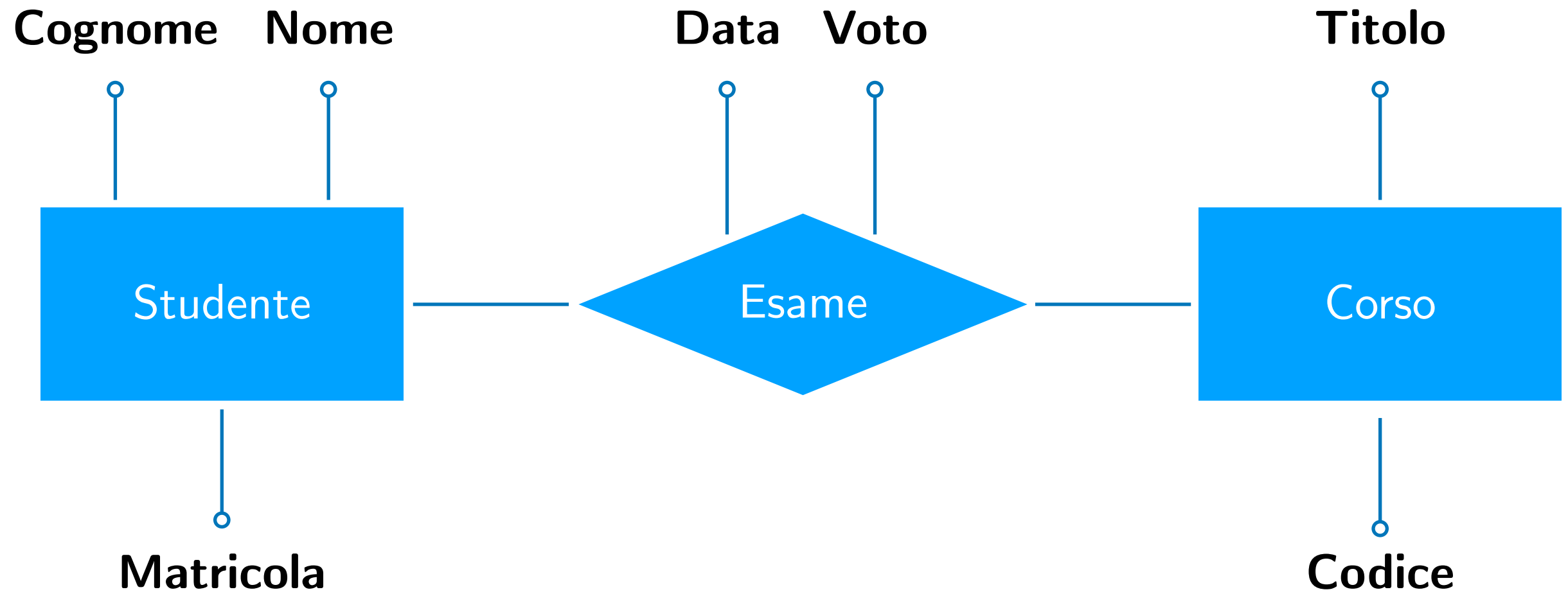
Relationship Mista



Attributo

- **Proprietà elementare** di un'entità o di una *relationship*, di **interesse** ai fini dell'applicazione
- Associa a **ogni occorrenza** di entità o *relationship* un **valore** appartenente a un insieme detto **dominio** dell'attributo

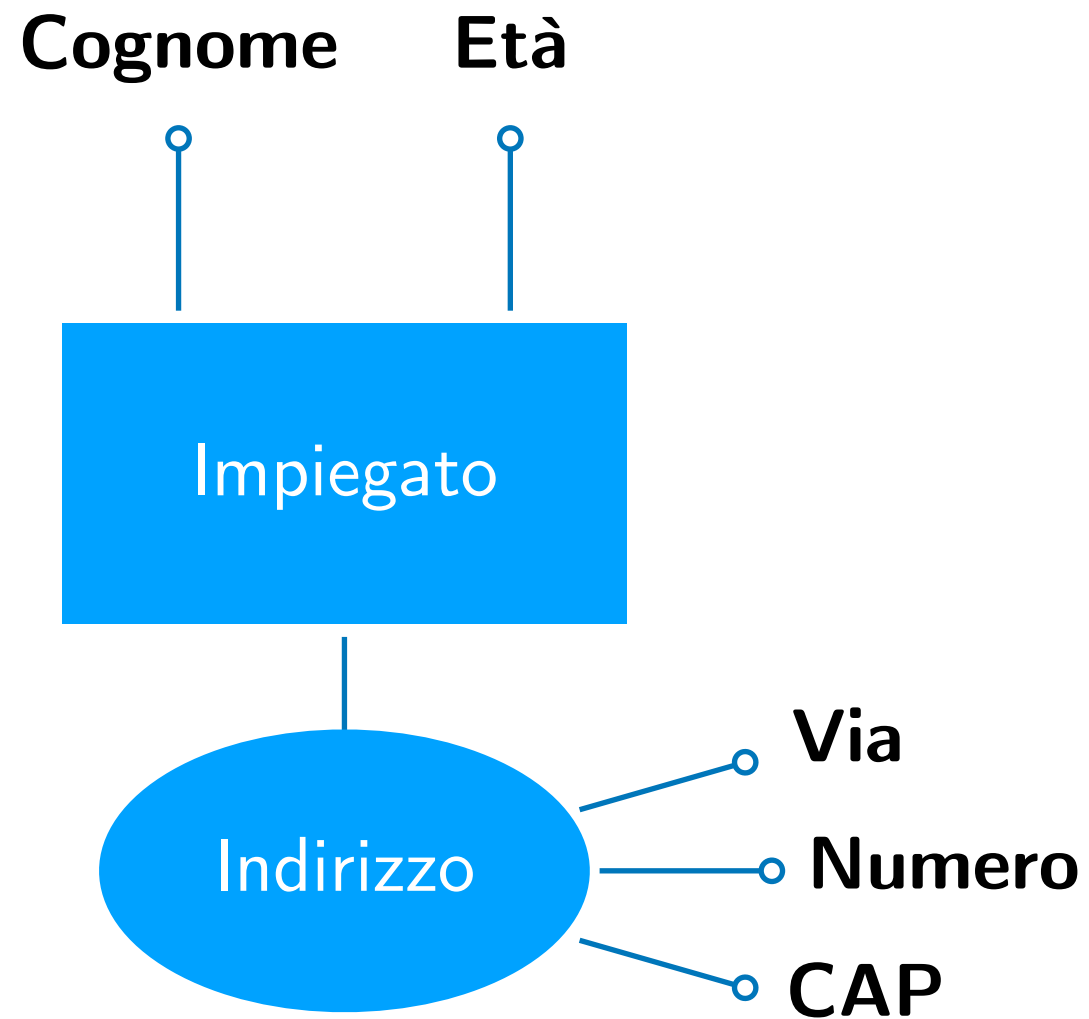
Rappresentazione Grafica di Attributi



Attributi Composti

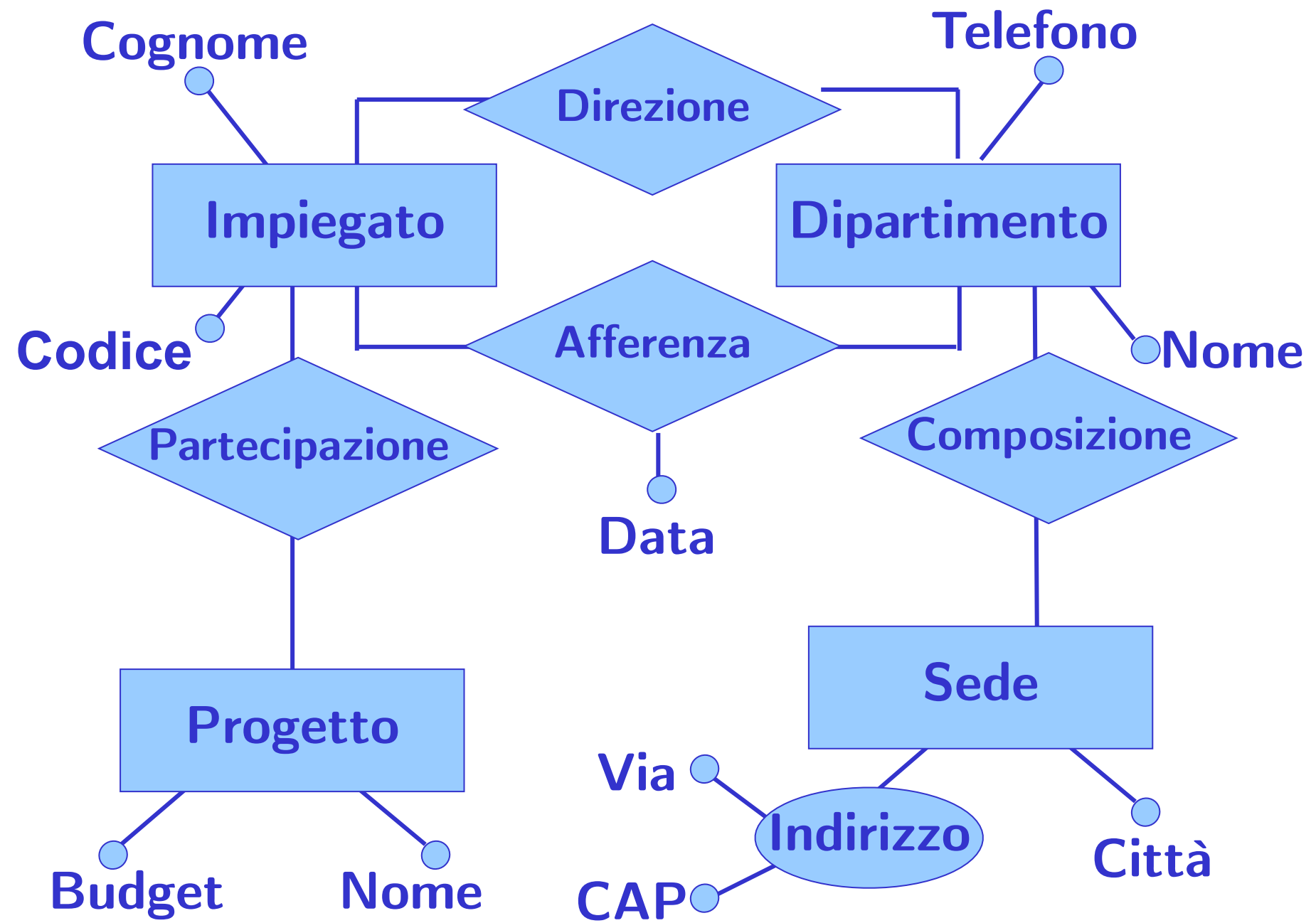
- **Raggruppano** attributi di una **medesima** entità o *relationship* che presentano **affinità** nel loro significato o uso
- Esempio:
 - Via, Numero civico e CAP formano un **Indirizzo**

Rappresentazione Grafica



Schema E-R con solo i costrutti base

- Si vuole descrivere l'organizzazione di un'azienda
 - Con sedi diverse
 - Ogni sede è composta di vari dipartimenti
 - Gli impiegati dell'azienda afferiscono ai vari dipartimenti e un impiegato li dirige
 - Gli impiegati lavorano su progetti
 - Ogni entità o *relationship* può avere vari attributi



Concetti Inesprimibili

- Un dipartimento ha un solo direttore?
- Un impiegato può afferire a un solo dipartimento?
- Il direttore di un dipartimento afferisce a quel dipartimento?
- ...

Altri Costrutti del Modello E-R

- **Cardinalità**
 - di *relationship*
 - di attributo
- **Identificatore**
 - interno
 - esterno
- **Generalizzazione**

Cardinalità di *relationship*

- **Coppia di valori** associati a ogni entità che partecipa a una *relationship*
- specificano il numero **minimo** e **massimo** di **occorrenze** della *relationship* cui ciascuna occorrenza di entità può partecipare

Cardinalità di *relationship*

- **Coppia di valori** associati a ogni entità che partecipa a una *relationship*
- specificano il numero **minimo** e **massimo** di **occorrenze** della *relationship* cui ciascuna occorrenza di entità può partecipare



Cardinalità di *relationship*

- 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**:
 - “**N**” non pone **alcun limite**

Cardinalità di *relationship*

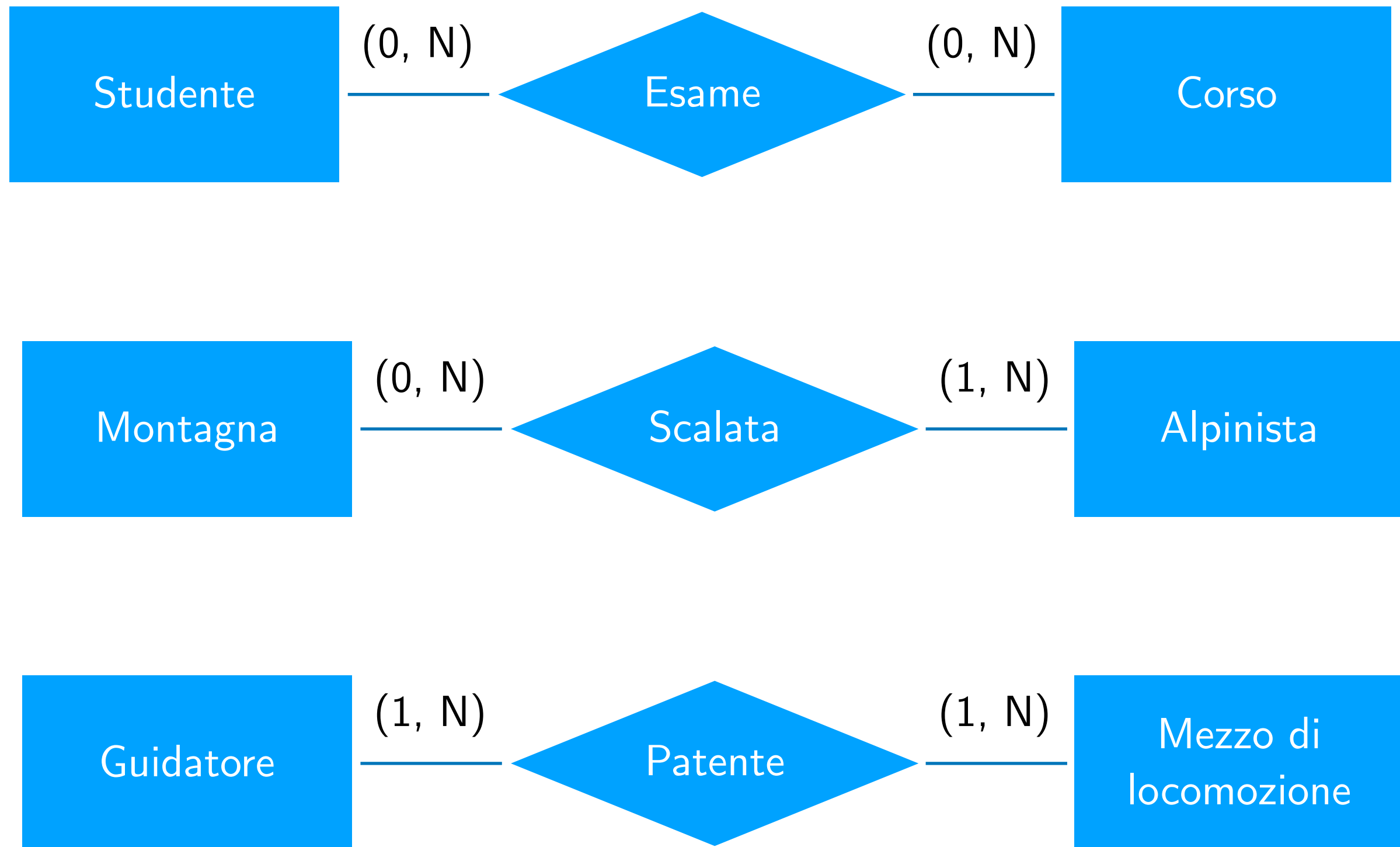
- Per semplicità usiamo solo tre simboli:
 - **0** e **1** per la cardinalità **minima**:
 - 0 = “partecipazione **opzionale**”
 - 1 = “partecipazione **obbligatoria**”
 - **1** e “**N**” per la **massima**:
 - “**N**” non pone **alcun limite**



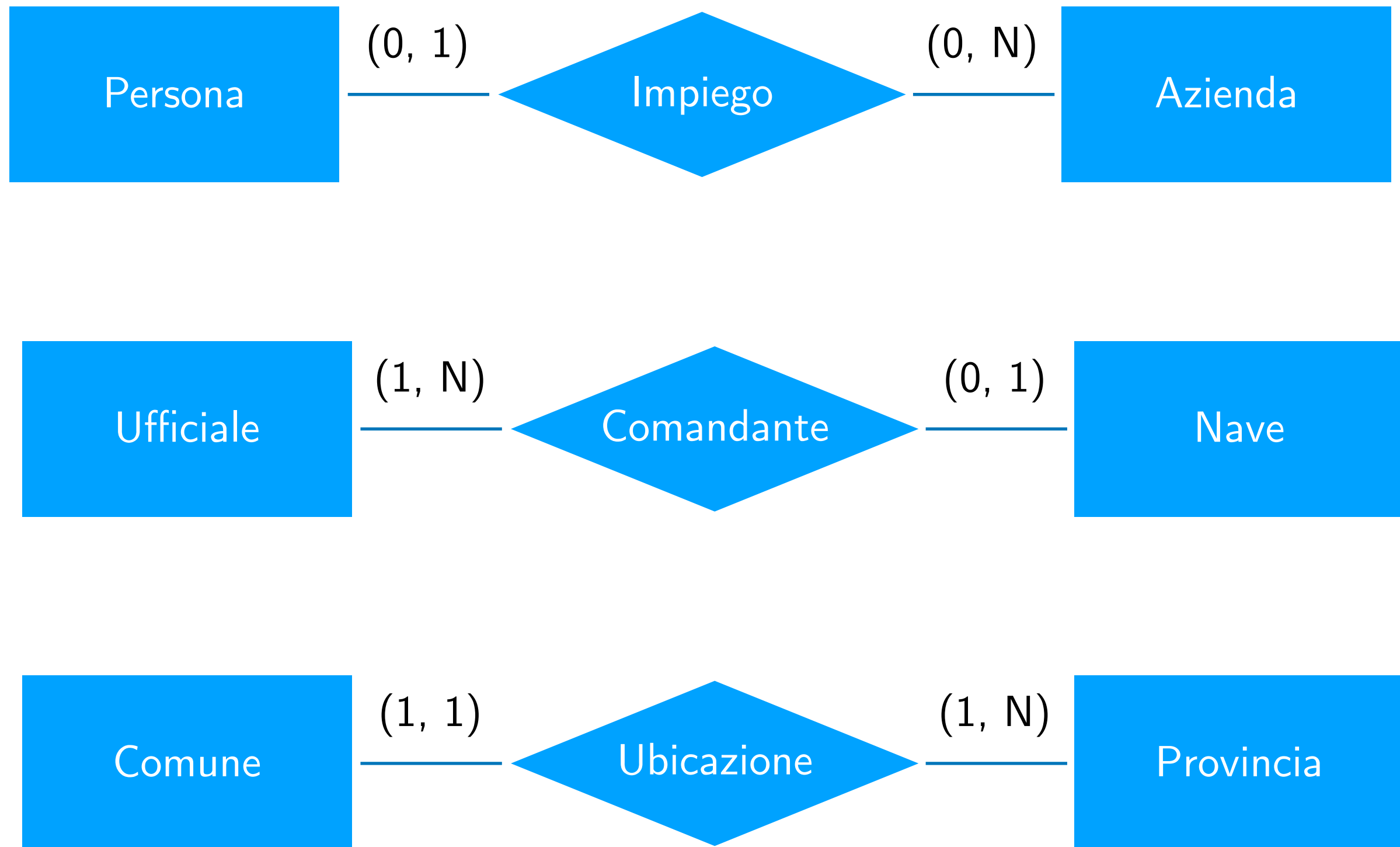
Tipi di *relationship*

- Con riferimento alle **cardinalità massime**, abbiamo *relationship*:
 - **uno a uno**
 - **uno a molti**
 - **molti a molti**

Relationship “molti a molti”

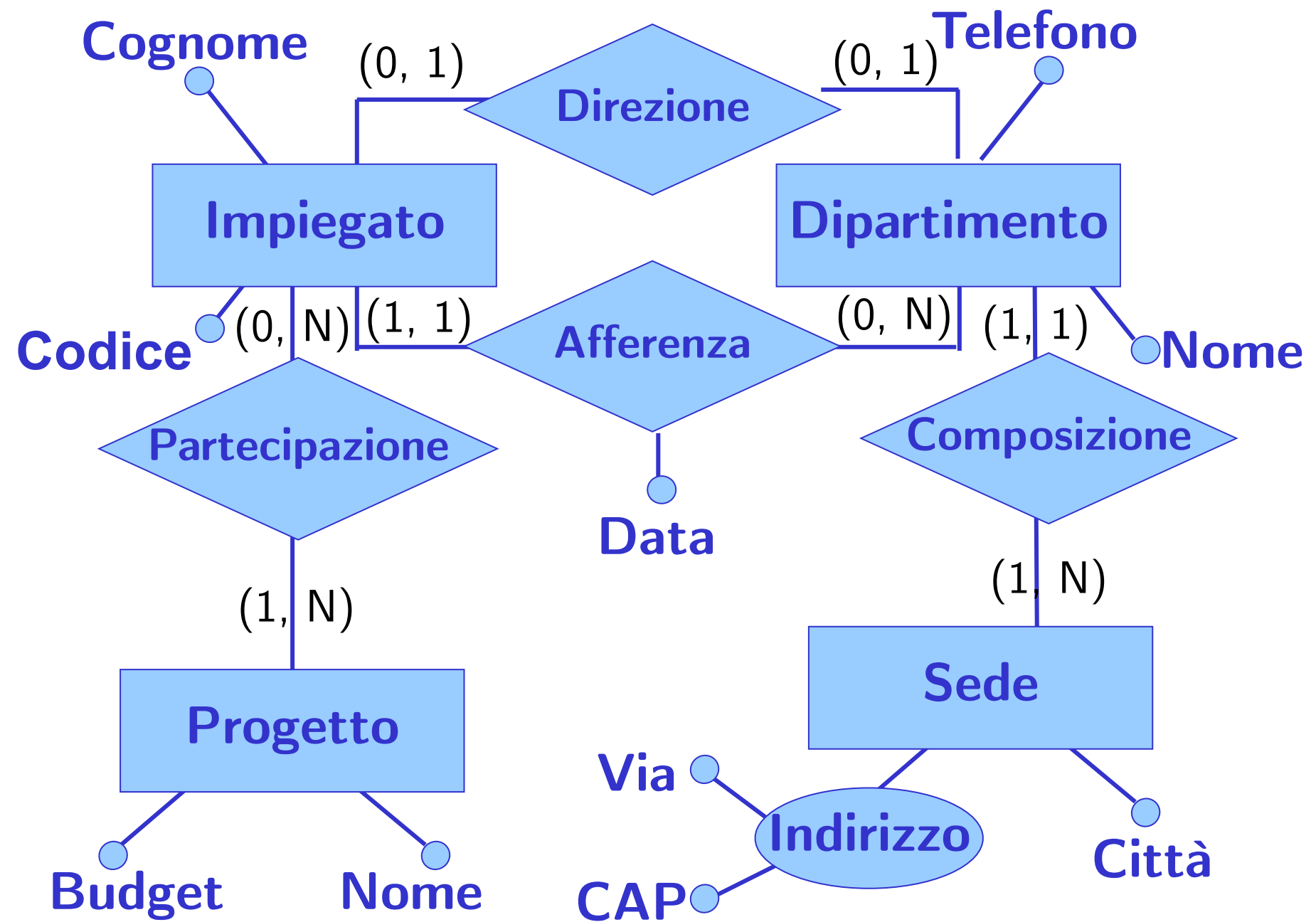


Relationship “uno a molti”



Relationship “uno a uno”

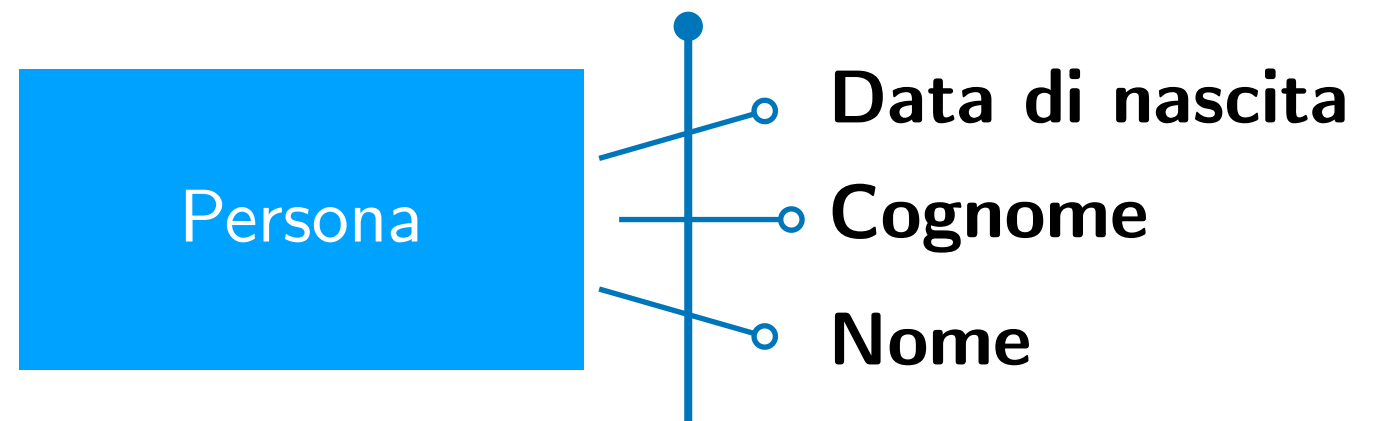
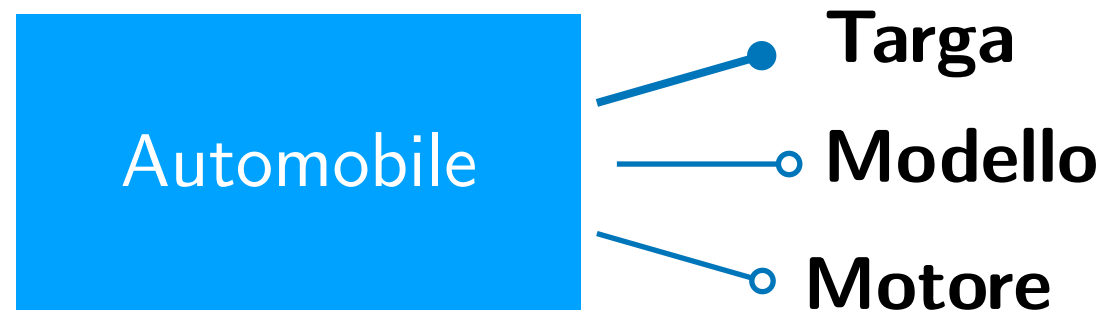




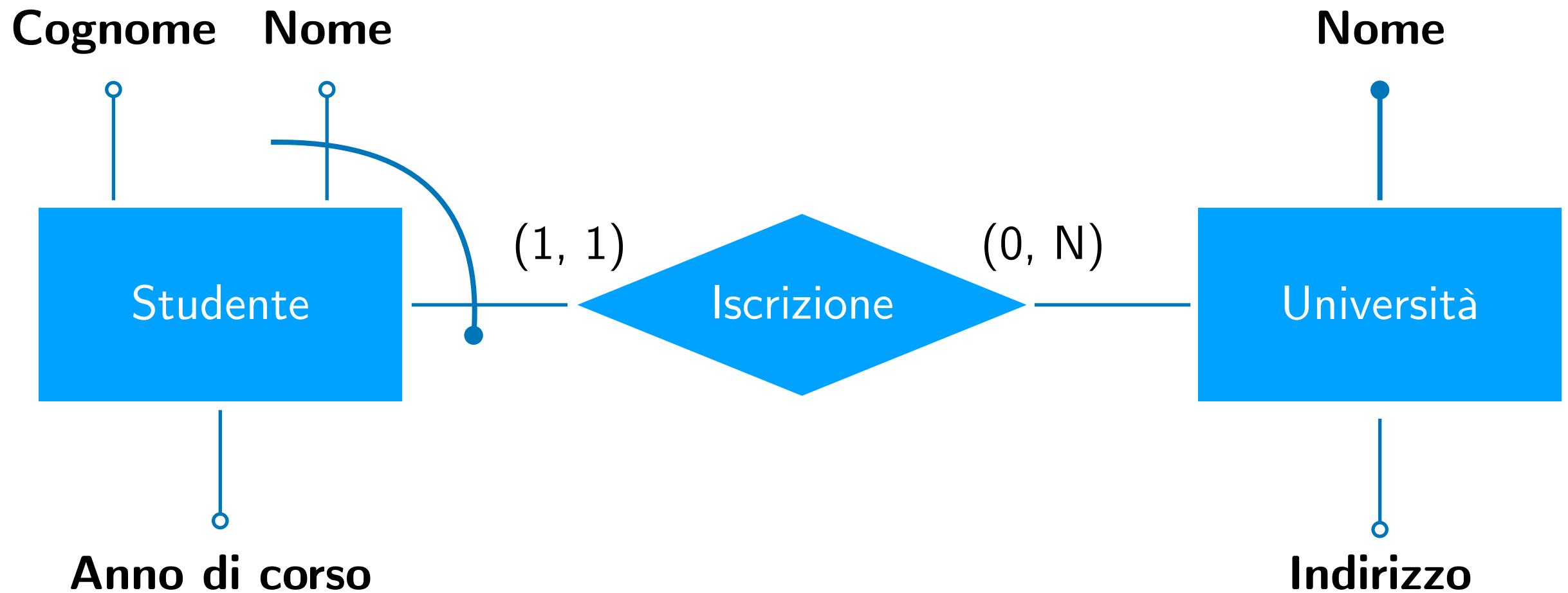
Identificatore di entità

- “Strumento” per l’identificazione univoca delle occorrenze di un’entità
- Costituito da:
 - attributi dell’entità
 - **identificatore interno** (o chiave)
 - (attributi +) la chiave di entità esterne raggiunta attraverso *relationship*
 - **identificatore esterno**

Identificatori Interni

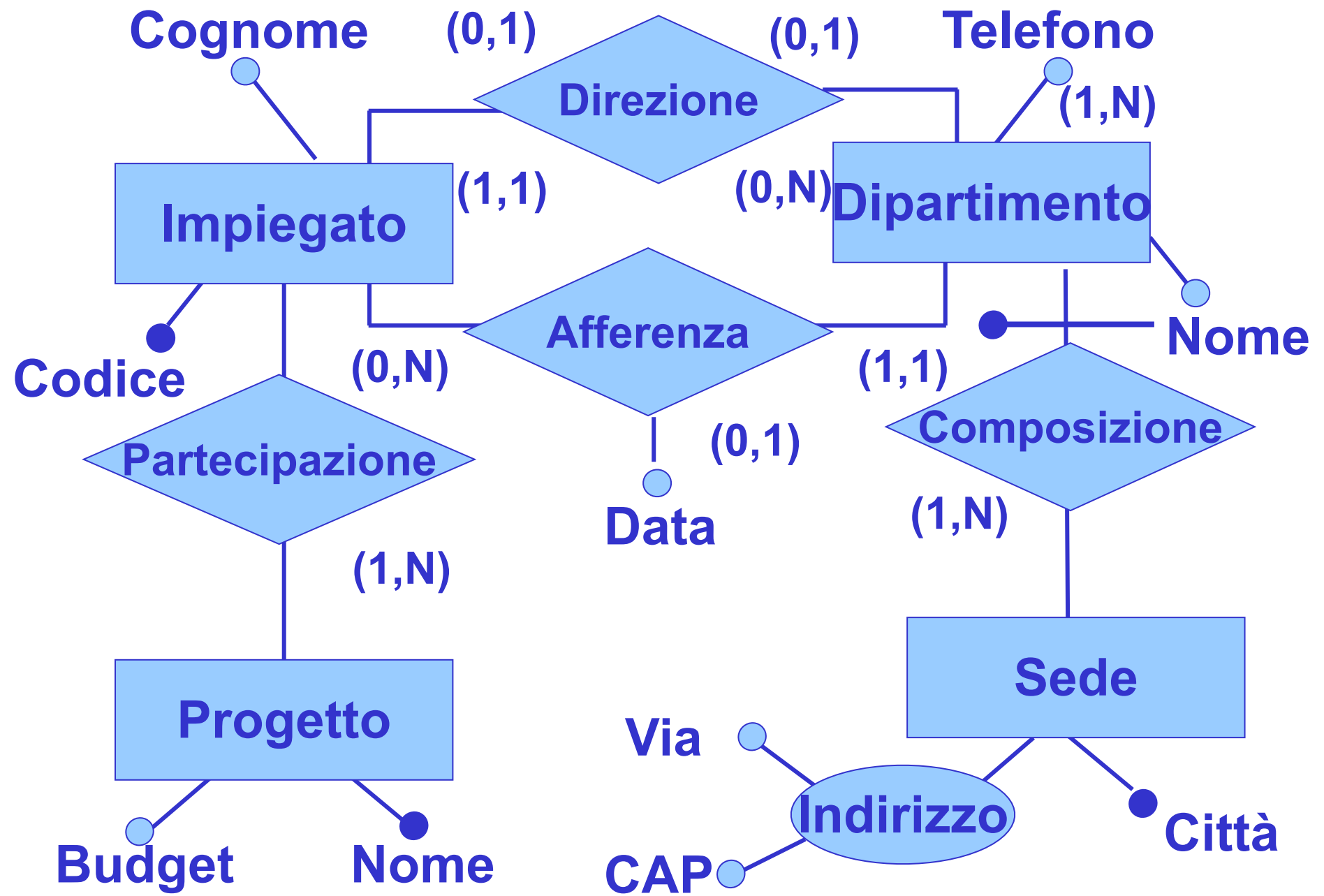


Identificatori Esterni



Caratteristiche degli Identificatori

- Ogni entità deve possedere **almeno un identificatore**, ma può averne in generale **più di uno**
- Una **identificazione esterna** è possibile solo attraverso una *relationship* a cui l'entità da identificare partecipa con **cardinalità (1, 1)**

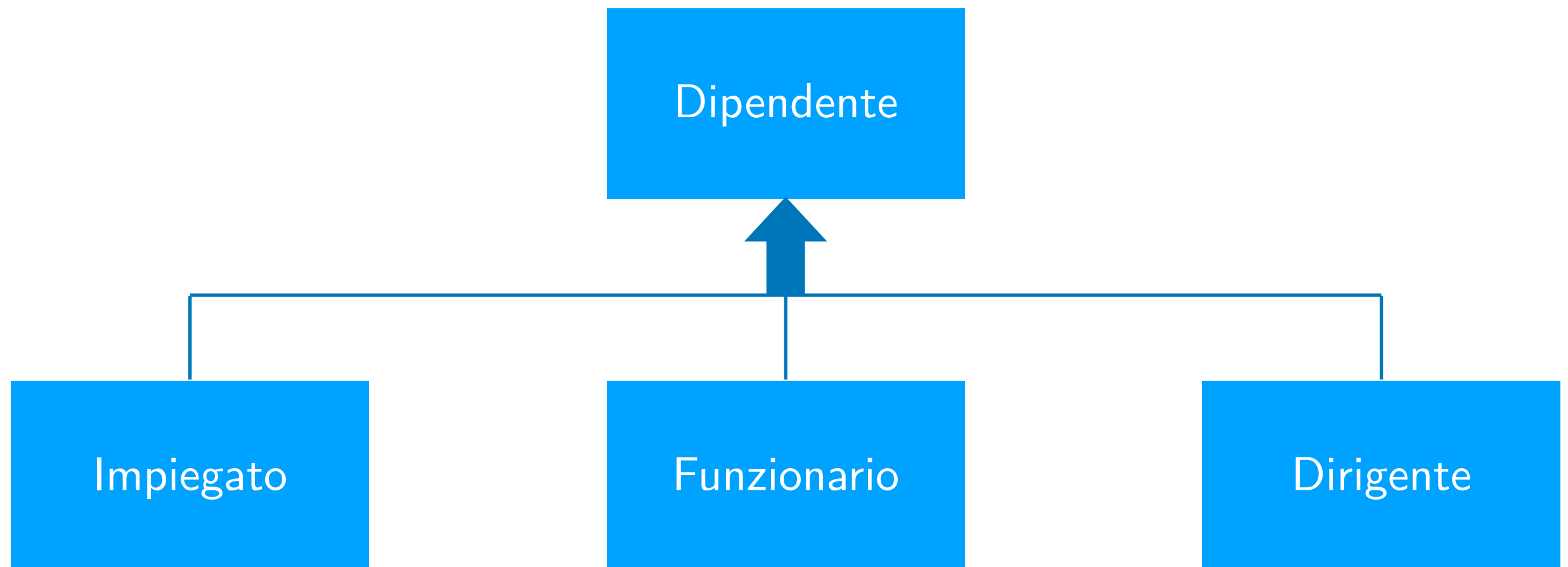


Generalizzazione

- Mette in relazione **una o più entità** E_1, E_2, \dots, E_n con una entità E , che le comprende come **casi particolari**
- E è una **generalizzazione** di E_1, E_2, \dots, E_n
- E_1, E_2, \dots, E_n sono **specializzazioni** (o sottotipi) di E

Generalizzazione

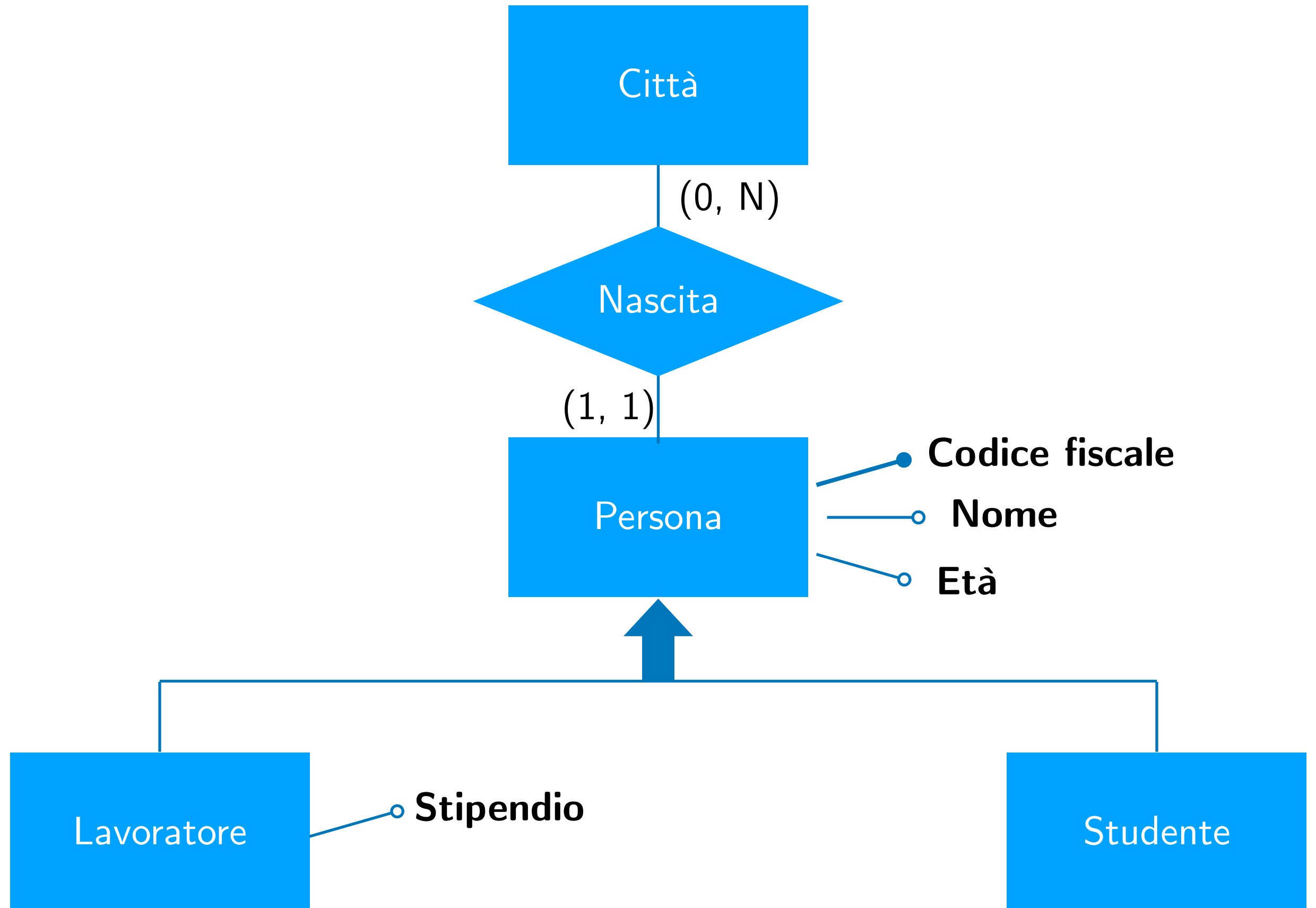
- Mette in relazione **una o più entità** E_1, E_2, \dots, E_n con una entità E , che le comprende come **casi particolari**
- E è una **generalizzazione** di E_1, E_2, \dots, E_n
- E_1, E_2, \dots, E_n sono **specializzazioni** (o sottotipi) di E



Proprietà delle generalizzazioni

- Se $E(\text{genitore})$ è generalizzazione di E_1, E_2, \dots, E_n (figlie):
 - ogni proprietà di E è significativa per E_1, E_2, \dots, E_n
 - ogni occorrenza di E_1, E_2, \dots, E_n è occorrenza anche di E

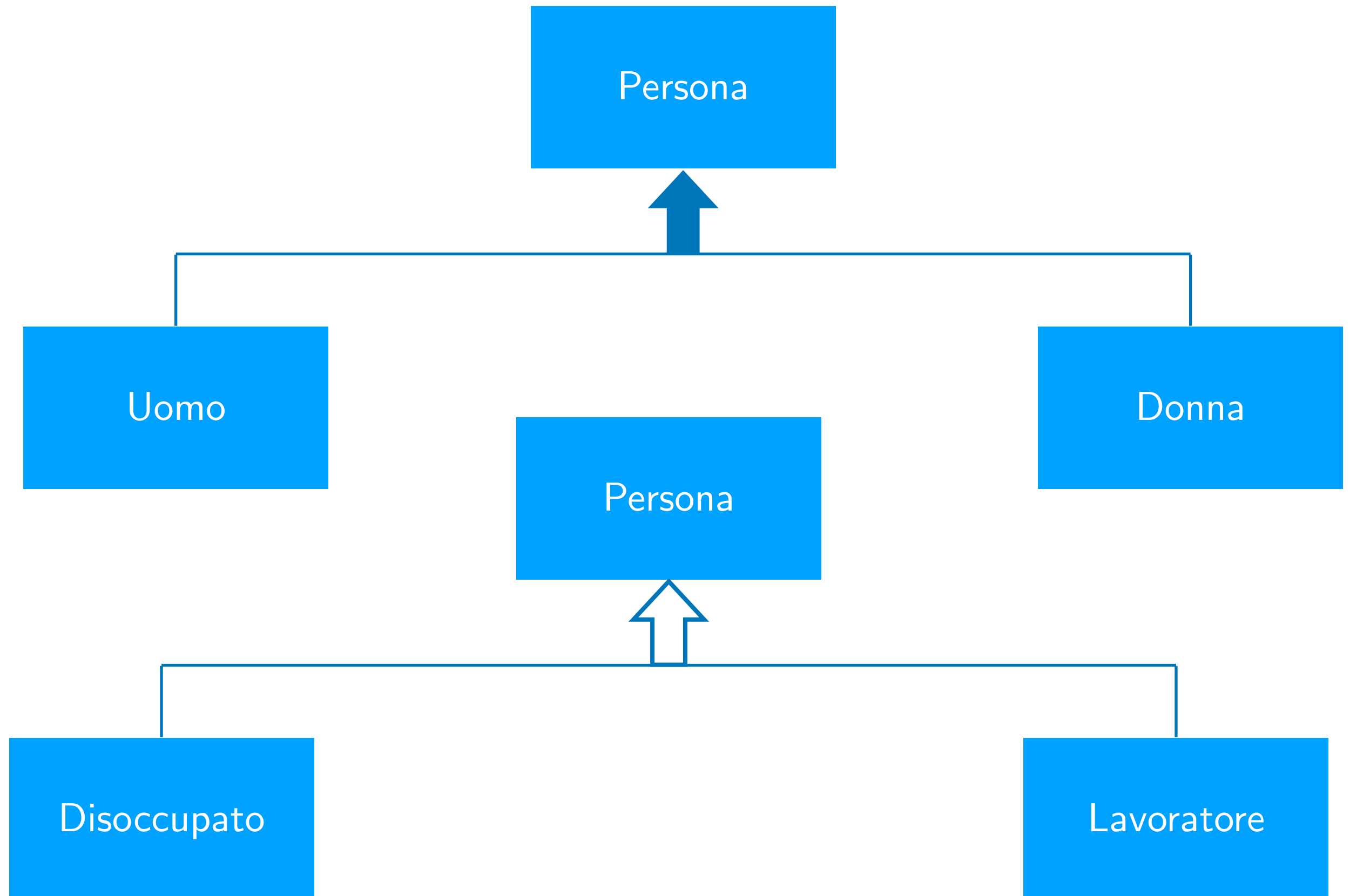
Esempio



Caratteristiche delle generalizzazioni

- **Ereditarietà**: tutte le **proprietà** (attributi, *relationship*, altre generalizzazioni) dell'entità genitore vengono **ereditate** dalle entità figlie e **non rappresentate esplicitamente**
- **Generalizzazione totale**: se ogni occorrenza dell'entità genitore è occorrenza di almeno una delle entità figlie, altrimenti è **parziale**
- **Generalizzazione esclusiva**: se ogni occorrenza dell'entità genitore è occorrenza di al più una delle entità figlie, altrimenti è **sovrapposta**

Generalizzazione Totale e Parziale

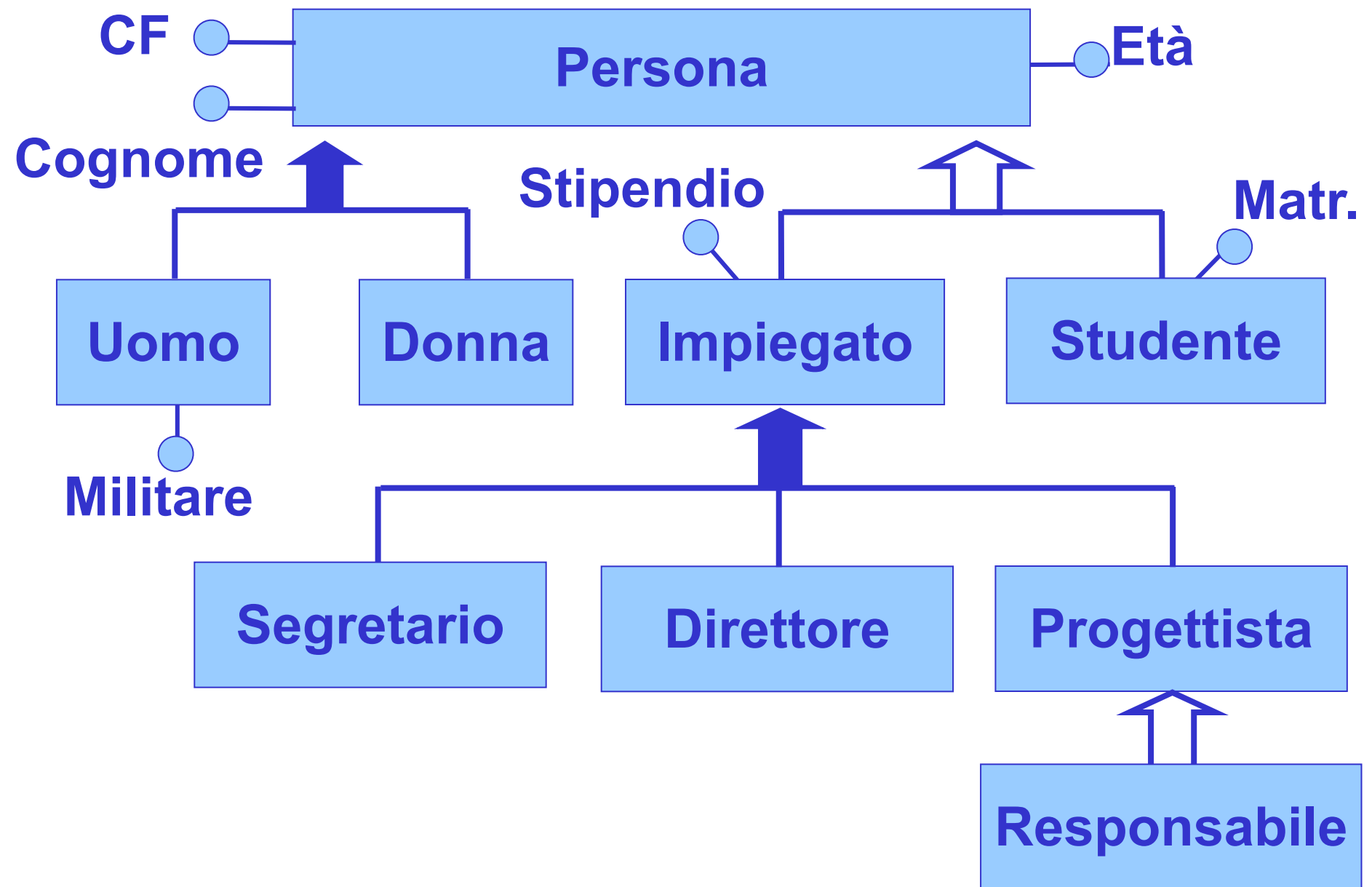


Altre proprietà

- 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**
- Il genitore di una generalizzazione totale può **non avere identificatore**, purché ...

Esercizio

- Le persone hanno codice fiscale, cognome ed età, gli uomini la posizione militare, le donne il numero dei figli
- Gli impiegati hanno lo stipendio e possono essere segretari, direttori o progettisti (un progettista può essere anche responsabile di progetto)
- Gli studenti (che non possono essere impiegati) hanno un numero di matricola;
- Esistono persone che non sono né impiegati né studenti (ma i dettagli non ci interessano)



Documentazione associata agli schemi concettuali

- **Dizionario dei dati:**
 - entità
 - *relationship*
- **Regole aziendali:**
 - Vincoli di integrità
 - Possibili derivazioni
- Uno schema E-R **non è quasi mai sufficiente da solo** a rappresentare tutti i dettagli di un'applicazione
- Ci sono **vincoli non esprimibili**
- È necessario associare una **documentazione di supporto**

Dizionario dei dati (entità)

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

Dizionario dei dati (*relationship*)

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	

Regole di vincolo

- | |
|--|
| (1) Il direttore di un dipartimento deve afferire a tale dipartimento |
| (2) Un impiegato non deve avere uno stipendio maggiore del direttore del dipartimento al quale afferisce |
| (3) Un dipartimento con sede a Roma deve essere diretto da un impiegato con più di dieci anni di anzianità |
| (4) Un impiegato che non afferisce a nessun dipartimento non deve partecipare a nessun progetto |

Regole di derivazione

- (1) Il numero di impiegati di un dipartimento si ottiene contando gli impiegati che afferiscono a tale dipartimento
- (2) Il budget di un progetto si ottiene moltiplicando per 3 la somma degli stipendi degli impiegati che vi partecipano