

# **Progettazione Concettuale**

Requisiti della base di dati

Progettazione  
concettuale

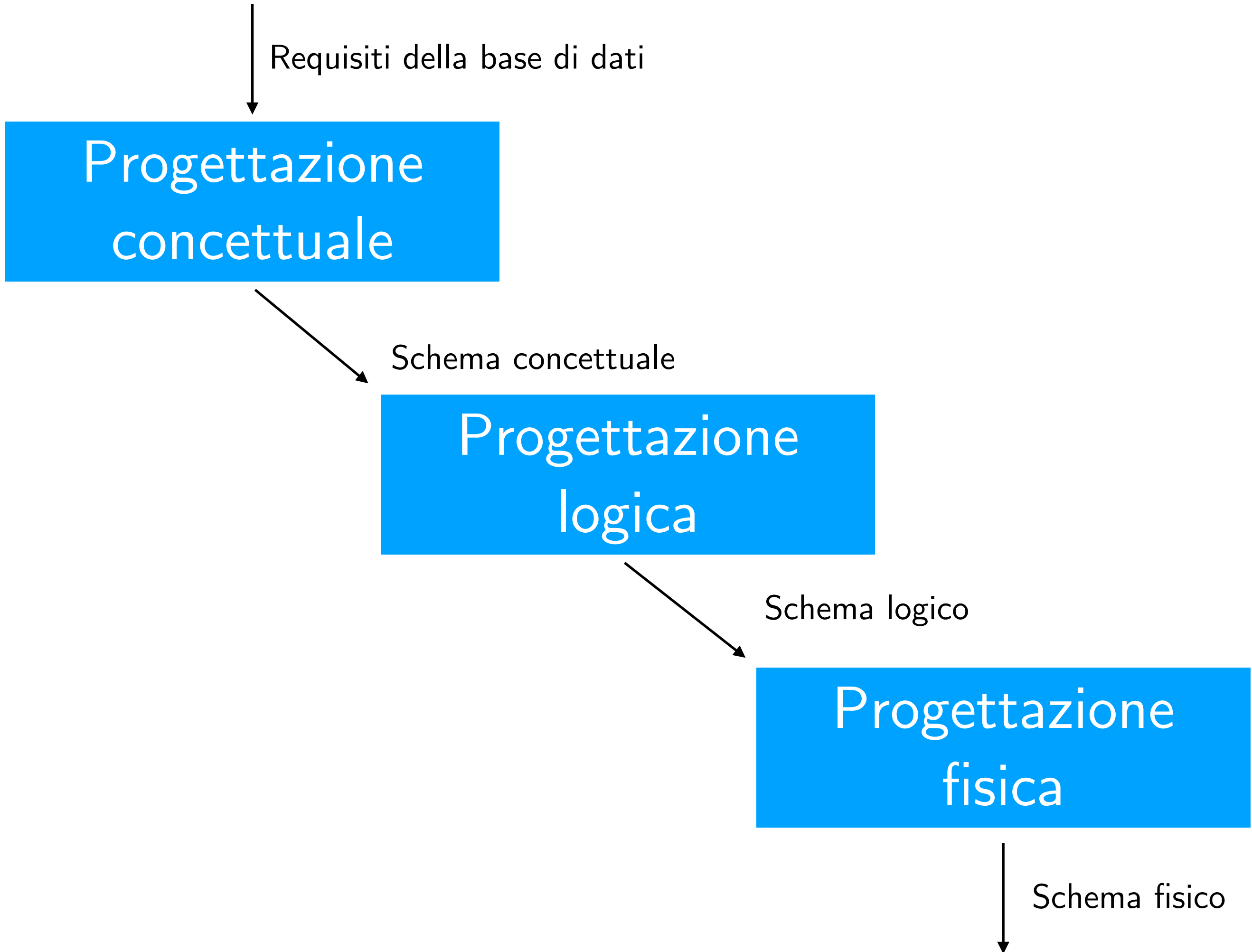
Schema concettuale

Progettazione  
logica

Schema logico

Progettazione  
fisica

Schema fisico

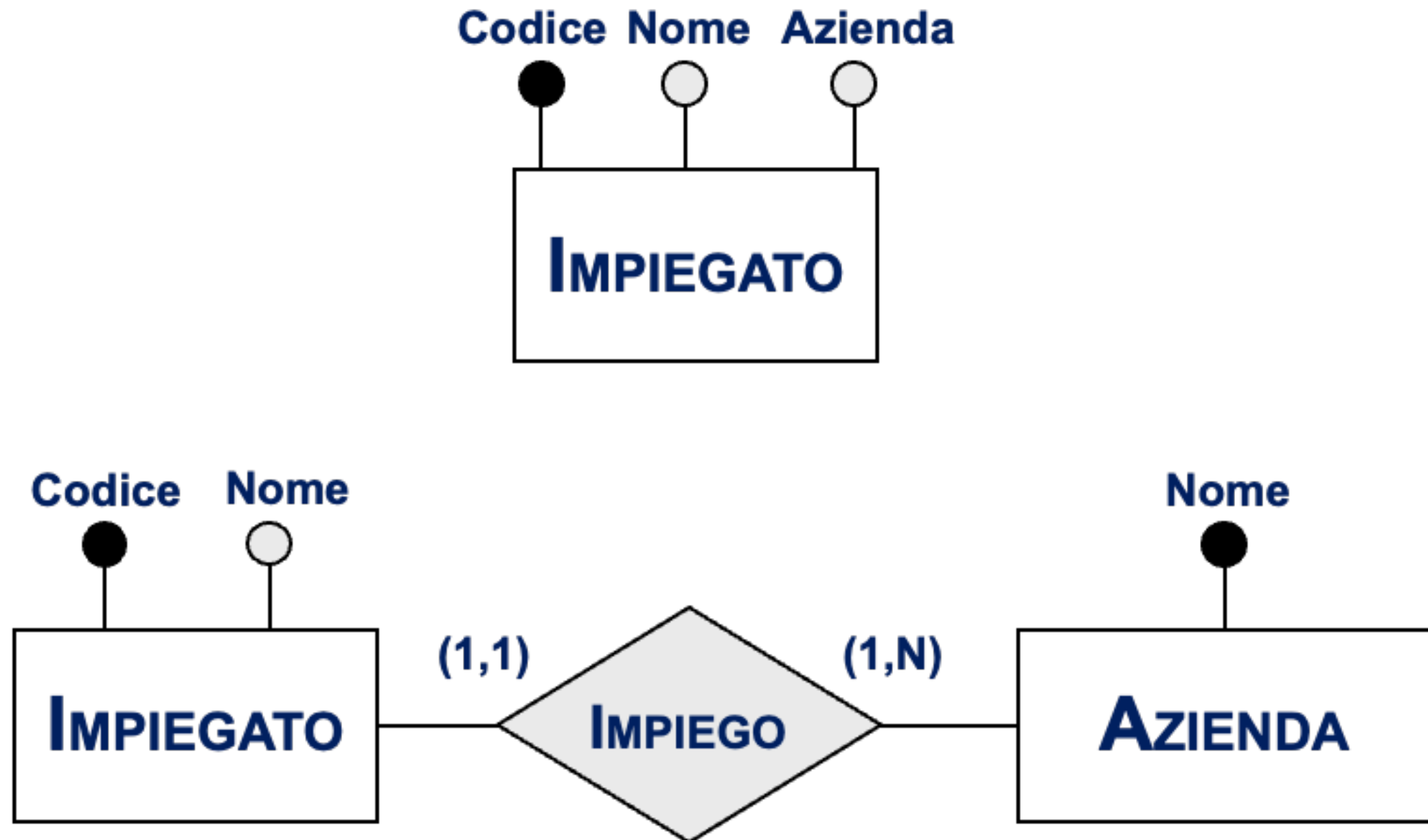


- **Quale costrutto E-R va utilizzato** per rappresentare un concetto presente nelle specifiche?
- **Bisogna basarsi sulle definizioni dei costrutti del modello E-R**
  - se ha proprietà significative e descrive oggetti con esistenza autonoma
    - **entità**
  - se è semplice e non ha proprietà
    - **attributo**
  - se correla due o più concetti
    - ***relationship***
  - se è caso particolare di un altro
    - **generalizzazione**

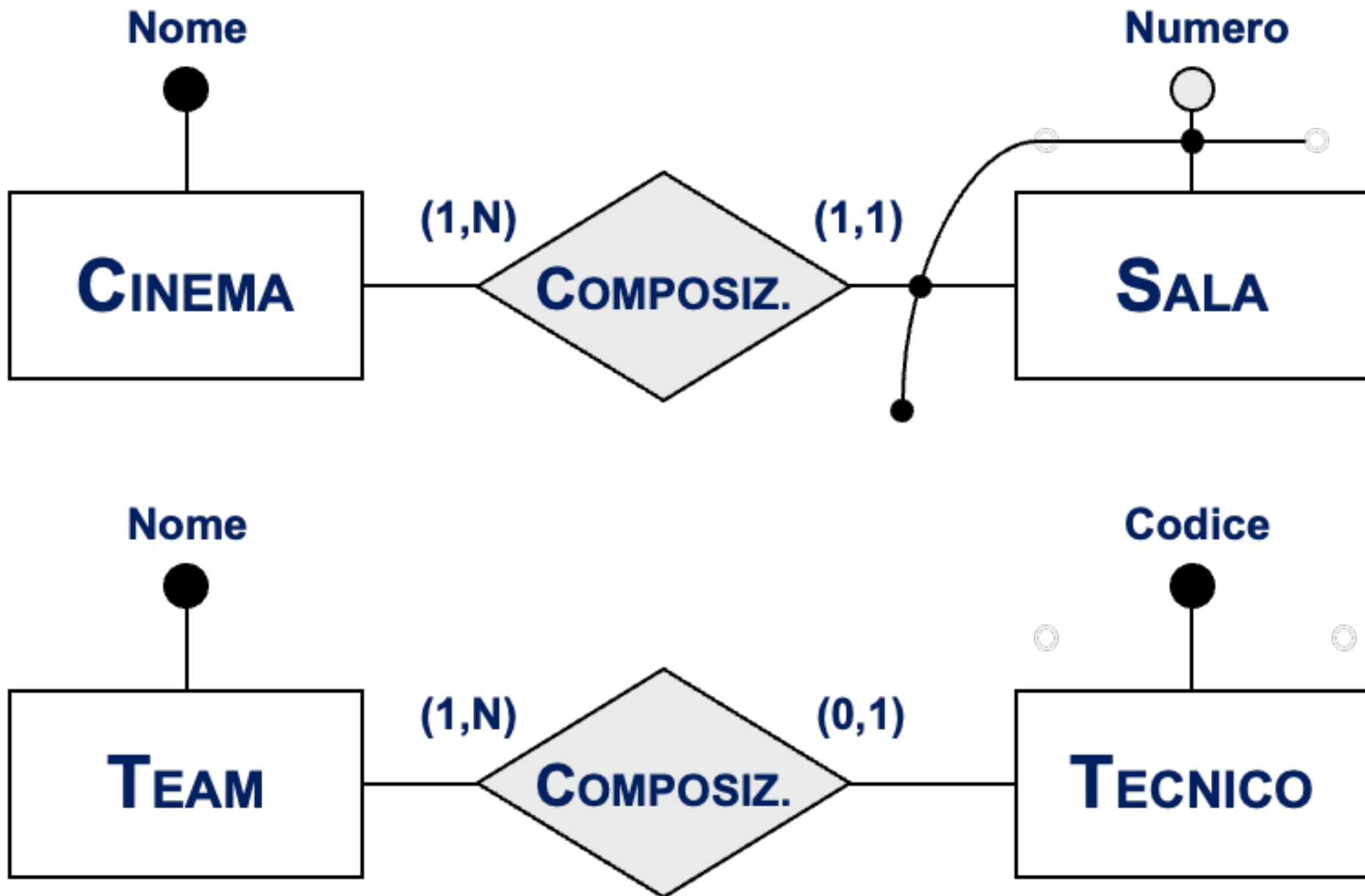
# Design Pattern

- Soluzioni progettuali a problemi comuni
- Largamente usati nell'ingegneria del software
- Vediamo alcuni pattern comuni nella progettazione concettuale di basi di dati

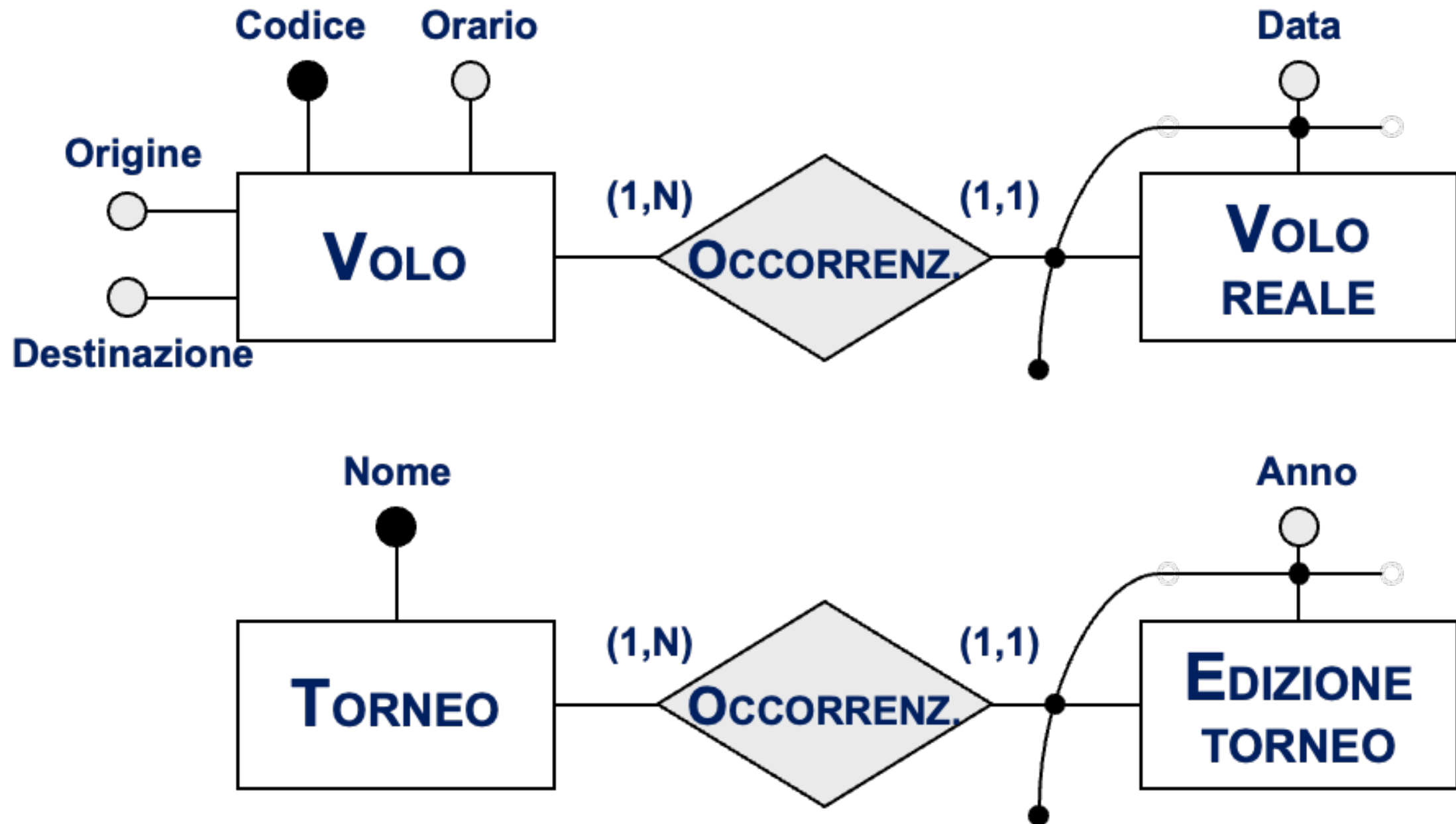
# Reificazione di attributo di entità



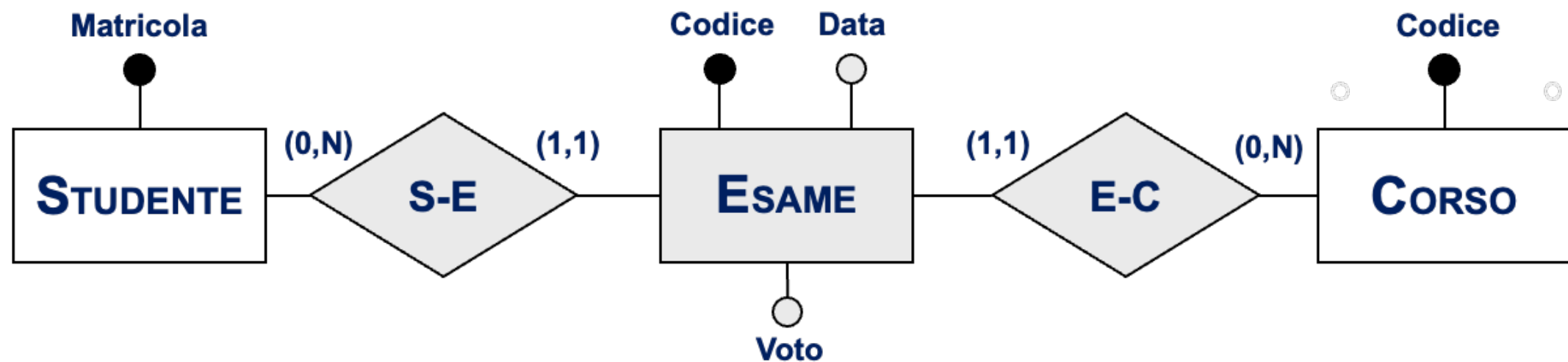
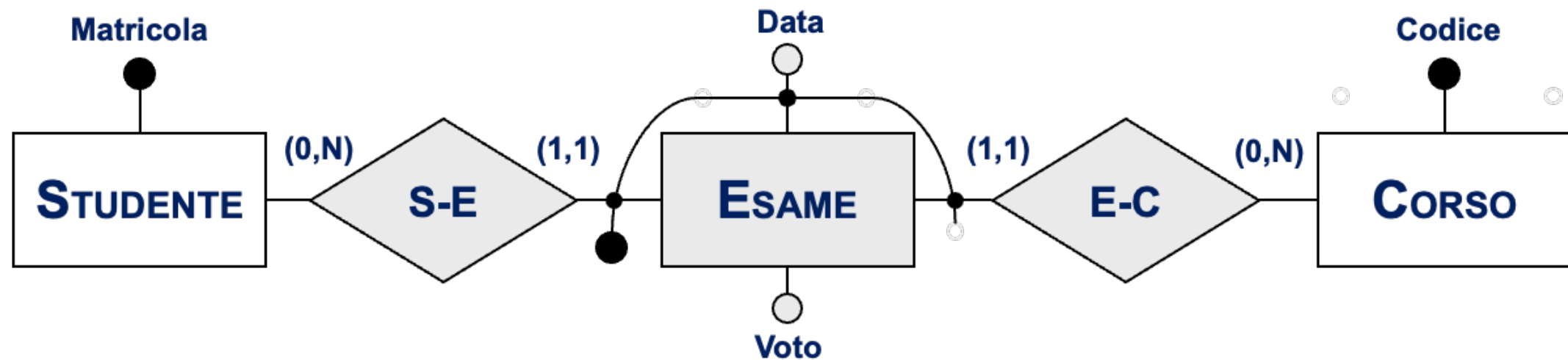
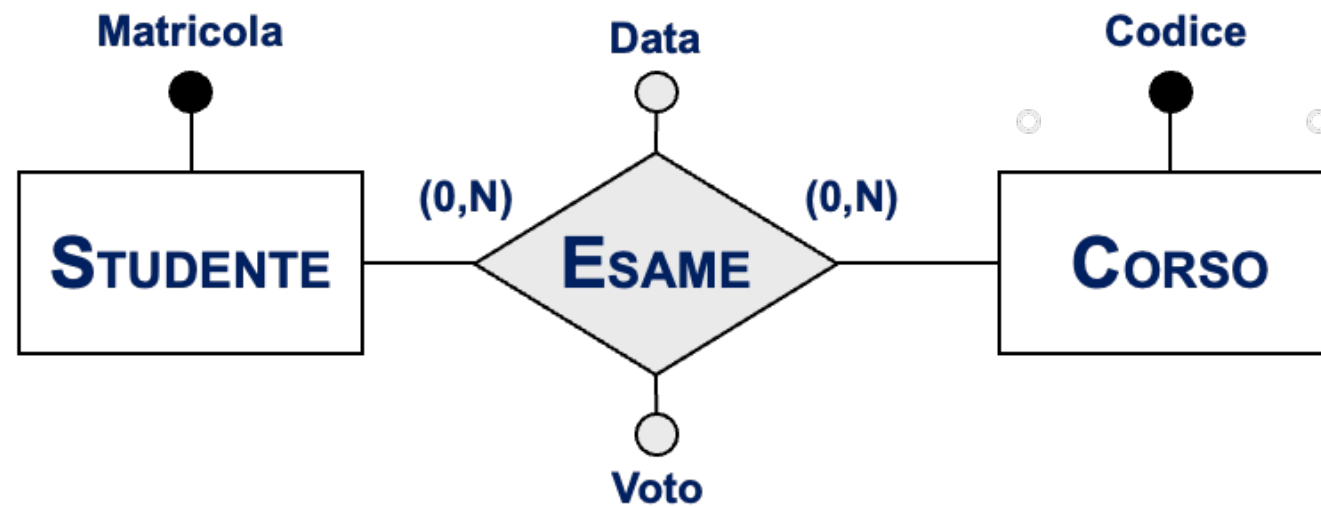
# Parte-di



# Istanza-di

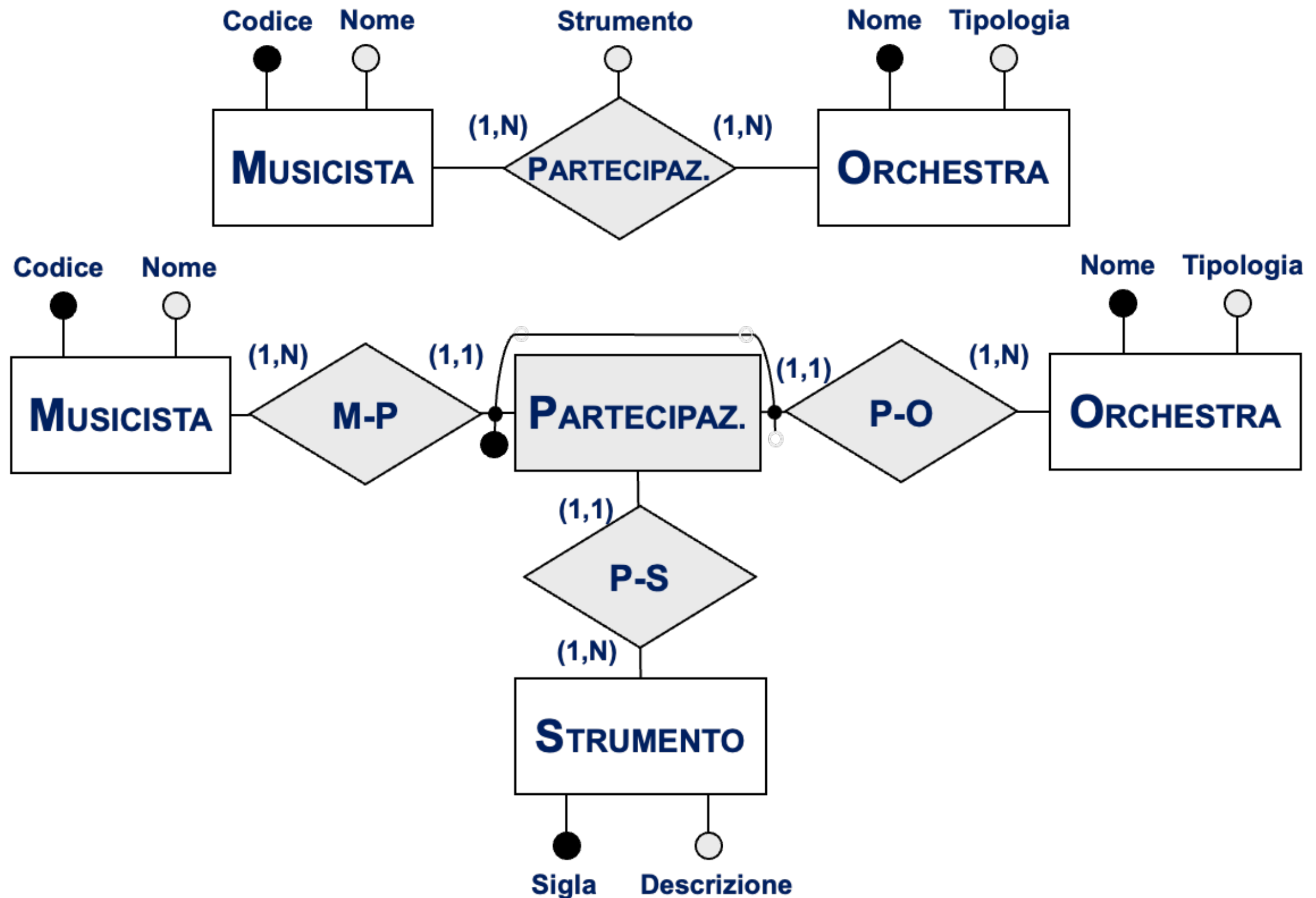


# Reificazione di *relationship* binaria

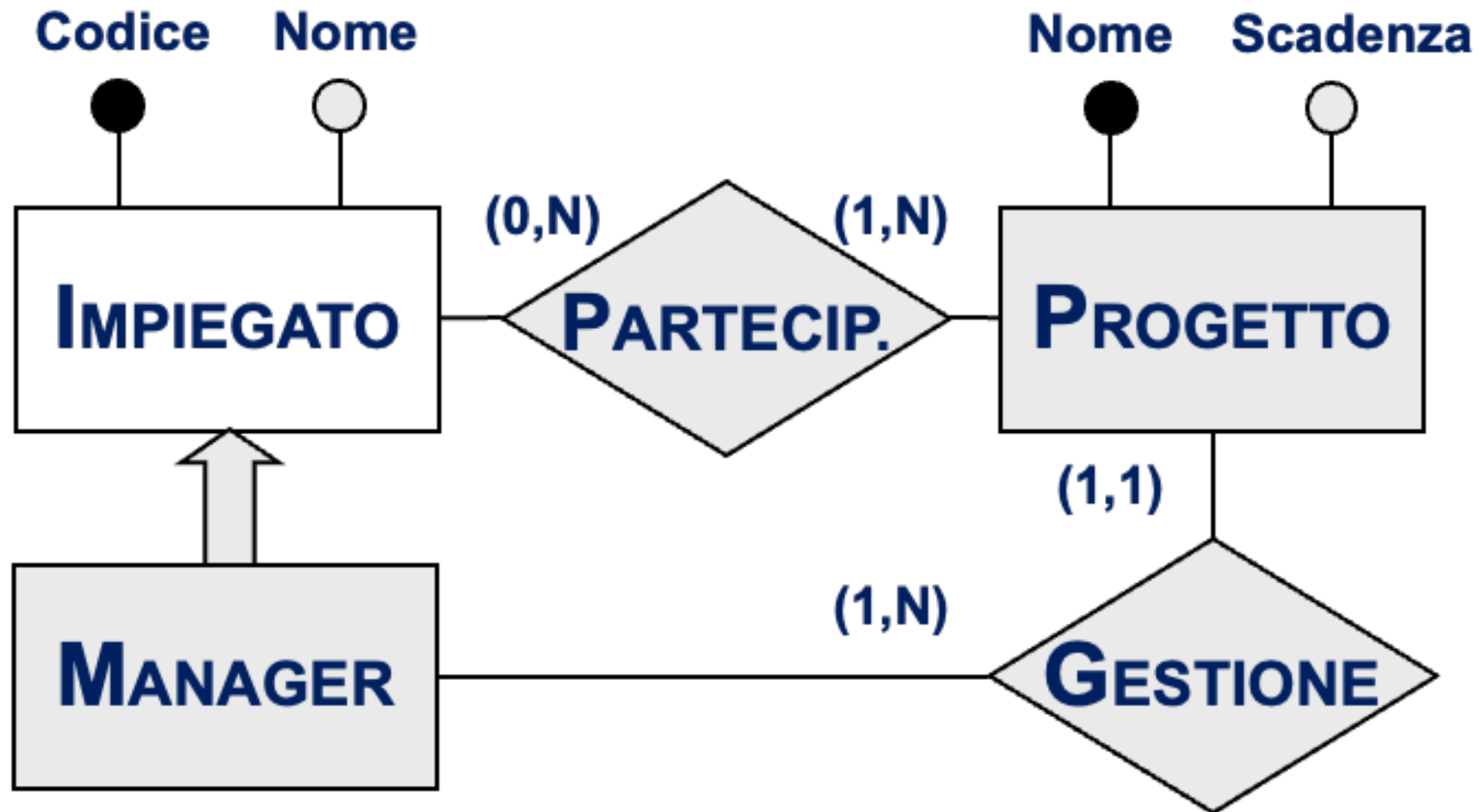




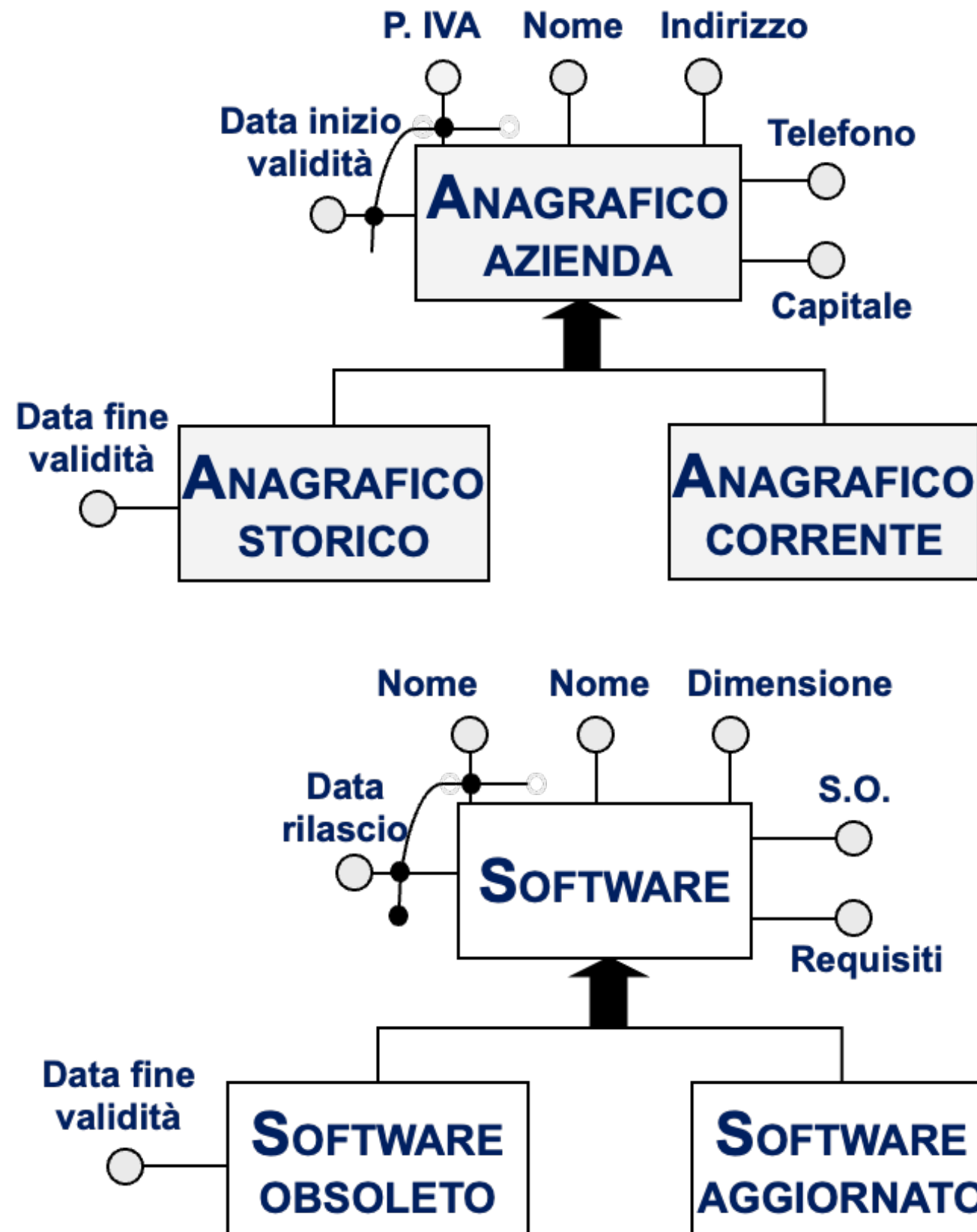
# Reificazione di attributo di *relationship*



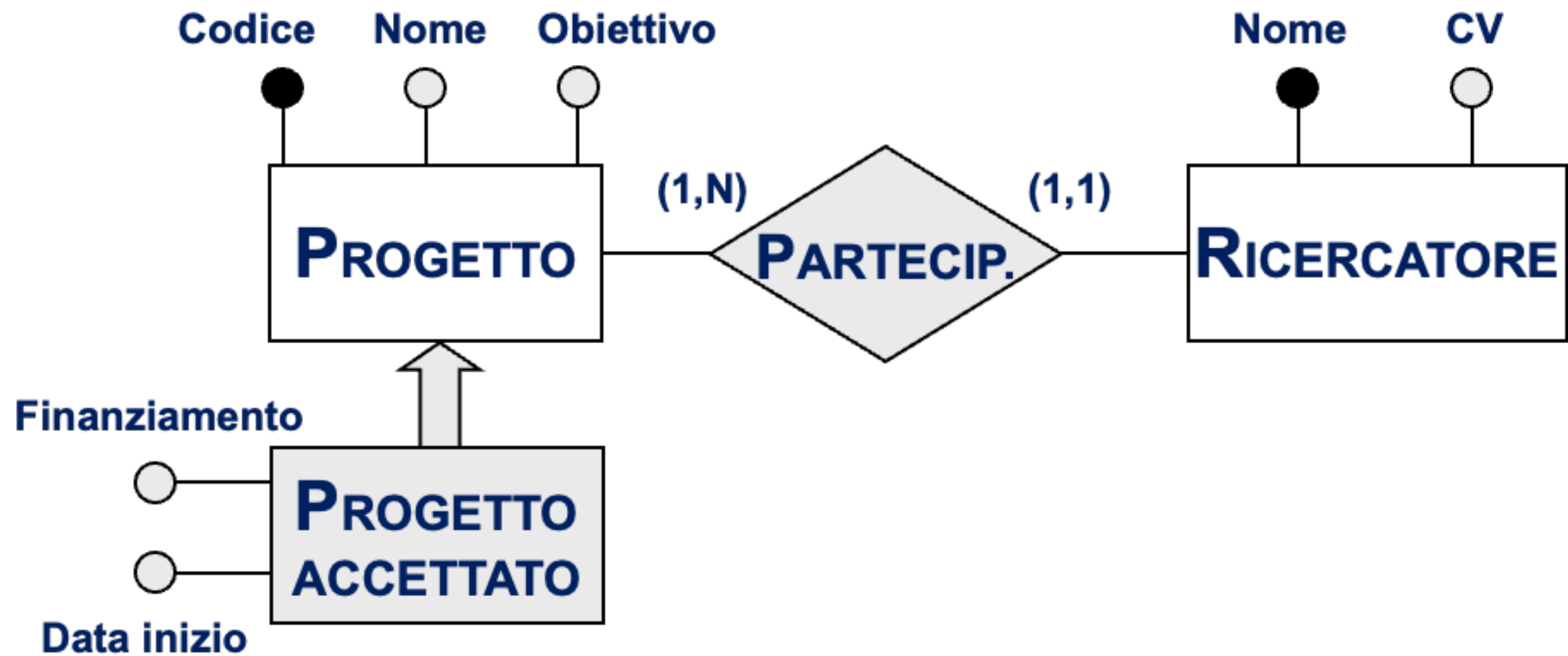
# Caso particolare di entità



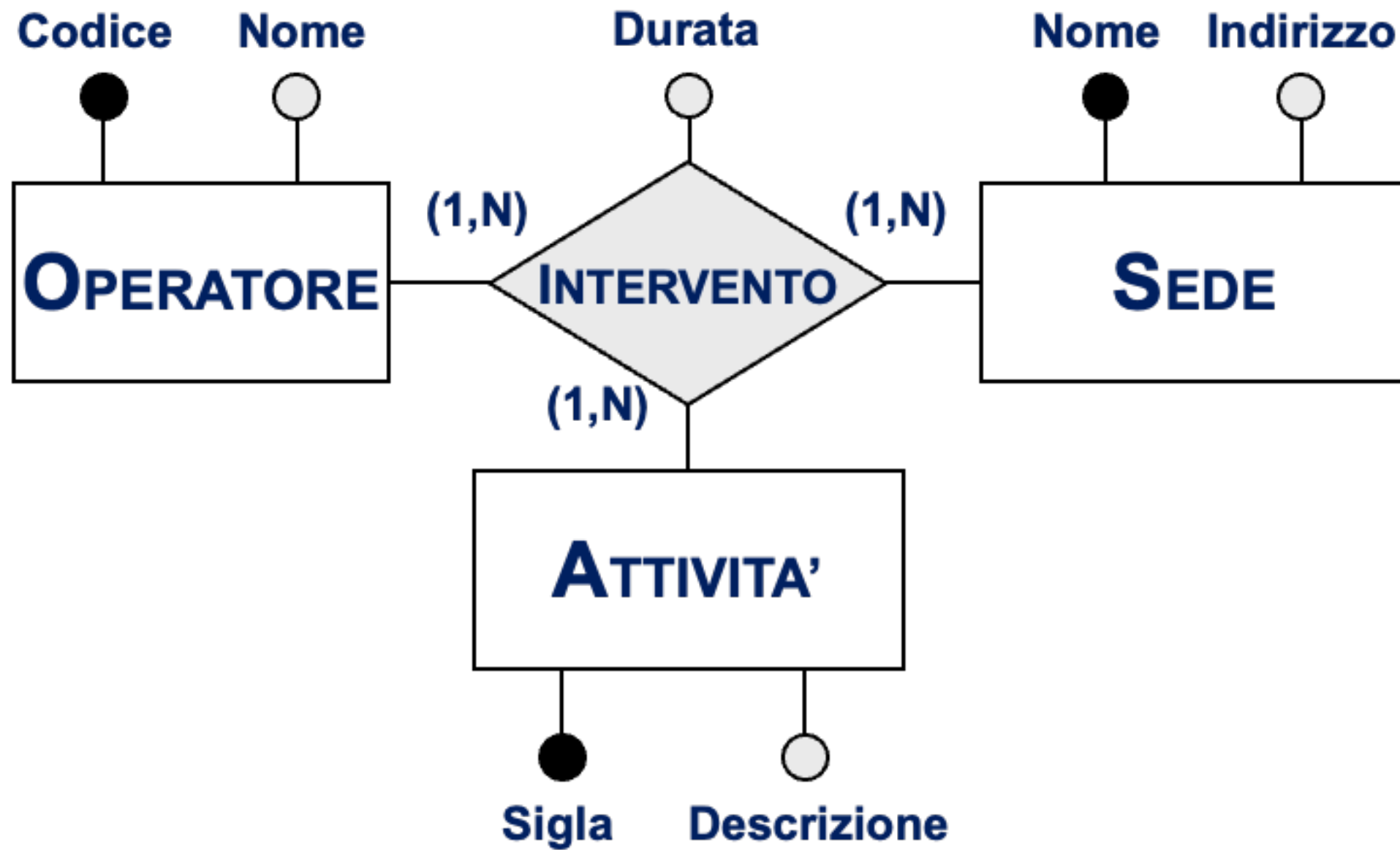
# Storicizzazione di concetto



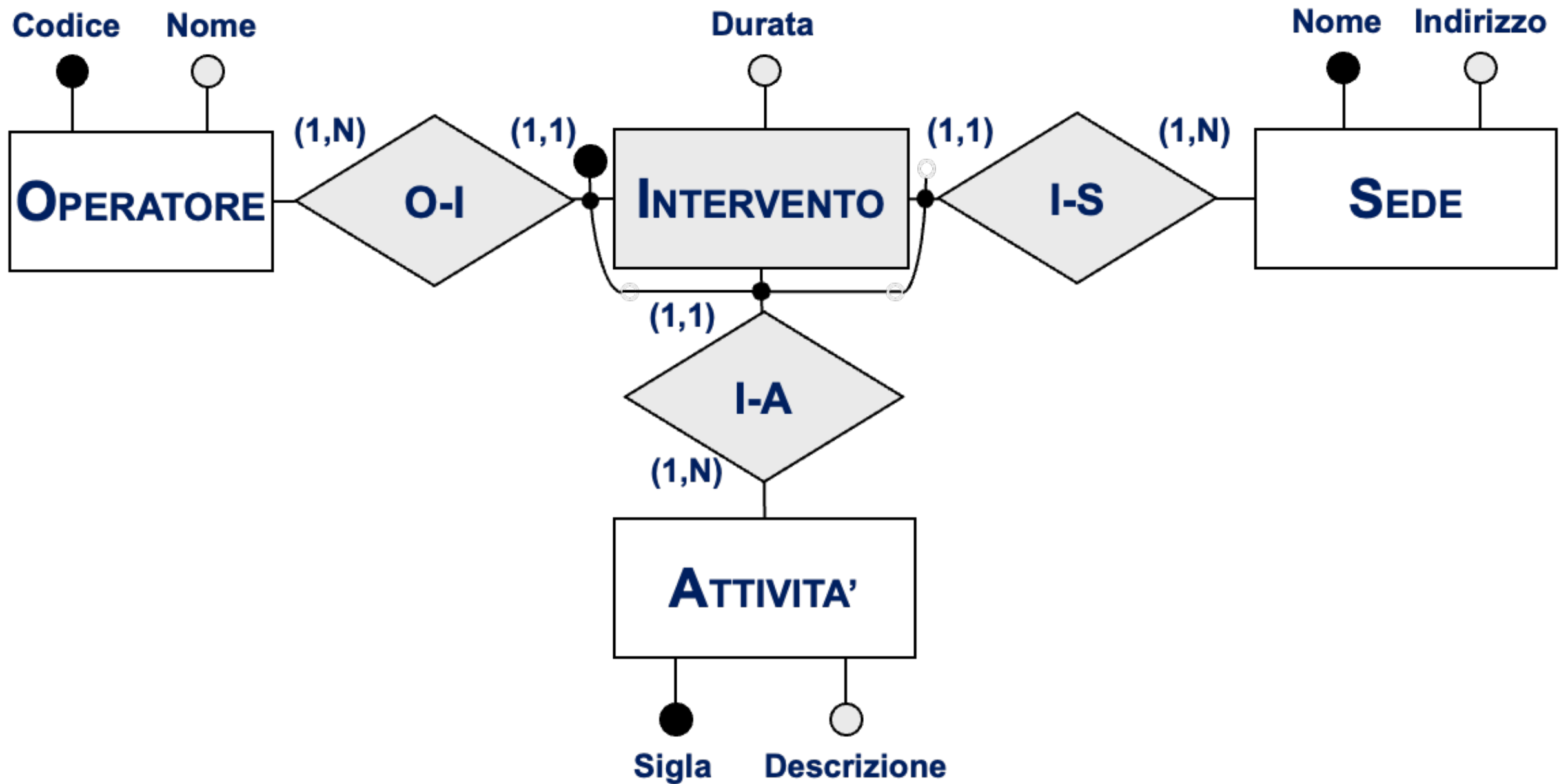
# Evoluzione di concetto



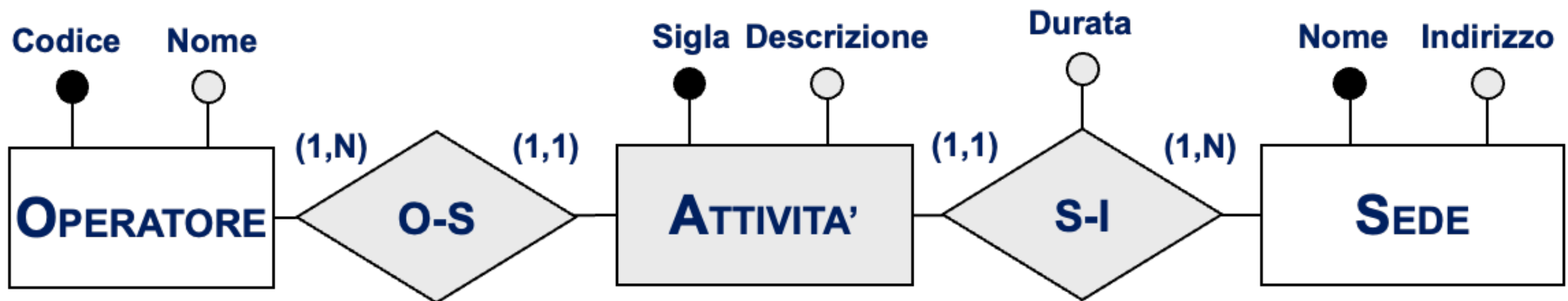
# *Relationship ternaria*



# Reificazione di *relationship* ternaria



# Reificazione di *relationship* ternaria



# Strategie di progetto

- Come procediamo con tante specifiche anche dettagliate? Come ci orientiamo?
- Strategie:
  - *top-down*
  - *bottom-up*
  - *inside-out*



# Strategia *top-down*

- Si parte da uno **schema iniziale** che viene **successivamente raffinato e integrato** per mezzo di **primitive** che lo trasformano in una serie di schemi **intermedi** per arrivare allo schema E-R **finale**
- **Primitive di raffinamento:**
  - Da entità a associazione tra entità
  - Da entità a generalizzazione
  - Da associazione a insiemi di associazioni
  - Da associazione a entità con associazioni
  - Introduzione di attributi su entità e associazioni

# Strategia *bottom-up*

- Si parte dalle **specifiche iniziali** e si suddividono fino a dare specifica ad una **componente minima** di cui si dà lo schema E-R
- Gli schemi prodotti vengono **fusi e integrati** fino ad ottenere lo **schema finale**
- **Primitive di trasformazione:**
  - Generazione di entità
  - Generazione di associazione
  - Generazione di generalizzazione

# Nella pratica...

- Si procede di solito con una **strategia mista**:
  - si individuano i **concetti principali** e si realizza uno **schema scheletro**
  - sulla base di questo si può **decomporre**
  - poi si **raffina**, si **espande**, si **integra**
- **Definizione dello schema scheletro**:
  - Si individuano i **concetti più importanti**, ad esempio perché più citati o perché indicati esplicitamente come cruciali e li si organizza in un **semplice schema concettuale**

# Una metodologia

- **Analisi dei requisiti**

- Analizzare i requisiti ed eliminare le ambiguità
- Costruire un glossario dei termini
- Raggruppare i requisiti in insiemi omogenei

- **Passo base**

- Definire uno schema scheletro con i concetti più rilevanti

- **Passo iterativo** (da ripetere finché non si è soddisfatti)

- Raffinare i concetti presenti sulla base delle loro specifiche
- Aggiungere concetti per descrivere specifiche non descritte

- **Analisi di qualità** (ripetuta e distribuita)

- Verificare le qualità dello schema e modificarlo

# Qualità di uno schema concettuale

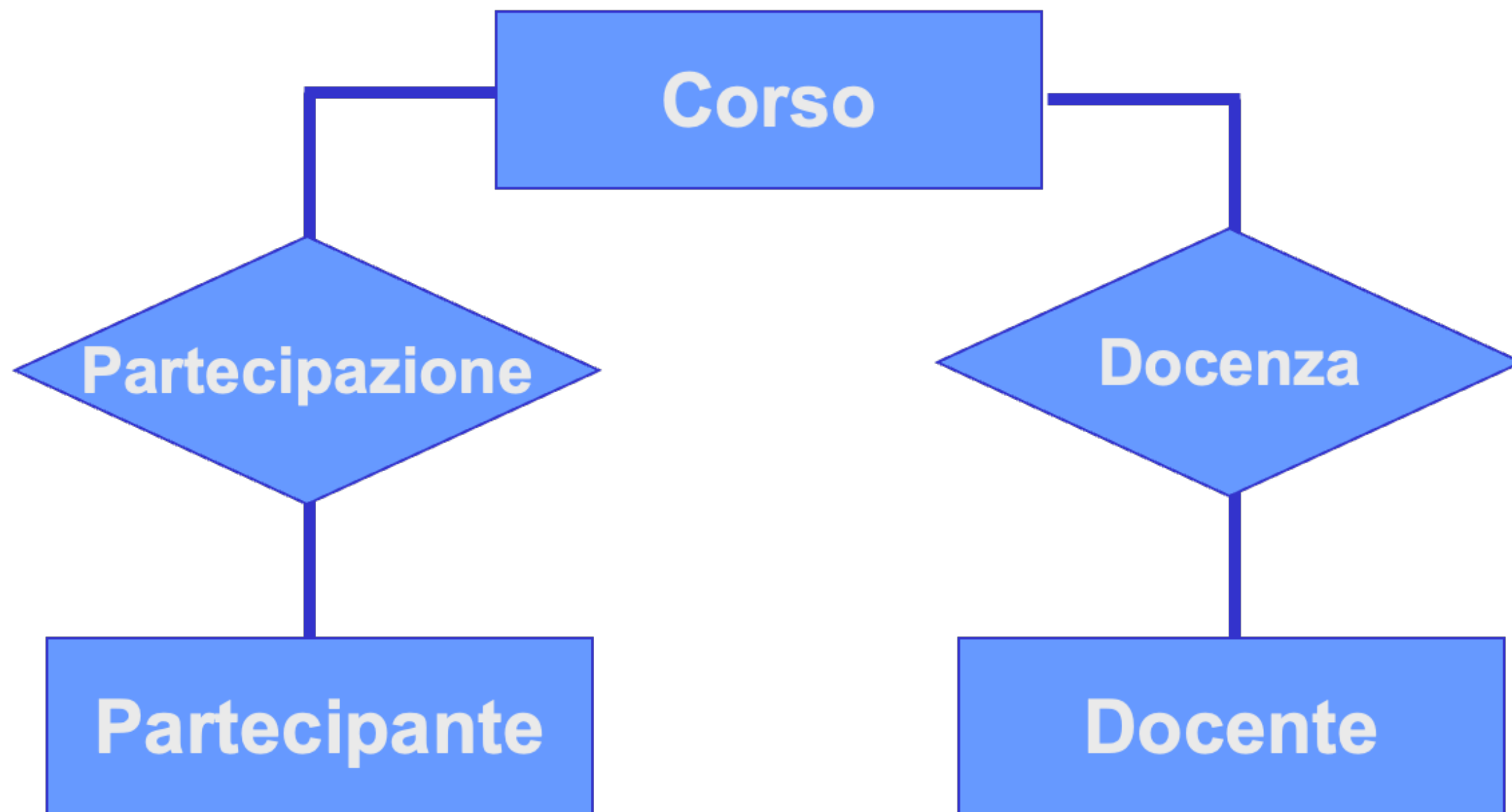
- correttezza
- completezza
- leggibilità
- **minimalità**

# Esempio

## Frasi 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.

# Schema scheletro



# Esempio

## **Frasi 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.



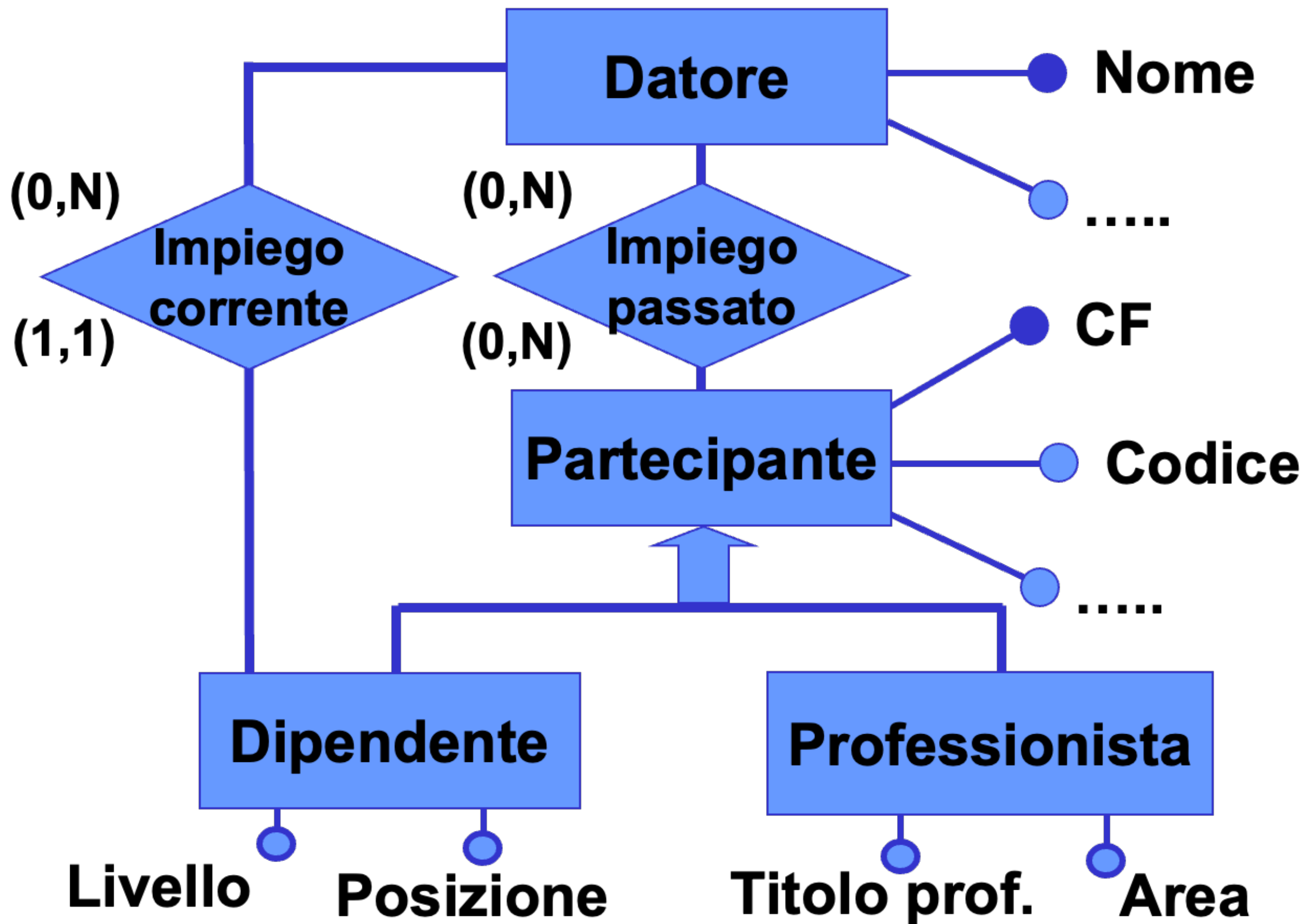
# Esempio

## **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 a tipi specifici di partecipanti**

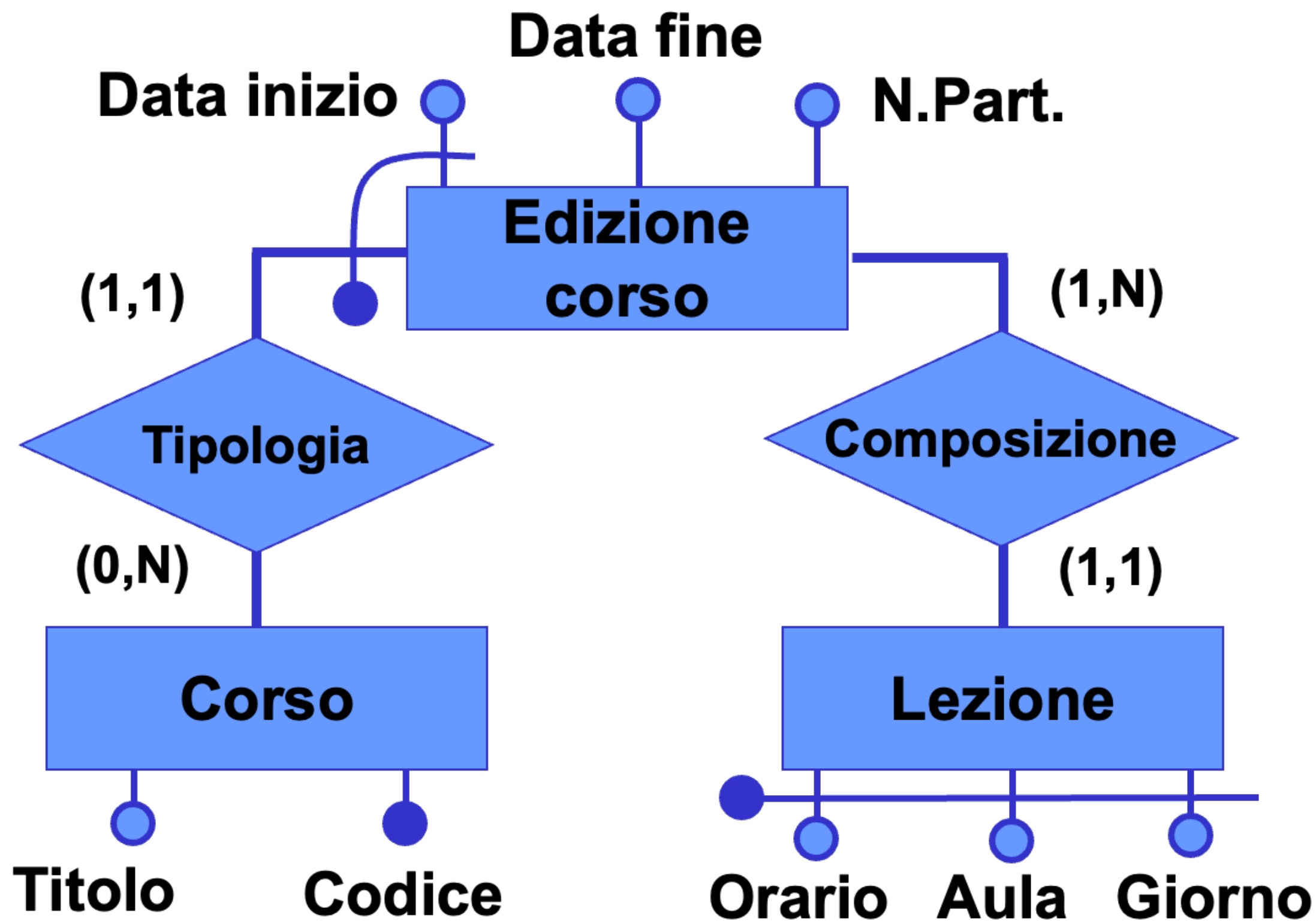
Per i partecipanti che sono liberi professionisti, rappresentiamo l'area di interesse e, se lo possiedono, il titolo professionale. Per i partecipanti che sono dipendenti, rappresentiamo invece il loro livello e la posizione ricoperta.



# Esempio

## **Frase relative ai corsi**

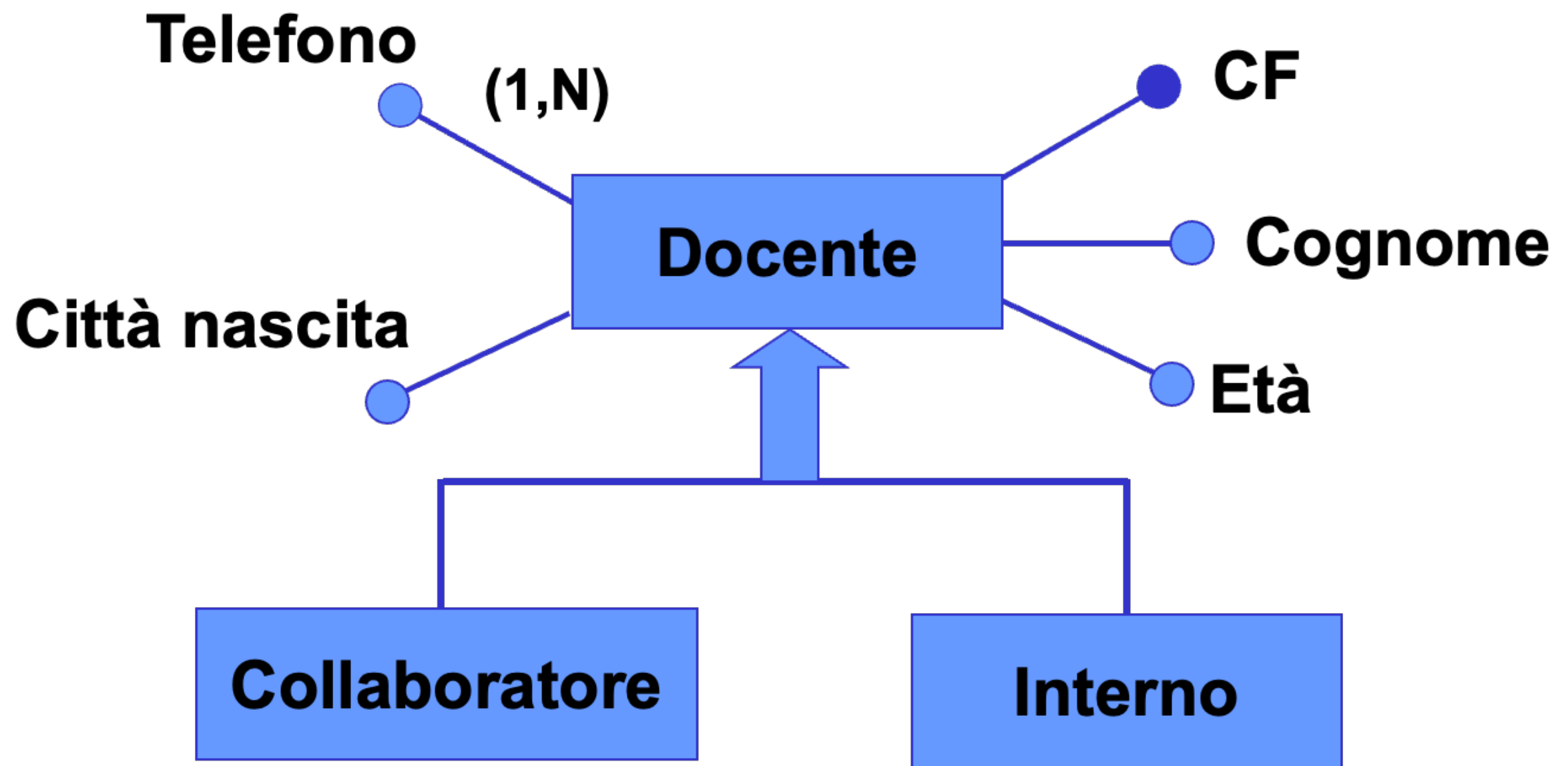
Per i corsi (circa 200), rappresentiamo il titolo e il codice, le varie edizioni con date di inizio e fine e, per ogni edizione, rappresentiamo il numero di partecipanti e il giorno della settimana, le aule e le ore dove sono tenute le lezioni.



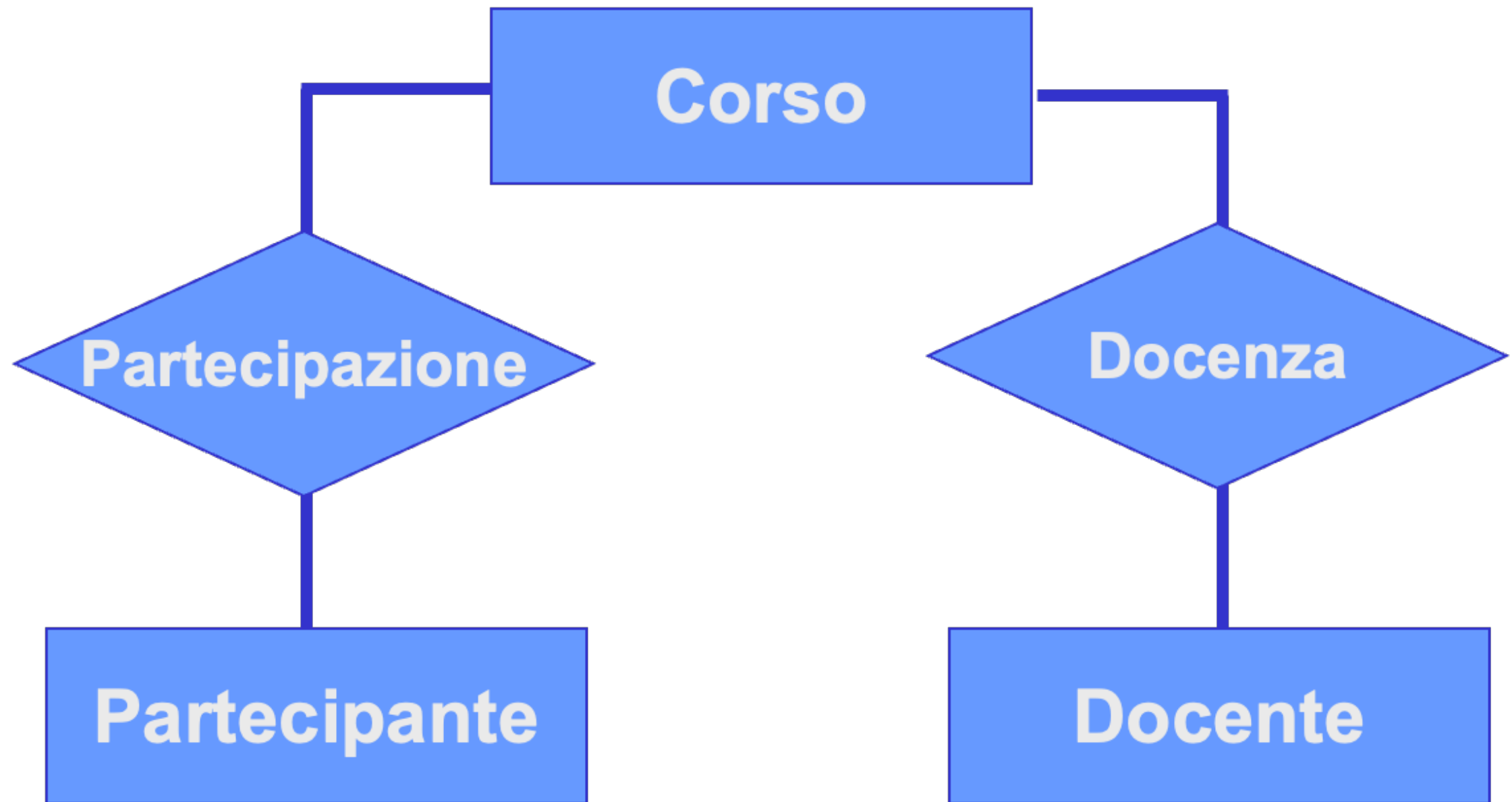
# Esempio

## **Frasi relative ai docenti**

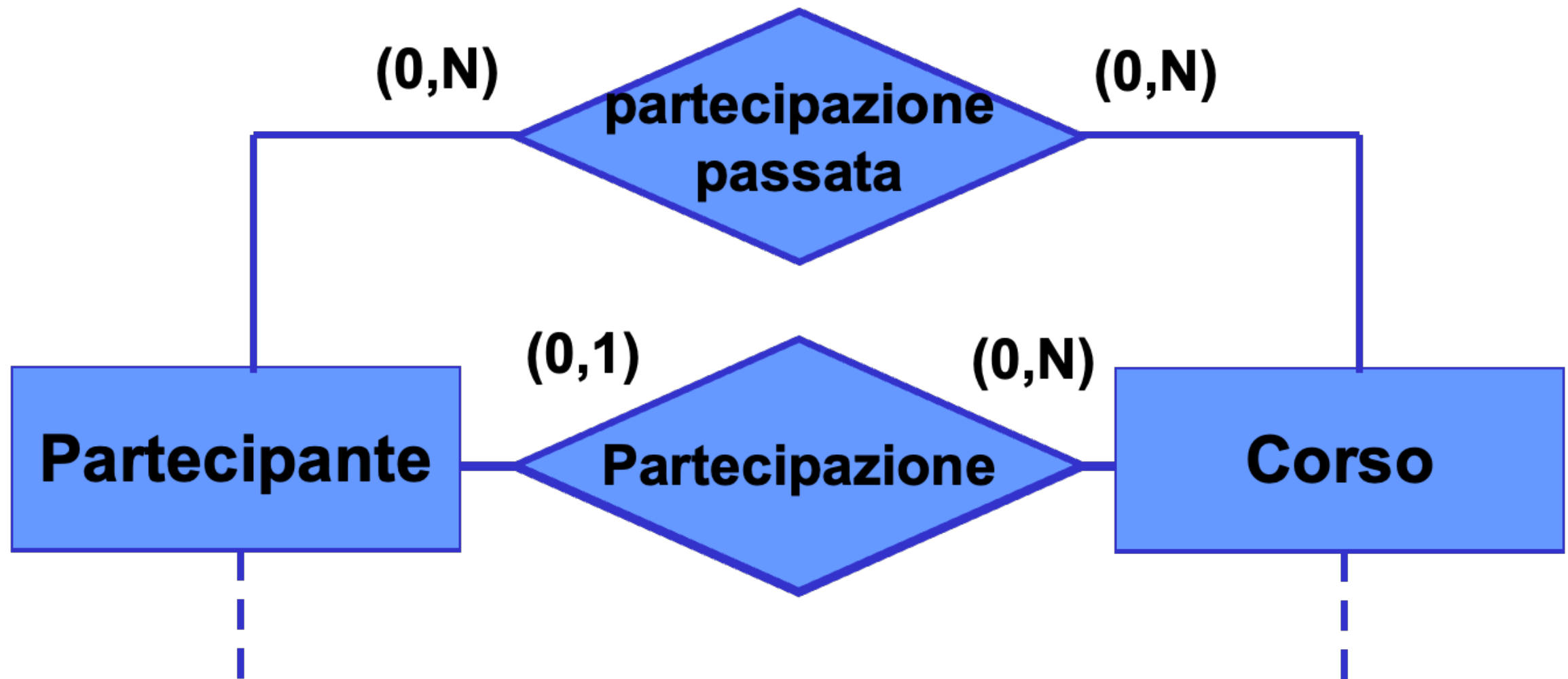
Per i docenti (circa 300), rappresentiamo il cognome, l'età, la città di nascita, tutti i numeri di telefono, il titolo del corso che insegnano, di quelli che hanno insegnato nel passato e di quelli che possono insegnare. I docenti possono essere dipendenti interni della società di formazione o collaboratori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



# Integrazione

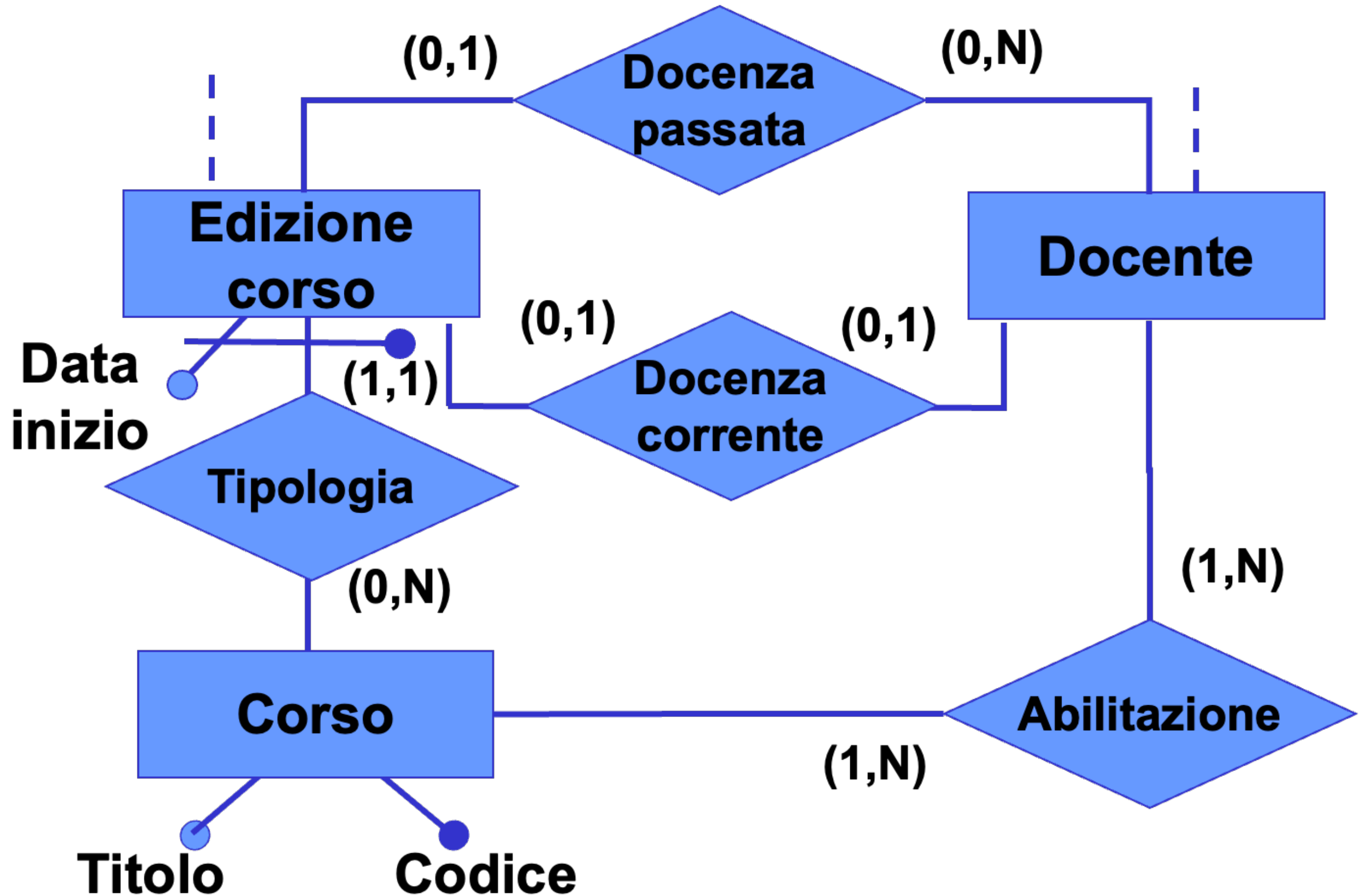


# Integrazione





# Integrazione



# **Progettazione Logica**

Requisiti della base di dati

Progettazione  
concettuale

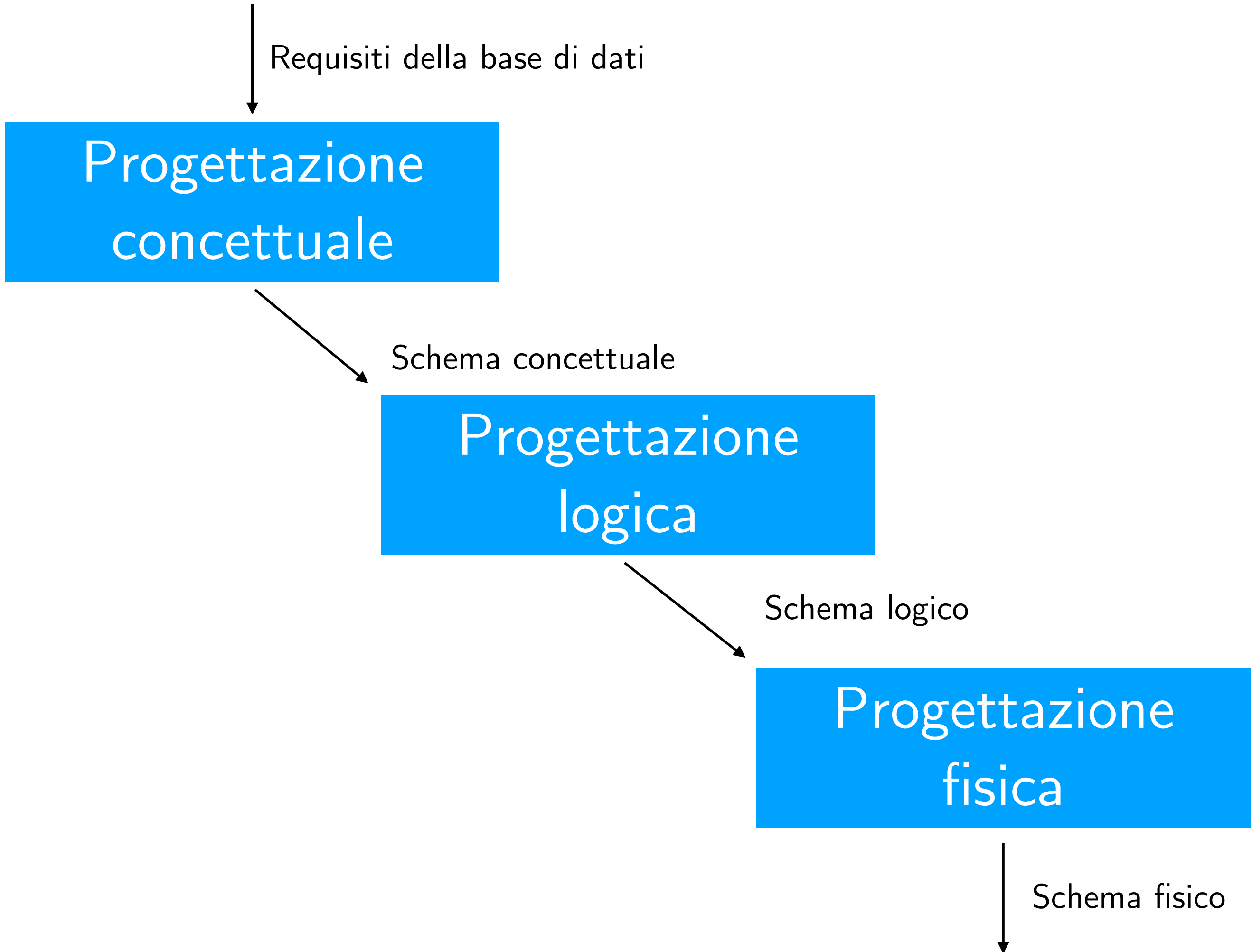
Schema concettuale

Progettazione  
logica

Schema logico

Progettazione  
fisica

Schema fisico

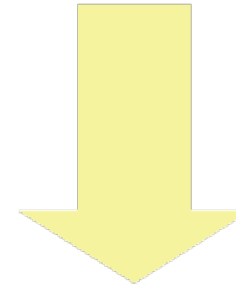


# Obiettivo

- “**Tradurre**” lo schema **concettuale** in uno schema **logico** che rappresenti gli stessi dati in maniera **corretta** ed **efficiente**
- Dati in **ingresso**:
  - schema concettuale
  - informazioni sul **carico applicativo** (dimensione dei dati)
  - modello logico
- Dati in **uscita**:
  - schema logico
  - documentazione associata

**Carico  
applicativo**

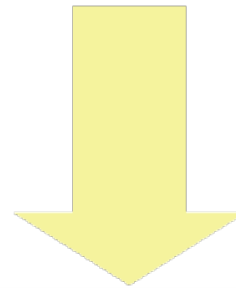
**Schema concettuale  
E-R**



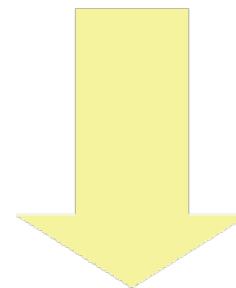
**Ristrutturazione dello  
schema E-R**

**Modello  
logico**

**Schema E-R  
ristrutturato**



**Traduzione nel  
modello logico**



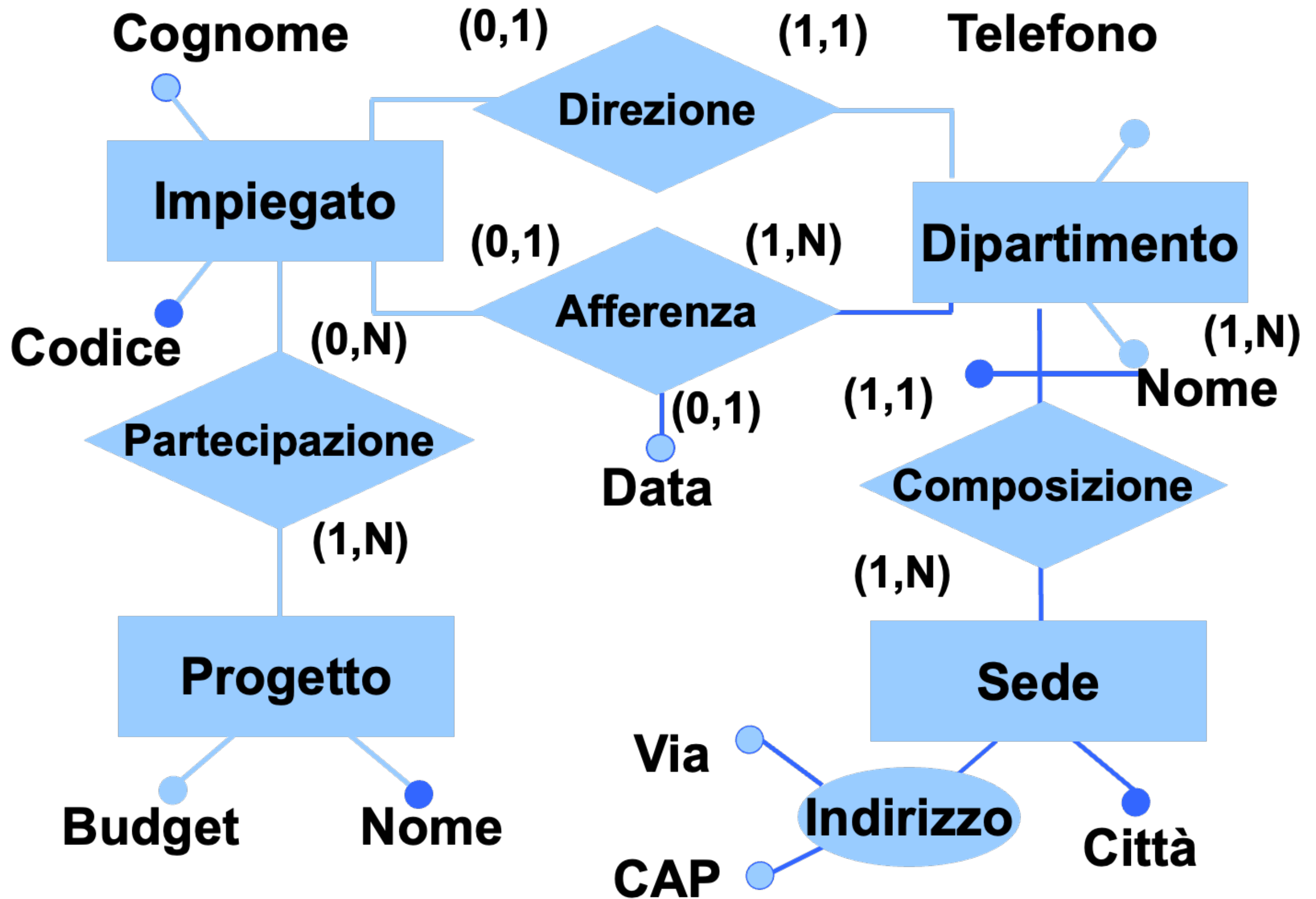
**Schema  
logico**

# Ristrutturazione di uno schema E-R

- **Motivazioni:**
  - **semplificare** la traduzione
  - **"ottimizzare"** le prestazioni
    - come **valutiamo** le **prestazioni**?
- **Osservazione:**
  - uno schema E-R ristrutturato non è (più) uno schema concettuale nel **senso stretto** del termine

# Indicatori per valutare le prestazioni

- Consideriamo degli “**indicatori**” dei parametri che caratterizzano le prestazioni
  - **spazio**: numero di occorrenze previste
  - **tempo**: numero di occorrenze (di entità e *relationship*) visitate per portare a termine un'operazione





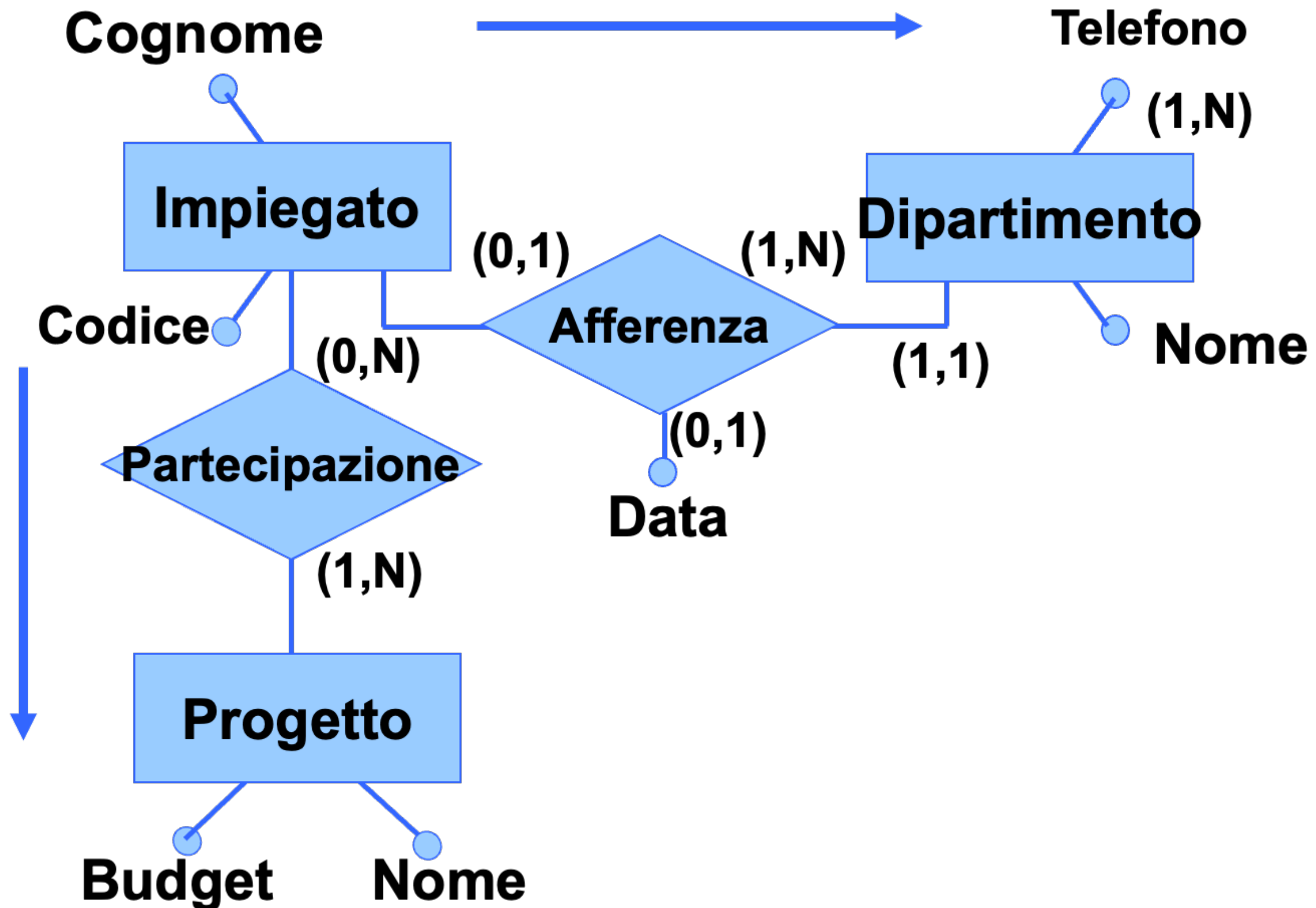
# Tavola dei volumi

Concetto	Tipo	Volume
Sede	E	10
Dipartimento	E	80
Impiegato	E	2000
Progetto	E	500
Composizione	R	80
Afferenza	R	1900
Direzione	R	80
Partecipazione	R	6000

# Indicatori per valutare le prestazioni

- Operazione:
  - trova tutti i dati di un impiegato, del dipartimento nel quale lavora e dei progetti ai quali partecipa
- Si costruisce una **tavola degli accessi** basata su uno **schema di navigazione**

# Schema di navigazione



# Tavola degli accessi

Concetto	Costrutto	Accessi	Tipo
Impiegato	Entità	1	L
Afferenza	Relationship	1	L
Dipartimento	Entità	1	L
Partecipazione	Relationship	3	L
Progetto	Entità	3	L

# Attività di ristrutturazione

- Analisi delle ridondanze
- Eliminazione delle generalizzazioni
- Partizionamento/accorpamento di entità e *relationship*
- Scelta degli identificatori primari

# Attività di ristrutturazione

- **Analisi delle ridondanze**
- Eliminazione delle generalizzazioni
- Partizionamento/accorpamento di entità e *relationship*
- Scelta degli identificatori primari

# Analisi delle ridondanze

- Una **ridondanza** in uno schema E-R è una **informazione significativa ma derivabile** da altre
- In questa fase si decide se **eliminare** le ridondanze eventualmente presenti o **mantenerle** (o anche di **introdurne** di nuove)
- **Vantaggi** delle ridondanze:
  - **semplificazione** delle interrogazioni
- **Svantaggi** delle ridondanze:
  - **appesantimento** degli aggiornamenti
  - maggiore occupazione di **spazio**

# Forme di ridondanza in uno schema E-R

- **Attributi derivabili:**

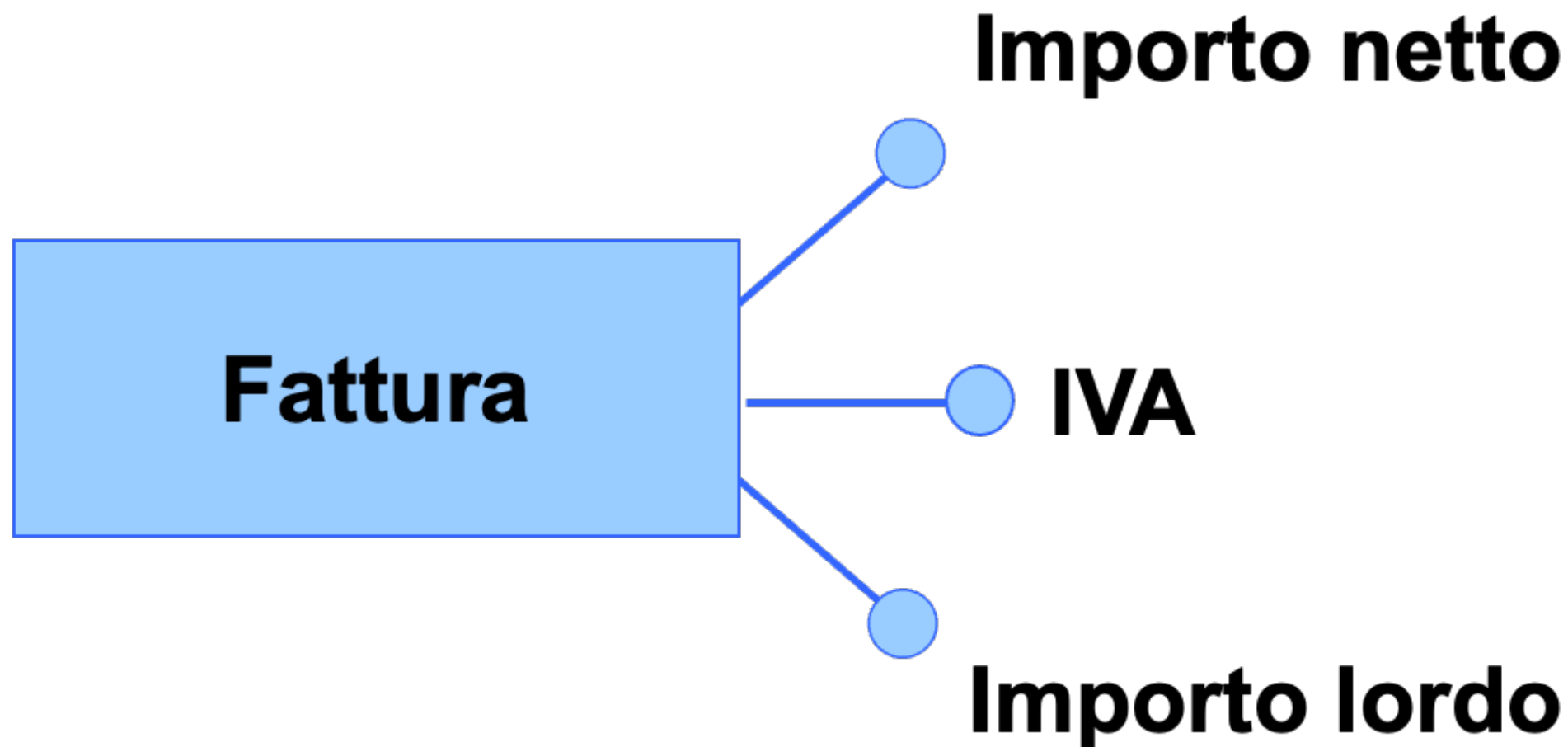
- da altri attributi della stessa entità (o *relationship*)
- da attributi di altre entità (o *relationship*)

- ***Relationship* derivabili:**

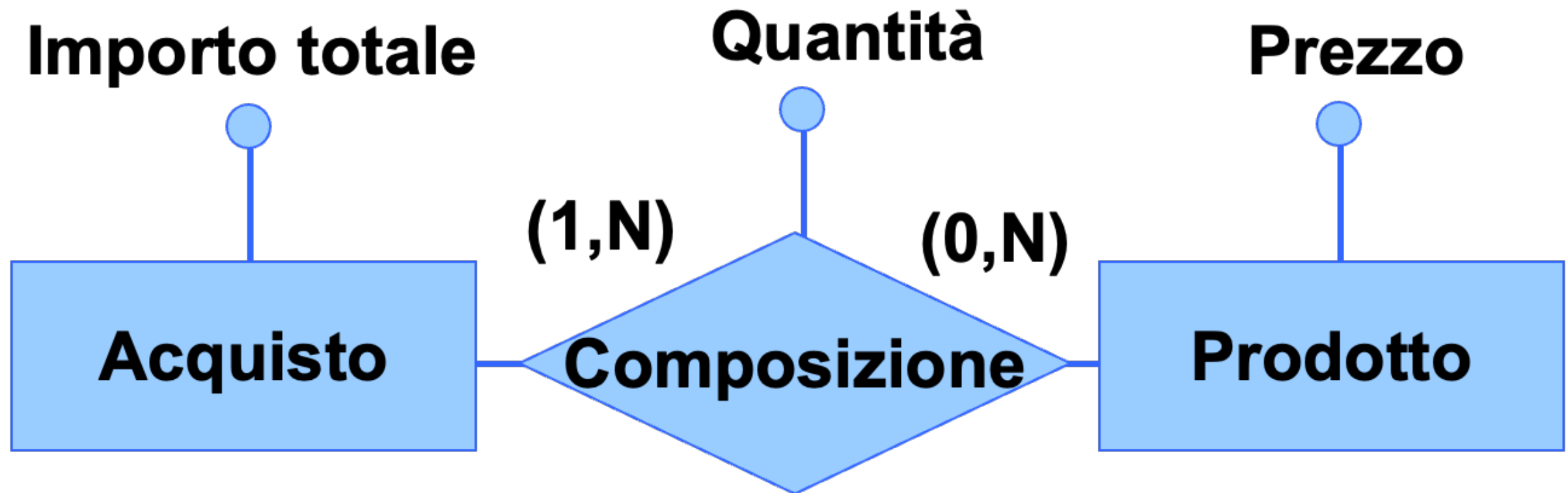
- dalla composizione di altre (più in generale: cicli di *relationship*)



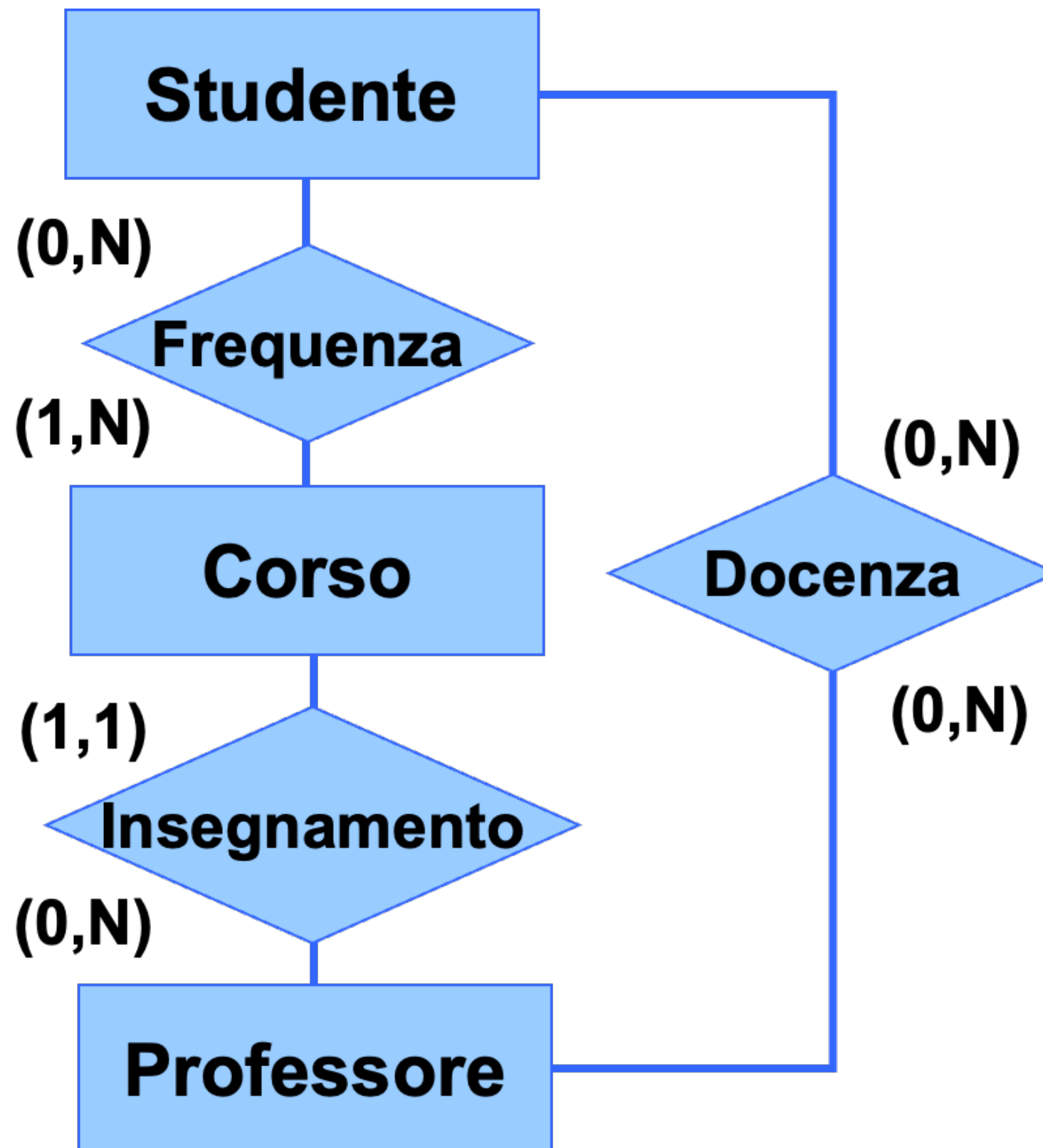
# Attributo derivabile dalla stessa entità



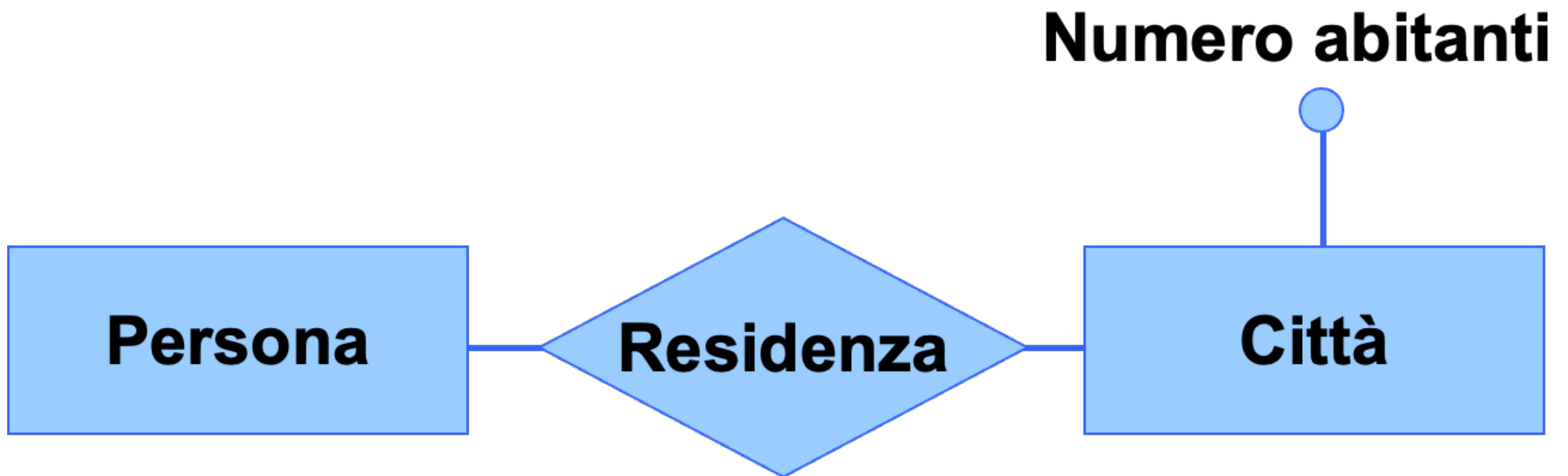
# Attributo derivabile da altra entità



# Ridondanza dovuta a ciclo



# Analisi di una ridondanza



# Tavola dei volumi e operazioni

Concetto	Tipo	Volume
Città	E	200
Persona	E	1000000
Residenza	R	1000000

- **Operazione 1:** memorizza una nuova persona con la relativa città di residenza (500 volte al giorno)
- **Operazione 2:** stampa tutti i dati di una città (incluso il numero di abitanti) (2 volte al giorno)

# Presenza di ridondanza

## Operazione 1

Concetto	Costrutto	Accessi	Tipo
Persona	Entità	1	S
Residenza	Relazione	1	S
Città	Entità	1	L
Città	Entità	1	S

## Operazione 2

Concetto	Costrutto	Accessi	Tipo
Città	Entità	1	L

# Assenza di ridondanza

## Operazione 1

<b>Concetto</b>	<b>Costrutto</b>	<b>Accessi</b>	<b>Tipo</b>
Persona	Entità	1	S
Residenza	Relazione	1	S

## Operazione 2

<b>Concetto</b>	<b>Costrutto</b>	<b>Accessi</b>	<b>Tipo</b>
Città	Entità	1	L
Residenza	Relazione	5000	L

# Costi

- Presenza di ridondanza:
  - Costi:
    - Operazione 1: 1500 accessi in scrittura e 500 accessi in lettura al giorno
    - Operazione 2: trascurabile
  - Contiamo doppi gli accessi in scrittura
    - Totale di 3500 accessi al giorno
- **Assenza di ridondanza:**
  - Costi:
    - Operazione 1: 1000 accessi in scrittura
    - Operazione 2: 10000 accessi in lettura al giorno
  - Contiamo doppi gli accessi in scrittura
    - Totale di 12000 accessi al giorno



# Attività di ristrutturazione

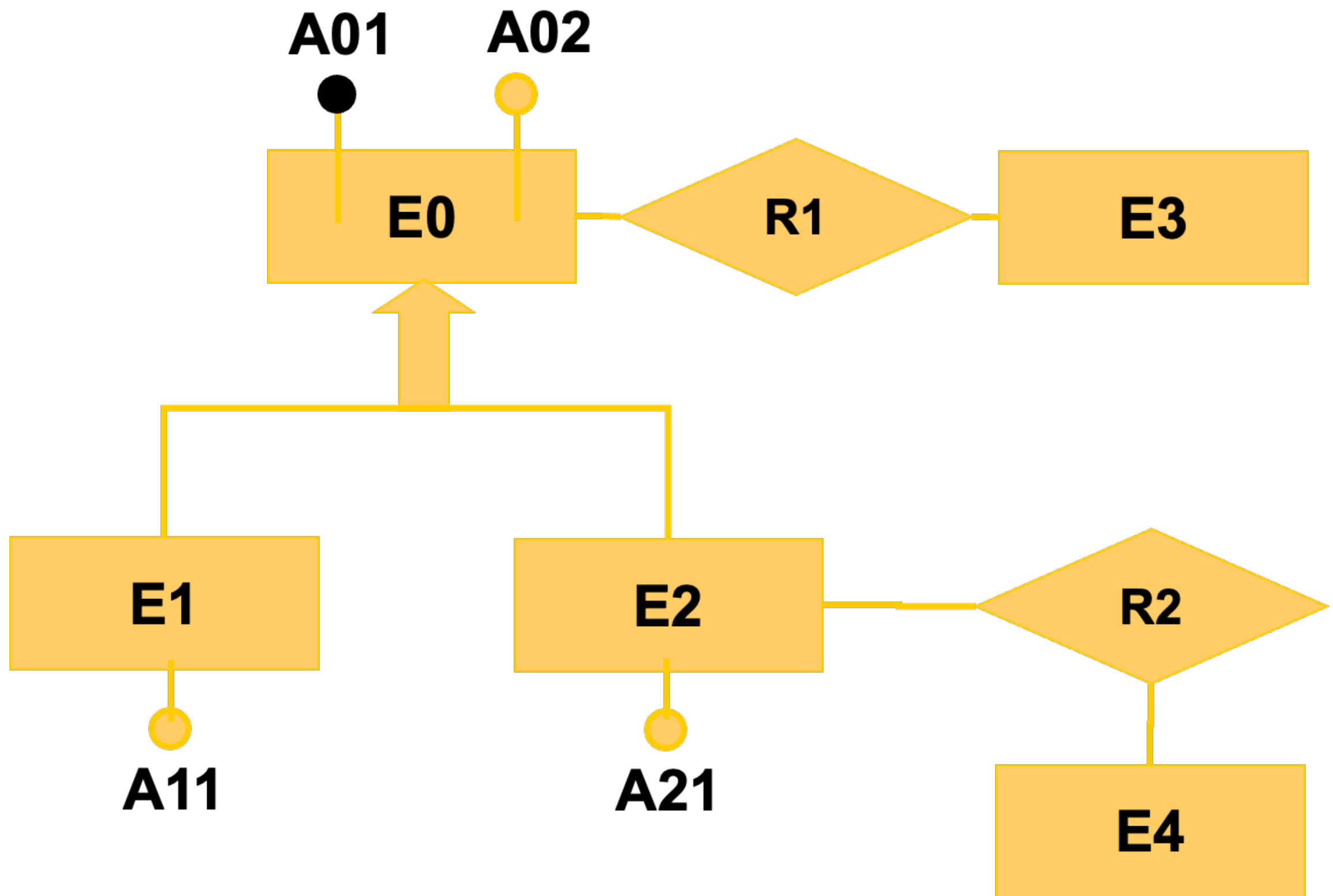
- Analisi delle ridondanze
- **Eliminazione delle generalizzazioni**
- Partizionamento/accorpamento di entità e *relationship*
- Scelta degli identificatori primari

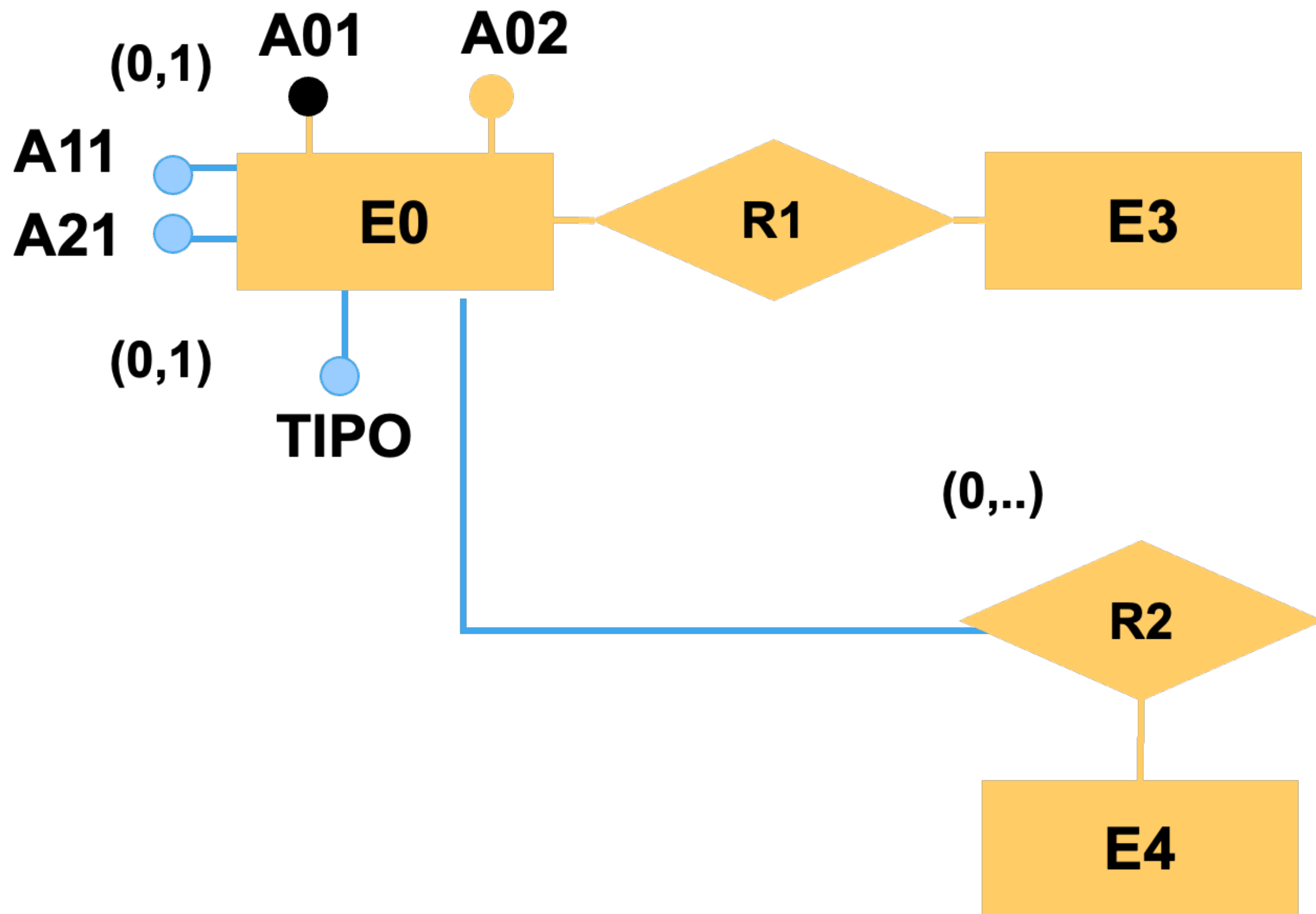
# Le gerarchie nel modello relazionale

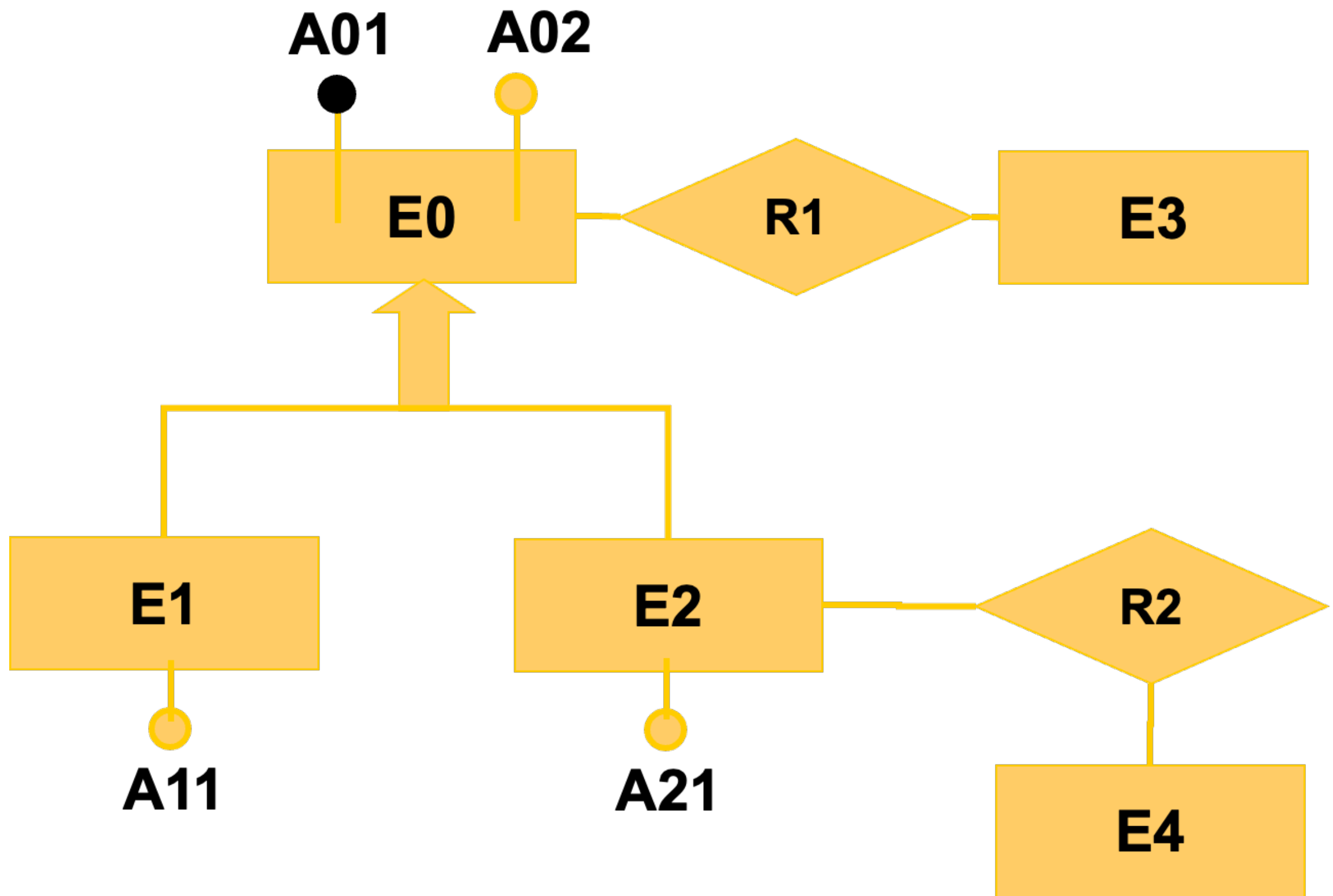
- Il modello relazionale **non può rappresentare** direttamente le generalizzazioni
- **Entità** e *relationship* sono invece direttamente rappresentabili
- Si **eliminano perciò le gerarchie**, sostituendole con entità e *relationship*

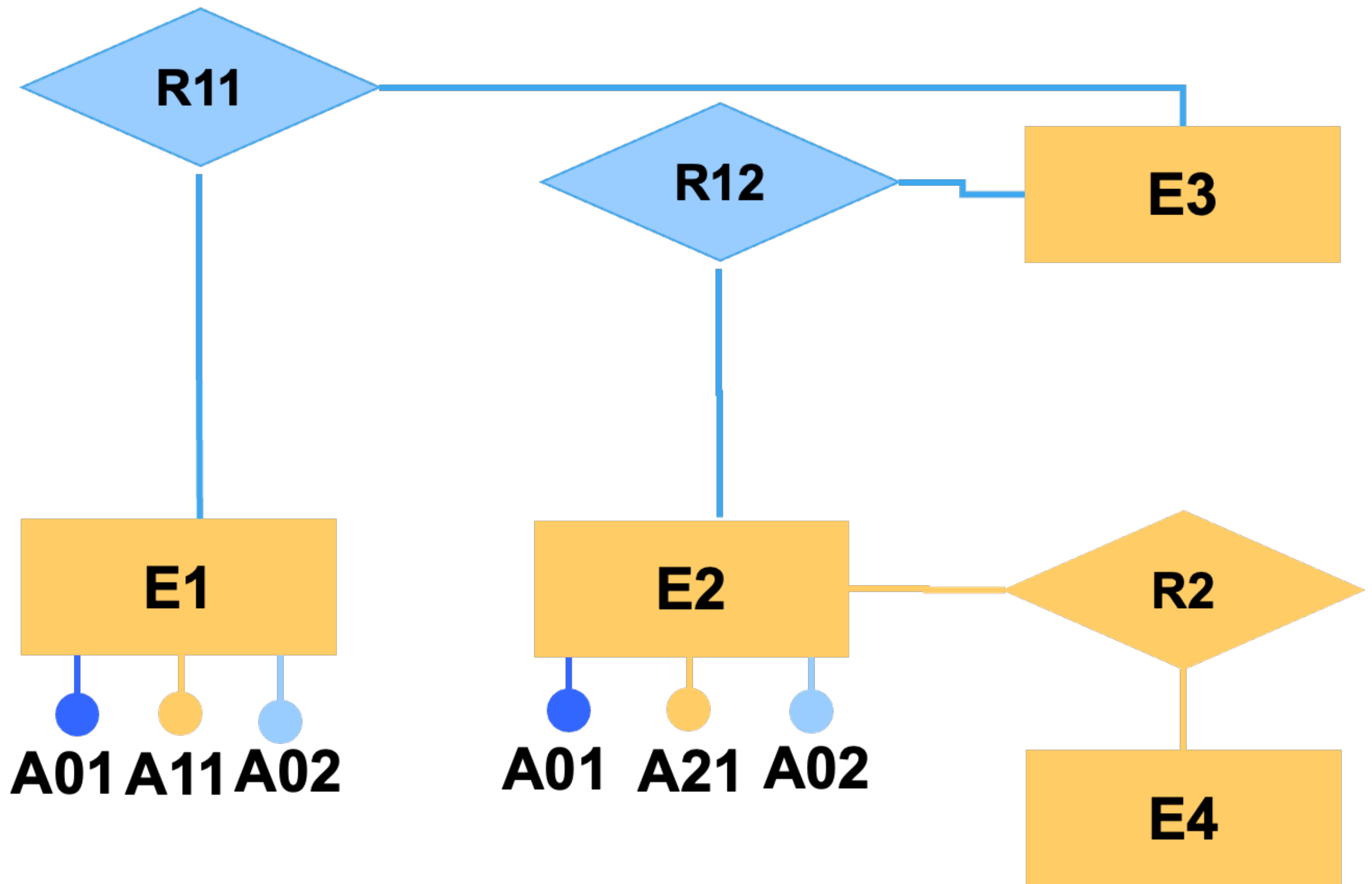
# Possibilità

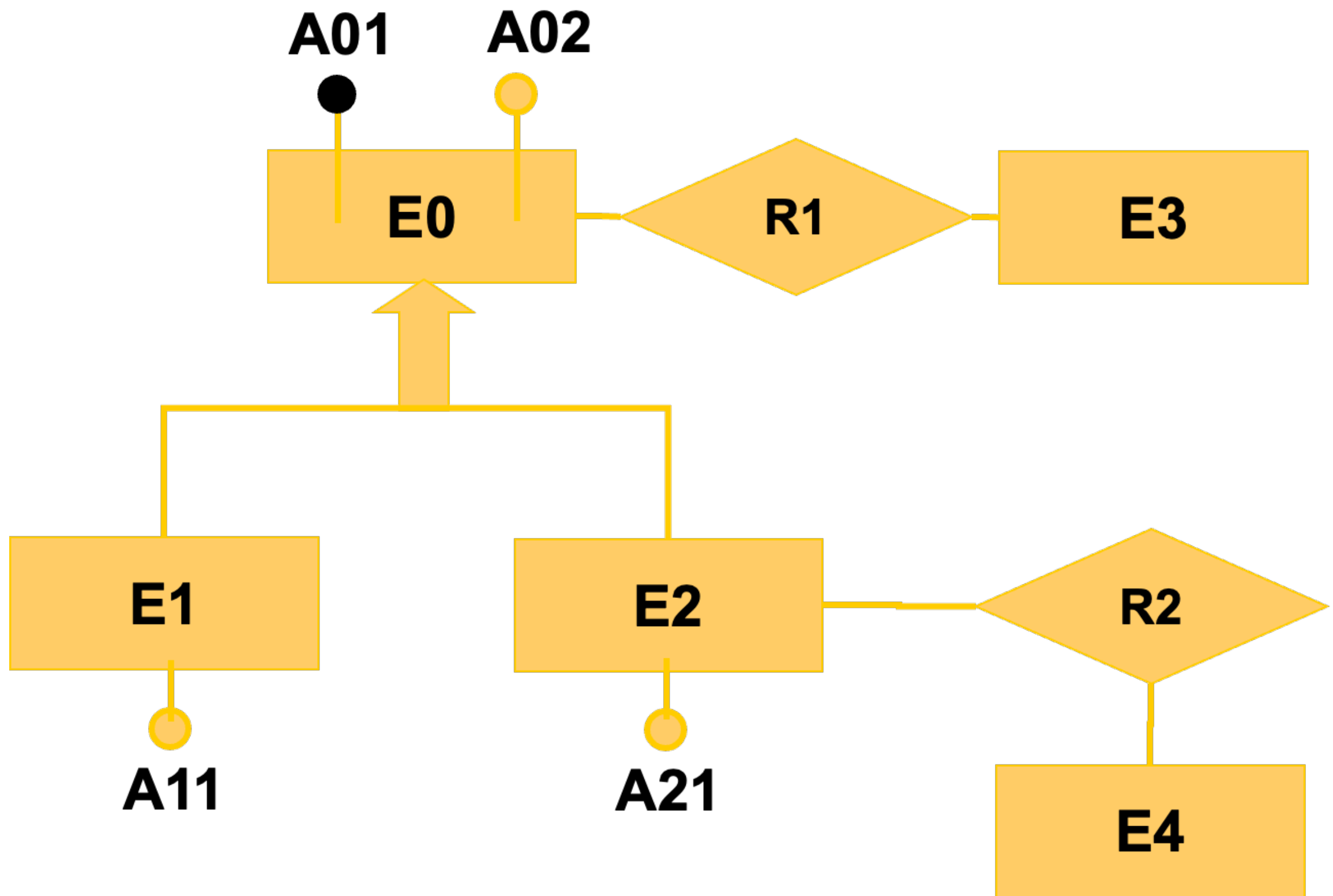
1. **Accorpamento delle figlie** della generalizzazione nel genitore
2. **Accorpamento del genitore** della generalizzazione nelle figlie
3. **Sostituzione** della generalizzazione con *relationship*



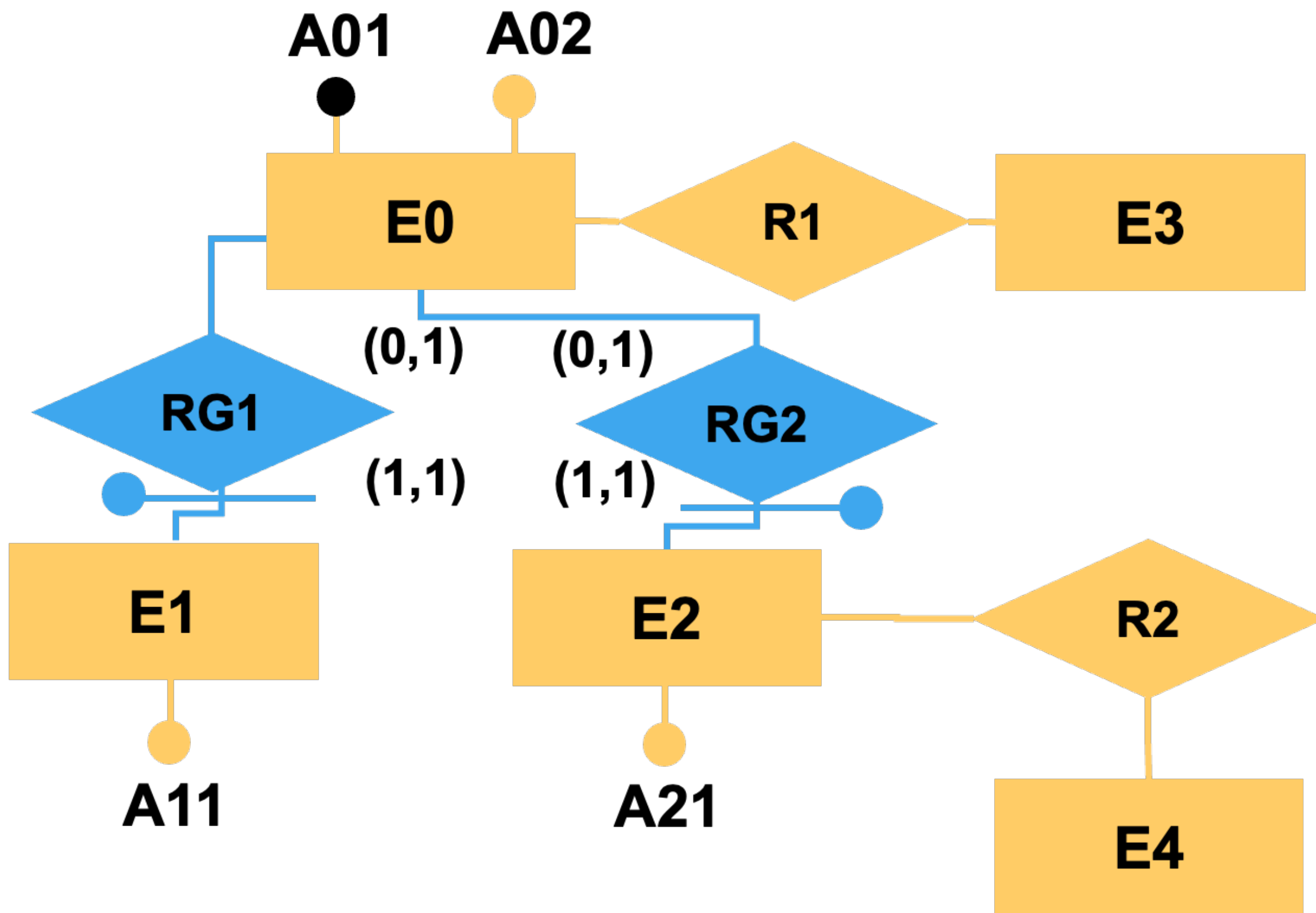






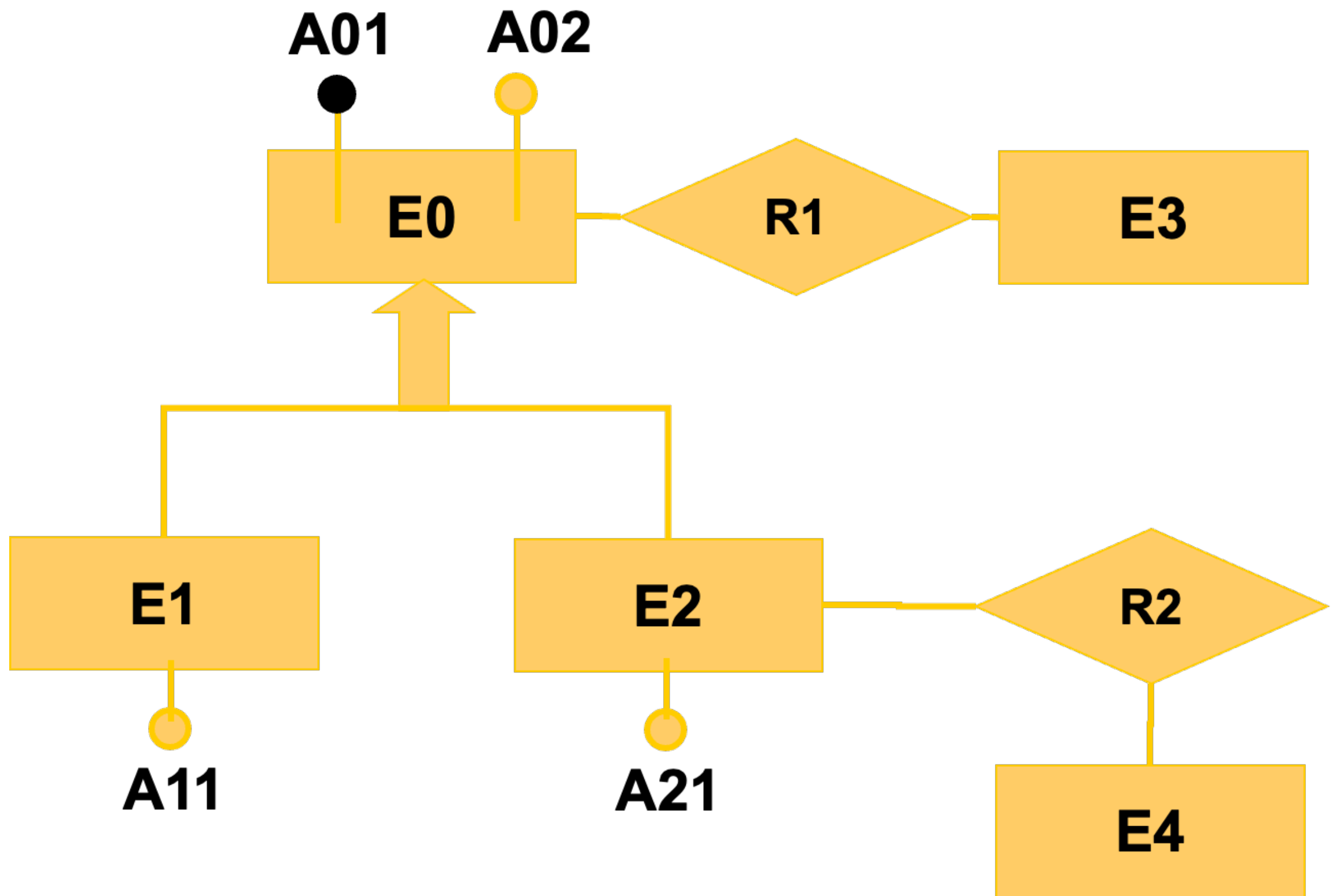


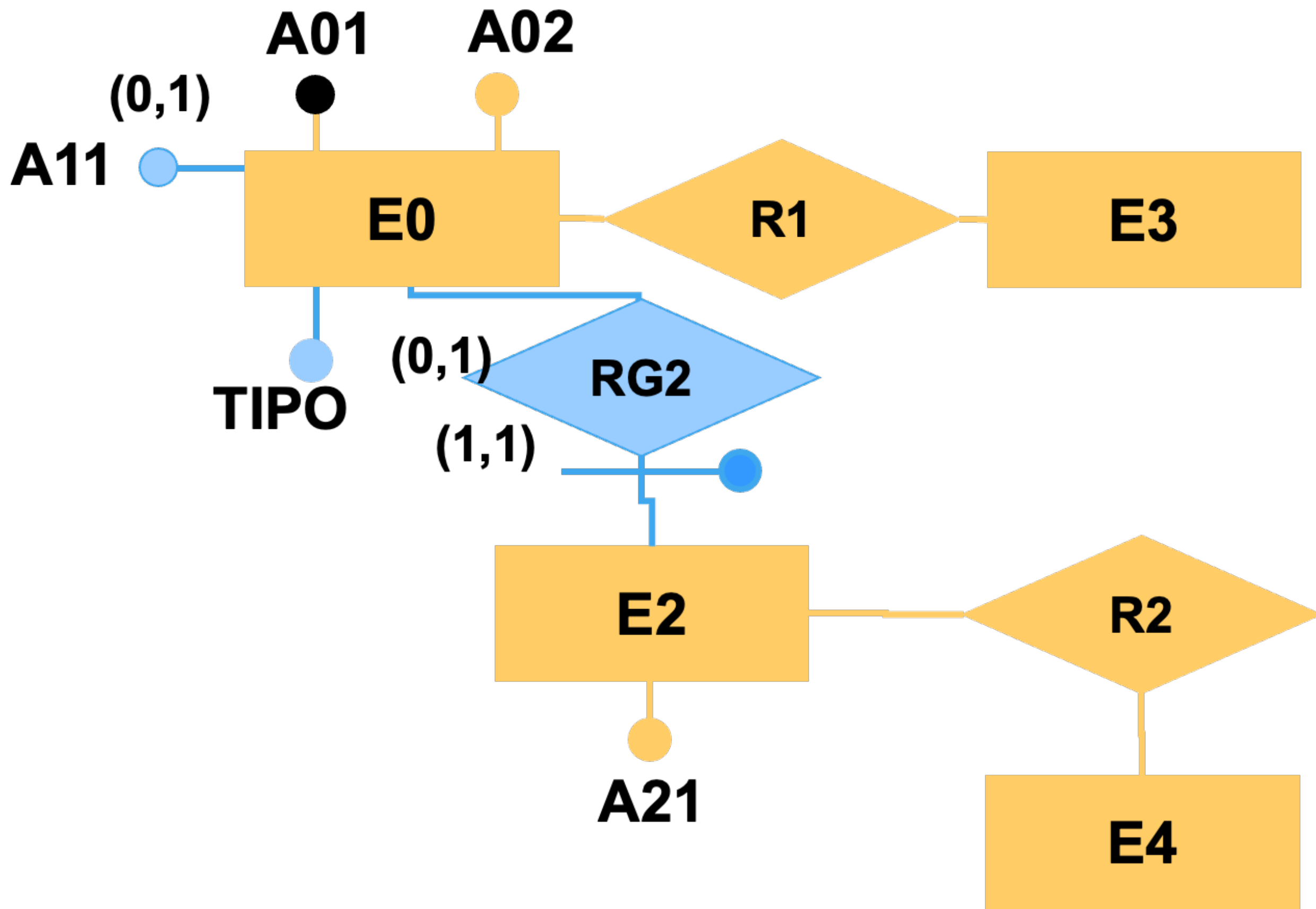




# Come scegliere?

- La scelta fra le alternative si può fare basandosi sul numero e il tipo degli accessi fatti alle singole entità per eseguire le operazioni
- È possibile seguire alcune semplici regole generali:
  - la prima conviene se gli accessi al padre e alle figlie sono contestuali;
  - la seconda conviene se gli accessi alle figlie sono distinti;
  - la terza conviene se gli accessi alle entità figlie sono separati dagli accessi al padre;
  - sono anche possibili soluzioni “ibride”, soprattutto in gerarchie a più livelli.





# Attività di ristrutturazione

- Analisi delle ridondanze
- Eliminazione delle generalizzazioni
- **Partizionamento/accorpamento di entità e *relationship***
- Scelta degli identificatori primari

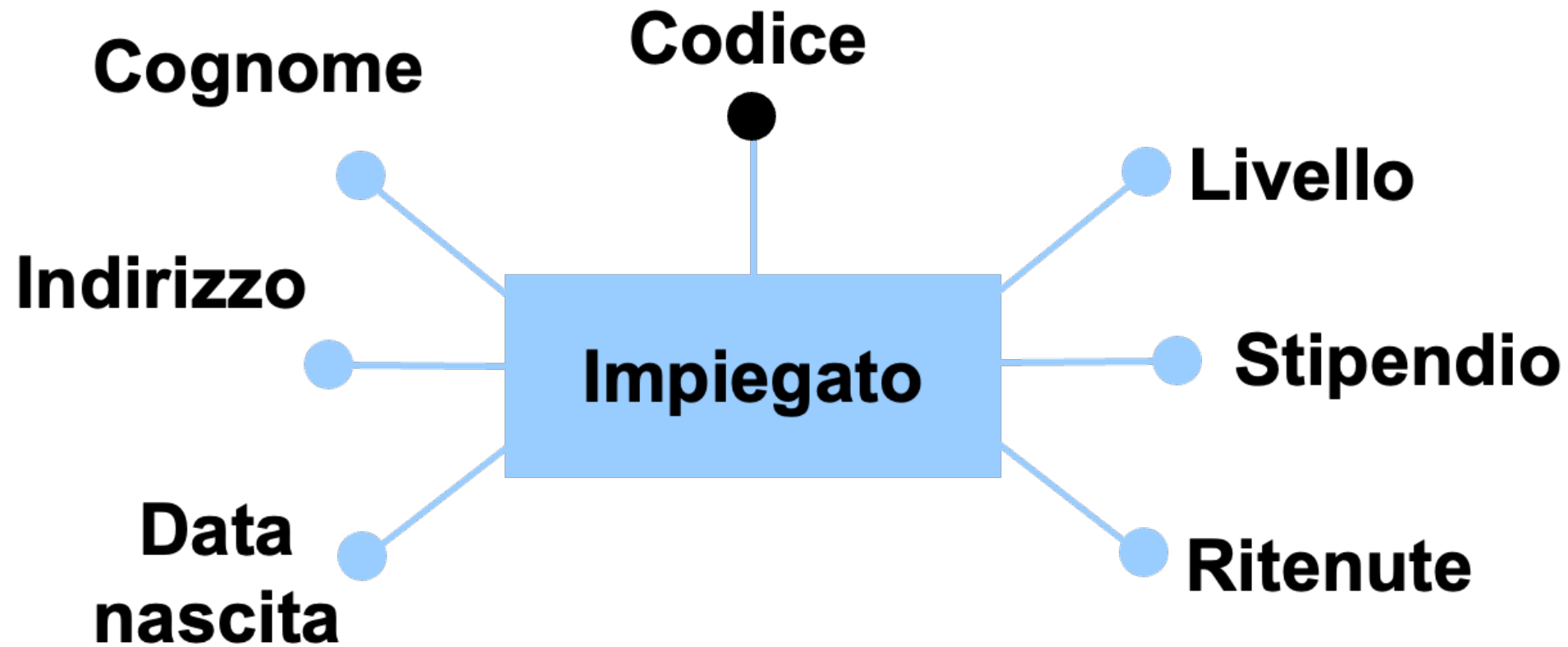
# Motivazione

- Ristrutturazioni effettuate per rendere **più efficienti** le operazioni in base al principio che:
  - **Gli accessi si riducono**
    - **separando attributi** di un concetto che vengono acceduti separatamente
    - **raggruppando attributi** di concetti diversi acceduti insieme
  - Si considera sempre che ad ogni accesso si **legge l'intera informazione**

# Casi principali

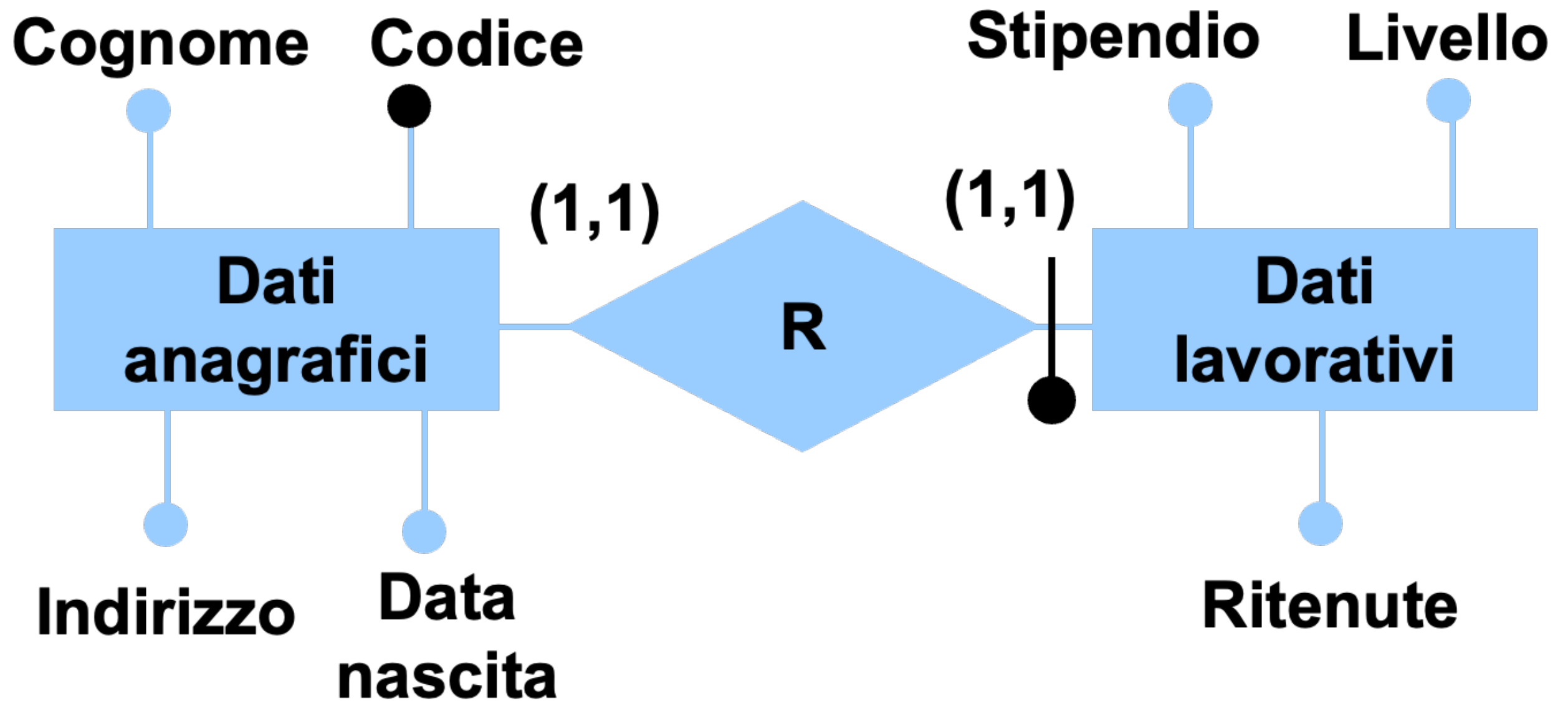
- **Partizionamento verticale** di entità
- **Partizionamento orizzontale** di *relationship*
- Eliminazione di **attributi multivalore**
- **Accorpamento** di entità/*relationship*

# Partizionamento verticale di entità

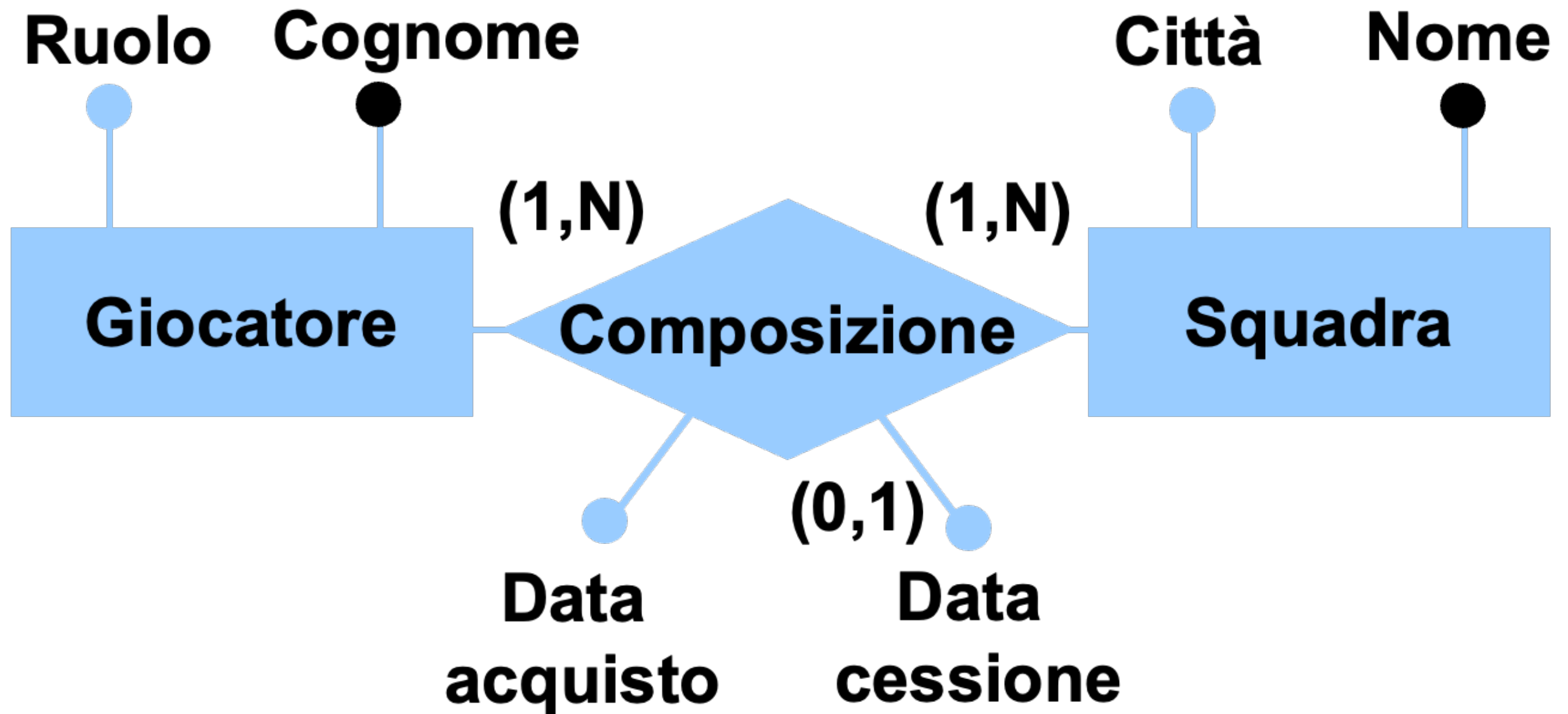




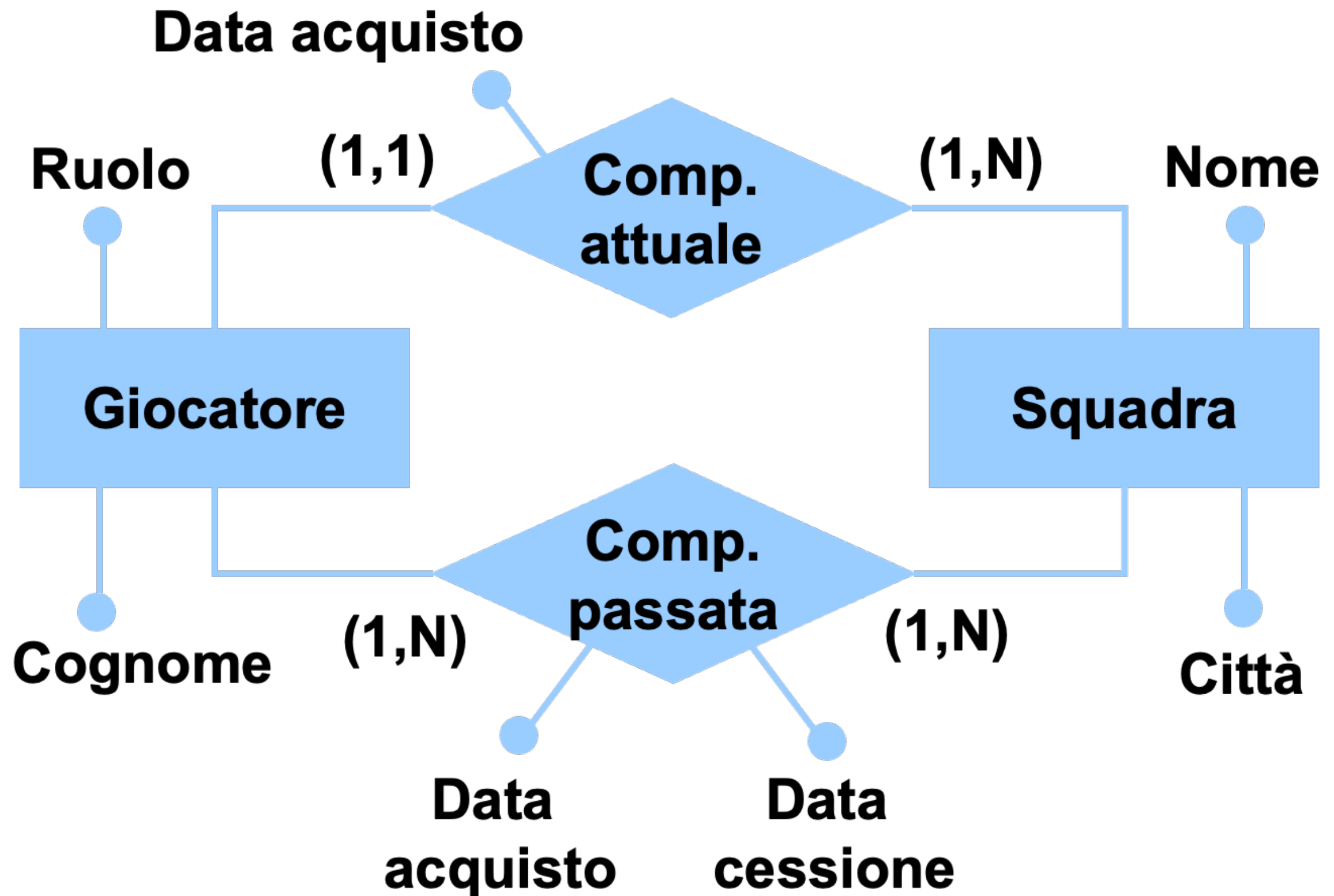
# Partizionamento verticale di entità



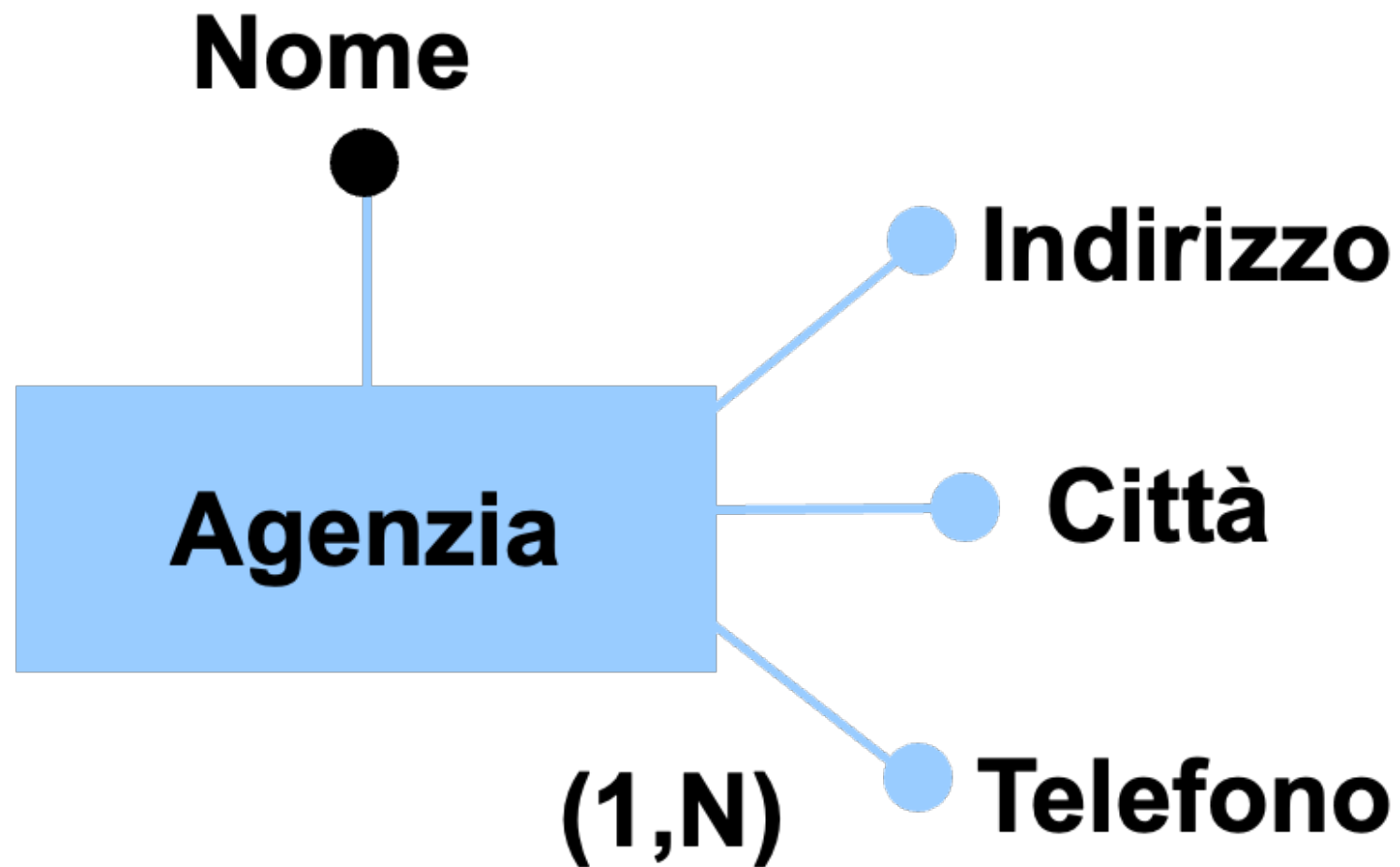
# Partizione orizzontale di *relationship*



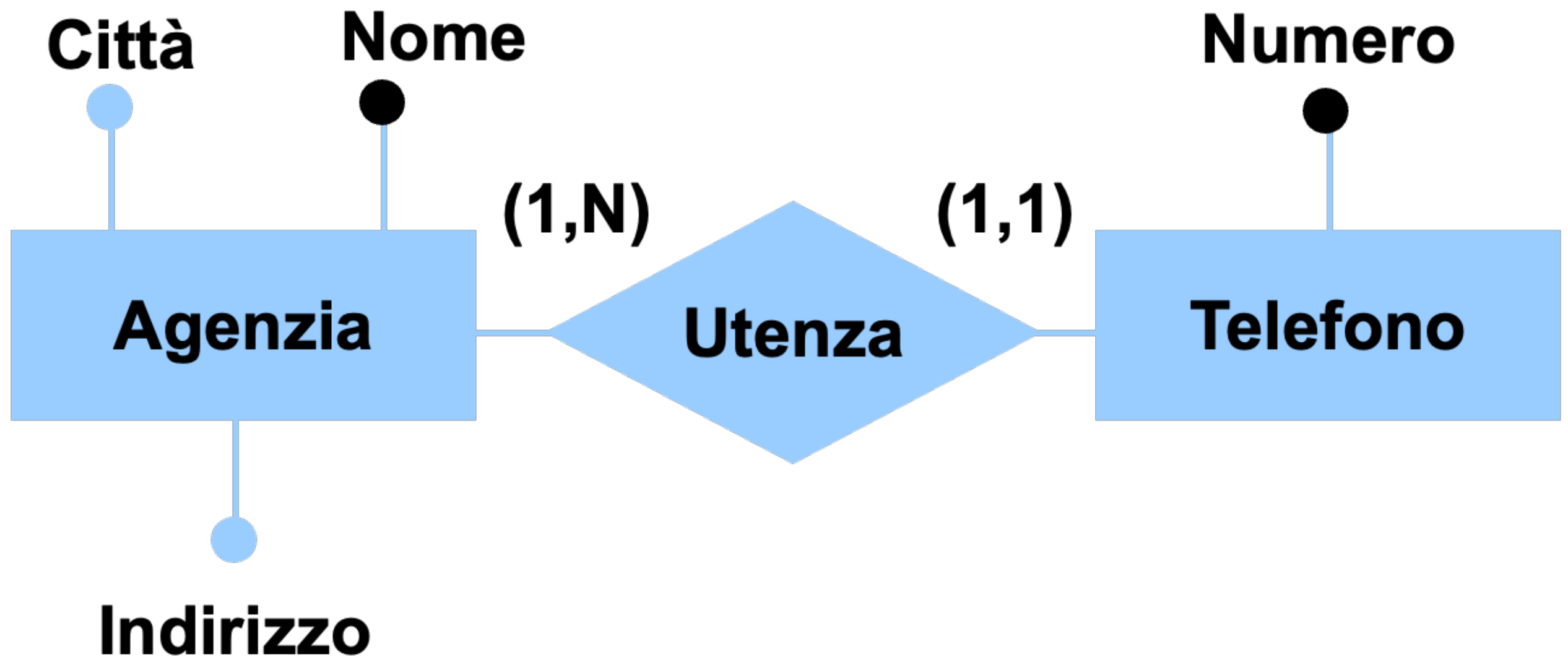
# Partizione orizzontale di *relationship*



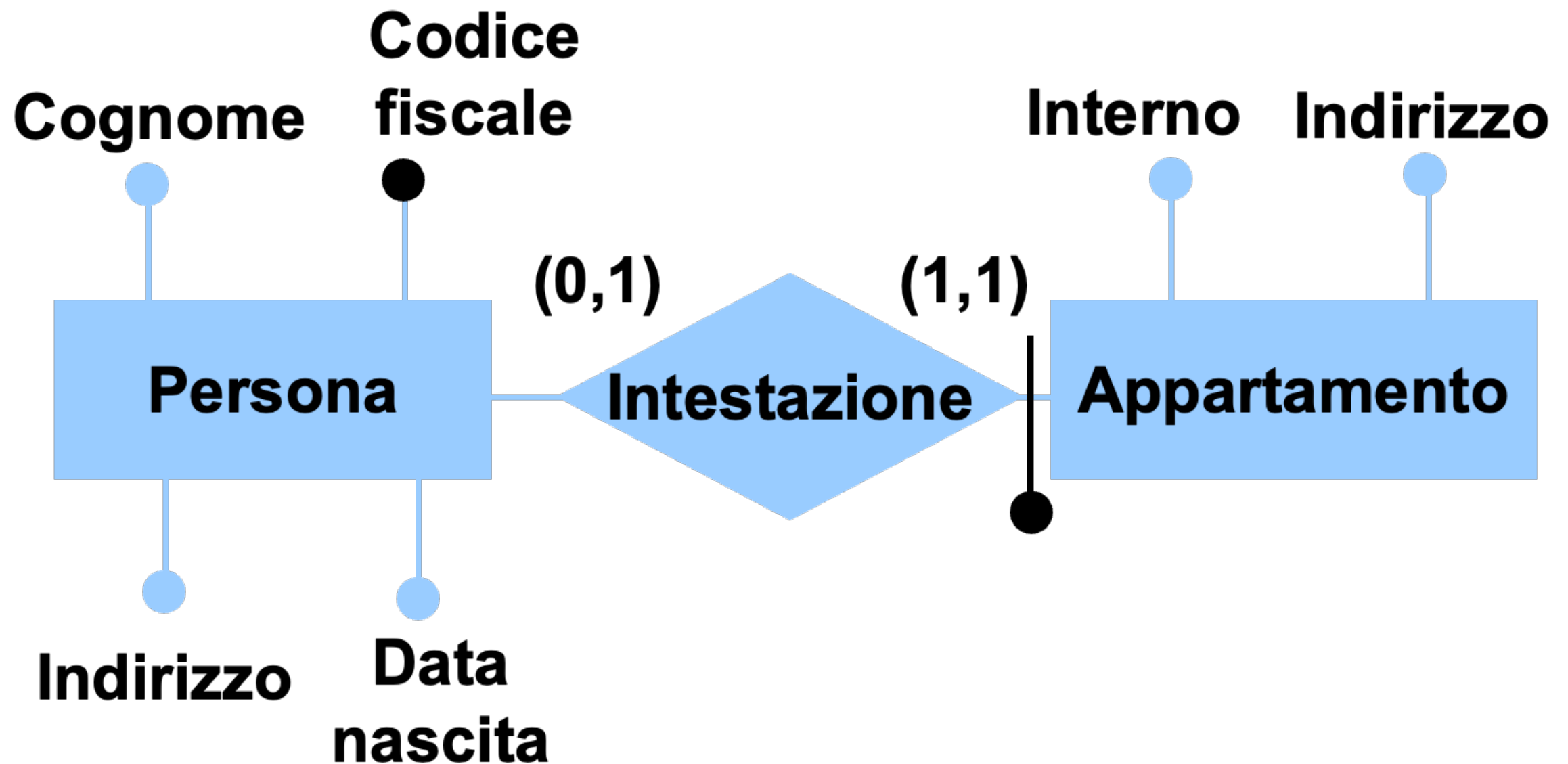
# Eliminazione di attributi multivalore



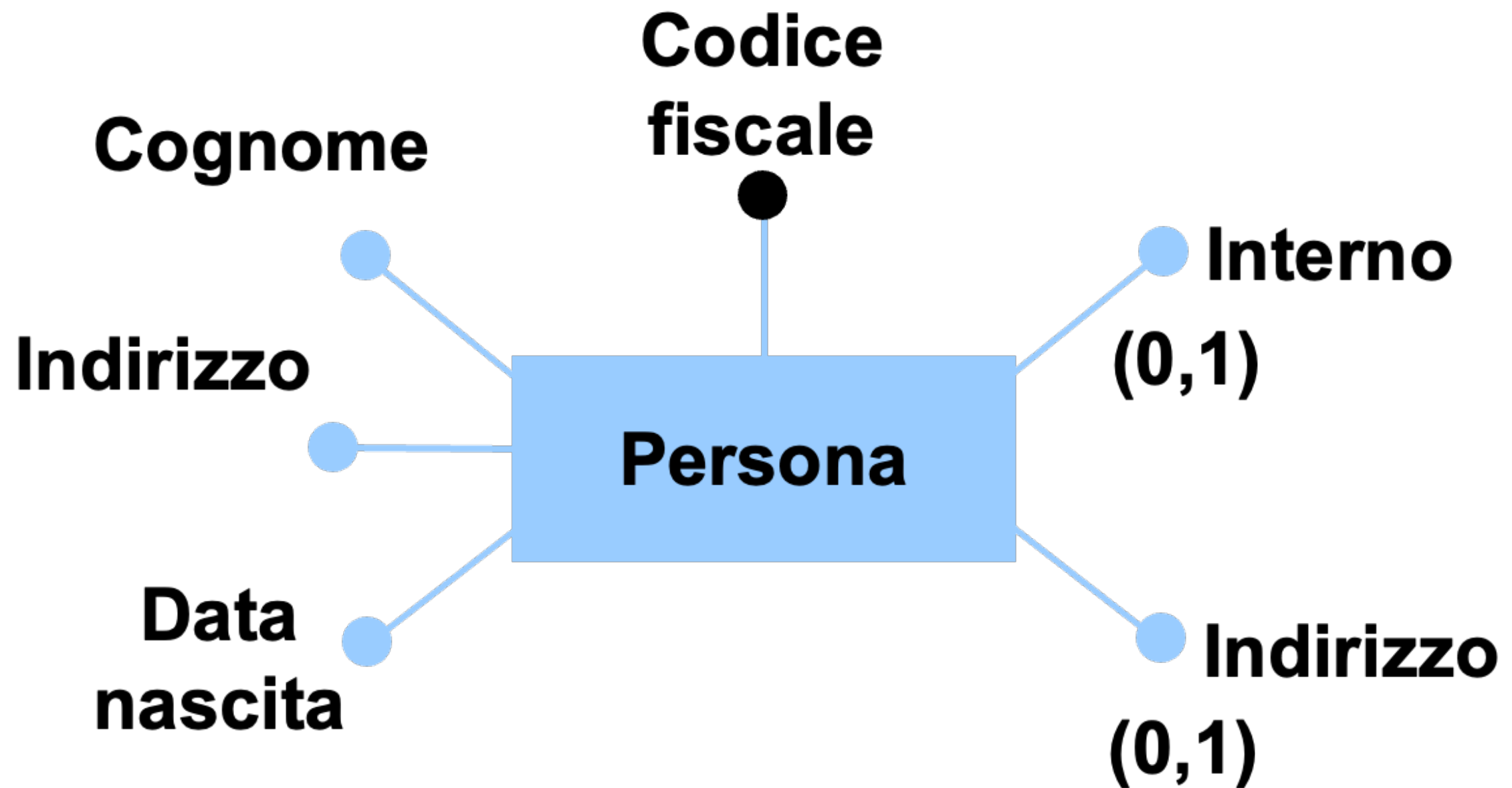
# Eliminazione di attributi multivalore



# Accorpamento di entità/*relationship*



# Accorpamento di entità/*relationship*



# Attività di ristrutturazione

- Analisi delle ridondanze
- Eliminazione delle generalizzazioni
- Partizionamento/accorpamento di entità e *relationship*
- **Scelta degli identificatori primari**



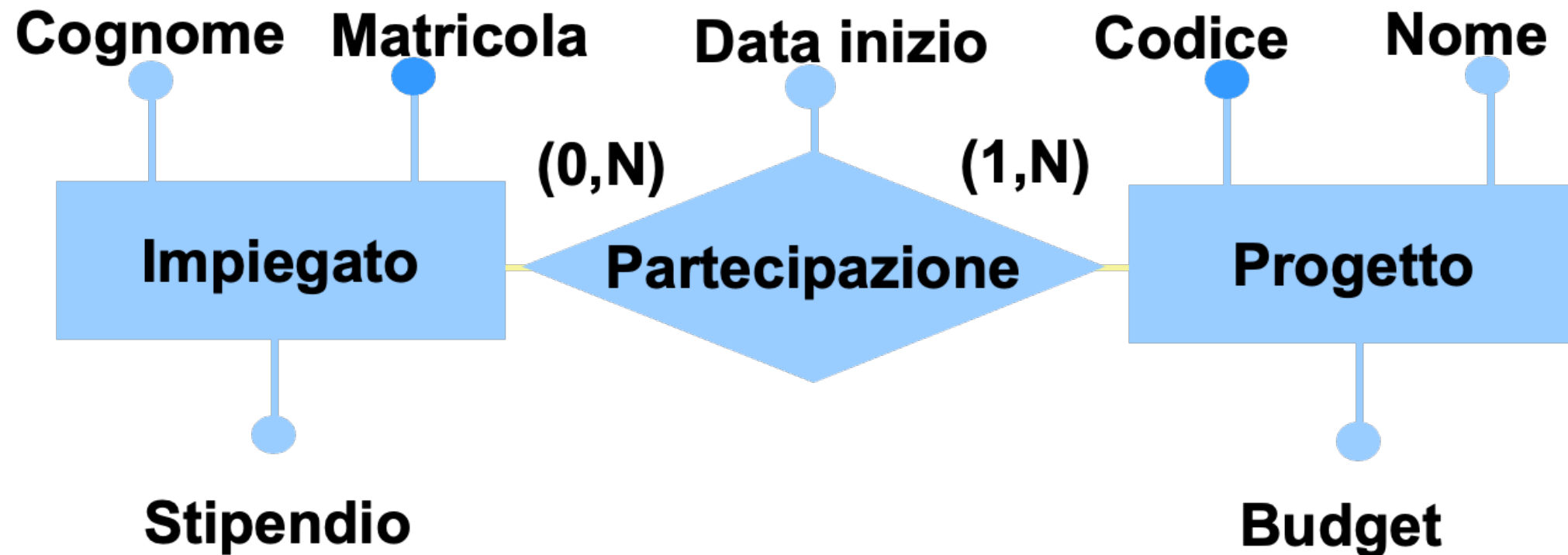
# Scelta degli identificatori principali

- **Operazione indispensabile** per la traduzione nel modello relazionale
- **Criteri:**
  - assenza di opzionalità
  - semplicità
  - utilizzo nelle operazioni più frequenti o importanti
- Se nessuno degli identificatori soddisfa i requisiti visti?
  - Si introducono nuovi attributi (**codici**) contenenti valori speciali generati appositamente per questo scopo

# Traduzione verso il modello relazionale

- Idea di base:
  - Le **entità** diventano **relazioni** sugli stessi attributi
  - Le ***relationship*** diventano **relazioni** sugli identificatori delle entità coinvolte (più gli attributi propri)

# Entità e *relationship* molti a molti

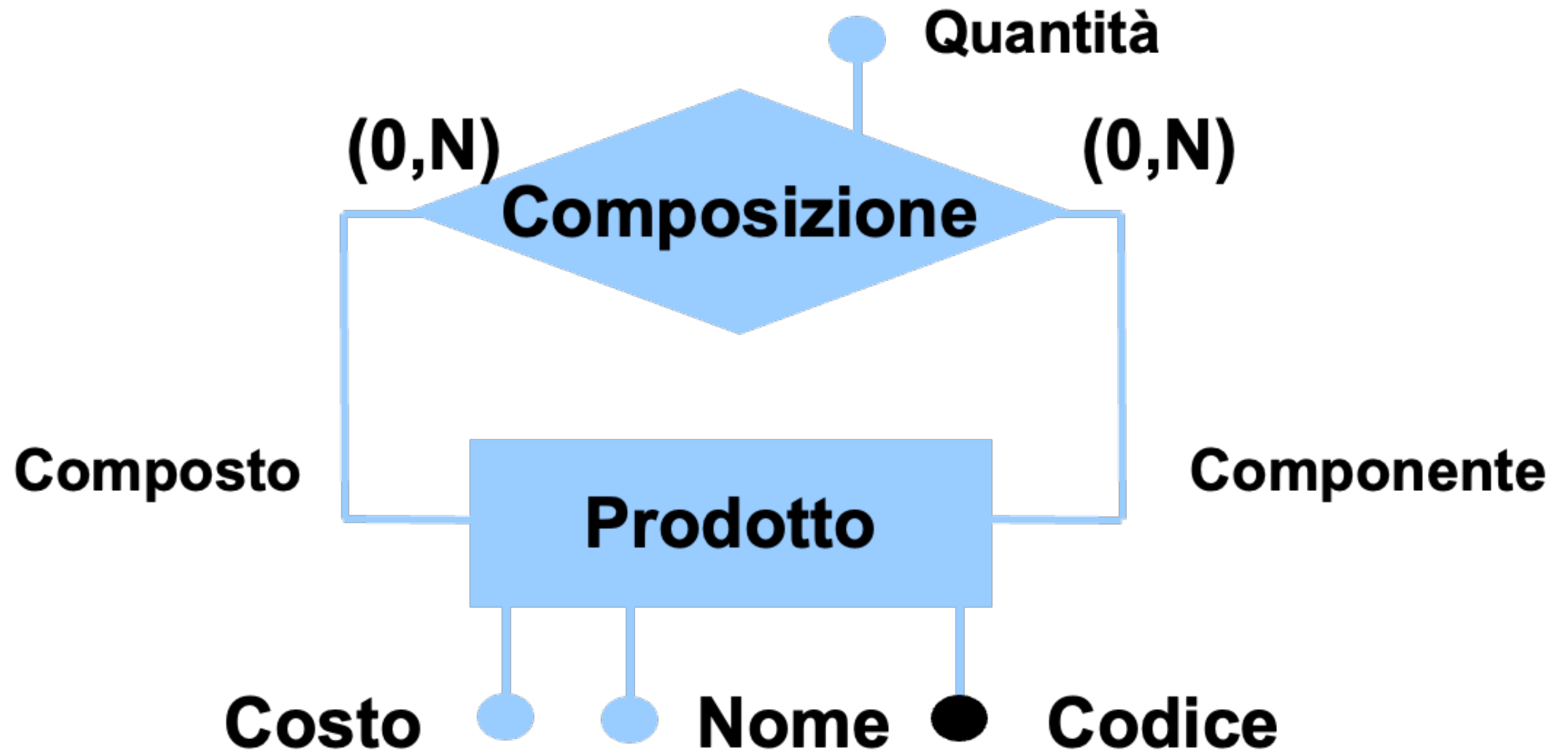


- **Impiegato**(Matricola, Cognome, Stipendio)
- **Progetto**(Codice, Nome, Budget)
- **Partecipazione**(Matricola, Codice, DataInizio)
- **Vincoli di integrità referenziale** fra:
  - Matricola in **Partecipazione** e (la chiave di) **Impiegato**
  - Codice in **Partecipazione** e (la chiave di) **Progetto**

# Entità e relationship molti a molti

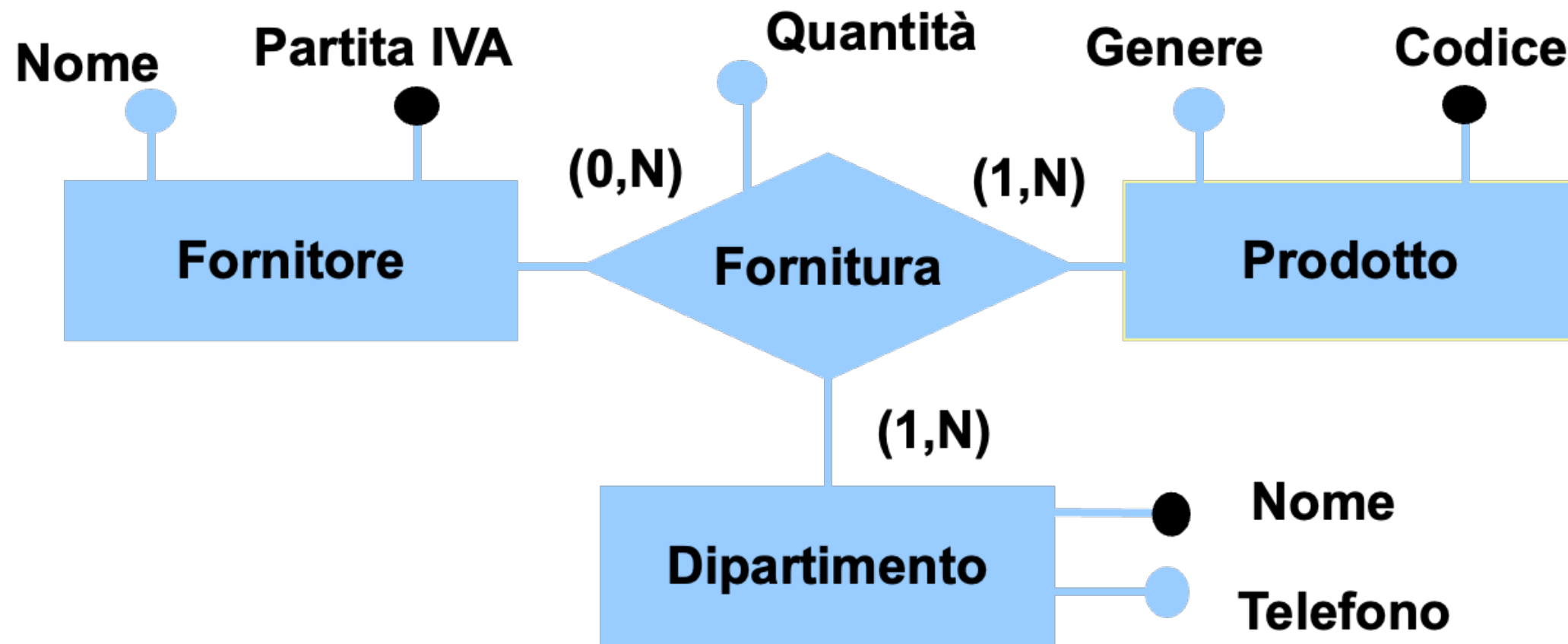
- Nomi più espressivi per gli attributi della chiave della relazione che rappresenta la relationship:
  - **Impiegato**(Matricola, Cognome, Stipendio)
  - **Progetto**(Codice, Nome, Budget)
  - **Partecipazione**(Matricola, Codice, DataInizio)
  - **Partecipazione**(Impiegato, Progetto, DataInizio)

# *Relationship* Ricorsive



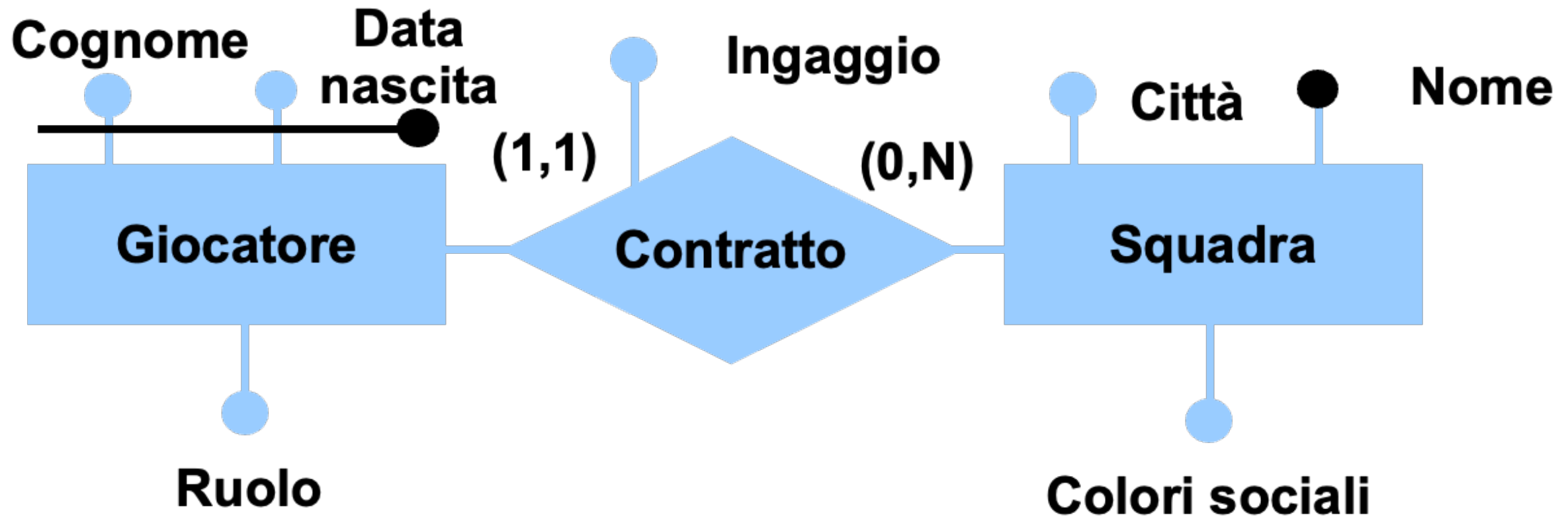
- **Prodotto**(Codice, Nome, Costo)
- **Composizione**(Composto, Componente, Quantità)

# *Relationship n-arie*



- **Fornitore**(Partita IVA, Nome)
- **Prodotto**(Codice, Genere)
- **Dipartimento**(Nome, Telefono)
- **Fornitura**(Fornitore, Prodotto, Dipartimento, Quantità)

# *Relationship uno a molti*



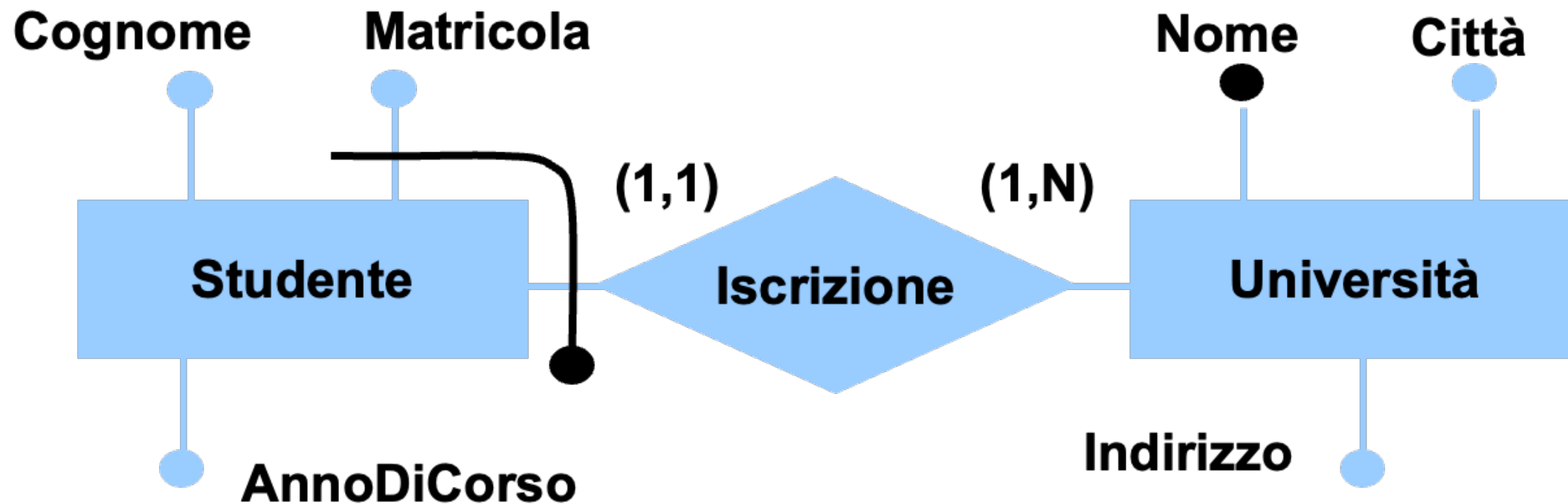
- **Giocatore**(Cognome, DataNascita, Ruolo)
- **Contratto**(CognGiocatore, DataNascG, Squadra, Ingaggio)
- **Squadra**(Nome, Città, ColoriSociali)
- È corretto?

# Soluzione più compatta

- **Giocatore**(Cognome, DataNasc, Ruolo, Squadra, Ingaggio)
- **Squadra**(Nome, Città, ColoriSociali)
- Con vincolo di integrità referenziale fra **Squadra** in **Giocatore** e (la chiave di) **Squadra**
- Se la cardinalità minima della *relationship* è 0, allora **Squadra** in **Giocatore** deve ammettere valore nullo
- La traduzione riesce a rappresentare efficacemente la cardinalità minima della partecipazione che ha 1 come cardinalità massima:
  - 0 : valore nullo ammesso
  - 1 : valore nullo non ammesso

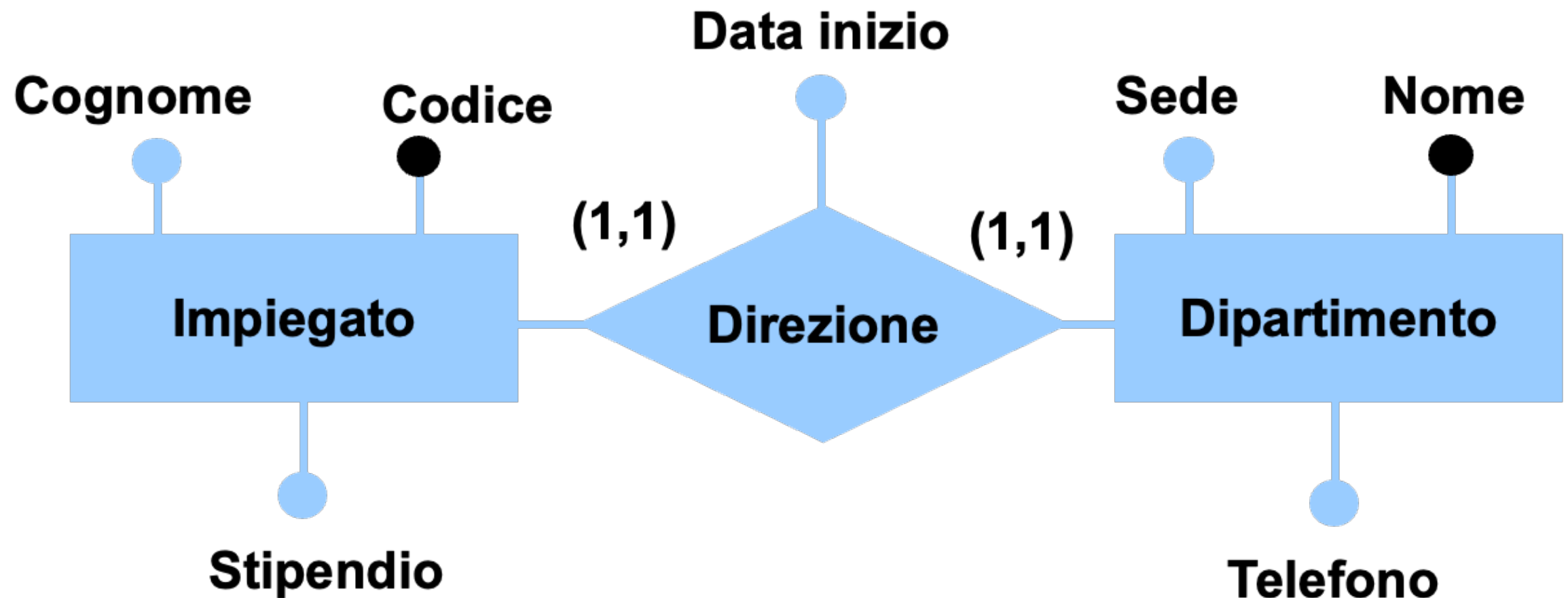


# Entità con identificazione esterna



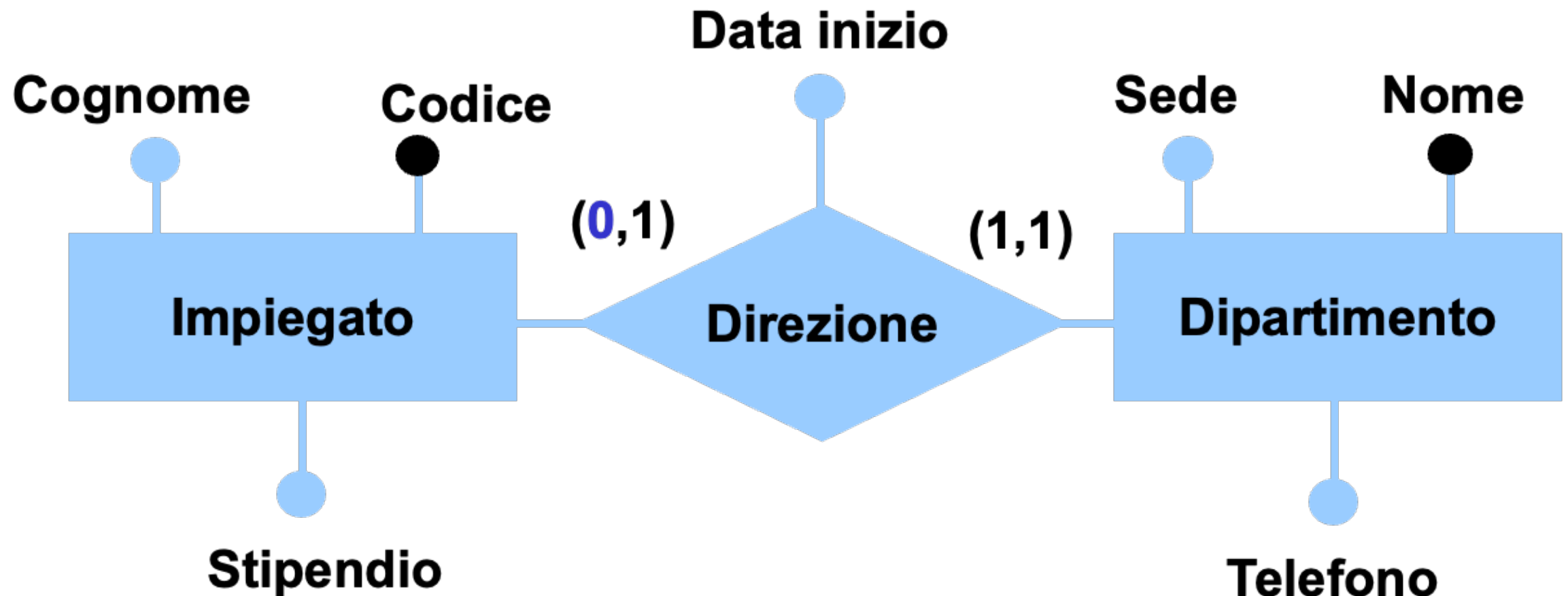
- **Studente**(Matricola, Università, Cognome, AnnoDiCorso)
- **Università**(Nome, Città, Indirizzo)
- con vincolo ...

# *Relationship* uno a uno



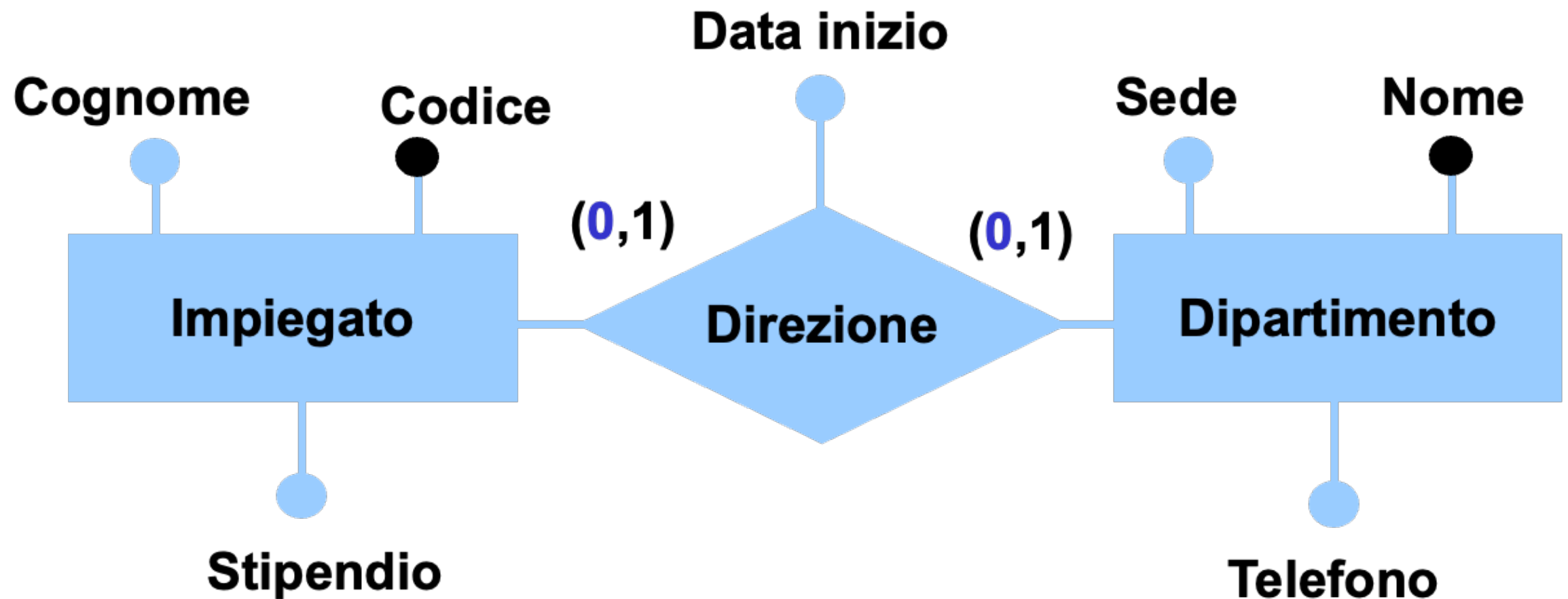
- Varie possibilità:
  - fondere da una parte o dall'altra
  - fondere tutto?

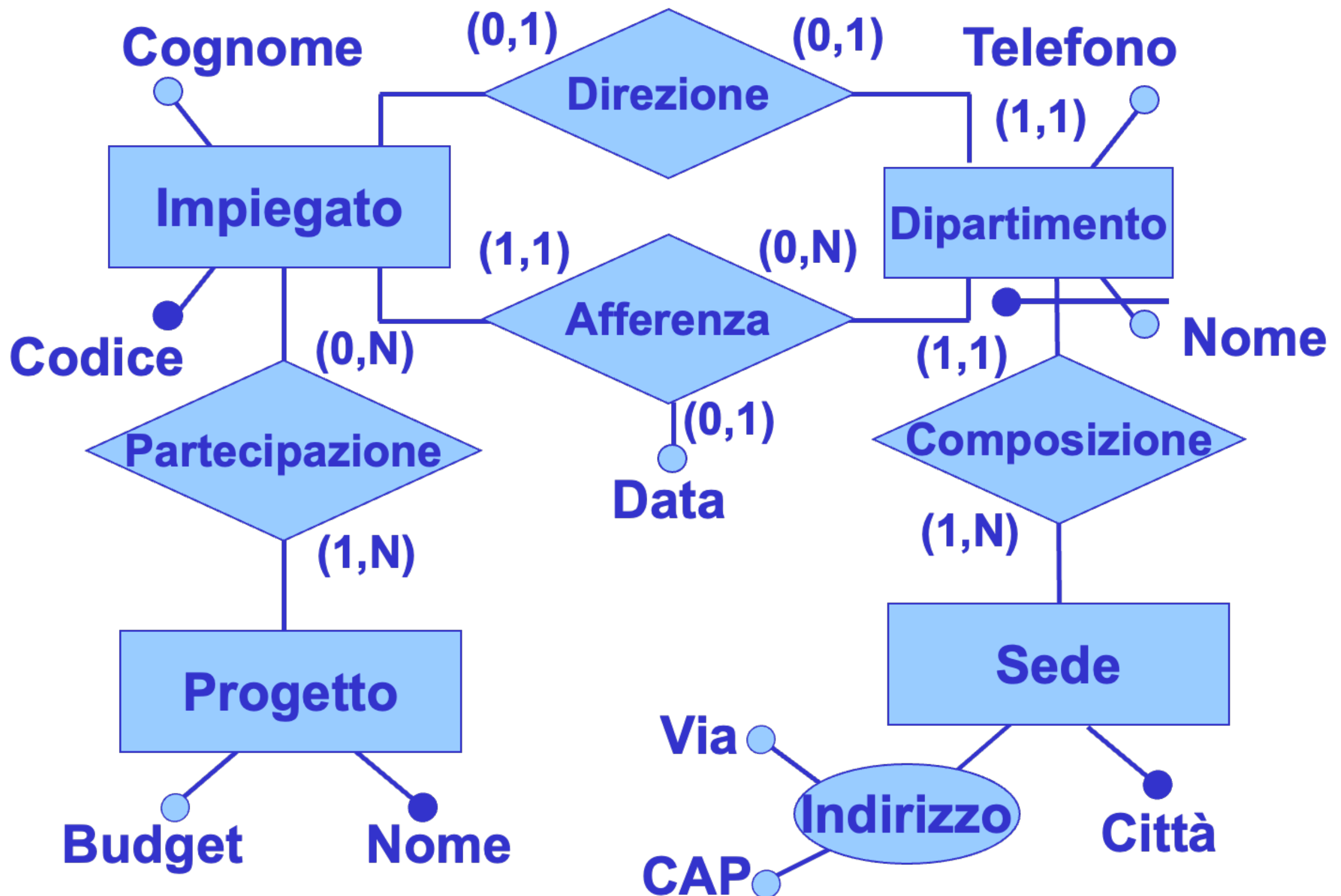
# Un caso privilegiato



- **Impiegato**(Codice, Cognome, Stipendio)
- **Dipartimento**(Nome, Sede, Telefono, Direttore, InizioD)
- con vincolo di integrità referenziale, senza valori nulli

# Un altro caso





# Schema Finale

- **Impiegato**(Codice, Cognome, Dipartimento, Sede, Data\*)
- **Dipartimento**(Nome, Città, Telefono, Direttore\*)
- **Sede**(Città, Via, CAP)
- **Progetto**(Nome, Budget)
- **Partecipazione**(Impiegato, Progetto)
- **ATTENZIONE**: differenze apparentemente piccole in cardinalità e identificatori possono cambiare di molto il significato ...

